

# Predicting Patient Mortality Using Graph Convolutional Neural Network Training

Jonathan Wu

Georgia Institute of Technology  
Atlanta, GA  
jswu72@gatech.edu

Xiao Luo

Indiana University-Purdue University Indianapolis  
Indianapolis, United States  
luo25@iupui.edu

Sanjay Gorur

The University of Texas at Austin  
Austin, TX  
sanjaygorur2001@utexas.edu

**Abstract**—Electronic health records contain valuable information in regards to patients’ features and can be manipulated to observe similarities where patients overlap to predict diagnoses. In order to detect such patient risk, a graph is constructed by extracting relevant features to relate patients in the EHRs and thus create a homogeneous graph to assess patient similarity. A novel GCN training algorithm alike Cluster-GCN is further used to predict mortality risk of patients by exploiting the graph clustering structure through node classification.

## I. INTRODUCTION

Patient mortality is a vast field in healthcare research that can often be forecasted using machine learning. Connecting patients to one another by examining relevant features such as demographics, interventions, vitals, etc., can be crucial to providing the opportunity to enhance proactive risk plans for groups of patients.

Large-scale electronic health record(EHR) databases generally possess the information of patients in intensive care units(ICUs) including the features outlined earlier. MIMIC-III, a data set used for the extensive purposes of this work, can be further simplified using an open-source pipeline or framework to solely extract relevant features for the mortality prediction task. From this approach, a patient graph is constructed such that patient nodes are related via edges using a similarity metric like Jaccard similarity [4], lending a hand to the prediction of mortality risk.

While previous works make use of training graph neural networks including the likes of gradient descent [5] and variance reduction [5], the new GCN training algorithm Cluster-GCN aids to train deep graph convolutional networks on formidably sized graphs, making efficient use of memory and achieving high accuracy when the model is utilized for node classification [2].

## II. BACKGROUND

### A. Graph Neural Networks

A common implementation for exploring the relationships between similar objects for prediction purposes is to represent the data as a graph neural network. [8] provides an in-depth explanation on graph neural networks and their applications to various domains, even to those outside the scope of this paper. Nodes in the graph will be assigned feature vectors which define each individual node through a series of values.

These values are propagated and aggregated amongst nodes connected by edges in a similar fashion to how the result of a node in the hidden layer is the result of the aggregation from the nodes in the layer before it. The duration and length of the propagation, as well as the weighting mechanisms are still an open area of research, especially since considerations must be made to temporal and spatial metrics due to a graph’s potentially large size. For the purposes of this paper, we elected to utilize Cluster-GCN, a type of graph neural network that has been proposed and shown by [2] to be an effective prediction model.

### B. LSTMs and Transformers

A large portion of medical documentation is in the form of handwritten or typed notes by doctors and nurses. While this documentation can provide more nuance in a patient’s condition, it is a difficult task for both a computer and observer to understand the connection and underlying meanings between two different sets of notes on patients. Understanding these similarities in these structureless formats is very similar to research in natural language processing (NLP). Many encoding schemes such as Long Short Term Memory Models and Transformers such as those described in [9] have been utilized with varying success towards these similarity matching schemes. [7] uses an LSTM in order to analyze the temporal features of the patients to achieve an embedding.

## III. METHODS

Our proposed model is composed of two main steps. The first step involves extracting relevant information from the MIMIC-III database in order to construct the patient graph. Leveraging nodes to represent patients in our graph, we then conduct a diagnosis-similarity analysis to determine if two nodes (patients) are similar enough to connect with an edge. Once the graph is constructed, it is then applied as input to the Cluster-GCN model provided and implemented by Rozemberczki<sup>1</sup>. Using Cluster-GCN, the model is trained and is then used to make per-node predictions on patient mortality. In the following sections we provide an in-depth explanation of our model. A graphical representation of our model can be seen in Fig. 1.

### A. Data

The dataset we use to create our graph is provided by MIMIC-III. According to [1], it is a large, publicly-available database that has anonymized patient records spanning 11 years. They provide records on 46,520 patients and provide information on relevant features such as vital sign measurements, medications, and procedures during that patient’s duration in the hospital. The data of these patients are dispersed among 20 CSV files in order to contain connected information in individual files. A *subject id* is used to distinguish patient records.

### B. Feature Extraction

In order to aid with training our model for patient mortality prediction, we constructed a feature vector matrix using elements taken from the MIMIC-III dataset. The features we elected to take from MIMIC-III were the patient’s age on admission, their ethnicity, their admission type, and their gender. Age on admission was calculated by extracting the patient’s admission time to the hospital and their date of birth. Ethnicity and admission type were categorical elements that we used a multi-hot encoded scheme for. Ethnicities were categorized into the most common categories provided by the US census. Ethnicity distribution and other data distributions for features used in our graph can be seen in Table 1. Gender was encoded as a trinary element. After the feature vector for all patients in the MIMIC-III database was calculated, we then applied a normalization scheme to create a better distribution of values for our matrix. A distribution of

1

TABLE I  
DATASET DISTRIBUTION

Ethnicity	Count
White	32380
Black	3849
Asian	1731
Hispanic or Latino	1659
Native American and Indigenous Peoples	46
Other or Unknown <sup>a</sup>	6855
Gender	Count
Male	26121
Female	20399
Admission Type	Count
Emergency	31495
Newborn	7752
Elective	6182
Urgent	1091
Patient Count	46520

<sup>a</sup>Empty entries, patients declining to answer, and multi-race ethnicities

### C. Edge Construction

In order to determine which edges would exist in our patient graph, we calculated diagnostic similarity scores amongst all pairs of patients in the graph. Diagnoses were encoded in the ICD-9 format, where the class of diagnosis is defined in the

<sup>1</sup><https://github.com/MLforHealth/MIMICExtract>

first three digits. If necessary, additional digits are provided after these three digits to define a subclass of diagnosis. For the purposes of our graph, we elected to truncate specific diagnoses and chose to only use the first three digits of the ICD-9 code.

A list of diagnoses for each patient was extracted from MIMIC-III. Each individual element in a patient’s diagnosis list would be checked if it existed in another list of diagnoses for a different patient. We assigned a value for each match, and if the aggregated values reached a threshold ( $c = 0.5$ ) then an edge would be constructed between these two nodes. The completed graph can be seen in Fig. 2, where the similar nodes have edges connecting them, and we have computed a feature vector for each node in this graph.

### D. Cluster-GCN

To train our graph we utilize Cluster-GCN, a graph neural network scheme proposed by Chiang et al[2], and implemented in PyTorch by Rozemberczki<sup>1</sup>. Cluster-GCN leverages a graph clustering scheme which allows it to reduce computation without sacrificing accuracy. In addition, it also provides a lightweight memory alternative for graph neural networks operating on large datasets. Inputting our graph into Rozemberczki’s implementation allows it to be trained on the prediction task as well as tested on it. Cluster-GCN provides us an F1 success score after its runtime, which allows our model to then be benchmarked against current state of the art models.

2

## IV. RESULTS

Because the model is still in its early stages, we have not yet run a robust series of tests to test the effectiveness of it. An example of a test is provided in Fig. 3.

Data points and graphs will be added in future editions of this paper. Currently the model needs to be effectively finetuned in order to properly test and evaluate it. We plan to benchmark our model against other state of the art models and include the metrics for that here. In addition, we will make sure to test it on various clustering methods so that we can get a more comprehensive understanding of its effectiveness amongst different orientations. Finally, we will analyze the results for all models tested and provide our own insight on the reasons behind each model’s performance.

## V. CONCLUSION

In this paper we present our patient graph model (yet to be named), which utilizes a diagnosis matching scheme amongst patients inspired by [7] in order to form the edges in our graph. We utilize several of the CSV files provided by the MIMIC-III database and normalize them to provide a better distribution. We then apply this graph to a PyTorch implementation of Cluster-GCN to train and test it on a patient mortality prediction task. We compare and analyze our model’s

<sup>2</sup><https://github.com/benedekrozemberczki/ClusterGCN>

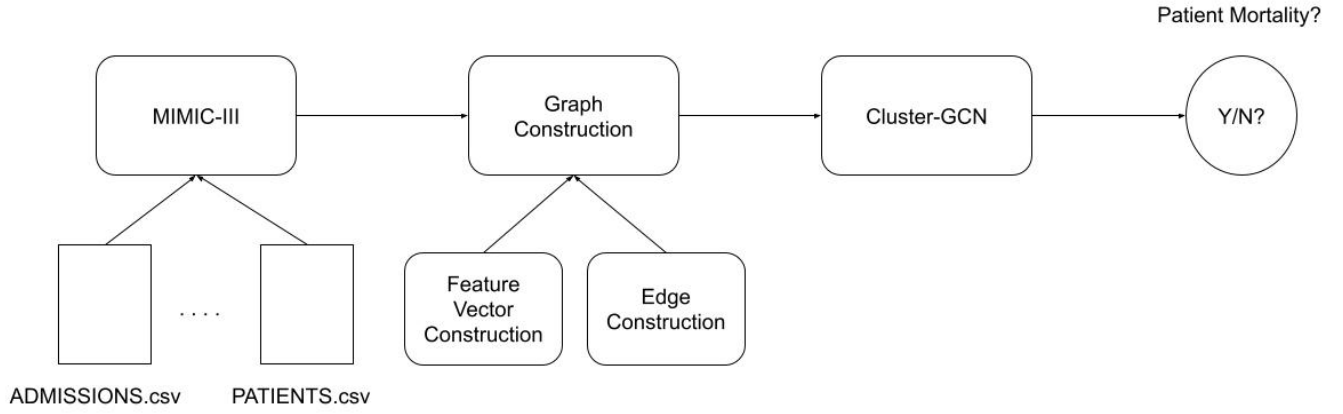


Fig. 1. The pipeline of our proposed model: relevant information is extracted from the MIMIC-III dataset. This data is used to construct feature vectors per-node, as well as calculate similarity between nodes for edge construction. After the graph is built we input it into a Cluster-GCN implementation to train and test it on the MIMIC-III dataset to predict patient mortality.

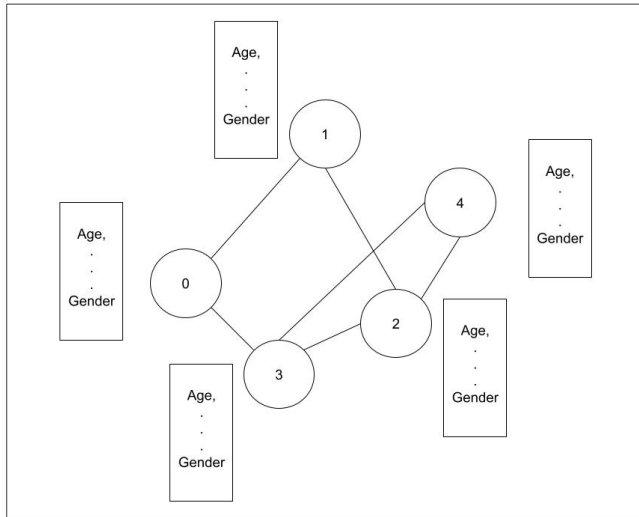


Fig. 2. Visual representation of the patient graph. Patients are represented by nodes and are assigned id numbers. In addition they have corresponding feature vectors, which are used to aid prediction when input into Cluster-GCN

performance to the performance of other state of the art models working on the same task.

## VI. RELATED WORK

Similar studies were conducted to predict mortality for patients using the MIMIC-III data set. One work predicts 30-day patient mortality by developing a machine learning approach using XGBoost for patients with a certain condition: sepsis-3, a significant cause of mortality in-hospital for ICU patients [3].

Moreover, additional studies observe graph neural networks through heterogeneous graphs to evaluate existing models for prediction tasks and propose new baselines. Another work explores the heterogeneity in EHRs and proposes a method to analyze records with a novel heterogeneous GNN. [4].

```

(kali@DESKTOP-0MMCKED) - [~/Desktop/ClusterGCN]
$ python src/main.py
+-----+-----+
| Parameter | Value |
+-----+-----+
| Cluster number | 10 |
+-----+-----+
| Clustering method | metis |
+-----+-----+
| Dropout | 0.500 |
+-----+-----+
| Edge path | ./input/edges.csv |
+-----+-----+
| Epochs | 200 |
+-----+-----+
| Features path | ./input/features.csv |
+-----+-----+
| Layers | [16, 16, 16] |
+-----+-----+
| Learning rate | 0.010 |
+-----+-----+
| Seed | 42 |
+-----+-----+
| Target path | ./input/target.csv |
+-----+-----+
| Test ratio | 0.900 |
+-----+-----+

Metis graph clustering started.
Training started.
Train Loss: 0.6923: 100%
F-1 score: 0.4910
  
```

Fig. 3. An example run of the ClusterGCN implementation provided by Rozemberczski. Options like clustering method, epochs, learning rate, test ratio can be modified at launch. F1-Score is used to signify the success rate of the prediction scheme.

In terms of constructing the patient graph, the consideration of semi-structured and unstructured data arises for the case of entity matching in healthcare. One study introduces a new benchmark that is capable of generalizing the research problem of entity matching by transforming homogeneous structured records into diverse structures as mentioned above.

Further, these studies facilitate future research to build on analyzing the health record data of patients.

## VII. FUTURE WORK

While the model we propose is functional, it is very rudimentary in its structure. In particular, we would focus on the preprocessing structure of the model. Currently, our model only has a simple diagnosis check between sets. In the future we would like for this checking scheme to provide more weight to diagnoses that are rarer than others. This would allow fewer edges to be constructed in our graph, making the edges in our graph a better representation of how similar two patients. In addition, we are only using the ICD-9 codes of the diagnoses. If possible we would like to additionally use the written names of these diagnoses and extract contextual information from them. Another area we could consider would be to factor in other elements besides diagnoses into the similarity consideration and include other relevant information like medications taken, admission reasons, or procedures done.

The data selected from MIMIC-III to construct the feature vector matrix is very sparse. Our feature vector contains four different elements for each node, whereas [6] uses 38 different elements for each of their nodes. While our feature vector is able to convey some information about the patient, it would be more helpful to collect more data points from the MIMIC-III dataset. This might mean accessing and parsing different CSV files than those explored in this paper. One possible source of information that was not explored in the creation of this model was in NOTEEVENTS.csv, which provides freehand documentation about a patient. While there is a vast potential amount of helpful information provided in this file, extracting this information would be quite difficult. In future editions of this paper, this information will be considered for its implementation into our graph construction process.

Finally, we need to conduct benchmark tests by comparing our model's performance to the performance of other models on the same prediction task of patient mortality. This will require identifying potential models to test against as well as reformatting their graphs to be utilized in the PyTorch implementation of Cluster-GCN.

## VIII. ACKNOWLEDGMENT

We would like to thank Dr. Xiao Luo for her guidance and assistance on this project. We would also like to thank those who provided feedback during the early stages of this paper.

## REFERENCES

- [1] Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3, 160035.
- [2] Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, Cho-Jui Hsieh. KDD, 2019.
- [3] Hou, N., Li, M., He, L. et al. Predicting 30-days mortality for MIMIC-III patients with sepsis-3: a machine learning approach using XGboost. *J Transl Med* 18, 462 (2020). <https://doi.org/10.1186/s12967-020-02620-5>
- [4] Zhu Z, Yin C, Qian B, Cheng Y, Wei J, Wang F. Measuring patient similarities via a deep architecture with medical concept embedding. In: 2016 IEEE 16th International Conference on Data Mining (ICDM). Barcelona, Spain: IEEE; 2016: 749–758.
- [5] J. Chen and J. Zhu. Stochastic training of graph convolutional networks. *arXiv preprint arXiv:1710.10568*, 2017.
- [6] Emma Rocheteau and Catherine Tong, Petar Veličković, Nicholas Lane, and Pietro Liò. Predicting Patient Outcomes with Graph Representation Learning/ *arXiv preprint arXiv:2101.03940*, 2021.
- [7] Isaac Ronald Ward, Jack Joyner, Casey Lickfold, Stash Rowe, Yulan Guo and Mohammed Bennamoun. A Practical Guide to Graph Neural Networks. *CoRR arXiv:2010.05234*, 2020.
- [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR arXiv:1810.04805*, 2018.