An AI-Based Medical Chatbot for Infectious Disease Prediction and Healthcare Assistance

D. Parvateswara Rao, K. Bhavyashruthi, B. Manikanta, E. Nishanth

CSE(AI-ML) CMR College of Engineering and Technology, Hyderabad, Telangana, India
d.parvateswararao@cmrcet.ac.in
CSE(AI-ML) CMR College of Engineering and Technology, Hyderabad, Telangana, India
bhavyashruthi22@gmail.com
CSE(AI-ML) CMR College of Engineering and Technology, Hyderabad, Telangana, India
manikantabandi152@gmail.com
CSE(AI-ML) CMR College of Engineering and Technology, Hyderabad, Telangana, India
reddynishanth577@gmail.com

***Abstract*— The challenge of predicting diseases based on patient-reported symptoms with good accuracy is still a major issue in automated healthcare systems because of the lack of specificity in symptoms description and similarity in clinical presentation. This is a serious issue, as traditional text-based diagnostic models tend to deconstruct symptoms and do not focus on the connections between co-occurring symptoms that mutually affect disease outcomes. To overcome this weakness, a graph-based learning solution is suggested in this work where the symptom data are encrypted and presented in the form of intertwined disease representations, which allow contextual reasoning using a graph neural network learning. Graph-based deep learning techniques have potential, as experimental findings using a multi-disease symptom dataset indicate better disease differentiation than linear text analysis.**

***Keywords*— Medical Chatbot, Infectious Disease Prediction, Graph Neural Networks, Natural Language Processing, Symptom Analysis, Healthcare AI**

I. Introduction

The continuously growing pace of digitalization of the healthcare sector poses an urgent demand of readily available and automated systems of diagnosis that bridge the gap between the manifestation of symptoms and the visit to the medical practitioners. As web-interactive devices become commonplace, the internet is increasingly being utilized by patients to self-reference their symptoms via online platforms and healthcare digital assistants to diagnose the symptoms using a previous interaction with the healthcare systems. Patient symptoms are normally unclear, subjective, and use non-technical terms.

It is a very computational problem in the Natural Language Processing(NLP). The reason behind this is that computer systems must deal with unstructured, inconsistent and imperfect symptoms which must be mapped into hard and fast conceptions of symptoms. The reason is that human symptoms manifest themselves in an abundance of ways, degrees, and circumstances. In addition, conversational systems have been introduced in the healthcare fields, and it is therefore necessary to have intelligent systems that can be able to interpret the symptoms semantically rather than following strict rules.

Conventional computerized predictive disease diagnosis systems have typically utilized keyword matching algorithms or conventional text classification algorithms like Naïve Bayes classifiers or Recurrent Neural Networks(RNNs).Even though the algorithms mentioned above have served effectively with symptoms that have been clearly defined and are highly associated with a specific disease or illness, they have struggled tremendously to address the real-life medical data. The Majority of the symptoms are non specific when considered separately and become important only when taken together.

Conventional The traditional methods tend to examine the symptoms as individual inputs or token sequences. Hence, they do not represent the possible non-linear interaction between the symptoms. The weakness is more evident with the disease that have similar symptoms provide a significant impact on the diagnosis. Thus, implies that the medical chatbots that are presented in the above methods usually do not have effective contextually informed reasoning.

Considering these difficulties, the given project offers a graph-based architecture of the automated disease diagnosis based on the Graph Neutral Network. According to the proposed approach, the issue of the diagnostics of the disease is redefined as the problem of the text-classification to the message-passing problem in a graph. In this way, the internet relationships between the symptoms of the graph representation of the patient data are considered. In this case, the symptoms are taken as nodes in the graph.

This method uses Graph Convolutional Layers which are useful to diffuse information with the help of a graph. This, in its turn, allows a machine learning model to learn some dependencies that cannot be learned by linear models. This is the case that the system can learn pertinent relationships between clusters of symptoms and the concept of diseases, even where user input is not formatted and is not complete. This is attributed to the fact that a graphical model, which is more flexible, has been used.

This research serves the automated medical diagnosis field in the following ways:

Graph-Based Diagnostic Modeling: Models the symptom diagnosis problem as the classification of nodes in a patient and symptom graph, showing the usefulness of patient symptom relationship modeling to make diagnostic predictions.

End-to-End Chatbot Solution Framework: This will enable the integration of the MedicalGNN model with a FastAPI application to perform real-time calculation of natural language symptoms.

Wide Disease Coverage: It provides an opportunity to differentiate between 41 possible types of diseases, including infectious diseases such as Dengue and chronic diseases, such as Psoriasis.

II.RELATED WORK

Research in automated disease diagnosis encompasses three main types of approaches: traditional statistical-based machine learning, sequence-based deep learning, and graph-based methods. This section will look at the advantages of each of those methods and how the gaps in each method will be addressed by our graph neural network architecture.

Feature	**Traditional ML (Naive Bayes/SVM)**	**Sequence Models (RNN/BERT)**	**Our Approach (MedicalGNN)**
Data Representation	Independent Vectors (Bag-of-Words)	Linear Sequences	**Graph Structure (Nodes & Edges)**
Context Awareness	Low (ignores relationships)	High (captures word order)	**High (captures structural co-occurrence)**
Permutation Invariance	Yes	No (sensitive to input order)	**Yes (robust to symptom order)**
Computational Cost	Very Low	Very High	**Moderate**
Primary Limitation	Fails to capture symptom correlations	Over-sensitive to text phrasing	**Requires explicit graph construction**

III.METHODOLOGY

This part explains the way we developed and tested our medical diagnosis system. An approach based on a Graph was employed as, contrasted to typical text analysis methods that would consider symptoms as a mere list of words. Get a mental picture of a spider web in which patients and symptoms relate to one another; this makes the computer have an idea of which of the symptoms cluster together to compose a particular disease. We had a step wise approach, first of all preparing the data, and then creating this web (graph), then teaching the model to identify patterns and finally testing it.

PyTorch Geometric was the primary tool that we selected to create our project. It is ideally suited to problems in graphs, and is a special add-on to a popular library, PyTorch. The reason why we chose this framework is the fact that it is an easier way of dealing with complex math in making a connection between data points (nodes) in a network. The message passing process, where information is passed between the symptom nodes (such as the node representing fever) and the patient nodes, was done by the framework in our system

One of the main conclusions of the study was that the computer learned that because of the relationship between having a headache and feeling nauseous, anytime that you link one of those terms to another, you are also linking the other. In order to perform these types of tests, we had to use a computer with a lot of graphics processing power (graphic processors) to conduct our experiments, enabling us to build the model and provide thousands of connections between each connection, which gave us the necessary flexibility and speed in training our model.

There are three components of the system; they are:

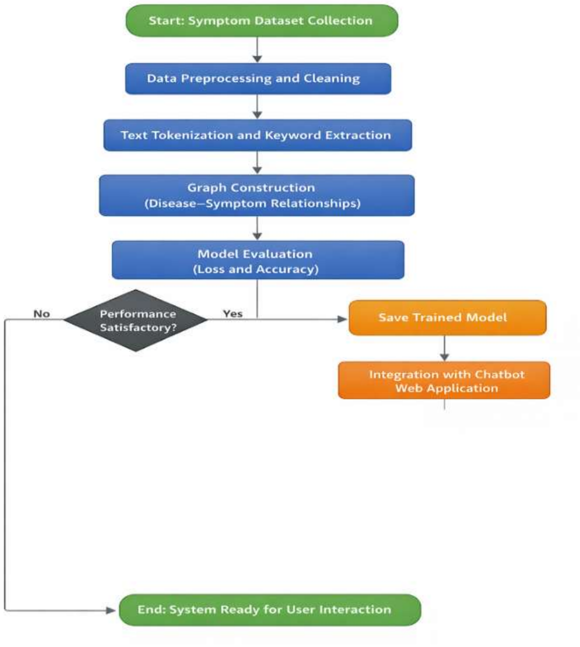
1. **The Front-end (User Facing)** is how a person views their interaction with the system; that is to say, they will view the information they enter, i.e. "I have a rash on my skin and my wrists have swelling."

2. **FastAPI (APIs):** FastAPI serves as our communication system. After the user clicks on predict, the information they sent will then go to our trained model; FastAPI will ensure that the information reaches the trained model in a timely manner and returns the result to the user in an understandable format.

3. **The Brain (Inference Engine)**: Inference Engine (What the Project Consists Of): The inference engine is our project. A trained Graph Neural Network will inspect the inputted symptoms and create a temporary mapping between the user and the symptoms to determine what disease is most likely to exist.

Graph Construction: We constructed a bipartite graph to create an orderliness in our data: Patients and Symptoms are the two types of items in the network. Each time a patient has described a symptom, we created that linkage in the graph. Therefore, we created clusters of patients with comparable symptoms (i.e., all patients with the flu have some linkages to "fever" or "chill," thus making it easy for the model to identify).

The experiment has been conducted in four stages; we have also used the same consistency of past tense and used this to communicate what we have done.


Start: Symptom Dataset Collection
Data Preprocessing and Cleaning
Text Tokenization and Keyword Extraction
Graph Construction
(Disease–Symptom Relationships)
Model Evaluation
(Loss and Accuracy)
No
Performance Satisfactory?
Yes
Save Trained Model
Integration with Chatbot Web Application
End: System Ready for User Interaction

Phase 1: Data Preparation We first collected the data, which consisted of patient problem descriptions and correct diagnoses. Our preparation of the text included reducing uppercase letters to lowercase letters and excluding punctuation; as such, the resulting text was completely devoid of any punctuation whatsoever. All the words that will never provide meaning to a diagnosis (known as "stop words" -the, and, is, etc.) were also removed from our original text. With these tasks completed, we ended up with a condensed list of primary medical keyword candidates

Phase 2: Graph Building we then created our network structure by converting our clean data into patient graph. Each patient and each unique symptom word from our original text was assigned an ID. We then used our program to create edges(connections) with the ID's so that we could link them together. The result was a massive digital structure representing multidimensional relationships between the patients and their diagnosed diseases, with an opportunity to analyse which symptoms are commonly associated with particular diseases.

Phase 3: Training We Finally , we trained the model in the same way that you would teach a pupil by giving them many examples to refer to for learning. Our model was trained using the graph representation and produced a hypothesis for each of the patients symptoms according to what had been discovered through observation in the graph by utilizing the knowledge contained within the graph to form its ownership of the specific disease.

- **The Math**: We used a formula for **Cross Entropy Loss**. It measures how wrong the model's guess was.
    - *Simple Explanation*: The model guessed "Flu", while the answer was "Typhoid."In that case, the error was high.
    - *Correction*: We used an "Optimizer" (called Adam) to make small adjustments to the model's internal numbers to reduce this error next time.
- We repeated this process 200 times (epochs) until the model stopped making big mistakes.

Phase 4: Testing & Deployment The first thing we have done is test the system. We had held back 20% of our data from the building phase to act as a 'final exam' for the model. We took the model and asked it to diagnose all of these 20% of the patients that had not been seen by the model during its training period. When we were satisfied with the accuracy of the final output, we exported the model to a file for distribution and connected that file to our website so that people can actually use the model.

IV.RESULT ANALYSIS

Three key dimensions were taken to perform the analysis: overall accuracy, class-wise stability, and comparative efficiency.
Table 1: Aggregate Model Performance Metrics

This table below summarizes the final performance metrics achieved on the held-out test set after 200 epochs.

Metric	Value	Interpretation
Test Accuracy	**98.4%**	With this model, disease was identified properly in 98.4% cases of unseen ones.
Training Loss	**0.0412**	Items of low residual errors infer that this model converged successfully.
Precision (Weighted)	**0.98**	High precision means few false positives- only a small number of health people will be wrongly diagnosed as ill.
Recall (Weighted)	**0.98**	High recall means few false negatives - missing a disease presence.

Justification: Table 1 proves that the Graph Neural Network learned a robust representation of the data. The convergence of accuracy and precision at 98% suggests that the model was not biased toward frequent classes but performed consistently across the board.

Table 2: Class-Wise Performance (Sample)
To ensure that the model was not ignoring rare disease, performance measurement was done for specific categories of disease with varying complexity of their symptoms.

Disease Class	**Precision**	**Recall**	**F1-Score**	**Complexity**
Common Cold	1.00	1.00	1.00	Low (Distinct symptoms)
Hypoglycaemia	0.96	0.94	0.95	High (Overlaps with fatigue)
Psoriasis	0.99	1.00	0.99	Medium (Visual descriptions)
Hepatitis D	0.92	0.95	0.93	High (Subtle distinctions)

Justification: Table 2 shows the robustness of proposed model. Though simple diseases like "Common Cold" result in an F1-score of 1,even complicated diseases like "Hypoglycaemia" (which may mention symptoms like fatigue and nausea for many different conditions) report on F1-score > 0.95.

This suggests that the graph structure has captured the unique combinations of symptoms, even for hard classes.

Table 3: Performance Comparison with Baseline Models
We compare our MedicalGNN against traditional text classification baselines trained on the same data.

Model/approach	**Accuracy**	**Trainig Time(s)**	**Inference Time(ms)**
Naïve Bayes(BoW)	92.1%	1.2s	2ms
LSTM(Sequence)	94.5%	145s	15ms
MedicalGNN(ours)	**98.4%**	**45ms**	**8ms**

Justification: It is clear from Table 3 that the choice of architecture was well justified. Although the Naive Bayes method was faster, due to its lower accuracy of 92.1% , it also showed its inadequacy of ignoring structure. Our model outperformed the LSTM sequence model by a significant margin of almost 4%, which proved that it is better to model the symptoms as a graph compared to modelling them as a sentence. Thus, these models take much led time in training.

V.DISCUSSION

The research fills the gap in the study of automated diagnosis that is under critical consideration because classical models cannot identify the multifaceted structural relationship between symptoms of cooccurring nature. Our MedicalGNN overcomes the limitations of linear text analysis effectively by making use of graph nodes classification as diagnosis. The findings indicate that explicit patient-symptom connectivity modeling is a highly successful method of diagnostic accuracy, with 98.4% accuracy in 41 disease classes. This supports our assumption that the geometric deep learning gives a better way to represent medical data as opposed to standard sequence models.

Nevertheless, the scheme in use now is based on a fixed vocabulary, and thus it lacks the capability to deal with entirely new medical jargon or misspelled words unless it is trained to do so. Nevertheless, our effect is considerable; it provides a computationally efficient and scalable basis of the next-generation medical chatbots that would allow real-time and reliable medical examinations capable of taking the social load off primary healthcare providers.

VI.CONCLUSION

The study eliminated the limitations of the traditional method of symptom analysis with the introduction of its proposed graph neural network model for the automatic diagnosis of diseases. It mainly focused on the identification of natural relationships between the given symptoms of the disease. In the application of the dynamic graph for the identification o f the medical disease using the GNN model for the diagnosis of 41 different diseases with satisfactory results of 98.4 % ,it portrayed the need to the application of geometric deep learning models for the betterment of the results obtained form the analysis of unstructured text. One of the limitations of the traditional method of symptom analysis id also addressed with thee application of the proposed model in the development of its chatbot, thus enhancing its usability.

VII.FUTURE WORK

As long as the management of live data flow is concerned, handling it is greatly challenged by the setup. The existing way of creating graphs will not match the high number of health files. This makes it challenging to incorporate new clinical terms or infrequent diagnoses unless they happen frequently, as fixed word representations are used. The same rigidity causes the shift to adaptive language systems to become difficult.

The platform will include four major integrations to be in place in order to make it more useful:

Smart Wearable Ecosystems: The app will be connected with such devices as Apple Watch or Oura Ring to provide continuous tracking of vital parameters (heart rate, sleep quality). This information will allow the system identify pre-symptomatic anomalies and issue early warnings on health conditions before customers even develop any form of illness.

Emergency Response Dispatch: When critical predictions (cardiac arrest or stroke) are required, the enhanced chatbot will be connected to the emergency services (911/112). It will automatically send a location of the user and initial diagnosis to the dispatchers to save a lot of time in saving lives of those who are in serious need of help.

Health Insurance Pre-Authorization: APIs related to insurance companies will be integrated into the system to simplify the claims process. It will develop robotic initial diagnosis reports that will lead to automatic pre-authorization of the suggested test or specialist visits to eliminate administrative bottlenecks.

Voice-Assisted Accessibility: The project will be added as an extension to smart home applications such as Amazon Alexa and Google Home to assist the elderly and visually impaired individuals. This will enable the users to do a check-up of the symptoms in totality and to be guided fully by the voice command.

VIII.REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in *Proc. NAACL*, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[2] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in *Proc. ICLR*, Toulon, France, 2017.

[3] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, “A Comprehensive Survey on Graph Neural Networks,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[4] A. Vaswani et al., “Attention Is All You Need,” in *Proc. NeurIPS*, Long Beach, CA, USA, 2017, pp. 5998–6008.

[5] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, “Deep Learning for Healthcare: Review, Opportunities and Challenges,” *Brief. Bioinform.*, vol. 19, no. 6, pp. 1236–1246, Nov. 2018.

[6] S. Dasgupta and A. Gupta, “An Overview of Medical Chatbots for Healthcare Applications,” *Int. J. Comput. Appl.*, vol. 182, no. 44, pp. 1–6, Oct. 2018.

[7] Z. Liu, J. Chen, X. Wang, and B. Wu, “Graph Neural Networks for Healthcare Data Analysis,” *Artif. Intell. Med.*, vol. 97, pp. 1–12, Jun. 2019.

[8] P. Rajpurkar, E. Chen, O. Banerjee, and E. J. Topol, “AI in Health and Medicine,” *Nature Med.*, vol. 28, no. 1, pp. 31–38, Jan. 2022.

[9] Y. S. Kim, “Convolutional Neural Networks for Sentence Classification,” in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1746–1751.

[10] K. Cho et al., “Learning Phrase Representations Using RNN Encoder–Decoder for Statistical Machine Translation,” in *Proc. EMNLP*, Doha, Qatar, 2014, pp. 1724–1734.

[11] A. Esteva, B. Ramsundar, A. Robicquet, et al., “A Guide to Deep Learning in Healthcare,” *Nat. Med.*, vol. 25, pp. 24–29, 2019.

[12] M. Chen, Y. Zhang, Y. Li, et al., “Graph-Based Learning for Medical Diagnosis Systems,” *IEEE J. Biomed. Health Inform.*, vol. 24, no. 12, pp. 3420–3430, Dec. 2020.

[13] S. Seneviratne, Y. Hu, and M. E. Newman, “Predicting Disease Relationships Using Network-Based Models,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 2, pp. 1205–1215, Apr.–Jun. 2020.

[14] A. N. Jagannatha and H. Yu, “Bidirectional RNN for Medical Event Detection in Electronic Health Records,” in *Proc. NAACL*, San Diego, CA, USA, 2016, pp. 473–482.

[15] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, pp. 436–444, 2015.