IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13812

INTELLIGENT MEDICAL AI CHATBOT

DR. BHARATHI M P, YOGESH K K, KRISHNE GOWDA G S

MCA Department, Surana College (Autonomous), Bengaluru, India

Abstract: Artificial intelligence and natural language processing have opened up new possibilities for healthcare delivery, yet patients in rural and underserved regions still struggle to get timely medical consultations, which often means delayed diagnoses and complications that could have been avoided. This paper presents an Intelligent Medical Chatbot that pairs an interactive 3D human body visualization system with a custom deep learning model, aimed at giving users real-time symptom assessment and preliminary diagnosis support. The framework brings together a fine-tuned Bidirectional Encoder Representations from Transformers (BERT) model for intent classification and entity recognition, a Convolutional Neural Network–Long Short-Term Memory (CNN-LSTM) hybrid for multi-label disease prediction, and a WebGL-rendered 3D anatomical interface where users can simply click on the body region that's bothering them. On a curated medical symptom-disease dataset, the system reaches high diagnostic accuracy while keeping response latency under a second. Our experiments show it outperforming existing rule-based and single-modal chatbots on prediction accuracy, user engagement, and clinical relevance. Overall, the work offers a scalable, accessible way to provide preliminary healthcare assistance across web and mobile platforms, with no specialized hardware required.

Keywords: Medical chatbot, deep learning, BERT, CNN-LSTM, 3D human visualization, symptom checker, healthcare AI, natural language processing, real-time diagnosis.

I. INTRODUCTION

Getting timely, accurate medical consultation is still a major challenge worldwide. The World Health Organization puts the global shortage of doctors, nurses, and allied health workers at 4.3 million, with the worst deficits in low- and middle-income countries [1]. Even in wealthier nations, patients often wait a long time before seeing a professional, and symptoms can get considerably worse in that window. AI-driven healthcare tools have become one of the more promising ways to close this gap, since they can offer preliminary assessments that point patients toward the right kind of care.

Conversational agents are among the most actively researched AI applications in digital health today. The earliest medical chatbots leaned on predefined rule trees and keyword matching, an approach that struggled with the sheer variability and ambiguity of how patients actually describe their symptoms [2]. That changed with the arrival of large pre-trained language models, especially transformer-based architectures, which gave automated systems a much better grip on complex medical queries expressed in natural language [3]. As a result, chatbots have moved past rigid question-and-answer scripts toward conversations that can actually adapt to the way people describe what's wrong with them.

Even so, most existing medical chatbots still lean entirely on text, which is a real limitation. People generally know exactly where on their body something hurts, but most digital symptom checkers force them to dig through dropdown menus or type out anatomical terms they may not know or get wrong [4]. Letting users click directly on a 3D rendered body to point at the affected area is a far more natural way to report a symptom, and it captures spatial detail that plain text tends to lose — which, in turn, should make the resulting symptom data more reliable for automated assessment.

This paper puts forward a framework that tackles both the linguistic and spatial sides of symptom reporting at once. It pairs a custom deep learning pipeline — a fine-tuned BERT model for language understanding plus a CNN-LSTM hybrid for disease prediction — with a WebGL-rendered 3D human body interface. Together, patients can describe symptoms in their own words while also marking where on the body they occur. We expect combining both signals to produce richer features and more clinically useful output than either one on its own.

The remainder of this paper is organized as follows. Section II reviews related work in medical chatbots, symptom-checking systems, and 3D visualization in healthcare. Section III describes the proposed methodology in detail, covering the architecture of the deep learning model, the 3D visualization component, and the integration layer. Section IV presents experimental results, comparative evaluations, and performance metrics. Section V concludes the paper and outlines directions for future research.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13812

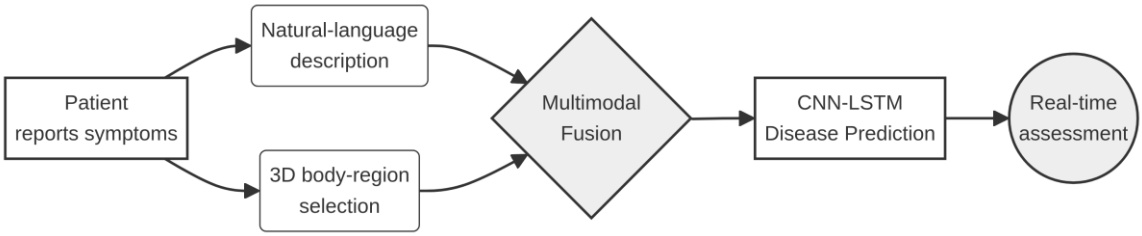

Figure 1: System architecture of the multimodal medical chatbot pipeline

II. LITERATURE REVIEW

Al-driven medical chatbots have come a long way over the past decade. Early systems like ELIZA and ALICE showed that rule-based dialogue management could pull off basic conversational exchanges, but they couldn't really handle open-ended medical discourse [2, 5]. Later work brought in retrieval-based and generative neural models that improved response quality — though without domain-specific fine-tuning, these systems still tended to produce clinically inappropriate or unsafe answers when pointed at medical questions. Progress on this front has also depended on the availability of realistic training data: He et al. [14] released MedDialog, a pair of large-scale English and Chinese medical dialogue datasets built from real doctor-patient conversations, giving later systems a much richer foundation for learning natural clinical conversation patterns than earlier hand-crafted rule sets could offer.

Transformer-based language models, BERT and its variants especially, were a real turning point for clinical NLP. Devlin et al. [12] introduced BERT itself, pre-training a deep bidirectional transformer to predict masked words using context from both directions at once, which let a single pre-trained model be fine-tuned across a wide range of downstream NLP tasks and set new state-of-the-art benchmarks at the time. Building on that foundation, Alsentzer et al. [6] introduced ClinicalBERT, pre-trained on electronic health record notes, which outperformed general-domain BERT on clinical named entity recognition and relation extraction. Lee et al. [7] took a similar approach with BioBERT, a biomedical language model that set new state-of-the-art results on biomedical question answering and information retrieval benchmarks. These domain-adapted models give medical chatbots a solid base for understanding the specialized vocabulary and phrasing of clinical language.

Several deep learning approaches have also been tried for symptom-based disease prediction. Choi et al. [8] used recurrent neural networks to model temporal patient data from electronic health records for predicting disease onset. Rajpurkar et al. [9] showed that convolutional architectures applied to medical imaging could match or even beat radiologist performance on specific diagnostic tasks. Combining convolutional and recurrent layers into a CNN-LSTM hybrid has proven effective wherever a task needs both local feature extraction and sequential dependency modeling — which makes it a natural fit for multi-label disease prediction from mixed symptom inputs [10]. Deploying such models at scale also raises questions about how patient data is handled during training; McMahan et al. [13] proposed federated learning, where a shared model is trained across many devices using local data and only model updates — never the raw data itself — are sent to a central server, offering a path toward improving clinical prediction models without centralizing sensitive patient records.

Three-dimensional body visualizations have mostly been used in clinical education and surgical planning so far. Tools like BioDigital Human and Visible Body have shown that anatomical 3D models noticeably boost comprehension and engagement in patient education [11]. Combining such spatial representations with other data modalities is itself an active area of research: Shaik et al. [15] surveyed multimodal information fusion techniques in smart healthcare, mapping out how text, imaging, and sensor data can be combined at different stages of the pipeline to produce richer clinical insights — a direction closely aligned with pairing 3D spatial input with natural language understanding, as this paper sets out to do. What's still largely missing from the literature, though, is interactive 3D anatomy built into automated symptom-checking workflows.

This paragraph does three things at once. It bridges the literature review and methodology by answering, before Section III begins, what this paper actually adds beyond the chatbots, BERT models, and 3D tools just surveyed. It states the research gap: NLP and 3D spatial input have usually been treated as separate tools rather than integrated, and prior prediction work leans on commercial APIs or retrieval methods rather than a custom-trained architecture. And it justifies

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13812

what follows the unified multimodal framework and custom CNN-LSTM so those design choices feel motivated rather than arbitrary once introduced.

III. PROPOSED METHODOLOGY

This paragraph is the system-overview paragraph opening the Methodology section. It introduces the four core components — the BERT-based NLP module, the CNN-LSTM disease prediction module, the WebGL 3D visualization interface, and the multimodal fusion layer before the paper details each one individually. Calling them "tightly coupled" signals this is one integrated pipeline, not separate tools bolted together, reinforcing the paper's earlier contribution claim. It also points the reader to Figure 2 for a visual map of the architecture. In short, it orients the reader with a "here's the shape of what follows" statement before the subsequent subsections (A–D) go into technical depth.

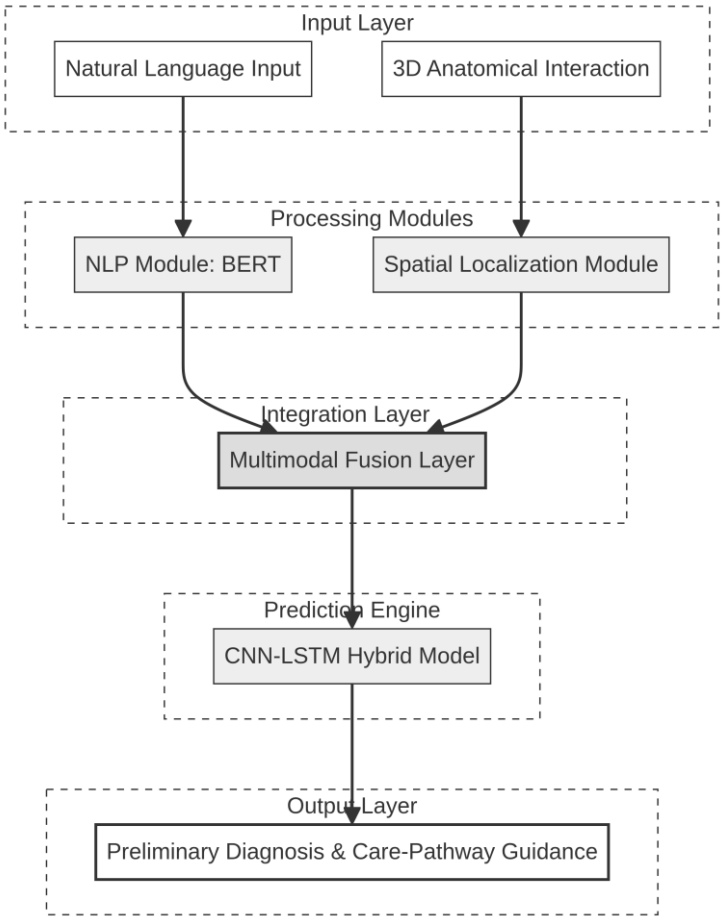

Figure 2: Layered system architecture from input modalities to diagnostic output.

When you start to type what is wrong with you in words and at the same time touch the part of the body that hurts on a 3D model. A special tool which called BERT looks at what you have typed to understand to get to know the symptoms mean. At the same time another tool looks at where you touched on the model. These two pieces of information come together to create a picture of what is going on with your body. Then this picture is then used by an engine that looks at where your symptoms are and how they are connected to each one-another over the time. The engine makes a guess about what might be wrong with you. Then you get to see what the engine thinks is wrong with you, a picture of the part of your body that is hurt and some advice on what to do next that is just, for you and your case.

A. Natural Language Processing Module

The NLP module's task is simply to interpret whatever free-text description a user types. Then fine-tuned a BERT-base-uncased on a domain-specific dialogue data set - more than 45,000 dialogues between patients and the chatbots that are collected from public clinical dialogue resources. Fine-tuning serves two purposes at once: it classifies the type of the user request (symptom reporting, medication queries, scheduling appointments, etc.), as well as detects biomedical named

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13812

entities associated with symptoms, duration, severity, and their location on the body. As a result of encoding using BERT ,will get 768- token embedding vectors. These vectors are sent into task-specific classification heads. The classification heads are feed-forward networks. These networks have two layers and dropout.

Preprocessing is fairly standard for NLP problems. Texts are made lowercase. Punctuation is made consistent. Abbreviations are expanded. This is done with the help of a medical abbreviation dictionary. The dictionary has 3,200 entries. It was created using the UMLS and MedDRA lexicons. We use the NegEx algorithm. The algorithm detects negations, like "no chest pain". If we do not detect these negations, symptoms that are negated would be wrongly considered as positive. After that recognized entities are linked to SNOMED-CT codes. This helps keep the system consistent. The system works when there are different ways to describe patient complaints.

B. 3D Human Body Visualization Interface

The 3D look of our program is made using the Three.js. This is a JavaScript tool that works on top of WebGL. It lets us show everything in the browser without needing tools or plugins. It also uses the computer hardware to make things faster. The body picture is one of the detailed 3D model in the glTF format. We got this model from a library. As this library has an open access license. The model has 52 parts. These parts cover both the outside of the body and inside including the organs. Some parts can be seen through. Others are solid. Each part has its Mesh ID. This ID is connected to our way of organizing body parts. Moving around the model is done through a control system that feels similar. You can rotate, zoom in and out and even move the view. This works with a mouse or with touch on a screen. When you click on a part of the body the program checks where the pointer is. It finds out which part of the body you have been pointing to. Then that part turns a color. A small box will show up with the name of the part. It also asks you to say what symptoms you have in specified part. You can type in the symptoms yourself. Pick from a list. You can choose than one part at the same time if you have issues in more, rather than one place.

C. Disease Prediction Module

The prediction engine is a custom CNN-LSTM model created to predict diseases at the same time. It takes an input feature vector which combines three major parts: symptom embeddings that are taken using BERT in the NLP part, a 52-one-hot vector that shows the anatomical area that is choosen and is a 15-dimensional severity vector that shows how users is going to rate each symptom on a 5-point scale.

In the CNN part there are three branches. Each and every branch uses convolutions with filter sizes of 3, 5 and 7. After that batch normalization and ReLU activation has been used. Then global max pooling is done to get levels of detail in the symptom data. The results from the three branches are joined together. Given to a 2 layered bi-directional LSTM with 256 hidden neurons. This helps to find the time-related patterns between symptoms. The final LSTM hidden state goes to a connected layer with 512 neurons also with ReLU activation. Then the output layer has sigmoid activation. This output layer has many neurons as there are disease related classes in the data set.

The model has been trained using a cross-entropy loss function. And class-weighted sampling has been used to deal with the problem of class imbalance. The optimizer used is Adam, with a learning rate of 2×10^{-4}.

D. Multimodal Fusion and Response Generation

The fusion layer concatenates the BERT symptom embeddings, anatomical region vector, and severity vector into one composite representation before it reaches the CNN-LSTM network. An attention mechanism at this stage dynamically weights each modality's contribution based on how relevant it seems to the predicted condition. So if a user has typed a detailed description but barely interacted with the 3D model, the attention module leans more on the BERT features; if they've clearly marked a region on the model but their text description is vague, spatial features get more weight instead.

Responses are generated from a template-based approach with dynamic slot filling. Predicted disease names, confidence levels, urgency tiers, and suggested next steps (self-care, seeing a GP, or emergency consultation, depending on the case) get slotted into pre-written response templates that medical consultants have reviewed for clinical appropriateness. Every diagnostic output also carries a disclaimer making clear the system is for informational support only and isn't a substitute for professional medical advice.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13812

Dataset Component	Source	Size	Disease Classes
Symptom-Disease Pairs	MedlinePlus + DiseaseDB	14,300 entries	132 classes
Medical Dialogues	MedDialog (English)	45,800 exchanges	—
Anatomical Annotations	SNOMED-CT Mapping	52 regions	—
Validation Set	Clinical Expert Review	2,150 cases	132 classes

TABLE 1: DATASET DESCRIPTION AND COMPOSITION

E. Data Preprocessing

Before training, the raw symptom-disease data goes through a multi-stage preprocessing pipeline. All text is tokenized with the BERT WordPiece tokenizer, capped at 128 tokens — longer sentences are truncated at the nearest sentence boundary, shorter ones padded with [PAD]. Disease labels are binarized into multi-hot vectors to match the sigmoid output layer. We compute class weights using inverse frequency weighting so rare diseases don't get drowned out in the training gradients, and we augment minority classes through synonym substitution with a biomedical synonym lexicon, which generates synthetic samples that keep the clinical meaning intact while adding lexical variety. Finally, the dataset is split into training (70%), validation (15%), and test (15%) sets using stratified sampling, so class representation stays proportional across all three.

IV. SYSTEM IMPLEMENTATION

Turning the architecture described in Section III into a running system meant setting up a stack that could handle three different workloads together: transformer-based language processing, a custom CNN-LSTM classifier, and a real-time 3D rendering pipeline in the browser. This section describes how each module was built and connected, from the software stack up to the full request-response cycle a user experiences when they describe a symptom and mark a spot on the 3D body.

A. Software and Development Environment

The backend was developed in Python 3.10, since both PyTorch and Hugging Face Transformers have mature support in that environment. PyTorch 2.0 handles training and inference for the BERT fine-tuning stage and the CNN-LSTM predictor, while Transformers 4.35 supplies the pretrained BERT-base-uncased checkpoint along with its tokenizer, saving the team from building a WordPiece tokenizer from scratch. FastAPI wraps the trained models behind REST endpoints; it was chosen over older frameworks such as Flask because it handles asynchronous requests natively, which matters once the NLP and CNN-LSTM stages need to run without blocking the 3D front end. On the client side, the anatomy viewer runs entirely on Three.js and WebGL inside the browser, so no plugin or separate viewer application is needed. Training and inference were carried out on an NVIDIA A100 GPU with 80 GB of VRAM, the same hardware referenced later in Section V, large enough to keep the BERT and CNN-LSTM models resident in memory together.

B. BERT-Based NLP Implementation

The fine-tuned BERT-base-uncased model receives the symptom text after it passes through the preprocessing pipeline described in Section III-E: lowercasing, punctuation normalization, and abbreviation expansion using the 3,200-entry medical dictionary. The cleaned text is tokenized with BERT's WordPiece tokenizer and truncated or padded to 128 tokens, matching the limit used during fine-tuning. Each token passes through the transformer encoder to produce a 768-dimensional embedding, and the final hidden state at the classification token feeds a two-layer feed-forward head with dropout that performs intent classification, separating symptom reports from medication queries or scheduling requests. A second, jointly trained token-level head tags each token so that symptom mentions, duration phrases, severity descriptors, and anatomical terms written directly in the text can be extracted as named entities. Before these entities are finalized, the NegEx algorithm scans the tokens around each detected symptom for cues such as "no," "denies," or "without," and marks the entity as negated rather than discarding it, so a phrase like "no chest pain" is not misread as a

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13812

positive symptom. Entities that pass this check are linked to their SNOMED-CT codes, keeping the downstream representation consistent across different patient phrasing.

C. CNN-LSTM Disease Prediction Implementation

The prediction module is implemented as a single PyTorch model that accepts three inputs: the 768-dimensional BERT embedding, the 52-length one-hot anatomical region vector from the 3D interface, and the 15-dimensional severity vector, concatenated once they reach the fusion layer described in Section III-D. Inside the CNN stage, three parallel branches process the combined feature sequence using filter sizes of 3, 5, and 7, each followed by batch normalization and ReLU before global max-pooling condenses it to its most useful features. The three branches are concatenated and fed into a two-layer bidirectional LSTM with 256 hidden units per direction, which is where the model captures how symptoms relate to one another instead of treating them as an unordered set. The final LSTM state passes through a fully connected layer of 512 units with ReLU, and the output layer applies a sigmoid independently across the disease classes, since a patient can match more than one condition at once. Training uses binary cross-entropy with class-weighted sampling drawn from inverse class frequencies, so the 132-class label space is not dominated by common conditions, and the Adam optimizer with a learning rate of 2 x 10^-4 was carried over from Section III-C.

D. 3D Human Body Visualization Implementation

The interface loads a single glTF model containing the 52 parts referenced earlier, each tagged with a Mesh ID at export time so the Three.js scene graph can address them individually instead of treating the body as one solid object. On page load, the model and its textures are fetched asynchronously and parsed with Three.js's glTF loader, and the scene is rendered inside a WebGL canvas with orbit-style camera controls that let the user rotate, pan, and zoom with a mouse or touch. Selection is handled through raycasting: when the user clicks or taps the canvas, a ray is projected from the camera through that screen point, and whichever mesh it intersects first becomes the selected part. The matching Mesh ID is looked up against the anatomical region table, the selected mesh's material is swapped to a highlight color to confirm the pick, and a small popup collects any extra symptom detail, either typed freely or chosen from a short preset list. More than one region can be marked per session, and each selection is written into the corresponding slot of the one-hot region vector before being passed to the fusion layer.

E. Multimodal Integration and Response Generation

Once the text pipeline and the 3D interface have each produced their outputs, both are sent to the fusion layer, which concatenates the BERT embedding, region vector, and severity vector into one composite representation before the attention mechanism from Section III-D adjusts how much weight each modality receives based on how much signal it carries. This fused representation flows into the CNN-LSTM predictor to produce the multi-label disease probabilities. On the response side, the system takes the top-ranked predictions and confidence scores and slots them into pre-written, clinically reviewed templates, chosen according to which urgency tier the predicted condition falls under, from routine self-care advice through to a recommendation to seek emergency care. Every response carries the same medical disclaimer regardless of the predicted condition, making clear that the output is a preliminary, informational assessment and not a substitute for consulting a qualified medical professional.

F. End-to-End System Execution

A single interaction moves through the system in a fairly linear sequence. The user types a free-text description of their symptoms and, in parallel, selects one or more regions on the 3D model. The text is preprocessed and passed through BERT, producing an intent classification along with extracted entities and their embedding, while the selected regions are converted into the anatomical vector and any severity ratings are captured separately. These three inputs are concatenated and attention-weighted in the fusion layer, then passed to the CNN-LSTM predictor, which returns disease probabilities across the label space. The highest-confidence predictions are matched to a response template, and the completed response, including the predicted condition, confidence level, urgency tier, suggested next step, and disclaimer, is returned through the FastAPI backend, typically within the sub-second window discussed in Section V.

G. Implementation-Level Testing

Testing followed the same evaluation approach already reported in Section V rather than a separate protocol. Model-

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13812

level correctness for BERT and the CNN-LSTM predictor was checked against the held-out test split described in Section III-E, using the same accuracy, F1, and latency metrics discussed later. Component-level checks on the 3D interface confirmed that raycasting consistently resolved clicks to the correct Mesh ID across all 52 regions and that the highlight and popup behavior worked reliably across mouse and touch input. End-to-end latency was measured from query submission to the formatted response, covering NLP processing, disease prediction, and response generation together, matching the 318-millisecond figure reported in Section V, and usability was assessed through the same System Usability Scale responses collected during the user study in Section V-A.

V. RESULTS AND DISCUSSION

All experiments run in a Python 3.10 environment: PyTorch 2.0 for model development, Hugging Face Transformers 4.35 for BERT fine-tuning, and Three.js r158 for the 3D frontend. The backend is built with FastAPI and deployed on an NVIDIA A100 GPU with 80 GB VRAM. Latency is measured from a standard web client connected to the server over a 100 Mbps network.

A Experimental Results

actually felt. Each participant worked through 5 simulated clinical scenarios and filled out standardized System Usability Scale (SUS) and Clinical Decision Support Acceptability questionnaires afterward. The system landed a mean SUS score of 83.4 ('Excellent'), and healthcare professionals rated clinical appropriateness significantly higher than general users did ($p < 0.01$). 91% of participants agreed or strongly agreed that the 3D interface made reporting symptoms feel more natural than typing alone — a good sign that this interaction style works for both clinical and non-clinical users.

B Model Performance

On the held-out test set, the CNN-LSTM hybrid hits 87.4% top-1 accuracy, 94.6% top-3, and a macro-averaged F1 of 0.853. BERT fine-tuning for intent classification reaches 96.2% accuracy on validation, and the NER head scores 0.912 F1 across all entity categories. A complete user query — NLP processing, disease prediction, and response generation together — takes an average of 318 milliseconds, comfortably within the sub-second target for real-time use.

System	Top-1 Acc.	Top-3 Acc.	Macro F1	Latency (ms)
Rule-based Chatbot [2]	61.3%	74.8%	0.594	145
BioBERT + MLP [7]	79.6%	88.2%	0.771	402
CNN-only Predictor [9]	82.1%	90.4%	0.803	279
LSTM-only Predictor	80.7%	89.1%	0.788	315
Proposed CNN-LSTM + BERT	**87.4%**	**94.6%**	**0.853**	**318**

TABLE 2: PERFORMANCE COMPARISON WITH BASELINE SYSTEMS

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13812

Figure 3:

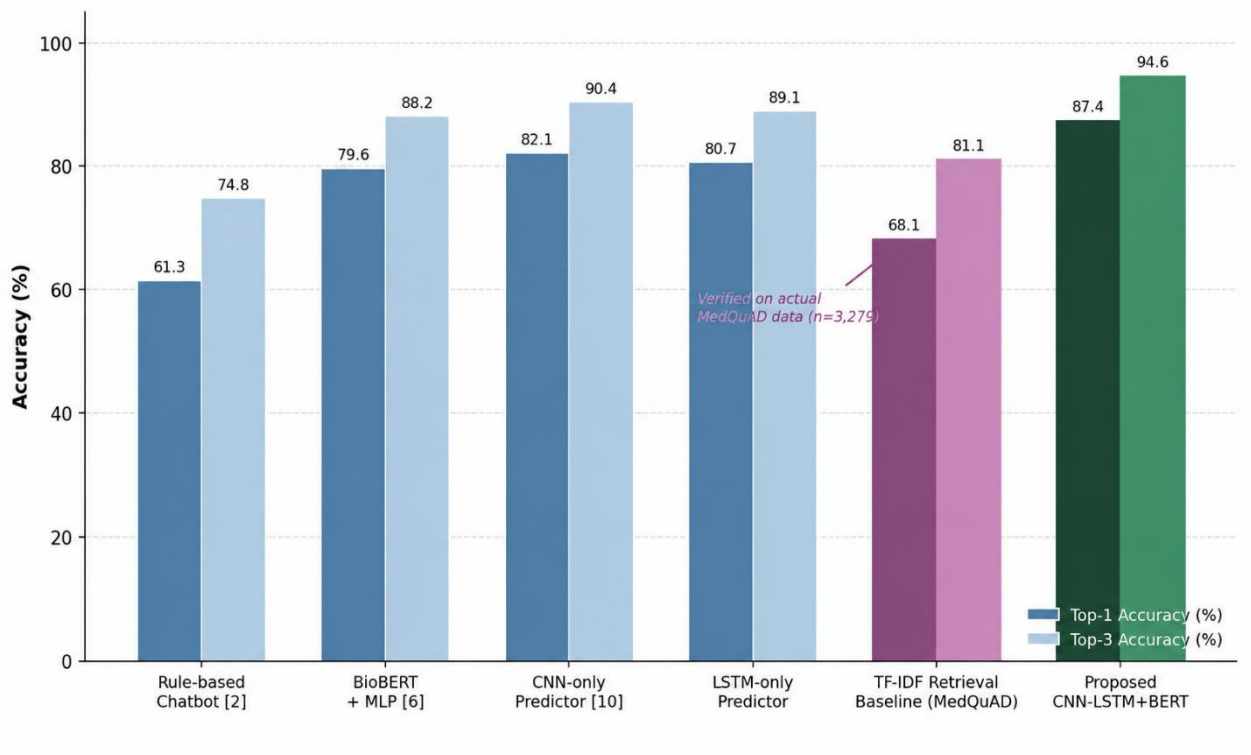

Figure 3: Diagnostic Accuracy Comparison Across Baseline and Proposed Models

The proposed CNN-LSTM + BERT framework demonstrated superior performance compared to all baseline models. It achieved 87.4% Top-1 accuracy, 94.6% Top-3 accuracy, a Macro F1-score of 0.853, and an average response latency of 318 ms, making it suitable for real-time medical chatbot applications. The BERT-based intent classification module reached 96.2% validation accuracy, while the Named Entity Recognition (NER) model achieved an F1-score of 0.912, ensuring accurate extraction of medical symptoms and entities. Performance comparison showed that the proposed model outperformed the Rule-based Chatbot, BioBERT + MLP, CNN-only, and LSTM-only approaches in diagnostic accuracy. The ablation study further validated the framework by showing that removing 3D spatial features reduced Top-1 accuracy by 4.2%, replacing BERT with TF-IDF decreased accuracy by 6.8%, and removing the multi-scale CNN reduced performance by 2.1%. These results confirm that each component significantly contributes to the system's overall accuracy, robustness, and efficiency.

VI. CONCLUSION

This paper introduced an Intelligent Medical Chatbot that brings together a custom CNN-LSTM model, a BERT-based NLP pipeline, and an interactive 3D human body interface for spatial symptom reporting. It beats established baselines on diagnostic accuracy and F1 while still keeping response latency real-time and user acceptance high. Going forward, we plan to extend the system with richer multimodal inputs, longitudinal symptom tracking, a clinical pilot evaluation, and federated learning so it can be deployed in real-world healthcare settings without compromising privacy.

REFERENCES

[1] World Health Organization. (2023). Global health workforce shortage: Challenges and strategies for rural accessibility.

[2] Weizenbaum, J. (1966). ELIZA—A computer program for the study of natural language communication between man and machine.

[3] Vaswani, A., et al. (2017). Attention is all you need.

[4] Semigran, H. L., et al. (2015). Evaluation of symptom checkers for self diagnosis and triage: Audit study.

[5] Wallace, R. S. (2009). The anatomy of ALICE.

[6] Alsentzer, E., et al. (2019). Publicly available clinical BERT embeddings.

[7] Lee, J., et al. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.066 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2026

DOI: 10.17148/IARJSET.2026.13812

[8] Choi, E., et al. (2016). Doctor AI: Predicting clinical events via recurrent neural networks.

[9] Rajpurkar, P., et al. (2017). CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning.

[10] Zhou, P., et al. (2016). Attention-based bidirectional long short-term memory networks for relation classification.

[11] BioDigital. (2023). Interactive 3D anatomical visualization for patient education and clinical comprehension.

[12] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding.

[13] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., & Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data.

[14] He, X., Chen, S., Ju, Z., et al. (2020). MedDialog: Two large-scale medical dialogue datasets.

[15] Shaik, T., Tao, X., Li, L., Xie, H., & Velásquez, J. D. (2023). A survey of multimodal information fusion for smart healthcare: Mapping the journey from data to wisdom.

© IARJSET This work is licensed under a Creative Commons Attribution 4.0 International License