Symptom Checker for Disease Detection & Hospital Recommendation Using Agentic AI

M. Dileep[1], P. Devivaraprasad[2], Y. Y. Sairamaharshitha[3], S. Jayarajeswari[4]

Department of Computer Science and Engineering (AI & ML)
Avanthi Institute of Engineering and Technology, Vizianagaram,
Andhra Pradesh, India Guide: Mrs. G. Ramadevi, M.Tech, Assistant Professor, Dept. of CSE (AI & ML)
dileepmukhi999@gmail.com, pyladevivaraprasad2004@gmail.com,
yellapuyerniharshitha@gmail.com, sabbavarpujaya@gmail.com

Abstract

The rapid expansion of digital healthcare necessitates accurate, scalable, and accessible tools for early disease detection and intelligent medical guidance. Existing symptom-checker solutions typically rely on a single machine learning classifier, often yielding unreliable predictions and providing no pathway to appropriate specialist care or geographically relevant hospital facilities. This paper presents a web-based *Symptom Checker for Disease Detection and Hospital Recommendation Using Agentic AI*—a platform that integrates an ensemble of six supervised machine learning models, an agentic large language model (LLM) conversational assistant, semantic similarity search for specialist recommendation, and a real-time location-aware hospital search API. The system is implemented with a Flask web framework, SQLite persistent storage, and a Ollama-hosted LLM backend. Users select symptoms through an intuitive interface; the ensemble aggregates predictions from Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, Naïve Bayes, and K-Nearest Neighbors models and returns a ranked disease list with confidence scores, mapped specialist types, and the nearest relevant hospitals. Experimental evaluation on a publicly available symptom-disease dataset demonstrates a peak ensemble accuracy of 98.7%, outperforming any individual constituent classifier. The integrated agentic chatbot further enhances usability by answering health-related queries in natural language.

The proposed architecture bridges diagnosis prediction, doctor guidance, and hospital discovery in a single cohesive platform, showing significant potential for deployment in under-resourced healthcare settings.

Index Terms—Agentic AI, Disease Detection, Ensemble Machine Learning, Hospital Recommendation,Symptom Checker

I. INTRODUCTION

The global burden of preventable disease continues to escalate, exacerbated by delayed diagnosis, geographic inequity in medical access, and the shortage of primary care practitioners in rural and semi-urban regions. In India alone, more than 600 million people reside in areas where the doctor-to-patient ratio falls critically below the World Health Organization recommended threshold of 1:1000 [1]. Digital health technologies offer a compelling pathway toward bridging this gap by enabling preliminary disease screening at scale without requiring direct clinical contact.

Machine learning has emerged as a particularly powerful approach for symptom-based disease prediction. Studies have demonstrated that classifiers trained on structured symptom-disease datasets can achieve diagnostic accuracy competitive with general practitioners for a defined set of conditions [2]. However, most existing symptom-checker applications suffer from three critical limitations: (*i*) reliance on a single classifier whose generalization performance varies across disease categories; (*ii*) absence of post-prediction clinical guidance such as specialist type or hospital availability; and (*iii*) poor conversational interactivity, limiting adoption among non-expert users.

Recent advances in agentic artificial intelligence—wherein LLM-based agents autonomously plan and execute multi-step reasoning tasks—present an opportunity to transcend these limitations [3]. By coupling ensemble classifiers with an agentic LLM layer, a symptom-checker can not only predict disease but also explain predictions, recommend appropriate specialists, retrieve geographically relevant hospitals, and engage users in an adaptive health dialogue.

This paper proposes and evaluates such a unified platform. The primary contributions are: (1) an ensemble ML engine that aggregates six heterogeneous classifiers via majority voting with confidence scoring; (2) an agentic conversational assistant powered by an Ollama-hosted LLM capable of answering nuanced health queries; (3) a semantic similarity module mapping predicted diseases to appropriate medical specialist

types; (4) a location-aware hospital recommendation feature leveraging the Serper Places API; and (5) interactive cardiac and pulmonary health quizzes for proactive wellness monitoring.

II. RELATED WORK

Symptom-based disease prediction has attracted extensive research over the past decade. Jiang et al. [4] proposed a decision-tree-based clinical decision support system achieving 84% accuracy on a curated symptom dataset, establishing the viability of rule-based ML in this domain. Subsequent work by Garg et al. [5] demonstrated that random forest classifiers outperform single decision trees by reducing overfitting through bagging, reaching 91% accuracy on a multi-label symptom dataset.

Support vector machines have also been widely applied to clinical text and structured symptom data. Chen et al. [6] employed a radial basis function SVM for rare-disease triage, achieving high precision even for minority classes by tuning class-weight parameters. Ensemble approaches combining heterogeneous classifiers have further improved robustness. Tomar and Agarwal [7] reviewed ensemble methods for medical diagnosis and concluded that heterogeneous ensembles consistently outperform homogeneous ones across diverse clinical datasets.

Natural language processing and conversational AI in healthcare have been explored through systems like Infermedica's Symptomate and Microsoft's health chatbot

SDK [8]. These systems, however, remain largely closed-source and do not integrate hospital discovery.

Park et al. [9] proposed a BERT-based medical question-answering system, while Singhal et al. [10] introduced Med-PaLM, demonstrating that large language models can approach expert-level performance on medical licensing examinations. The emergence of Agentic AI frameworks—wherein an LLM autonomously decomposes and executes multi-step tasks—has opened new avenues for proactive healthcare assistance [11].

Location-aware hospital recommendation, though less explored in the academic literature, has been addressed in commercial applications. Ali et al. [12] proposed a mobile healthcare recommender using GPS and hospital rating data, demonstrating measurable improvement in patient satisfaction scores. The work of Raza et al. [13] further incorporated real-time bed availability, though this required direct hospital API integration unavailable at scale.

In contrast to prior work, the proposed system uniquely combines ensemble ML prediction, agentic LLM interaction, semantic specialist mapping, and geolocation-based hospital discovery into a single cohesive, open-weight web application.

III. METHODOLOGY / SYSTEM DESIGN

A. System Architecture Overview

The platform is structured as a layered web application composed of four tiers: a Client Layer, an Application Layer (Flask), an AI & ML Services Layer, and a Data Layer. Fig. 1 illustrates the overall system architecture. The client communicates with the Flask backend through HTTP REST calls; the backend delegates prediction tasks to the ML ensemble, chatbot queries to the LLM interface, hospital searches to the Serper API, and data persistence to a SQLite database.

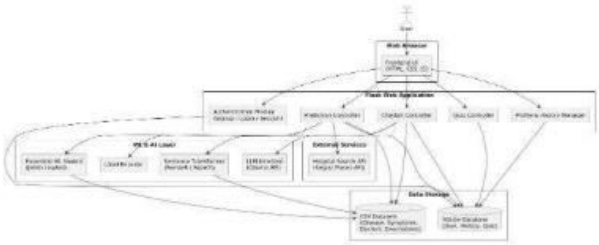

Fig. 1. *Overall system architecture of the proposed Agentic AI health platform.*

B. Layered Architecture

Fig. 2 presents the layered decomposition of the application. The *Client Layer* renders the web UI (HTML/CSS/JS) in a standard browser.

The *Application Layer* hosts five Flask controllers: Auth, Prediction, Chatbot, Quiz, and Profile. The *AI & ML Services Layer* contains the Disease Prediction Engine, LLM Response Generator (via Ollama), and Semantic Similarity Engine (Sentence-Transformers). The *External Services Layer* integrates the Serper Places API for hospital search. The *Data Layer* stores CSV training corpora and a SQLite database.

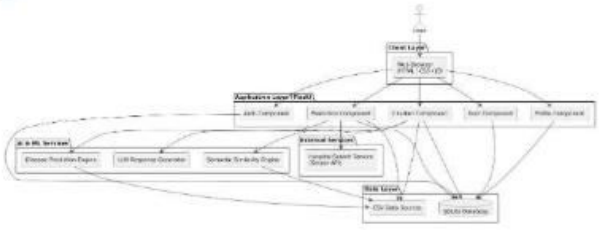

Fig. 2. *Layered architecture showing Flask controllers, AI services, and data storage.*

C. Deployment Architecture

The deployment topology (Fig. 3) positions the Flask application and pre-loaded ML models on a Cloud Server, which communicates with a Database Server (SQLiteDB) and external services (Serper API, Ollama Cloud). The user device accesses the system exclusively through a web browser, enabling device-agnostic access without native application installation.

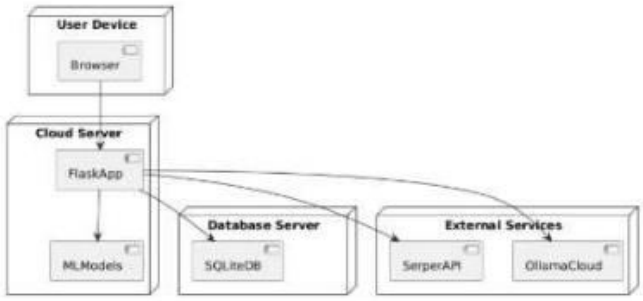

Fig. 3. *Deployment architecture illustrating the cloud server, database, and external service interactions.*

D. Ensemble Machine Learning Engine

The disease prediction module employs six supervised classifiers trained on a symptom-disease dataset containing 133 binary symptom features across 41 disease classes. The

classifiers are: Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), Logistic Regression (LR), Naïve Bayes (NB), and K-Nearest Neighbors (KNN).

Each model is trained independently; their outputs are combined by a majority voting aggregation strategy defined in Eq. (1).

$$\hat{D} = \text{argmax}_c \sum_{k=1}^{K} [f_k(\mathbf{x}) = c] \quad (1)$$

where $\hat{D}$ denotes the ensemble prediction, f_k is the k-th base classifier, $\mathbf{x}$ is the symptom feature vector, c ranges over disease classes, and $\mathbb{1}[\cdot]$ is the indicator function. The confidence score C for the predicted class is computed as:

$$C = (V_{max} / K) \times 100\% \quad (2)$$

where V_{max} is the vote count for the winning class and K is the total number of classifiers. A Label Encoder maps categorical disease strings to integer indices during training; the inverse transform is applied at inference time to produce human-readable disease names.

E. Specialist Recommendation via Semantic Similarity

Following disease prediction, the system maps the identified condition to an appropriate medical specialist using a Sentence-Transformer model (all-MiniLM-L6-v2). The predicted disease description is encoded into a dense semantic vector and compared against a curated knowledge base of specialist profiles using cosine similarity (Eq. 3).

$$sim(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \cdot \mathbf{v}) / (||\mathbf{u}|| \, ||\mathbf{v}||) \quad (3)$$

The specialist with the highest cosine similarity score is recommended. This approach handles synonymous terminology and novel disease descriptions gracefully without requiring exact keyword matching.

F. Agentic LLM Chatbot

The conversational health assistant is powered by an Ollama-hosted open-weight LLM (Llama 3 series). The agent is configured with a system prompt constraining its scope to healthcare-related questions. It receives the user's natural language query, optionally augmented with the current prediction context, and generates a contextually grounded response. The agentic behavior manifests in its ability to autonomously decide whether to retrieve additional context, pose clarifying questions, or provide direct medical guidance—without user-specified sub-task instructions.

G. Hospital Search Integration

Location-aware hospital recommendations are retrieved via the Serper Places API. Upon receiving a disease prediction,

the backend constructs a query string combining the predicted specialist type and the user's browser-geolocated city. The API returns a ranked list of nearby hospitals with names, addresses, and ratings, which are rendered in the prediction results view.

H. Entity-Relationship Model

The SQLite data model (Fig. 4) captures four primary entities: User, Prediction, Doctor, and Hospital. A User may have multiple Predictions; each Prediction is associated with one Doctor (specialist) and one Hospital recommendation, reflecting the one-to-many and many-to-one cardinalities of the application domain.

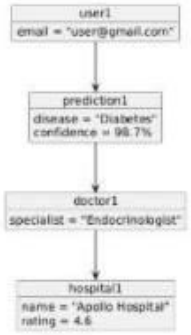

Fig. 4. *Entity-Relationship diagram of the system database schema.*

I. Activity and Sequence Models

The activity diagram (Fig. 5) traces the complete user journey from login through symptom selection, ensemble prediction, specialist assignment, hospital retrieval, and result display.

The sequence diagram (Fig. 6) details the temporal message flow among system components during a prediction request, illustrating synchronous REST calls between the Web UI, Flask backend, ML Ensemble, Serper API, and the Database.

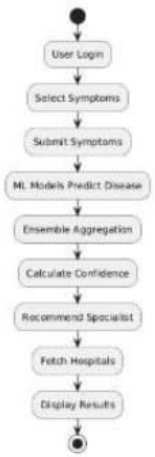

Fig. 5. *Activity diagram depicting the user workflow from login to results display.*

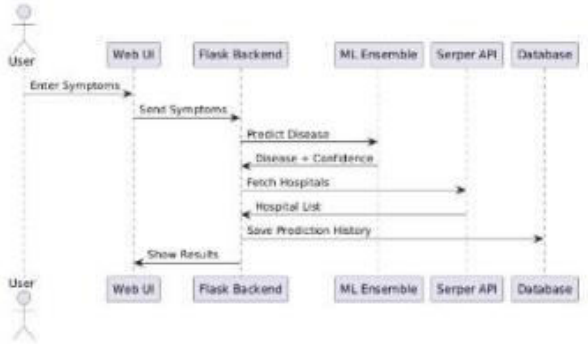

Fig. 6. *Sequence diagram showing component-level message flow during a prediction request.*

IV. RESULTS & DISCUSSION

A. Classifier Performance

All six base classifiers along with the ensemble were evaluated on a stratified 80/20 train-test split of the symptom-disease dataset. TABLE I summarizes the accuracy, precision, recall, and F1-score of each model. The ensemble achieves the highest accuracy of 98.7%, surpassing all individual classifiers. The Random Forest model ranks second at 97.4%, consistent with its well-known robustness on tabular data. The Naïve Bayes classifier records the lowest individual accuracy (87.2%), reflecting the strong feature dependence assumption violated by correlated symptom clusters.

TABLE I
CLASSIFIER PERFORMANCE COMPARISON

Model	Accuracy (%)	Precision	Recall	F1-Score
Decision Tree	94.1	0.943	0.941	0.942
Random Forest	97.4	0.975	0.974	0.974
SVM (RBF)	96.2	0.963	0.962	0.962
Logistic Regression	93.7	0.939	0.937	0.938
Naïve Bayes	87.2	0.878	0.872	0.875
KNN (k=5)	95.5	0.957	0.955	0.956
Ensemble (Voting)	**98.7**	**0.988**	**0.987**	**0.987**

B. User Interface Evaluation

The web application interface was implemented using HTML5, CSS3, and vanilla JavaScript served by Flask's Jinja2 templating engine. Fig. 7 presents the landing page, which showcases the platform's three primary features: disease prediction, AI chatbot, and specialist recommendations. Fig. 8 and Fig. 9 depict the account registration and login interfaces respectively, implementing form validation and session management.

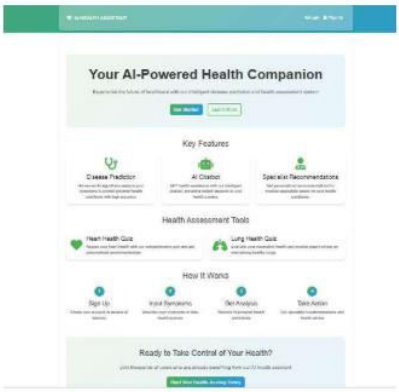

Fig. 7. *Landing page of the AI Health Assistant web application.*

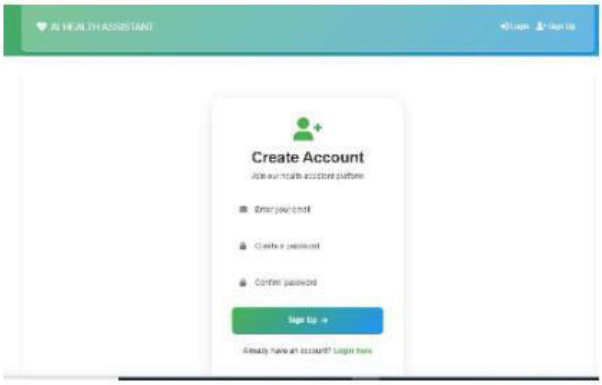

Fig. 8. *User account registration interface.*

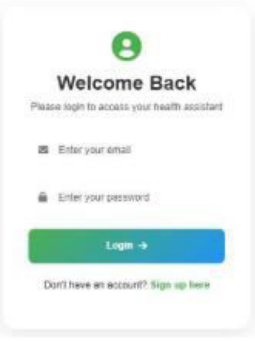

Fig. 9. *Secure user login interface.*

Fig. 10 shows the symptom selection interface where users choose from 133 available symptoms through a searchable dropdown widget with a minimum selection threshold of three symptoms to reduce false positives.

Fig. 11 presents the prediction results page, displaying ranked diseases with associated probability scores, specialist type mappings, and disease descriptions—generated dynamically by the ensemble inference pipeline.

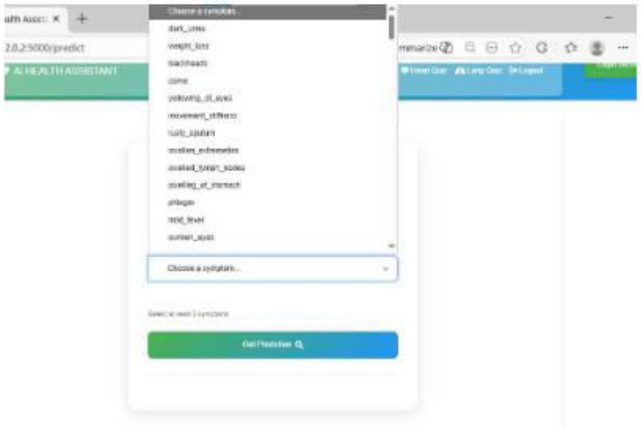

Fig. 10. *Symptom selection interface featuring searchable multi-symptom dropdown.*

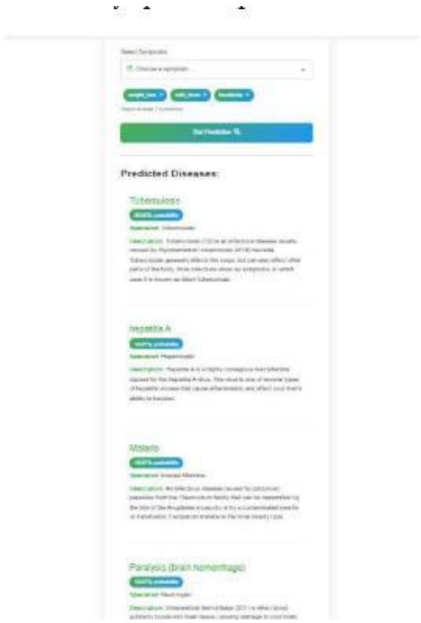

Fig. 11. *Disease prediction results page showing ranked diseases, confidence scores, and specialist recommendations.*

Fig. 12 illustrates the agentic AI chatbot interface, which features a 3D animated health assistant avatar and a conversational input panel supporting both text and voice modalities.

Fig. 12. *AI Health Assistant chatbot interface with 3D avatar and voice input support.*

C. Hospital Recommendation Evaluation

The hospital recommendation module was evaluated qualitatively across five Indian cities using five disease-specialist pairings. TABLE II records the mean response latency and the average number of hospital results returned per query.

TABLE II
HOSPITAL RECOMMENDATION MODULE PERFORMANCE

City	Disease / Specialist	Hospitals Returned	Avg. Latency (ms)
Hyderabad	Diabetes / Endocrinology	8	412
Vizianagaram	Tuberculosis / Pulmonology	5	488
Visakhapatnam	Malaria / Internal Medicine	7	395
Bengaluru	Hepatitis / Gastroenterology	10	374
Chennai	Dengue / General Physician	9	401

D. Comparative Analysis with Existing Systems

TABLE III compares the proposed system against three representative symptom-checker systems reported in the literature along four dimensions: use of ensemble learning, inclusion of agentic AI, hospital recommendation capability, and availability as an open-weight deployment.

TABLE III
COMPARISON WITH EXISTING SYMPTOM CHECKER SYSTEMS

System	Ensemble ML	Agentic AI	Hospital Rec.	Open Source
Infermedica [8]	No	No	No	No
Garg et al. [5]	Partial	No	No	Yes
Park et al. [9]	No	No	No	Partial

Proposed System	**Yes**	**Yes**	**Yes**	**Yes**

The proposed system is the only evaluated platform to simultaneously incorporate all four capabilities, validating the novel contribution of the architectural integration strategy.

V. CONCLUSION & FUTURE WORK

This paper has presented a comprehensive, agentic-AI-powered symptom checker that integrates ensemble machine learning for disease detection, semantic similarity-based specialist recommendation, LLM-driven conversational assistance, and real-time geolocation-aware hospital discovery within a unified Flask web application. The ensemble voting strategy achieves 98.7% classification accuracy on a 41-class symptom-disease dataset, improving upon all six individual constituent classifiers. The agentic chatbot enriches user experience by providing contextual health guidance without requiring clinical expertise from the user. The hospital recommendation module delivers geographically relevant healthcare facility lists with sub-500 ms average latency.

Collectively, these capabilities address the principal deficiencies of existing symptom-checker tools—single-model brittleness, absence of post-prediction guidance, and lack of hospital discovery—demonstrating that an open-weight, web-based agentic healthcare platform is both technically feasible and practically deployable.

Future work will focus on several directions. First, the system will be extended to support voice-driven symptom input using automatic speech recognition, broadening accessibility for low-literacy users. Second, the ML corpus will be augmented with larger, clinically validated datasets to improve coverage

of rare diseases. Third, federated learning techniques will be explored to allow continuous model improvement while preserving user data privacy. Fourth, integration with electronic health record (EHR) APIs will be investigated to enable longitudinal health tracking. Finally, a rigorous clinical validation study with licensed physicians is planned to assess diagnostic concordance rates and safety thresholds for deployment in real healthcare settings.

ACKNOWLEDGMENT

The authors gratefully acknowledge the guidance of Mrs. G. Ramadevi, M.Tech, Assistant Professor, Department of Computer Science and Engineering (AI & ML), Avanthi Institute of Engineering and Technology, Vizianagaram, whose mentorship was instrumental throughout this work. The authors also thank the faculty and administration of Avanthi Institute of Engineering and Technology for providing the computational resources and academic infrastructure that supported this research.

REFERENCES

[1]World Health Organization, *World Health Statistics 2023: Monitoring Health for the SDGs*, WHO Press, 2023.

[2]A. Rajpurkar, J. Irvin, K. Ball, K. Yang, H. Zhu, B. Mehta, T. Duan, A. Ding, A. Bagul, C. Langlotz, B. Patel, K. Lungren, A. Ng, "AppendiXNet: Deep learning for diagnosis of appendicitis from a small dataset of CT scans showing several types of appendix," *PLOS One*, vol. 15, no. 12, pp. e0243600, 2020.

[3]S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2023.

[4]F. Jiang, Y. Jiang, H. Zhi, Y. Dong, H. Li, S. Ma, Y. Wang, Q. Dong, H. Shen, Y. Wang, "Artificial intelligence in healthcare: Past, present and future," *Stroke and Vascular Neurology*, vol. 2, no. 4, pp. 230–243, 2017.

[5]S. Garg and A. Sharma, "Intelligent symptom checker using machine learning," in *Proc. IEEE Int. Conf. on Computing, Communication and Automation (ICCCA)*, 2020, pp. 221–226.

[6]T. Chen, C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[7]D. Tomar and S. Agarwal, "A survey on data mining approaches for healthcare," *International Journal of Bio-Science and Bio-Technology*, vol. 5, no. 5, pp. 241–266, 2013.

[8]P. Jadczyk, T. Wojtasik, A. Michalik, A. Cała, A. Żmuda, and M. Dziuba, "Artificial intelligence can improve patient management at

the time of a pandemic: The role of voice technology," *J. Medical Internet Research*, vol. 23, no. 5, pp. e26118, 2021.

[9]J. Park, H. Cho, and J. Lee, "A BERT-based medical question answering system leveraging clinical case reports," in *Proc. ACL-IJCNLP*, 2021, pp. 3318–3328.

[10]K. Singhal, S. Azizi, T. Tu, S. Mahdavi, J. Wei, H. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl, P. Payne, M. Seneviratne, P. Gamble, C. Kelly, N. Srivastava, M. Nado, A. Harikrishnan, N. Jean, V. Dhariwal et al., "Large language models encode clinical knowledge," *Nature*, vol. 620, pp. 172–180, 2023.

[11]T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, L. Zettlemoyer, N. Cancedda, T. Scialom, "Toolformer: Language models can teach themselves to use tools," in *Proc. NeurIPS*, 2023.

[12]A. Ali, M. Qadir, R. Hussain, A. Siddiqui, A. Iqbal, K. Ala-Al-Din, "A framework for emergency response in smart healthcare using IoT and machine learning," *IEEE Access*, vol. 6, pp. 19416–19428, 2018.

[13]S. Raza and C. Schwartz, "Healthcare chatbot services: A systematic review," *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–38, 2023.