International Journal of Creative and Open Research in Engineering and Management
ISSN: 3108-1754 (Online)
Volume 02 Issue 11 November-2026 | Impact Factor: 3.5



An Intelligent Multimodal and Multilingual Clinical Decision Support System Using LLM and RAG with Explainable Medical Insights

Gangadhar R
Dept. of Computer Science and Engineering RV Institute of Technology and Management Bengaluru, Karnataka, India
gangadharr414@gmail.com

M G Nikhil
Dept Computer Science and Engineering RV Institute of Technology and Management Bengaluru, Karnataka, India
mgnikhil9456@gmail.com

Mallikarjunayya S
Dept of Computer Science and Engineering RV Institute of Technology and Management Bengaluru, Karnataka, India
salimallu56@gmail.com

Gururaj S H
Dept Computer Science and Engineering RV Institute of Technology and Management Bengaluru, Karnataka, India
hiremathgururaj073@gmail.com

Mrs. Bhavya N Javagal
Assistant Professor
Dept. of Computer Science and Engineering
RV Institute of Technology and Management
Bengaluru, Karnataka, India
bhavyanj.rvitm@rvei.edu.in

How to Cite this Article:
R, G., Nikhil, M. G., S, M. & H, G. S. (2026). An Intelligent Multimodal and Multilingual Clinical Decision Support System Using LLM and RAG with Explainable Medical Insights. International Journal of Creative and Open Research in Engineering and Management, 02(11), 1-9. https://doi.org/10.55041/ijcope.v2i11.022

License:
This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.
© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.







https://doi.org/10.55041/ijcope.v2i11.022

***Abstract*--Modern healthcare environments generate heterogeneous clinical information, including patient history, symptoms, laboratory reports, and medical images. Managing these diverse data sources through separate systems can make it difficult to obtain a unified view of patient information and supporting medical evidence. This paper presents an intelligent multimodal and multilingual Clinical Decision Support System (CDSS) that integrates machine learning, deep learning, large language models (LLMs), and Retrieval-Augmented Generation (RAG) within a unified web-based application.**

The proposed system combines patient and clinical visit management, structured symptom-based screening using XGBoost, chest X-ray analysis using DenseNet121, and fracture-related image classification using EfficientNet-B0. Laboratory reports are processed using text extraction and configurable interpretation rules. A RAG pipeline based on Sentence Transformers and FAISS retrieves relevant information from an indexed medical knowledge base to support LLM-generated explanations and report-related question answering. The application also incorporates multilingual interaction, structured clinical report generation, authentication, appointment management, and aggregated GeoHealth analytics.

The integrated platform is designed to bring multiple clinical information-processing functions together in a single workflow, supporting the organisation and interpretation of patient information while providing accessible explanations of generated outputs. The system is intended to assist healthcare professionals rather than replace clinical judgment. Although the application integrates its principal functional modules, further systematic evaluation of model performance, retrieval quality, multilingual reliability, and clinical usefulness is required before clinical deployment.

Keywords: Clinical Decision Support System, Machine Learning, Deep Learning, Large Language Models, Retrieval-Augmented Generation, Multimodal Analysis, Medical Image Classification, Multilingual Healthcare.

I. INTRODUCTION

Healthcare organisations generate large volumes of heterogeneous data: patient-reported symptoms, clinical notes, laboratory measurements, vital signs, prescriptions and radiological images. Each source carries part of the clinical picture, yet they are frequently stored in separate systems and require different skills to interpret. Reviewing them together is time-consuming and makes a consistent patient overview hard to obtain, particularly in busy outpatient settings.

A Clinical Decision Support System (CDSS) is a computer-based system that assists healthcare professionals in organising patient information and evaluating possible clinical findings [17]. Machine Learning (ML) can detect patterns in structured data, Deep Learning (DL) can process selected medical im-



ages, Natural Language Processing (NLP) can handle textual clinical information, and Large Language Models (LLMs) can produce readable summaries and explanations [11], [18]. However, language models can generate fluent but incomplete or unsupported statements, a phenomenon documented as hallucination in the medical domain [15]. Retrieval-Augmented Generation (RAG) mitigates this by retrieving passages from configured knowledge sources and supplying them as context, so that a generated answer can be traced back to evidence [19]. A preliminary survey of this problem space was reported by the authors in [16].

Two further challenges motivate this work. First, patients often describe symptoms in a regional language while clinical records and medical knowledge sources are predominantly in English; a multilingual interface can reduce this barrier only if its language-processing components are implemented and validated. Second, AI outputs that give a prediction without context are hard for clinicians to evaluate, so interpretability and explicit communication of uncertainty are essential.

This paper presents *Clinical Intelligence OS*, a unified web-based prototype that brings these technologies into a single clinical workflow. It provides interfaces for secure login, patient registration, clinical visit entry, laboratory and image upload, AI-assisted analysis, report generation and report-based questions. It is designed as a decision-support tool, not an autonomous diagnostic system

A. Contributions

The main contributions of this work are as follows:

1. *Integrated system architecture*: A modular web-based architecture that brings together patient record management, multimodal clinical inputs, machine-learning-based screening, LLM-assisted explanations, RAG-based evidence retrieval, multilingual interaction, and structured reporting.
2. *Report-grounded clinical assistance:* A Clinical Report Assistant designed to answer questions using the selected clinical report and relevant retrieved medical evidence, helping users relate generated explanations to the available information.
3. *Safety-oriented decision support:* A design that distinguishes preliminary screening outputs from confirmed diagnoses, communicates uncertainty where appropriate, and positions the system as an aid to professional clinical review rather than a replacement for clinical judgment.
4. *Aggregated GeoHealth analytics:* A regional analytics view intended to support awareness of aggregated health trends while avoiding the display of individually identifiable patient information.
5. *Functional integration and evaluation framework:* An implementation that brings the principal clinical information-processing modules into a unified workflow, accompanied by a testing and evaluation framework for assessing functional behaviour, model performance, retrieval quality, and multilingual reliability.

B. Organisation of the Paper

The remainder of this paper is organised as follows. Section II reviews the background and relevant literature. Section III presents a comparative analysis of related work. Section IV identifies the research gaps, and Section V states the research objectives. Section VI describes the proposed methodology and system architecture. Section VII discusses the implementation. Sections VIII, IX, and X describe the RAG and LLM pipeline, medical-image analysis, and security design, respectively. Section XI presents the testing and evaluation methodology, while Section XII discusses the results. Sections XIII, XIV, and XV present the limitations, conclusion, and future scope, respectively.

II. Background and Literature Review

A. Clinical Decision Support Systems

Clinical Decision Support Systems (CDSSs) have evolved from rule-based alerting mechanisms to data-driven tools that support clinical assessment, triage, medication safety, and care coordination. Their potential benefits include improved access to relevant clinical information, support for guideline adherence, and assistance in organising complex patient data. However, documented challenges include alert fatigue, over-reliance on automated recommendations, and inadequate integration into clinical workflows [17]. These challenges highlight the importance of keeping healthcare professionals involved in decision-making and making the basis of system-generated recommendations accessible for review.

B. Machine Learning for Structured Clinical Screening

Tree-based ensemble methods, including Random Forest [20] and gradient-boosting algorithms such as XGBoost [21], are widely studied for structured clinical prediction tasks. These methods can capture nonlinear relationships among input variables, although their effectiveness depends on dataset size, feature quality, class distribution, missing-data handling, and the evaluation procedure. Reliable assessment requires appropriate validation using data not used during model training, along with transparent reporting of limitations. Model-generated screening outputs should not be interpreted as confirmed diagnoses without appropriate clinical validation. Studies combining structured prediction models with LLM-based components, including research on chronic-disease prediction [9], illustrate the broader direction of integrating predictive models with language-based interfaces. In the proposed system, XGBoost is used for structured symptom-based screening.

C. Deep Learning for Medical Image Analysis

Convolutional Neural Networks (CNNs) learn hierarchical visual representations and can be adapted to medical-image classification through transfer learning. DenseNet121 connects each layer to subsequent layers within a dense block, allowing feature maps from preceding layers to be reused and helping improve feature propagation [22]. DenseNet-based architectures have been applied to chest-radiograph classification, including CheXNet [24], which was developed using the ChestX-ray14 dataset [25]. EfficientNet-B0 uses compound scaling of network depth, width, and input resolution to balance model complexity and



computational requirements [23]. These architectural properties make it suitable for consideration in image-classification applications, although performance depends on the training data and evaluation methodology. Reviews of LLMs for medical imaging [10] and multimodal LLMs in healthcare [5] describe growing interest in combining visual analysis with language-based interaction. The proposed system uses DenseNet121 for chest X-ray analysis and EfficientNet-B0 for fracture-related image classification.

D. Large Language Models in Healthcare

Large Language Models (LLMs) can support clinical-record summarisation, question answering, and the generation of explanations in accessible language. Singhal *et al.* demonstrated that language models can encode substantial medical knowledge while also exhibiting limitations that require careful evaluation [18]. Domain-adapted models such as GatorTron [13] and Meditron-70B [14] investigate the use of clinical text and medical knowledge in model development. Surveys of medical LLMs [11], [12] discuss their applications, reliability challenges, and research directions. The Med-HALT benchmark [15] examines hallucination and reasoning failures in medical LLM applications, highlighting the need to evaluate generated responses carefully. General-purpose models, including LLaMA [26] and subsequent releases [27], are available through different deployment and hosted-inference options. The specific LLM and inference provider used by the proposed system should be identified according to its actual backend configuration.

E. Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) combines information retrieval with text generation. Rather than relying exclusively on information encoded in a model's parameters, a RAG pipeline retrieves relevant passages from an external or indexed knowledge corpus and supplies them as context for response generation [19]. In healthcare, RAG has been explored for clinical copilots using knowledge graphs [1], multimodal electronic health record analysis [2], reducing clinician information overload [8], and structured RAG development and evaluation [4]. Surveys further summarise developments in this area [3].

Resources such as PubMed and MedlinePlus can provide medical information for a knowledge base, whereas PubMedQA [28] is a biomedical question-answering dataset. These resources have different roles and should be described accordingly. The usefulness of a RAG system depends on the relevance and reliability of retrieved passages, the coverage of the knowledge base, and the extent to which generated responses remain faithful to the retrieved information. Consequently, retrieval relevance, source traceability, and the handling of unsupported claims are important evaluation considerations. In the proposed system, Sentence Transformers and FAISS are used to support retrieval from an indexed medical knowledge base.

F. Multilingual Clinical Information Processing

Patients may describe symptoms using colloquial expressions or regional languages that differ from standard clinical terminology. Multilingual Natural Language Processing (NLP) can support language identification, expression normalisation, translation, and mapping of patient descriptions to representations suitable for downstream processing. However, translation errors can alter clinically meaningful details, including symptom duration, laterality, and negation. Therefore, each supported language should be evaluated using representative examples. The reliability of multilingual clinical processing depends on the actual language-processing pipeline and its interaction with downstream prediction and explanation components. A language-selection interface alone does not establish the accuracy or reliability of multilingual functionality.

G. Explainability and Clinical Safety

Explainable Artificial Intelligence (XAI) aims to make model behaviour and outputs more understandable to users. Depending on the model and task, feature-attribution methods [29] and visual-saliency approaches [30] can help indicate which input features or image regions influenced a prediction. Retrieved medical references can also provide textual context for LLM-generated explanations. However, an explanation does not guarantee that a prediction is correct, and confidence scores from neural networks may be poorly calibrated [31]. Such scores should not automatically be interpreted as the probability that a patient has a particular disease. Clinical decision-support applications should communicate limitations and uncertainty and should not present preliminary model outputs as definitive diagnoses. The specific explainability methods implemented in a system should be distinguished from methods discussed in the broader literature.

H. Multimodal and Conversational Clinical Systems

Recent research has explored LLM-based multimodal clinical chatbots [6], adaptive multi-agent systems for clinical decision support [7], and broader applications of LLMs in healthcare [11]. These approaches investigate different aspects of clinical assistance, including conversation, retrieval, medical-image interpretation, and decision support. Integrating these capabilities into a coherent clinical workflow presents engineering and evaluation challenges. The proposed *Clinical Intelligence OS* brings together patient information management, structured symptom screening, medical-image classification, laboratory report processing, retrieval-supported explanations, and report generation within a unified web-based application. Its contribution is the integration of these functions into a common workflow. Any claim of superiority over existing systems would require a systematic comparison using clearly defined criteria and supporting evidence.

III. Related Work and Comparative Analysis

Table I summarises the studies cited in this paper according to their research themes, approaches, and relevance to the proposed system. The comparison is qualitative and reflects the scope of the cited works. Numerical performance results are not directly compared because the studies may use different datasets, prediction tasks, evaluation metrics, and



experimental protocols.

The reviewed literature covers several important areas, including clinical decision support, structured machine learning, medical-image analysis, large language models, retrieval-augmented generation, and multimodal healthcare applications. These studies provide a foundation for developing systems that combine clinical data processing with language-based interaction and evidence retrieval. However, the extent to which existing approaches support a complete workflow—from patient intake and multimodal data processing to report generation, professional review, and regional analytics—must be assessed against the individual studies included in Table I. The review also highlights the importance of evaluating the relevance of retrieved evidence and the reliability of generated explanations, particularly in healthcare applications where unsupported statements may have significant consequences.

IV. Identified Research Gaps

The literature review identifies the following challenges that motivate the development of the proposed Clinical Intelligence OS.

A. Fragmented Multimodal Information

Patient information may include symptoms, vital signs, laboratory results, clinical history, and medical images. When these inputs are processed through separate tools or workflows, obtaining a consolidated patient overview can become difficult. A unified system should organise these different data types and associate them with the appropriate patient and clinical visit to support consistent information management.

B. Unsupported Generated Statements

LLMs can generate fluent and understandable responses that may nevertheless contain incomplete or unsupported information. Retrieval-Augmented Generation can help address this challenge by supplying relevant passages from a configured knowledge base as context for answer generation. Systems should preserve the association between generated explanations and retrieved evidence where retrieval and source attribution are implemented. However, retrieval alone does not guarantee that the generated response is correct or fully supported by its sources.

C. Superficial Multilingual Support

Multilingual clinical interaction involves more than providing a language-selection control. Patient input must be processed appropriately in each supported language, while clinically important details such as symptom duration, negation, and laterality must be preserved. Representative evaluation is therefore necessary to assess whether language-processing components function reliably across the languages claimed by a system.

D. Presentation and Interpretation of Model Outputs

Machine-learning and deep-learning models can produce screening classifications and confidence scores that may be misinterpreted as definitive clinical conclusions. Systems should distinguish preliminary screening outputs from confirmed diagnoses, explain the limitations of model-generated results, and make clear that clinical decisions require appropriate professional assessment. Confidence values should not automatically be interpreted as calibrated disease probabilities.

E. Integrated Workflows and System Evaluation

Combining patient intake, clinical data processing, medical-image analysis, laboratory report interpretation, evidence-supported explanations, and report generation within a single application presents both integration and evaluation challenges. A useful prototype should be assessed through functional testing, including workflow completion, correct association of information with patient records, error handling, and appropriate access controls. Model performance, retrieval quality, multilingual reliability, and clinical usefulness require separate evaluation procedures. Successful integration of software modules does not, by itself, establish clinical validity.

F. Population-Level Awareness and Privacy

Aggregated regional health information can support awareness of population-level trends and may help inform further investigation by healthcare and public-health professionals. However, such analytics should minimise the disclosure of personal information through appropriate aggregation, access controls, and privacy safeguards. Anonymisation and aggregation methods should be documented and evaluated rather than assumed to eliminate every possibility of re-identification.

V. Research Objectives

The primary objective of this work is to develop an intelligent multimodal and multilingual Clinical Decision Support System (CDSS) that integrates Machine Learning (ML), Deep Learning (DL), Large Language Models (LLMs), and Retrieval-Augmented Generation (RAG) within a unified web-based application. The specific objectives are as follows:

1. To develop a web-based interface for patient registration, authentication, and clinical visit management.
2. To organise patient symptoms, vital signs, medical history, and other available clinical information in a structured manner.
3. To provide interfaces for uploading and processing laboratory reports and supported medical images.
4. To integrate an XGBoost-based screening component for structured symptom-based clinical assessment.
5. To implement medical-image classification for chest X-rays using DenseNet121 and fracture-related images using EfficientNet-B0.
6. To develop a RAG workflow that retrieves relevant information from an indexed medical knowledge base to support LLM-generated explanations and report-related question answering.



7. To provide multilingual interaction for the languages supported by the implemented language-processing components.
8. To generate structured clinical reports that consolidate available patient information and AI-assisted screening outputs.
9. To integrate a Clinical Report Assistant that enables users to ask questions about a selected report and receive contextually relevant responses.
10. To incorporate appropriate authentication, privacy safeguards, evidence traceability where supported, uncertainty communication, and professional clinical review into the system's design.

The objectives focus on developing and integrating the system's functional components. The clinical validity, model performance, retrieval reliability, and multilingual accuracy of the resulting application require systematic evaluation and should not be inferred from functional integration alone.

VI. PROPOSED METHODOLOGY AND SYSTEM ARCHITECTURE

A. Overall Methodology

Clinical Intelligence OS follows a modular methodology that integrates patient-data management, clinical prediction, medical-image analysis, laboratory report processing, Retrieval-Augmented Generation (RAG), Large Language Model (LLM)-based assistance, and clinical report generation within a unified web application.

The system follows a client–server architecture. The frontend provides the interface for patient registration, clinical data entry, file uploads, analysis, and report review. The backend receives requests through API endpoints and directs them to the appropriate processing modules. The outputs are returned to the frontend and consolidated into the clinical workflow where applicable.

B. System Architecture

The architecture consists of a presentation layer, backend API layer, clinical processing layer, knowledge retrieval and language layer, and report-generation layer. The presentation layer manages user interaction, while the backend coordinates data processing and access to the relevant modules. Structured clinical information is processed using XGBoost, chest X-rays using DenseNet121, and fracture-related images using EfficientNet-B0. Laboratory reports are processed through text extraction and configurable interpretation rules.

The RAG component retrieves relevant passages from the configured medical knowledge collection using sentence embeddings and FAISS. The retrieved context is supplied to the configured LLM service to support natural-language explanations and question answering. The available patient information and processing outputs can then be organised into a structured clinical report.

Table I. Technologies Used in the Proposed System

Layer	Technology	Purpose
Frontend	React.js, JavaScript, Vite	User interface and interaction
Backend	Python, FastAPI, Uvicorn	API handling and module coordination
Authentication	JWT	Authentication for protected API routes
Structured prediction	XGBoost	Clinical feature-based prediction
Chest X-ray analysis	DenseNet121	Chest X-ray classification
Fracture image analysis	EfficientNet-B0	Fracture-related image classification
Text embedding	Sentence Transformers (all-MiniLM-L6-v2)	Text vector representation
Vector retrieval	FAISS	Similarity search over the indexed knowledge collection
Knowledge files	medical_index.faiss, medical_docs.jsonl	Vector index and associated document metadata
Language-model assistance	Configured LLM service	Context-assisted explanations and question answering

C. Workflow and Data Flow

The system workflow begins with patient registration or record selection, followed by clinical visit entry and the upload of supported documents or images. The backend validates and routes the submitted information to the relevant processing module. The resulting outputs are presented through the interface and may be included in the generated report. For questions requiring medical information retrieval, the RAG component searches the indexed knowledge collection and supplies relevant passages to the LLM.

Table II. Proposed System Workflow

Stage	**Processing activity**	**Output**
Patient intake	Validate and register patient information	Patient record
Clinical visit	Record symptoms and available clinical details	Visit record
File upload	Accept supported laboratory reports and images	Uploaded-file information



Preprocessing	Prepare structured inputs, extracted text, or images	Prepared inputs
Clinical analysis	Execute the relevant prediction model or laboratory rules	Module-specific outputs
Evidence retrieval	Search the indexed medical knowledge collection	Relevant passages
Explanation	Generate a response using the available context and configured LLM	Natural-language explanation
Report generation	Consolidate available clinical information and results	Structured clinical report
Professional review	Review the report and available findings	Information for clinical assessment

Not every stage is required for every request. The stages executed depend on the selected workflow and the availability of relevant information.

D. Patient Data Management

The application organises patient records, clinical visits, uploaded files, and analysis outputs to support information retrieval and report generation. Associating the available information with the appropriate patient and visit is important for maintaining a coherent clinical record.

The conceptual entities below illustrate the information relationships relevant to the proposed system. Their names and fields should be checked against the deployed database schema before being described as the exact implementation.

Table III. Conceptual Data Entities

Entity	Purpose
User	Stores user identity and authentication-related information
Patient	Maintains patient records
Clinical Visit	Associates symptoms and visit details with a patient
Uploaded File	Organises uploaded laboratory reports and medical images
Screening Result	Associates model outputs with the relevant clinical record
Evidence Record	Represents retrieved passages and their metadata
Generated Report	Organises available clinical information and analysis outputs
Appointment	Supports appointment-related information

E. Clinical Prediction and Medical-Image Analysis

The structured clinical prediction module uses XGBoost to process supported clinical features and generate preliminary prediction outputs. The input information is prepared in the format expected by the trained model, and the resulting output is returned to the application for display and review.

For medical-image analysis, DenseNet121 is used for chest X-ray classification, while EfficientNet-B0 is used for fracture-related image classification. Each model processes its supported image input according to the preprocessing requirements of the trained classifier and returns a model-generated output.

These components enable structured clinical information and selected medical images to be processed within the same application. However, the generated predictions and image classifications should be treated as preliminary computational outputs rather than confirmed diagnoses. Their clinical reliability must be established through appropriate evaluation on labelled datasets.

F. Laboratory Report Processing

The laboratory report processing module accepts supported reports and extracts or organises the available information using the implemented text-processing workflow. Configurable interpretation rules are applied to the supported laboratory tests where the required values and reference ranges are available.

The extracted information can contribute to the consolidated clinical view and report-generation process. Its reliability depends on document quality, extraction accuracy, and the coverage of the configured rules. The original report and appropriate clinical guidance remain important when interpreting the results.

G. Retrieval-Augmented Generation and LLM Assistance

The RAG component supports medical-information retrieval and natural-language question answering. It uses the Sentence Transformers model sentence-transformers/all-MiniLM-L6-v2 to represent text as dense vector embeddings. FAISS searches the configured vector index, while medical_docs.jsonl stores associated document metadata.

When a user submits a supported question, the system generates a query embedding and searches the index for relevant passages. The retrieved context is then supplied to the configured LLM service, which generates a natural-language response. The application also supports questions relating to available clinical reports.

The relevance and reliability of the response depend on the quality of the indexed knowledge base, the relevance of the retrieved passages, and the language model's behaviour. RAG can provide contextual support but does not guarantee that every generated statement is correct. Responses must therefore be reviewed appropriately and should not replace



professional medical judgment.

H. Multilingual Interaction

The system is designed to support interaction in English, Hindi, and Kannada. This functionality aims to make supported application interactions more accessible to users with different language preferences.

The quality of multilingual interaction depends on the implemented language-processing components and their handling of medical terminology. Translation errors or ambiguous symptom descriptions may affect interpretation; therefore, support for a language should not be taken as proof of equivalent performance across languages.

I. Clinical Report Generation

The report-generation component organises the information available for a selected patient and visit together with the outputs of the applicable processing modules. Depending on the available information, the report can include a clinical summary, preliminary screening findings, laboratory information, image-analysis outputs, explanatory content, safety messages, and doctor-review status.

The report provides a consolidated view of the information processed by the application. It is intended to support review and communication, not to independently confirm a diagnosis.

J. Authentication and Security

Authentication and protected API access are incorporated into the application architecture. The frontend attaches an authentication token to requests that require access control, and the backend validates the token before permitting access to protected routes.

This approach separates backend access-control responsibilities from frontend navigation. Additional considerations for handling clinical information include input validation, access permissions, secret management, confidentiality, error handling, and the protection of information sent to external services. The use of an external LLM service with clinical text requires particular attention to applicable data-protection and data-handling requirements.

K. Functional Integration and Evaluation

Functional integration and model performance are separate evaluation concerns. Functional testing checks whether supported workflows—such as authentication, patient management, file processing, prediction requests, retrieval, and report generation—operate as intended.

Model-level evaluation requires separate experiments. Predictive models should be assessed using appropriate labelled test data and relevant metrics, such as precision, recall, F1-score, and accuracy. RAG should be evaluated for retrieval relevance and consistency between generated answers and retrieved evidence. Multilingual performance requires representative test queries in each supported language.

Only metrics obtained from documented experiments should be reported as measured results. The successful operation of integrated modules does not by itself establish clinical validity or suitability for real-world deployment.

L. Summary

The proposed methodology integrates structured clinical prediction, medical-image classification, laboratory report processing, vector-based knowledge retrieval, LLM assistance, multilingual interaction, and report generation within a unified web application. The workflow and supporting tables describe how the main components interact and how relevant information is organised. Further systematic evaluation is required to establish the reliability, safety, and clinical usefulness of the generated outputs.

VII. IMPLEMENTATION

A. Implementation Overview

Clinical Intelligence OS was implemented as a web-based clinical decision support prototype that combines patient record management, clinical data entry, medical file processing, machine-learning-based screening, medical image analysis, retrieval-augmented generation (RAG), and clinical report generation. The application provides a unified interface through which users can enter patient information, submit clinical inputs, review analysis outputs, and obtain an organised report for professional review.

The implementation separates the user interface from backend processing and model-related operations. This modular design allows individual components to perform their specific functions while supporting an integrated clinical workflow. The system is intended to assist healthcare professionals in reviewing clinical information and does not replace professional diagnosis or treatment decisions.

B. Frontend Implementation

The frontend was developed using React.js and Vite, with reusable interface components for patient registration, clinical visits, file uploads, analysis results, report generation, and the clinical report assistant. Axios is used for communication between the frontend and backend APIs, while React Markdown supports the presentation of formatted explanatory responses where applicable.

The interface organises the major functions into separate views so that patient information and analysis outputs can be accessed through a consistent workflow. Input forms collect relevant demographic information, symptoms, vital signs, and other available clinical details. The application also provides interfaces for viewing imaging results, reading generated reports, and interacting with the report assistant.



Fig 1: Secure login interface.



Fig 2: Patient registration or patient details interface

C. Backend and API Implementation

The backend was implemented using Python and FastAPI, with Uvicorn used to run the application during development. The backend coordinates requests from the frontend, validates incoming data, manages application workflows, and connects the relevant processing components.

API-based communication separates the presentation layer from the processing layer. When a user performs an operation, the frontend sends the corresponding request to the backend. The backend processes the request through the relevant application logic and returns a response for display in the interface. This structure supports modular development and makes it easier to maintain the different clinical functions.

D. Patient and Clinical Visit Management

The patient management module provides an interface for entering and accessing patient details. Clinical visit management records information associated with an encounter, including symptoms, symptom duration, severity, vital signs, and other available clinical inputs. Uploaded files and generated outputs are associated with the relevant patient or visit through the application's data-management workflow.

Maintaining these associations is important because the screening results and generated report must correspond to the correct clinical encounter. The application presents patient and visit information alongside the relevant outputs, helping users review the information within its clinical context.



Fig 3: Clinical visit form showing the available symptom and clinical-input fields

E. Laboratory Report Processing

The laboratory report module accepts supported laboratory documents for processing. Depending on the input format and the configured extraction pipeline, text or laboratory values can be extracted and prepared for subsequent processing. Extracted information can then be used by the applicable analysis and report-generation components.

The processing workflow depends on the quality and readability of the uploaded document, the supported file format, and the availability of relevant laboratory fields. Therefore, extracted values should be checked against the original report before being used for clinical interpretation. The system's output is intended to support review rather than establish a diagnosis independently.

F. Machine-Learning-Based Clinical Screening

The structured-data screening component uses XGBoost to process supported clinical and symptom-related inputs. The model produces screening outputs based on the features available to it. These outputs are presented as preliminary decision-support information and should be interpreted in conjunction with the patient's clinical context and professional assessment.

The implementation also includes medical-image analysis components for selected image types. DenseNet121 is used for the chest X-ray analysis component, while EfficientNet-B0 is used for the bone-fracture classification component. These models address different image-analysis tasks and should not be interpreted as interchangeable classifiers.

For chest X-ray analysis, the application can display model-generated findings and associated confidence scores. Such scores represent model outputs; unless calibration and clinical validation have been performed, they should not be interpreted as calibrated probabilities of disease. Similarly, fracture-classification outputs require appropriate clinical interpretation and independent evaluation before being used to support clinical decisions.

G. Retrieval-Augmented Generation and Clinical Report Assistant

The RAG component is designed to support explanatory responses by retrieving relevant passages from an indexed



medical knowledge collection and supplying them as context to a language model. The retrieval pipeline uses Sentence Transformers with the all-MiniLM-L6-v2 embedding model and a FAISS vector index. The indexed document metadata is maintained in a JSON Lines file.

For a user question, the retrieval component identifies relevant indexed content that can be used to contextualise the response. The language model then generates a readable explanation based on the available context and the application's configured prompting workflow. This approach is intended to improve the relevance of responses by connecting generated explanations to retrieved information. It does not, by itself, guarantee that every response is factually correct or fully supported by the retrieved evidence.

The clinical report assistant provides a question-and-answer interface for users to ask about information in a selected report. Example questions may concern the meaning of reported findings, the interpretation of available laboratory information, or the reasons for discussing particular results with a healthcare professional.

The exact language-model provider and external medical sources should be identified from the deployed configuration. Only sources that are actually included in the running knowledge base should be described as active retrieval sources.



Fig 4: Clinical report assistant showing a real question and response associated with a report

H. Clinical Report Generation

The report-generation module organises available patient information, clinical inputs, screening outputs, imaging findings, explanatory content, and applicable safety messages into a structured report. The report interface allows users to review the generated information and, where supported, download the report as a PDF.

The generated document is intended to improve the readability and organisation of the available information. It should clearly distinguish patient-provided details, model-generated outputs, extracted laboratory information, and explanatory text. A doctor-review status or confirmation message communicates that professional assessment remains necessary.



Fig 5 : Final clinical report interface.

I. Authentication and Access Control

The application uses token-based authentication for protected operations. The frontend stores the authentication token under the cdss_token key in local storage and uses an Axios interceptor to attach the token as a Bearer token to applicable API requests. The backend contains JWT validation logic in backend/app/core/deps.py, which is used to validate authentication for protected routes.

This separation allows the frontend to send authenticated requests while the backend enforces access to protected resources. The security of the complete workflow also depends on token expiry and handling, credential storage, authorisation checks, secret management, transport security, and appropriate protection of patient information. These properties should be reported as verified safeguards only where they have been implemented and tested.

J. Module Integration

The application brings the major user-facing and analysis functions into a common workflow. Patient information and visit inputs provide the context for the relevant processing operations, while the file-processing, machine-learning, image-analysis, retrieval, and report-generation components produce outputs for review through the interface.

Table V summarises the principal implementation modules and their roles.

Table V. Implemented Modules and Their Responsibilities

Module	Implementation or role
User interface	React.js, Vite, and reusable application views
Backend API	FastAPI endpoints and workflow coordination
Authentication	JWT-based authentication for protected API routes
Patient and visit management	Collection and organisation of patient and encounter information
Structured clinical screening	XGBoost-based prediction from supported clinical features
Chest X-ray	DenseNet121-based image analysis



analysis	
Fracture classification	EfficientNet-B0-based image classification
Laboratory processing	Supported document extraction and preparation of available laboratory information
Evidence retrieval	Sentence Transformers embeddings and FAISS vector search
Language-model assistance	Contextual explanation using the configured language-model provider
Report generation	Structured clinical summary and PDF output
Clinical report assistant	Question-and-answer interface for report-related queries

K. GeoHealth Analytics

GeoHealth , a public-health surveillance view: a map-like area with coloured indicators for locations or condition groups, and cards showing regional categories, condition groups and the number of anonymised cases in a selected period. It is intended to help researchers and public-health professionals understand regional trends without track-ing individual patients; accuracy depends on the availability, quality and representativeness of the underlying data.



Fig 6: GeoHealth public-health surveillance view using anonymised, aggre-gated condition groups.

L. Implementation Summary

The implementation provides a common web interface for organising patient information, entering clinical details, processing supported medical inputs, viewing model-generated outputs, retrieving contextual information, and generating structured reports. The modular architecture supports the integration of conventional machine-learning models, deep-learning-based image analysis, and language-model-assisted explanations within a single application.

Although the working prototype demonstrates the intended application workflow, its clinical reliability must be assessed separately through appropriate datasets, documented evaluation procedures, security testing, and review by qualified healthcare professionals. The following section evaluates the system's functionality and discusses the scope and limitations of the observed results.

VIII. RESULTS AND DISCUSSION

A. Functional Results

The Clinical Intelligence OS prototype brings multiple clinical-support functions into a unified web application. The implemented workflow supports patient information management, clinical visit entry, medical file submission, AI-assisted analysis, clinical report generation, and report-related question answering. The application also provides interfaces for multilingual interaction and additional functions such as GeoHealth analytics and appointment management.

The main functional outcome is the availability of these capabilities within a common application rather than as separate standalone demonstrations. The interface organises patient information and analysis outputs in a way that supports review of the available clinical information. The generated report provides a structured presentation of the available findings and includes a doctor-review status.

The prototype demonstrates an integrated application workflow. However, functional operation should be distinguished from the clinical validity of the models, the reliability of generated explanations, and the effectiveness of the system in a real healthcare setting.

TABLE X. OBSERVED APPLICATION OUTPUTS

Component	Observed Output
Login	Email and password fields, role selection, and language controls
Patient Intake	Patient details form and patient identifier display
Clinical Visit	Fields for symptoms, severity, duration, vital signs, and laboratory information
File Upload	Upload controls for laboratory reports and medical images
AI Analysis	Fusion-engine interface and review-priority labels
Medical Image Analysis	Chest X-ray findings accompanied by model-generated confidence scores
Clinical Report	Structured summary, possible conditions, recommended checks, and safety alerts
Clinical Report Assistant	Suggested questions and free-text input for report-related queries



Doctor Review	Pending review status and confirmation-required message
GeoHealth Analytics	Geographic healthcare analytics interface and displayed healthcare-related information
Appointments	Appointment-related navigation and booking interface

B. Results of the Major Modules

The structured-data screening module uses XGBoost to process supported clinical features and produce preliminary screening outputs. This component provides a machine-learning-based view of the entered clinical information. Its usefulness depends on the relevance and completeness of the input features, the representativeness of the training data, and independent evaluation of the model.

The medical-image analysis components address two different tasks: chest X-ray analysis using DenseNet121 and bone-fracture classification using EfficientNet-B0. Their outputs provide additional information for the clinical review workflow. The confidence scores displayed by the image-analysis component indicate model outputs and should not be interpreted as calibrated probabilities without supporting calibration evidence.

The laboratory-processing module accepts supported documents and prepares available information for downstream use. The RAG-based assistant is designed to combine report context with passages retrieved from the configured medical knowledge index. This provides a mechanism for producing explanations that are informed by retrieved material rather than relying exclusively on the language model's internal knowledge.

The report-generation module brings relevant patient details and available analysis results into a structured document. Together, these modules illustrate how structured clinical information, medical images, retrieved knowledge, and generated explanations can be organised within a common decision-support workflow.

C. Analysis of the Generated Clinical Report

The generated clinical report is an important output because it consolidates information from the application into a format that can be reviewed by a healthcare professional. The sample report includes a clinical summary, possible conditions, imaging findings, suggested follow-up checks, safety messages, and a doctor-review status.

The sample report displays the following chest X-ray findings and confidence values.

Table VI. Chest X-Ray Findings Displayed in the Sample Report

Imaging finding	**Displayed confidence (%)**
Lung opacity	75.8
Pleural effusion	63.0
Edema	50.0
Cardiomegaly	39.9

The displayed values illustrate how multiple model outputs can be presented together to support review. They do not establish that a finding is present in a patient. Clinical interpretation requires consideration of the original image, the patient's history, the model's limitations, and assessment by an appropriately qualified professional.

The report's doctor-review status also supports the intended human-in-the-loop approach. The generated document is intended to organise preliminary information for review, not to independently establish a diagnosis or determine treatment.



Fig 6 : The actual generated clinical report, showing the findings and doctor-review status.

D. Discussion of Retrieval-Augmented Generation

RAG provides a structured approach for connecting a user question with relevant material in a medical knowledge collection. In the project, sentence embeddings generated with all-MiniLM-L6-v2 and a FAISS vector index support semantic retrieval from the indexed documents. Retrieved passages can then be supplied as context for language-model-assisted explanations.

This architecture is useful because a report-related question can be considered alongside the available report context and retrieved information. For example, a user may ask for clarification of a reported finding or the meaning of a recommended follow-up check. The assistant is designed to make such information easier to understand through a conversational interface.

Nevertheless, the presence of a retrieval component does not guarantee that every response is supported by the retrieved passages. Retrieval relevance, source quality, completeness, factual consistency, and the handling of missing evidence



must be evaluated independently. Quantitative claims about retrieval precision, mean reciprocal rank, or language-model factuality should therefore be reported only after the corresponding tests have been conducted.

E. Multimodal and Multilingual Interaction

The system's multimodal design brings together structured clinical information, laboratory documents, and medical images within a single application workflow. Each input type contributes a different form of information: structured fields provide explicitly entered clinical data, laboratory documents provide extracted measurements where supported, and images provide inputs for the relevant image-analysis models.

The intended multilingual interface aims to make clinical information more accessible to users who prefer different languages, including English, Hindi, and Kannada. However, the availability of language-selection controls alone is not sufficient to establish that medical meaning is preserved across all supported languages. A rigorous assessment should test representative symptom descriptions, medical terminology, negation, symptom duration, and translated explanations in each language.

Accordingly, the multimodal architecture and available language interface demonstrate the system's intended breadth, while quantitative claims about multilingual effectiveness require a separate evaluation.

F. Functional Integration and Reliability

The integration of the frontend, backend, data-management functions, analysis components, and report-generation workflow provides a common environment for accessing the system's major capabilities. The modular organisation also makes it possible to identify the role of each component and evaluate its contribution separately.

Reliable clinical support depends not only on the successful execution of individual modules but also on correct association of outputs with the relevant patient and visit, appropriate handling of unavailable inputs, controlled error handling, and clear presentation of uncertainty. The application should therefore be evaluated through repeatable end-to-end tests in addition to demonstrations of individual screens.

Functional testing and clinical evaluation address different questions. Functional testing establishes whether a workflow behaves as expected. Clinical evaluation assesses whether model outputs and generated explanations are sufficiently accurate, reliable, and safe for their intended use. Both are important, but successful functional integration does not itself establish clinical readiness.

G. Limitations of the Observed Results

The results demonstrate the capabilities of the working application prototype, but several limitations remain relevant to interpreting its outputs.

First, the predictive performance of the XGBoost, DenseNet121, and EfficientNet-B0 models must be established using appropriate labelled datasets and documented evaluation procedures. The confidence values displayed by the application are not substitutes for measurements such as sensitivity, specificity, precision, recall, F1-score, and ROC-AUC.

Second, the RAG and language-model components require systematic evaluation for retrieval relevance, evidence consistency, unsupported claims, readability, and safe handling of questions for which supporting information is unavailable. The reliability of the output may depend on the quality and coverage of the indexed knowledge collection.

Third, the multilingual functionality requires language-specific assessment to determine whether translated inputs and explanations preserve clinical meaning. Finally, privacy, access control, and the handling of potentially sensitive clinical information must be evaluated before the prototype is used with real patient data in a clinical environment.

These limitations concern the evidence required to establish clinical reliability; they do not imply that the application's functional modules are not operating.

H. Overall Discussion

Clinical Intelligence OS demonstrates an approach to integrating patient and visit management, structured clinical screening, medical-image analysis, retrieval-supported explanations, and structured reporting in a web-based clinical decision-support application. Its principal contribution is the organisation of these functions within a common workflow, allowing different forms of clinical information and model-generated outputs to be reviewed together.

The report-generation and report-assistant components provide a way to present preliminary findings in a more accessible format, while the doctor-review status reinforces the need for professional assessment. At the same time, the displayed outputs must be interpreted cautiously until the predictive models, retrieval pipeline, language-model responses, and multilingual functions have undergone appropriate evaluation.

The prototype therefore provides a foundation for further technical and clinical assessment. Future work should prioritise reproducible model evaluation, systematic testing of evidence-grounded explanations, multilingual assessment, and privacy and safety testing before any claim of clinical effectiveness is made.

IX. LIMITATIONS

Although the proposed Clinical Decision Support System integrates structured clinical prediction, medical-image analysis, laboratory report processing, Retrieval-Augmented Generation (RAG), multilingual interaction, clinical report generation, and GeoHealth Analytics within a unified web-based application, several limitations must be considered.

A. Need for Clinical Validation

The system is designed to support healthcare professionals by organising clinical information and providing preliminary AI-assisted insights. However, functional implementation



alone does not establish diagnostic accuracy or clinical effectiveness. The machine-learning models require systematic evaluation using appropriate, representative, and independently held-out datasets. Further validation by qualified healthcare professionals is necessary before the system can be considered for practical clinical use.

B. Limitations of Medical Image Analysis

The performance of chest X-ray and bone fracture classification depends on the quality, diversity, and representativeness of the images used for model development. Variations in image acquisition, image quality, patient populations, and disease presentation may affect prediction reliability. Model-generated confidence scores should not be interpreted as calibrated probabilities or definitive diagnoses unless supported by appropriate validation and calibration studies.

C. Limitations of LLM-Based Responses

The LLM-based assistant can generate explanations and answer questions using retrieved medical information and available clinical context. Nevertheless, generated responses may contain inaccurate statements, omit relevant details, or misinterpret the retrieved evidence. Human review remains important, particularly when responses concern symptoms, potential conditions, or clinical recommendations.

D. Scope and Maintenance of the Medical Knowledge Base

The RAG component depends on the coverage, quality, and freshness of the documents indexed in its medical knowledge base. If relevant evidence is missing, outdated, or poorly represented, retrieval may produce incomplete or less relevant context. Regular maintenance, source verification, and systematic evaluation of retrieval relevance and answer faithfulness are therefore necessary.

E. Reliability of Multilingual Processing

The system is designed to support multilingual interaction, including English, Hindi, and Kannada. However, medical terminology, regional expressions, spelling variations, and differences in sentence structure can affect interpretation and translation. Language-specific evaluation is required to determine whether symptoms, negation, duration, and other clinically important details retain their intended meaning across supported languages.

F. Laboratory Report Processing

Laboratory report processing depends on the quality and structure of uploaded files, the accuracy of text extraction, and the rules used to interpret supported laboratory parameters. Scanned documents, unclear text, inconsistent layouts, and differences in units or reference ranges may affect extraction and interpretation. The module's outputs should therefore be reviewed against the original laboratory report and appropriate clinical reference ranges.

G. Privacy, Security, and Deployment Considerations

The application incorporates authentication and protected API routes to support access control. However, deployment in real healthcare environments requires broader security assessment, secure handling of sensitive patient information, appropriate data-retention policies, and protection of information transmitted to external services. Compliance with applicable healthcare and data-protection requirements must also be assessed for the intended deployment environment.

H. Limitations of GeoHealth Analytics

GeoHealth Analytics provides a geographical perspective on healthcare-related information within the application. The usefulness of its outputs depends on the availability, accuracy, completeness, and update frequency of the underlying geographical data. Incomplete facility information or outdated records may limit the reliability of location-based insights. The module should not be interpreted as guaranteeing real-time facility availability or access to healthcare services unless these capabilities are independently verified.

I. Generalisability and Future Evaluation

The integrated architecture demonstrates how multiple clinical information-processing modules can be brought together within a common application. However, the reliability of the complete workflow must be evaluated across different patient populations, input formats, languages, and operating conditions. Future evaluation should include documented model-performance metrics, retrieval-quality measurements, response-faithfulness assessment, security testing, and review by healthcare professionals.

In summary, these limitations identify the need for further validation and evaluation rather than negate the functional integration of the proposed system. The CDSS is intended to assist clinical decision-making by presenting organised information and preliminary AI-generated insights; it is not intended to replace qualified healthcare professionals or independently determine definitive diagnoses and treatment decisions.

X. CONCLUSION

This paper presented Clinical Intelligence OS, an integrated multimodal and multilingual clinical decision-support prototype that combines patient record management, structured visit entry, laboratory and medical-image upload, machine-learning-based screening, LLM-based explanation, Retrieval-Augmented Generation (RAG), structured reporting with safety alerts and doctor-review status, a report-grounded assistant, and anonymised geographical analytics. The prototype demonstrates the integration of these functionalities into a unified web-based application and provides a foundation for accessible, evidence-informed clinical decision support.

The integration of machine-learning models, medical-image analysis, laboratory report processing, and retrieval-grounded explanations enables the system to organise diverse clinical information and present preliminary insights in a structured format. Multilingual interaction and GeoHealth



Analytics further extend the system's intended accessibility and healthcare information support capabilities.

Although the implemented workflows demonstrate the system's functional capabilities, further systematic evaluation is required to establish model accuracy, image-analysis reliability, multilingual performance, retrieval quality, and the factual consistency of LLM-generated responses. The system is intended to assist qualified healthcare professionals by organising clinical information and presenting preliminary AI-generated insights; it does not replace professional judgement or independently establish definitive diagnoses or treatment decisions.

XI. FUTURE SCOPE

Although Clinical Intelligence OS integrates multiple clinical decision-support functionalities into a unified application, several opportunities remain to improve its clinical reliability, explainability, accessibility, and practical applicability.

A. Explainable Artificial Intelligence

The interpretability of machine-learning predictions can be improved by incorporating explainable AI techniques. SHapley Additive exPlanations (SHAP) could be investigated for explaining structured-data predictions, while Gradient-weighted Class Activation Mapping (Grad-CAM) could be explored to highlight image regions that contribute to chest X-ray or bone-fracture classification outputs. These explanations would require validation to ensure that they accurately represent model behaviour and are useful to healthcare professionals.

B. Enhancement of Retrieval-Augmented Generation

The RAG pipeline could be extended using a larger, regularly updated collection of authoritative medical resources. Improvements to document chunking, semantic retrieval, reranking, and source attribution may help retrieve more relevant evidence for clinical questions. Future versions could also identify insufficient or conflicting evidence and communicate uncertainty rather than generate unsupported responses.

C. Evaluation of LLM-Generated Explanations

A systematic evaluation framework could be developed to assess the factual correctness, relevance, completeness, readability, and evidence support of LLM-generated explanations and clinical summaries. Additional safeguards could be investigated to reduce hallucinations, distinguish retrieved evidence from model-generated content, and ensure that recommendations remain within the intended scope of clinical decision support.

D. Improved Multilingual Capabilities

The multilingual interface could be extended and evaluated across English, Hindi, Kannada, and additional regional languages. Future evaluation should examine translation quality, medical terminology preservation, speech recognition accuracy, and the consistency of clinical explanations across languages. Particular attention should be given to ensuring that translation does not alter clinically significant details such as symptoms, medication names, units, or test results.

E. Advanced Laboratory Report Processing

Laboratory report processing could be enhanced to support a broader range of report layouts, test names, units, reference intervals, and document formats. Improved OCR, structured data extraction, and validation rules may help handle variations in scanned and digital reports. Future work could also investigate methods for flagging missing values, ambiguous units, and results that require professional review.

F. Security, Privacy, and Interoperability

Further development could strengthen role-based access control, encryption, audit logging, secure secret management, data-retention policies, and backup and recovery procedures. Integration with electronic health record systems and established healthcare data-exchange standards could also be explored to support secure information sharing between compatible healthcare applications, subject to appropriate permissions and regulatory requirements.

G. Expansion of GeoHealth Analytics

The GeoHealth Analytics module could be enhanced using verified, anonymised, and regularly updated geographical healthcare data. Future functionality may include regional summaries of healthcare resources, identification of underserved areas, and visualisation of healthcare-related patterns. The reliability of these analyses would depend on data completeness, freshness, geographical coverage, and appropriate privacy safeguards.

In summary, future development of Clinical Intelligence OS will focus on more transparent model outputs, stronger evidence retrieval, improved multilingual processing, enhanced privacy protection, and reliable integration with healthcare workflows. These extensions could strengthen the system's usefulness as a clinical decision-support tool while retaining the requirement for qualified healthcare professionals to make final diagnostic and treatment decisions.

XII. REFERENCES

[1] X. Zhao, S. Liu, S.-Y. Yang, and C. Miao, “MedRAG: Enhancing Retrieval-Augmented Generation with Knowledge Graph-Elicited Reasoning for Healthcare Copilot,” in *Proceedings of the ACM Web Conference 2025*, pp. 4442–4457, 2025, doi: 10.1145/3696410.3714782.

[2] Y. Zhu, C. Ren, S. Xie, S. Liu, H. Ji, Z. Wang, T. Sun, L. He, Z. Li, X. Zhu, and C. Pan, “REALM: RAG-Driven Enhancement of Multimodal Electronic Health Records Analysis via Large Language Models,” *arXiv preprint arXiv:2402.07016*, 2024.

[3] R. AlSaad *et al*., “Multimodal Large Language Models in Health Care: Applications, Challenges, and Future Outlook,” *Journal of Medical Internet Research*, vol. 26, Art. no. e59505, 2024, doi: 10.2196/59505.



[4] Y. Ke *et al*., "Development and Testing of Retrieval Augmented Generation in Large Language Models—A Case Study Report," *arXiv preprint arXiv:2402.01733*, 2024.

[5] Q. Liu *et al*., "A Review of Applying Large Language Models in Healthcare," *IEEE Access*, vol. 13, pp. 6878–6892, 2025.

[6] Ç. U. Öğdü, K. Arslanoğlu, and M. Karaköse, "An Adaptive Multi-Agent LLM-Based Clinical Decision Support System Integrating Biomedical RAG and Web Intelligence," *IEEE Access*, vol. 13, pp. 167390–167404, 2025, doi: 10.1109/ACCESS.2025.3613340.

[7] A. Shetgaonkar *et al*., "Mitigating Clinician Information Overload: Generative AI for Integrated EHR and RPM Data Analysis," in *Proceedings of the IEEE International Conference on Computers, Software, and Applications (COMPSAC)*, 2025.

[8] H. M. Serugunda *et al*., "Using Large Language Models for Chronic Disease Management Tasks: Scoping Review," *JMIR Medical Informatics*, vol. 13, Art. no. e66905, 2025, doi: 10.2196/66905.

[9] L. Liu *et al*., "A Survey on Medical Large Language Models: Technology, Application, Trustworthiness, and Future Directions," *arXiv preprint arXiv:2406.03712*, 2024.

[10] X. Yang *et al*., "A Large Language Model for Electronic Health Records," *npj Digital Medicine*, vol. 5, Art. no. 194, 2022, doi: 10.1038/s41746-022-00742-2.

[11] Z. Chen *et al*., "MEDITRON-70B: Scaling Medical Pretraining for Large Language Models," *arXiv preprint arXiv:2311.16079*, 2023.

[12] A. Pal, L. K. Umapathi, and M. Sankarasubbu, "Med-HALT: Medical Domain Hallucination Test for Large Language Models," in *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 314–334, 2023, doi: 10.18653/v1/2023.conll-1.21.

[13] R. T. Sutton *et al*., "An Overview of Clinical Decision Support Systems: Benefits, Risks, and Strategies for Success," *npj Digital Medicine*, vol. 3, Art. no. 17, 2020, doi: 10.1038/s41746-020-0221-y.

[14] K. Singhal *et al*., "Large Language Models Encode Clinical Knowledge," *Nature*, vol. 620, pp. 172–180, 2023, doi: 10.1038/s41586-023-06291-2.

[15] P. Lewis *et al*., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[16] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[17] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016, doi: 10.1145/2939672.2939785.

[18] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017, doi: 10.1109/CVPR.2017.243.

[19] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, pp. 6105–6114, 2019.

[20] P. Rajpurkar *et al*., "CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning," *arXiv preprint arXiv:1711.05225*, 2017.

[21] X. Wang *et al*., "ChestX-ray8: Hospital-Scale Chest X-Ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2097–2106, 2017, doi: 10.1109/CVPR.2017.369.

[22] H. Touvron *et al*., "LLaMA: Open and Efficient Foundation Language Models," *arXiv preprint arXiv:2302.13971*, 2023.

[23] A. Dubey *et al*., "The Llama 3 Herd of Models," *arXiv preprint arXiv:2407.21783*, 2024.

[24] Q. Jin, B. Dhingra, Z. Liu, W. W. Cohen, and X. Lu, "PubMedQA: A Dataset for Biomedical Research Question Answering," in *Proceedings of EMNLP-IJCNLP*, pp. 2567–2577, 2019, doi: 10.18653/v1/D19-1259.

[25] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, vol. 30, pp. 4765–4774, 2017.

[26] R. R. Selvaraju *et al*., "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017, doi: 10.1109/ICCV.2017.74.

[27] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On Calibration of Modern Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, vol. 70, pp. 1321–1330, 2017.