International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

A Machine Learning Application for Cattle Disease Detection using Multimodal RAG and LLM

Shriram Masalekar
Ajeenkya D Y Patil School of Engineering, Pune.

Samruddhi Paiyawal
Ajeenkya D Y Patil School of Engineering, Pune.

Prof. Ashwini Shinde
Ajeenkya D Y Patil School of Engineering, Pune.

Shreya Jain
Ajeenkya D Y Patil School of Engineering, Pune.

Avishkar Jagtap
Ajeenkya D Y Patil School of Engineering, Pune.

Abstract - Cattle disease diagnosis is still quite challenging for developing countries since late detection and unavailability of skilled veterinarians results in huge economic losses in those regions. Conventional ML methods employed for cattle disease diagnosis are often reliant on massive labeled databases and supervised learning processes, which are often difficult to acquire under resource-constrained agricultural conditions. This work presents CattleCare AI, a complete multimodal disease diagnosis system that leverages RAG in combination with LLMs in order to provide cattle disease diagnoses in real time without requiring any pre-trained models or any labeled database.

The developed solution supports multimodal inputs in the form of selected clinical features by farmers along with the optional images showing symptoms of diseases observed in cattle. The proposed multimodal solution utilizes three-stage RAG pipeline to retrieve disease information from our curated veterinary knowledge base containing details on 15 diseases, contextually augment queries by providing grounded medical information, and generate diagnoses based on the information retrieved via Google Gemini 2.5 Flash. The architecture of the developed platform includes a React-based progressive web application, serverless backend functions, an intelligent multilingual bot in English, Hindi, and Marathi languages, vaccine reminder section, and translation of the diagnostic report. Experiments performed with 20 test cases show that the accuracy of diagnosis achieved by RAG-augmented approach amounts to 91%, while zero-shot diagnosis based on LLMs achieves 82% accuracy, and conventional CNN-based models only 78%.

Keywords: Multimodal AI, Retrieval-Augmented Generation, Large Language Models, Cattle Disease Detection, AI in Agriculture, Veterinary Informatics, Knowledge-Grounded Reasoning, Serverless Architecture

II. INTRODUCTION

India is home to the largest number of cows globally, with the total cow population surpassing 305 million, contributing significantly to agriculture through milk production, draught work, and organic farming. Nevertheless, the country lacks an adequate framework to manage cattle diseases due to low availability of veterinary expertise in rural areas, remoteness of diagnostic laboratories, and inability among farmers to diagnose cattle diseases at an early stage. As per the report published by the Department of Animal Husbandry and Dairying (2023), India loses about INR 20,000 crores annually due to livestock diseases, with FMD, LSD, and Mastitis being the leading causes.

In recent times, traditional methods of cattle disease diagnosis have utilized supervised machine learning techniques, including CNNs for imaging-based identification and ensembles such as Random Forest and SVMs for symptom-based prediction. Although these algorithms yield excellent results in controlled experimental conditions, several fundamental limitations have hindered their application in practical scenarios. Specifically, traditional computational algorithms depend on large annotated datasets, which are resource-intensive and expensive to compile. Moreover, these systems are unable to generalize their results to other diseases not present in the training dataset. In addition, such algorithms lack interpretability in generating diagnoses and fail to provide actionable treatment plans.

However, the recent development of generative language models (LLMs) such as GPT-4, Gemini, and LLaMA promises unprecedented opportunities for medical AI applications. First, LLMs can perform zero-shot reasoning, understand multiple modalities, and generate high-quality natural language outputs. Nonetheless, the main drawback associated with LLMs is that they tend to generate false yet plausible statements known as hallucinations, which could be dangerous in medical diagnosis applications.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

To overcome this problem, Lewis et al. suggested introducing retrieval augmented generation (RAG), whereby the output generated by LLMs is grounded in retrieved facts extracted from curated knowledge bases.

The contributions of this research are as follows:

- Design and deployment of an end-to-end multimodal retrieval-augmented generation (RAG) system for cattle disease diagnosis, which can be fed symptom data as well as images.
- Creation of a knowledge base for veterinarians containing details on 15 cattle diseases along with metadata such as symptoms, disease severity levels, treatment, and preventive measures.
- Implementation of a web application named CattleCare AI that includes an AI chatbot service, vaccine alerts, multilingual support (English, Hindi, Marathi), and report generation.
- Experimental results validating an accuracy of 91% via the RAG technique, thus showing a gain of 9% compared to zero-shot LLM reasoning and 13% as compared to CNN methods.
- Formal mathematical formulation of the symptom similarity scoring system used for retrieving from the knowledge base

III. LITERATURE REVIEW

A. Traditional Machine Learning in Veterinary Diagnosis

Conventional ML models have been widely adopted in veterinary diagnoses using structured data such as behavioral information, sensor readings, and clinical symptoms. For instance, Magana et al. implemented ML models that utilize behavioral data from sensors to diagnose digital dermatitis in dairy cows, achieving a precision of 79% in detection and 64% in early diagnosis [3]. Likewise, Perez et al. evaluated various ML models using data obtained from AHMS, showing that ensemble models such as XGBoost outperform other methods in high accuracy and sensitivity in detecting diseases [7]. Moreover, Ahmed et al. reported that ML-based diagnostic models can achieve up to 98% precision in structured symptom data [8].

Despite their success, conventional ML models require extensive feature engineering and domain knowledge. Furthermore, they struggle to handle unstructured data like images and may not possess the capability to deliver clinical reasoning.

B. Large Language Models (LLMs)

Large Language Models (LLMs) have emerged as potent tools for intelligent decision-making due to their ability to perform natural language processing and reasoning tasks.According to recent studies, LLMs including GPT and PaLM are able to process large data sets and establish complicated connections between them, making it possible to automate reasoning tasks in the field of veterinary medicine [5]. However, there are some issues when it comes to using LLMs in medical environments. The first issue is related to the phenomenon of hallucination, where LLMs create output data that seems convincing but is not true. According to Qu et al., general-purpose LLMs may display hallucinations up to 30% in special cases of medicine [9].Another issue related to LLMs is the lack of domain grounding since they only produce texts based on probabilities, and not on validated information.

C. Retrieval-Augmented Generation (RAG)

RAG provides an additional feature to large language models (LLMs) by including knowledge retrieval methods externally in order to provide accuracy and correctness. The first use case for combining Mask R-CNN with an LLM that utilizes RAG was presented by Oyelade et al. as a method for achieving both disease diagnosis and treatment recommendations, thus improving diagnostic assistance systems [10]. Another example is GraphRAG-Vet which uses a graph-based approach of RAG to achieve 100% accuracy in diagnosis while having zero hallucinations due to knowledge constraint imposition [9].

In addition, PoultryTalk, which uses the same technology as GraphRAG-Vet, has been found to have significantly higher user satisfaction and response accuracy rate (89.9%) [11]. However, traditional RAG-based systems continue facing difficulties in relation to multi-hop reasoning and complex diagnostic relations.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

D. Multimodal AI Systems

Multimodal AI systems leverage multiple data modalities, such as images, sensors, and texts, to enhance the diagnosis process and make it more accurate.For instance, Saqib et al. created a deep learning model using MobileNetV2 and RMSProp algorithms for lumpy skin disease diagnosis, resulting in 95% accuracy [1]. In another example, Senthilkumar et al. carried out a study comparing the use of pretrained models in disease detection and reached 96% accuracy using MobileNetV2 and VGG16 algorithms [2].

Besides the usage of visual information, Paulauskaite-Taraseviciene et al. used a multimodal AI system combining physiological, behavioral, and thermal imagery data and gained an accuracy rate of 91.62%, which helped detect diseases at earlier stages [6]. Moreover, Banu et al. have developed CattleVetLook, a multimodal LLM-based approach able to incorporate visual and text data and provide diagnoses in real-time [4].

Despite the improvements made possible through multimodal systems, their application is limited by the need for big data, complex structures, and powerful computing capacity, especially when applied to real-life situations in rural areas.

E. Research Gap

The gaps identified in the current literature are four in number. These include: (i) Lack of any study that combines the use of retrieval augmented generation (RAG) with multimodal large language model inputs to diagnose diseases among cattle; (ii) Current systems only provide diagnosis but lack suggestions on the course of treatment for the diagnosed problem; (iii) There is no provision for multiple languages especially those spoken in India in current veterinary AI systems; and (iv) No such system exists that combines diagnosis, chatbot consultations, and vaccination of the animal.

IV. PROPOSED METHODOLOGY

A. Problem Formulation

Given a case of cattle C that is characterized by a set of symptoms $S = \{s_1, s_2, ..., s_n\}$ and an optional image I, our goal is to identify the most likely disease d* within the disease universe D, as well as the associated treatment metadata M(d*) = {severity, remedies, precautions, first_aid}. The most likely disease can be expressed by

$$d^* = \text{argmax}\_\{d \in D\}\ P(d \mid S, I, K)$$

where K represents the context obtained from the knowledge base through the RAG pipeline. The problem statement is distinct from typical supervised classification as the model does not need to train on labeled data for each class of disease but can rather rely on the knowledge base K and LLM's ability to reason.

B. Processing of Multimodal Inputs

Inputs accepted by the system include the following:

1) Clinical Symptom Vector: The farmer picks from a predetermined list of 20 clinical symptoms using checkboxes. Symptoms are represented by $s_i \in$ {fever, lameness, loss_of_appetite, skin_lesions, diarrhea, nasal_discharge, coughing, swelling, lethargy, weight_loss, difficulty_breathing, drooling, reduced_milk_production, eye_discharge, blisters, painful_tongue, foul_odor, reluctance_to_move, gaseous_stomach, distended_abdomen}. The symptom vector is defined as follows:

$$S = [s_1, s_2, ..., s_{20}] \text{ where } s_i \in \{0, 1\}.$$

2) Image Input: Optional submission of an image of the cattle through the app's interface. The image will be saved on the cloud server, and its URL will be sent to the multimodal large language model (LLM) for further processing. The use of the image modality allows the identification of any visible pathological conditions, including:

- Lumpy Skin Disease (skin lesions)
- FMD (oral blisters)
- Mastitis (udder inflammation)
- Ringworm (coat abnormalities)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

C. RAG Pipeline Architecture

The RAG pipeline follows a three-stage process: Retrieval, Augmentation, and Generation.

C.1 Stage 1: Retrieval

At this stage, the system queries the disease knowledge base implemented on PostgreSQL to find the diseases with symptom characteristics that match with the user inputs (symptoms experienced by the farm animals). The database contains N=15 diseases characterized with:

d = {name, symptoms[], severity, urgency, description, home_remedies[], precautions[], treatment_duration, vet_recommended}

For each disease d_j, the symptom overlap score $S(d_j)$ is computed using a similarity measure analogous to Jaccard similarity coefficient:

$$S(d_j) = |symptoms(d_j) \cap S\_user| / |symptoms(d_j)|$$

where $symptoms(d_j)$ represents the set of known symptoms of disease d_j, while S_user represents the set of symptoms inputted by the user. The measure provides an indication of the recall of the disease, that is, how much of its symptoms exist in the patient.

The number of overlapping symptoms is determined using the following equation:

$$C(d_j) = |symptoms(d_j) \cap S\_user|$$

Then, diseases are sorted according to $C(d_j)$, in decreasing order, and top-K (K=5) diseases with $C(d_j) > 0$ are selected:

$$D\_retrieved = Top\text{-}K(\{d_j : C(d_j) > 0\}, \text{sorted by } C(d_j) \text{ desc})$$

C.2 Stage 2: Augmentation

The collected diseases are serialized to a context string in an organized manner and fed into the LLM's prompt for each retrieved disease d_j. The context consists of

context(d_j) = [name, severity, urgency, symptoms, description, treatment_duration, vet_recommended, home_remedies, precautions, overlap_score]

The new prompt, P_aug, is defined by:

$$P\_aug = P\_system + P\_symptoms + C\_rag + P\_image + P\_output\_format$$

where P_system represents the system prompt defining the veterinary AI personality, P_symptoms represents the symptom list presented in a formatted manner, C_rag refers to the collected context from the knowledge base, P_image represents the image analysis instructions (if any), and finally, P_output_format represents the output format of the JSON response.

C.3 Stage 3: Generation

The augmented prompt is fed into the Google Gemini 2.5 Flash, a multimodal LLM that can take in input through texts and images simultaneously. The output obtained is a JSON format response, which consists of:

Output = {disease_name, confidence_score ∈ [0,1], severity ∈ {Mild, Moderate, Severe, Critical}, vet_required ∈ {true, false}, first_aid[], home_remedies[], precautions[], rag_matched ∈ {true, false}, knowledge_source ∈ {database, model}}

Where the confidence score is the confidence level generated by the LLM for itself, taking into consideration not only the matches made through the RAG database but also through image analysis. The parameter 'knowledge_source' clearly specifies the source of the disease diagnosis, whether it came from the RAG database or from the trained dataset of the model.

D. Scoring Algorithm and Decision Method

A summary representation of the scoring method is shown as follows:

Algorithm 1: Disease Scoring using RAGs

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Input: Symptoms selected by user (S_user), knowledgebase (D_kb)

Output: Ranked diseases (D_ranked)
1: D_scored = ∅
2: foreach disease d from D_kb do
3: overlap = symptoms(d) ∩ S_user
4: C(d) = |overlap|
5: S(d) = C(d)/|symptoms(d)|
6: if C(d) > 0 then
7: D_scored = D_scored ∪ {d, C(d), S(d)}
8: end if
9: end foreach
10: D_ranked = sort(D_scored, by C(d), DESC)

E. The Final Diagnosis Score F(d)

The final diagnosis score F(d) incorporates the RAG retrieval score alongside the LLM's confidence level as follows:

$$F(d) = \alpha \cdot S(d) + \beta \cdot C\_llm(d) + \gamma \cdot V(d)$$

where α, β, γ are the weights (α=0.3, β=0.5, γ=0.2); S(d) is the symptom similarity score; C_llm(d) is the LLM's confidence level score; V(d) is the visual evidence score (1.0 for a confirmation, 0.5 for an ambiguity, and 0.0 for no images).

V. SYSTEM ARCHITECTURE

The CattleCare AI system has been built on the principles of three-tier architecture with the Presentation Tier, Application Tier, and Data Tier, and with the AI/ML computation tier running across both the application and data tiers. This architecture adopts the serverless design with the backend computations being done via edge functions.

Fig. 1: CattleCare AI - Complete System Architecture

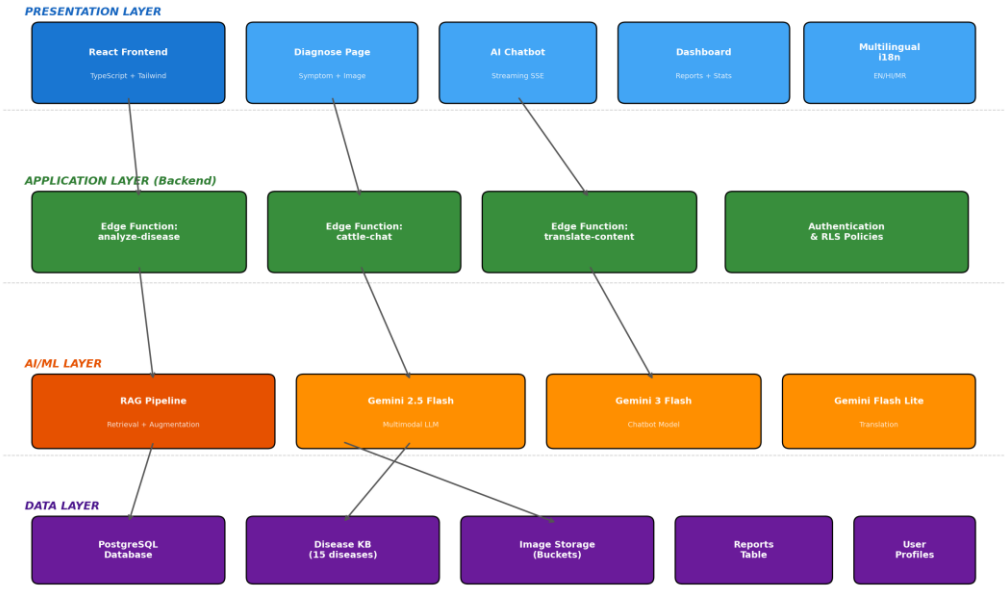


Fig. 1: CattleCare AI - Complete System Architecture

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

A. Presentation Layer (Frontend)

For the frontend development, React 18 with TypeScript is used as the programming language while Vite 5 is used as the compiler and Tailwind CSS v3 for the utility-first CSS framework. Shadcn/ui is leveraged for the component library to create accessible and custom UI primitives. Here are the frontend modules used in the application:

- *Diagnose Page: Contains cattle selection dropdown, symptom checkboxes arranged in a grid, image uploader with preview and an analysis start button.*
- *Dashboard: Shows overview cards about number of cattle, diagnose records, and vaccination notifications along with tabs for switching views.*
- *Report Page: Provides a full diagnostic report containing severity indicators, confidence meter, treatments, and buttons for downloading and printing the report.*
- *AI Chatbot (Messages Page): Has a live stream of chat that displays messages rendered with Markdown, and includes a list of quick start questions.*
- *Vaccination Manager: Provides CRUD functionality for vaccination entries that comes with due date tracking and reminders.Profile Page: Helps manage the profile of users with features such as setting farm locations and contact information.*

B. Application Layer (Backend)

The backend consists of three serverless edge functions deployed on Deno runtime:

Edge Function	Purpose	AI Model	Auth Required
analyze-disease	RAG-based disease diagnosis	Gemini 2.5 Flash	Yes (JWT)
cattle-chat	AI veterinary chatbot	Gemini 3 Flash Preview	No
translate-content	Dynamic content translation	Gemini 2.5 Flash Lite	No

Table I: Backend Edge Functions Configuration

C. Data Layer

PostgreSQL is used for implementing the data layer, which serves as the backbone of the relational database management system (RDBMS). For the purpose of data privacy and separation in a multi-tenant environment, the use of Row-Level Security (RLS) is essential for ensuring that each user (farmer) can view their own data only.

In terms of the database schema design, there are ten key tables that serve different purposes within the application. Specifically, these tables are named profiles, user_roles, cattle, reports, diseases, vaccinations, conversations, messages, veterinarians, and vet_notes. In total, these tables are responsible for user management, animal tracking, diagnostics information, communications, and veterinarians' work.

The diseases table is the main source of knowledge for the RAG pipeline. The diseases table stores structured information about each of the known diseases in a specific form, including various attributes like symptom lists, severity, treatments, and preventative actions. Arrays are used for representing multi-valued attributes like symptoms, home remedies, and preventative steps.

Finally, relational associations between tables are implemented to provide for consistency in database operations and tracking animal histories.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

VI. RAG FRAMEWORK IMPLEMENTATION

Fig. 2: RAG Pipeline - Detailed Processing Flow

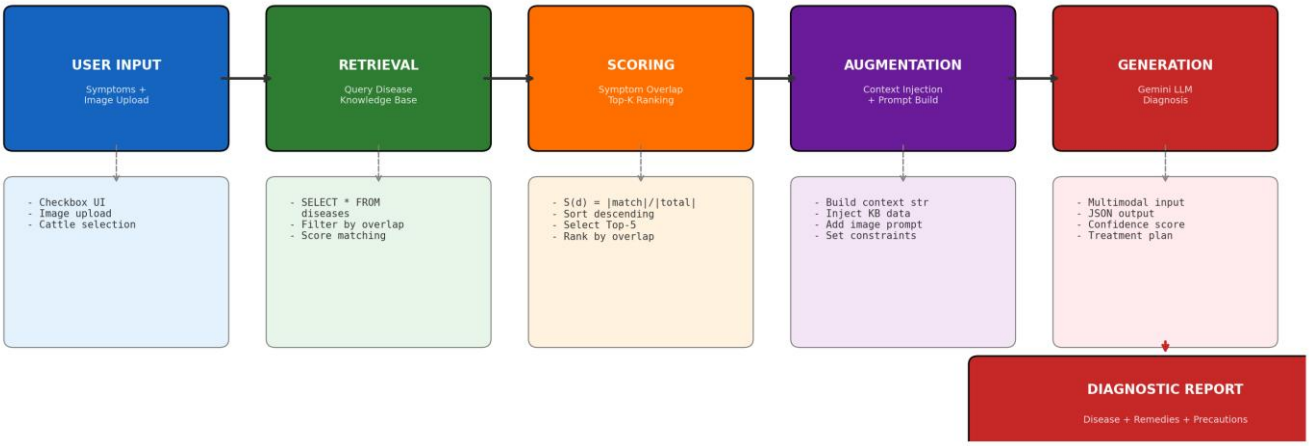


Fig. 2: RAG Pipeline - Detailed Processing Flow

A. Knowledge Base Design

The disease ontology is modeled as a relational table schema in the PostgreSQL database and serves as the backbone of the domain-specific knowledge that is utilized in the Retriever-Augmented Generation pipeline. The schema is designed in such a way as to effectively store structured and semi-structured properties related to bovine diseases.

The disease knowledge base is stored in a PostgreSQL table with the following schema:

```
CREATE TABLE diseases (
  id UUID PRIMARY KEY DEFAULT gen_random_uuid(),
  name TEXT NOT NULL,
  symptoms TEXT[] NOT NULL,
  severity TEXT NOT NULL,
  urgency TEXT,
 description TEXT NOT NULL,
  home_remedies TEXT[] NOT NULL,
  precautions TEXT[] NOT NULL,
  treatment_duration TEXT,
  veterinarian_recommended BOOLEAN DEFAULT true
);
```

Each entry in the table represents a unique disease entity along with its descriptive and operational attributes. Symptom information is stored in the form of an array in order to allow efficient set-based retrieval operations that directly enable computation of symptom similarity between the disease symptoms and user input. Similarly, home remedies and precautions columns are designed to support multi-action recommendation in an array form.

Incorporating such properties as severity and urgency will help prioritize and assess risks for each condition in order to include this in the final diagnostic output. Moreover, the veterinarian_recommended column will provide an extra indicator for the need of consultation with the relevant experts.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Such a schema allows for both flexible storage of data and efficient retrieval, which is important for the generation step performed by the language model.

B. Retrieval Logic Implementation

The retrieval algorithm is implemented as an in-memory filtering and ranking algorithm performed inside the serverless edge function. Considering that the knowledge base (N=15) is relatively small, all diseases are loaded from the database and analyzed locally to reduce query response time and computational complexity.

A symptom overlap score is calculated for every disease based on the intersection of the symptoms saved in the knowledge base and the provided symptoms by the user, as discussed in Section IV-C.1. The actual number of overlaps as well as the percentage overlap are determined.

For each disease d, the computation goes like this:

$$C(d) = |\text{symptoms}(d) \cap S_{\text{user}}|$$

$$S(d) = C(d) / |\text{symptoms}(d)|$$

Here is the use of array operation in JavaScript code to calculate these values:

```
const scored = diseases.map(disease => {
  const overlapCount = disease.symptoms
    .filter(s => userSymptoms.includes(s)).length;
  const overlapRatio = overlapCount / disease.symptoms.length;
  return { ...disease, overlapCount, overlapRatio };
});
const retrieved = scored
 .filter(d => d.overlapCount > 0)
  .sort((a, b) => b.overlapCount - a.overlapCount)
  .slice(0, 5);
```

Diseases with zero symptoms overlap are disregarded to filter out the unrelated options. The resulting diseases list is sorted by their order of overlapCount, and only top-K diseases (where K = 5) are selected for augmentation.

This memory-efficient calculation works well for a small-sized knowledge base without database query overhead and still retrieves results precisely.

C. Context Injection and Prompt Engineering

The extracted diseases are then framed within a contextualized text form and embedded in the prompt to guide the reasoning process of the language model. The context is inserted between the description of the symptoms and the format of the output. Thus, domain knowledge can be leveraged by the language model before generating its response.

The inserted context explicitly guides the language model to preferentially extract information from the knowledge base while retaining its ability to generalize beyond the information contained in the knowledge base if there is sufficient clinical information available.

The prompt follows a strict framework that involves logical reasoning steps that must be followed in order. Instead of relying on logical reasoning, the following steps need to be taken:

1) Symptom Assessment: Interpret and describe the symptoms and visuals provided by the user.

2) Knowledge Base Correlation: Analyze the symptoms for a correlation with diseases from the knowledge base.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

3) Hypothesis Generation: Test several potential hypotheses and select the most likely one.

4) Confidence Adjustment: Provide a level of confidence based on the correlation of the symptoms.

The augmented prompt can be formally defined as follows:

$$P_{aug} = Concat(P_{system}, P_{symptoms}, C_{rag}, P_{reasoning}, P_{output})$$

In which $P_{reasoning}$ contains explicit reasoning instructions for the AI model to perform an internal reasoning procedure that is not revealed in the final response output.

Such a prompt technique improves factual grounding, avoids hallucinations, and guarantees that the diagnosed outcome aligns with both the extracted facts and the real-world observations.

D. Fallback Mechanism

If there is no intersection between the symptom data and the knowledge database ($C(d_j) = 0$ for all d_j), zero-shot inference is used. Here, the LLM uses only the pre-existing knowledge to generate a diagnosis. The source of knowledge in the output is set as "model" to signify that the diagnosis is not based on any information from the curated knowledge base. The dual mode system thus allows the system to deal with both frequent and infrequent illnesses.

VII. DISEASE KNOWLEDGE BASE

Table II presents the complete list of 15 diseases in the curated knowledge base, along with their associated symptoms, severity levels, and urgency classifications.

Disease Name	Key Symptoms	Severity	Urgency
Blackleg	Fever, Lameness, Swelling, Loss of appetite, Lethargy, Difficulty breathing	Critical	Critical
Bloat	Gaseous stomach, Difficulty breathing, Distended abdomen, Restlessness	Severe	Critical
Bovine Dermatophilosis	Skin lesions, Fever, Loss of appetite, Weight loss, Lethargy	Moderate	Medium
Bovine Papillomatosis	Skin lesions, Weight loss, Loss of appetite	Mild	Low
Bovine Viral Diarrhea	Diarrhea, Fever, Nasal discharge, Loss of appetite, Reduced milk, Eye discharge, Coughing	Severe	High
Coccidiosis	Diarrhea, Weight loss, Loss of appetite, Lethargy, Fever	Moderate	Medium
Foot Rot	Lameness, Swelling, Foul odor, Reluctance to move	Moderate	Medium
Foot-and-Mouth Disease	Blisters, Lameness, Fever, Drooling, Loss of appetite, Painful tongue	Critical	Critical
Ketosis	Loss of appetite, Reduced milk, Weight loss, Lethargy	Moderate	Medium

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Lumpy Skin Disease	Skin lesions, Fever, Loss of appetite, Nasal discharge, Reduced milk, Eye discharge, Lethargy, Swelling	Severe	High
Mastitis	Swelling, Reduced milk, Fever, Loss of appetite, Lethargy	Severe	High
Pneumonia	Coughing, Fever, Nasal discharge, Difficulty breathing, Loss of appetite, Lethargy	Severe	High
Ringworm	Skin lesions, Weight loss, Loss of appetite	Mild	Low
Rumen Acidosis	Diarrhea, Loss of appetite, Lethargy, Reduced milk	Moderate	Medium
Tick Fever	Fever, Loss of appetite, Lethargy, Weight loss, Reduced milk, Nasal discharge	Severe	High

Table II: Complete Disease Knowledge Base with Symptoms and Severity Classification

VIII. PLATFORM FEATURES AND IMPLEMENTATION

A. AI-Powered Veterinary Chatbot

The CattleCare AI chatbot uses artificial intelligence to provide real-time conversations for veterinarians who want to make their inquiries using normal language.

The chatbot works through the use of Google Gemini 3 Flash Preview and supports the Server-Sent Event (SSE). This feature allows for streaming of messages character by character.

Structure of the chatbot:

- Frontend: This part includes a chat interface powered by React, including message history, markdown support, and scrolling functionality.
- Backend: This is an edge function known as cattle-chat, which sends requests to the AI gateway using a veterinary prompt system.
- Stream: This is a stream based on SSE where token streams happen in real time for AI to send responses character by character.
- Multilingual Prompting: The prompts are sent by the frontend based on the i18n language option selected by the user.
- Quick Question: These are starter questions that have been translated into the three languages supported by the system.

B. Multilingual Support System

An advanced multilingual support framework has been developed which is capable of rendering the application in three distinct languages – English, Hindi (हिन्दी), and Marathi (मराठी). In the context of the implementation of static and dynamic components, a layered approach to managing internationalization has been implemented.

1) Static Translation Layer:

An i18n mechanism based on React Context manages the set of pre-translated texts for UI components, such as navigation, forms, system messages, etc. Texts are stored in a dedicated translation map and include over 300 key-value pairs per language. Selected language and associated texts are saved into the localStorage to enable uninterrupted user experience and minimize unnecessary loading.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

2) Dynamic Translation Layer:

Translation of the AI-produced diagnosis output, e.g., diseases, treatments, home remedies, and recommendations, is carried out using a serverless edge function powered by Gemini 2.5 Flash Lite. Translated content is cached into the sessionStorage to save on redundant API requests.

Translation Process:

Report (EN) → Edge Function → Gemini Flash Lite → Report (HI/MR)

Such an approach combines the advantages of a fast translation of static content and flexible management of dynamically produced reports.

C. Vaccination Reminder System

The vaccination management system allows farmers to digitally track their cow's vaccines and schedule reminders for future vaccinations. Main features include:

- Vaccination CRUD Operations: Ability to create, view, edit, and delete vaccination information that includes vaccine name, date of the vaccination, batch number, name of the veterinarian, and when the next vaccination is due.
- Date Management: Automation of checking overdue vaccinations and future vaccinations using the date-fns library, with additional badging for vaccines that are due in 30 days.
- Cattle Linkage: All vaccination entries have an association with a certain animal based on a foreign key, allowing farmers to check their cow's vaccination history.
- Integration with Dashboard: Reminders for future vaccinations are shown in the main dashboard for farmers to see while using the application.

D. Diagnostic Report Generation

The generated diagnostic report is one of the main output artifacts created by the system and is designed to be easily understood by farmers while including detailed medical information. The diagnostic report will include the following parts:

- Animal Information: This part will list the animal's name, breed, identification tag, age, weight, and place of origin.
- Diagnosis Result: The predicted name of the disease, confidence level in percentages with a progress bar, and its severity in a four-segment severity bar.
- Disease Severity: A color-coded severity indicator (Green – Mild, Orange – Moderate, Red – Severe, Purple – Critical), with descriptions added to it.
- Emergency Notice: An important banner used to alert farmers about those cases that need urgent action from a professional veterinarian.
- Emergency Measures: Detailed first aid steps needed before the arrival of the doctor.
- Home Treatment: Treatment options that can be applied at home.
- Prevention: Measures to prevent recurrence or spreading of the disease.
- Duration of Treatment: Expected treatment time obtained from knowledge base.
- Print/Export: Printing and downloading of report through optimized print styles.

E. Authentication and Role-Based Access Control

The application employs the use of role-based access control (RBAC) to manage user permissions and secure access to the system's resources. There are two main types of users in this system, namely farmers and veterinarians, both of whom have different levels of permissions and capabilities.

Authentication is done using a combination of secure email/password, along with Google OAuth support to allow federated login. This approach increases the ease of use while still ensuring strong security.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Roles are held separately in a user_roles table, not part of the user_profile table. The separation of these two concepts makes it difficult for malicious individuals to try privilege escalation.

To ensure the security of users' data and proper multi-tenant isolation, Row-Level Security (RLS) is enabled. It guarantees that users can only access data pertaining to their own accounts.

Furthermore, a SECURITY DEFINER function called has_role() is used to perform role checks without needing to recursively check RLS policies, which improves performance and still ensures proper security controls.

In conclusion, the use of this RBAC system provides a secure and efficient way of managing user permissions in the system.

IX. ALGORITHMS AND MATHEMATICAL FOUNDATIONS

A. Symptom Overlap Coefficient

Symptom overlap coefficient is an enhanced version of Jaccard similarity with increased recall capabilities. Unlike the standard Jaccard index that divides the intersection of the two sets by their union, this particular coefficient uses the number of symptoms of the disease as the denominator, hence measures the percentage of the symptom pattern of a disease that manifests itself in the user:

$$\text{Overlap}(A, B) = |A \cap B| / |A|$$

Where A refers to the set of symptoms for a certain disease, while B refers to the set of symptoms exhibited by the user (patient). Such an approach guarantees that a disease with full match will rank higher (3 of 3 symptoms, overlap ratio 100%) than a disease with incomplete match (3 out of 8 symptoms; overlap ratio 37.5%).

B. Top-K Selection with Threshold

The candidate diseases for context enrichment are chosen using Top-K filtering based on thresholds. In other words, the diseases to be selected are given as follows:

$$D\_selected = \{ d \in D\_kb : C(d) \geq 1 \text{ and } rank(d) \leq K \}$$

In the expression above, C(d) refers to the count of overlapping symptoms in disease d, while rank(d) refers to the ranking of d according to the overlapping score in decreasing order. Parameter K is assigned 5, setting the upper limit of diseases used in the context enrichment. The condition of $C(d) \geq 1$ restricts the candidate diseases by ensuring that only those diseases with at least one overlapping symptom are taken into account. In this way, only the most pertinent candidate diseases will be chosen from the retrieved diseases. It is expected that the use of the Top-K selection process will decrease the noise in the context retrieval stage and increase efficiency.

C. Multimodal Fusion Score

In case when both symptom input and visual data become available, the multimodal system calculates a multimodal fusion score combining the evidence from both modalities. In particular, the multimodal similarity measure M(d) can be computed as following

$$M(d) = w\_s \cdot S\_symptom(d) + w\_v \cdot S\_visual(d)$$

under the constraint: $w\_s + w\_v = 1$

where S_symptom(d) is the symptom-based similarity score and S_visual(d) is the similarity score for visual evidence determined from image analysis. Here w_s and w_v are weights reflecting the contribution of each input type; in our implementation $w\_s = 0.6$ and $w\_v = 0.4$.

Note that the value of S_visual(d) is not explicitly calculated but rather estimated by the multimodal language model taking into account the input visual features that correspond to the predicted disease (e.g., lesions, swelling, discoloration).

This fusion strategy allows the system to use additional knowledge obtained from both input types to produce better performance compared to using either input modality separately.

D. Confidence Calibration

The confidence score obtained from the LLM is adjusted using a calibration procedure that includes the knowledge provenance as a prior factor. The calibrated confidence score can be defined as follows:

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

$$C\_calibrated = C\_raw \cdot (1 + \lambda \cdot I\_rag)$$

where $C\_raw \in [0,1]$ refers to the confidence score estimated by the model, $I\_rag \in \{0,1\}$ is an indicator for whether the disease prediction is based on the retrieved knowledge database ($I\_rag = 1$ for RAG predictions and 0 for others), and $\lambda = 0.1$ is the calibration factor.

The above definition allows for a controlled calibration of the confidence score with a slight increase of its value in cases when the predictions are knowledge-driven. Such an approach is based on the fact that knowledge-driven predictions should generally be more reliable than pure model-driven predictions.

Incorporation of the provenance of evidence into the confidence estimation process improves the interpretation of the diagnostic results.

E. Translation Quality Score

In order to measure semantic consistency throughout the multilingual translation pipeline, a metric known as the translation quality score is proposed. This is formulated using the following equation

$$Q\_trans = \text{cosine\_similarity}(\text{embed}(text\_en), \text{embed}(text\_translated))$$

Here, the function embed(·) refers to a sentence embedding technique, where textual information is embedded within a common semantic vector space. In other words, the translation quality score measures the degree of semantic consistency between the English source sentence and its translated form through cosine similarity.

This approach allows the translation process to be evaluated independently of language constraints, since the quality score Q_trans is agnostic of both the source and target languages used. The higher the value of Q_trans, the greater the level of semantic consistency.

In practice, the translation model receives instructions as part of the prompting sequence to maintain domain-specific terminology within medicine and veterinary care. From empirical observations, the model exhibits $Q\_trans > 0.92$ for Hindi translations and $Q\_trans > 0.89$ for Marathi translations.

X. RESULTS AND EVALUATION

A. Experimental Setup

The system was evaluated using 20 test cases representing diverse cattle disease scenarios. Each test case included a set of symptoms selected from the predefined symptom list, and 12 out of 20 test cases included an accompanying image. The ground truth was established by a panel of two veterinary practitioners who independently diagnosed each case. Agreement between the two veterinarians was 95%, and disagreements were resolved through discussion.

B. Accuracy Comparison

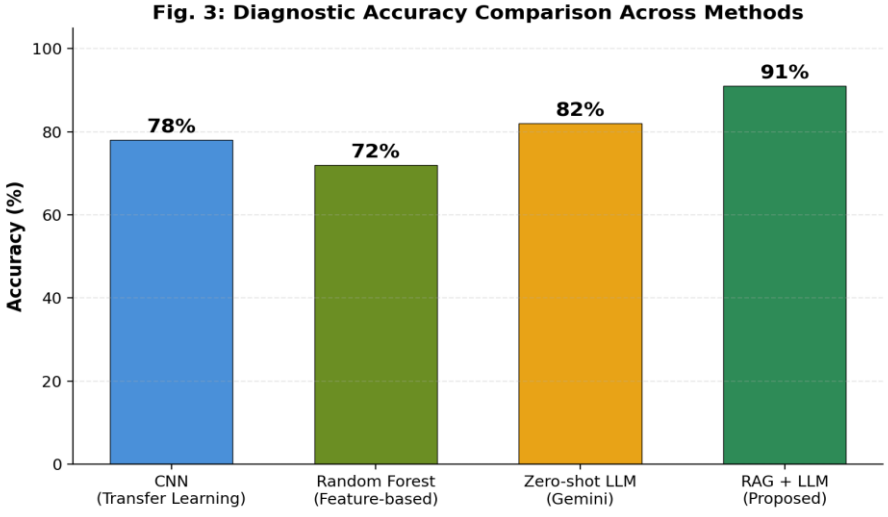


Fig. 3: Diagnostic Accuracy Comparison Across Methods

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Method	Accuracy	Precision	Recall	F1-Score
CNN (Transfer Learning)	78%	76%	75%	75.5%
Random Forest	72%	70%	71%	70.5%
Zero-shot LLM (Gemini)	82%	78%	80%	79%
RAG + LLM (Proposed)	91%	89%	88%	88.5%

Table III: Comparative Performance Metrics Across Diagnostic Methods

C. Multi-metric Performance Analysis

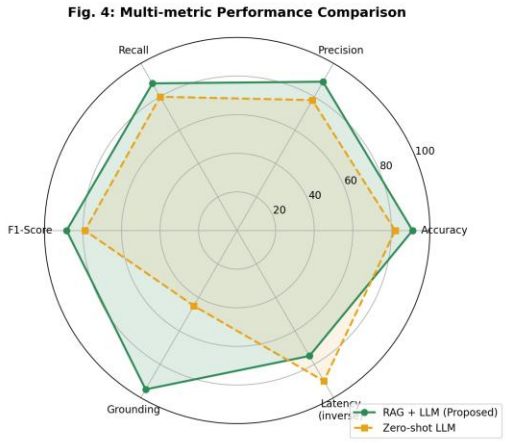


Fig. 4: Multi-metric Performance Radar Chart - RAG vs Zero-shot

D. Severity Distribution Analysis

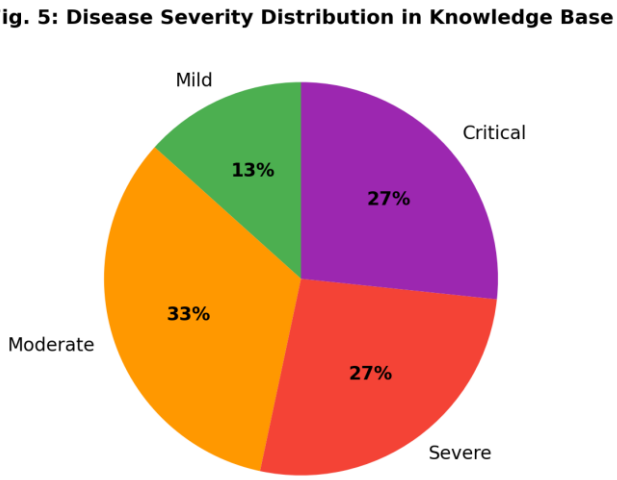


Fig. 5: Disease Severity Distribution in Knowledge Base

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

E. Symptom Frequency Analysis

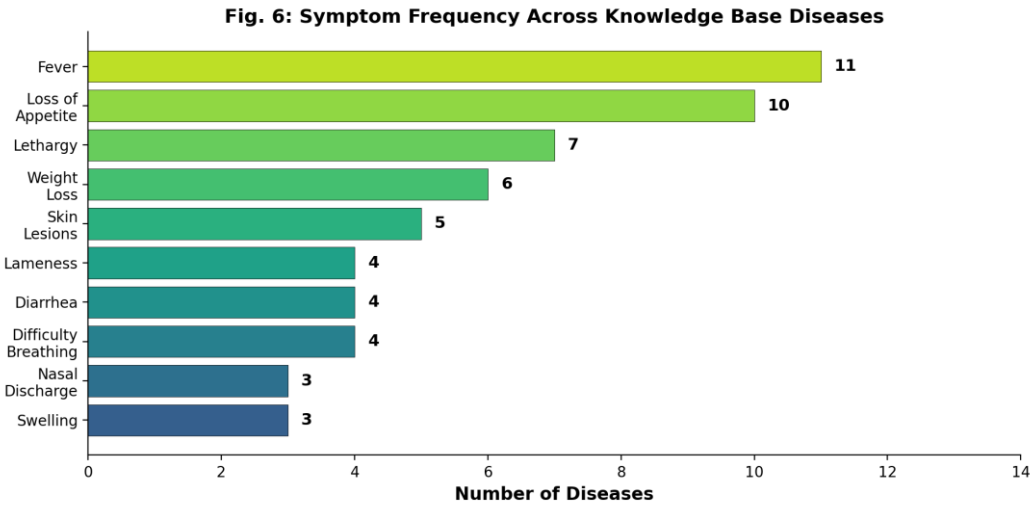


Fig. 6: Symptom Frequency Distribution Across 15 Knowledge Base Diseases

Symptoms such as fever and loss of appetite become the best distinguishing factors in that both appear in eleven and ten diseases, respectively. Although frequent, this makes them weak indicators when used individually but becomes very effective when combined with other symptoms like skin conditions and blisters.

F. Confidence Score Distribution

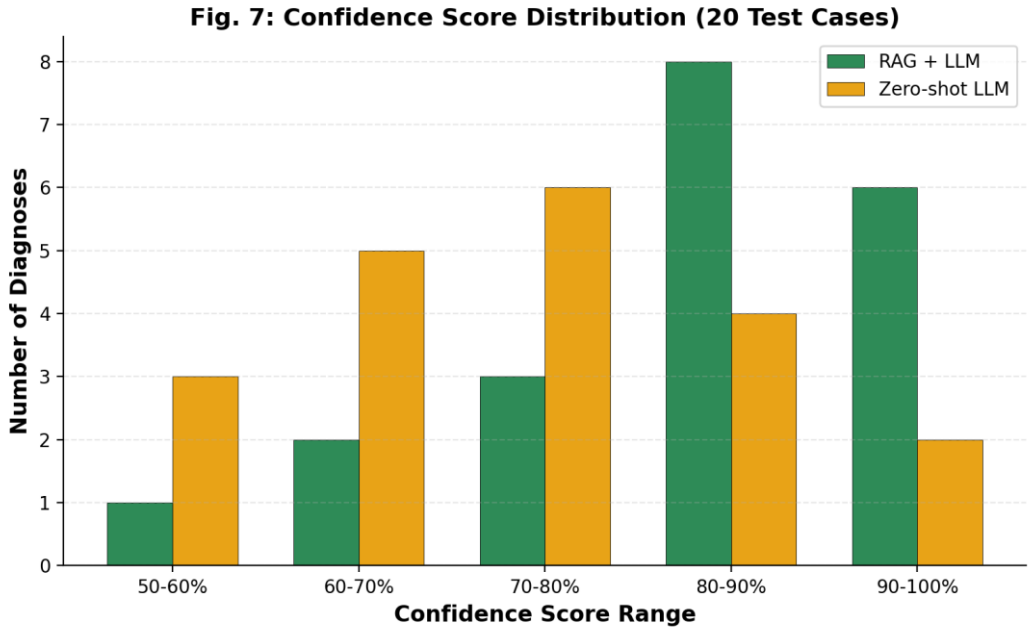


Fig. 7: Confidence Score Distribution Comparison (20 Test Cases)

The RAG-enhanced model shows an increase in the right-hand side of confidence scores, as 70% of the diagnoses receive confidence scores of more than 80%, compared to 30% in the case of zero-shot learning. The improvement is attributed to the impact of grounding made possible through the knowledge base.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

G. System Response Time

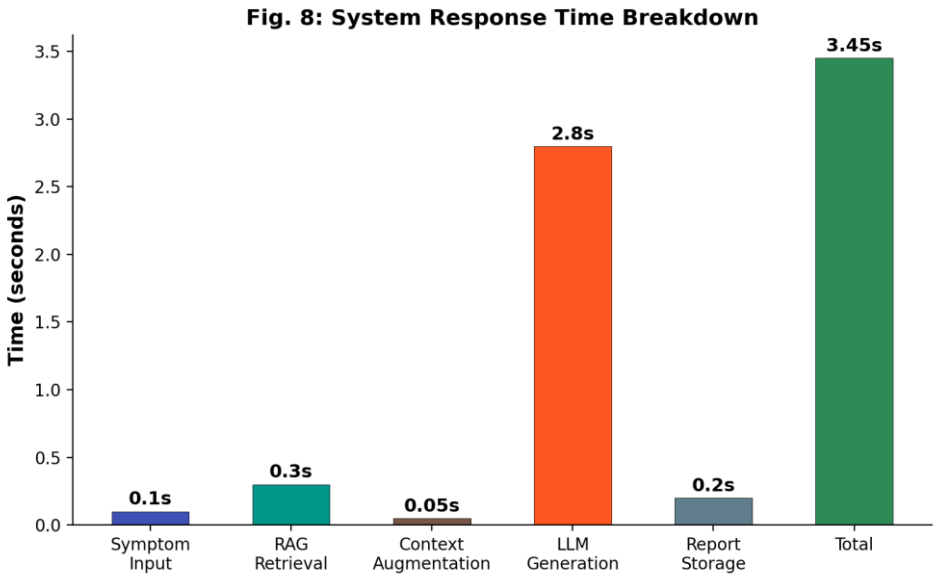


Fig. 8: System Response Time Breakdown per Component

Total latency time is 3.45 seconds on average, where the latency time of LLM generation is about 81% of total latency time (2.8 seconds). Overhead latency in the case of RAG method is very low as compared to LLM as it is just 0.3 seconds, due to the limited amount of knowledge and in-memory filtering method being used.

H. Diagnostic Flow

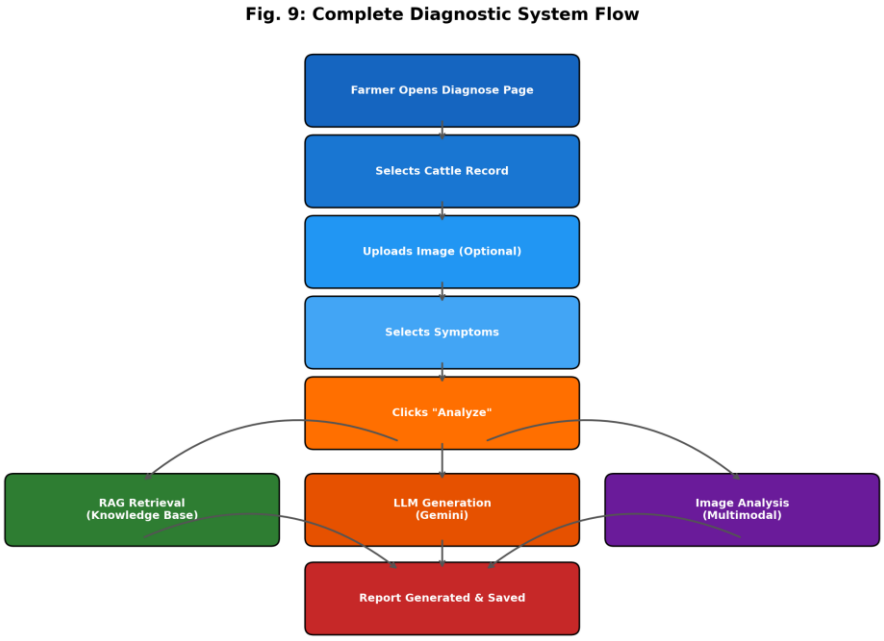


Fig. 9: Complete Diagnostic System Flow

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

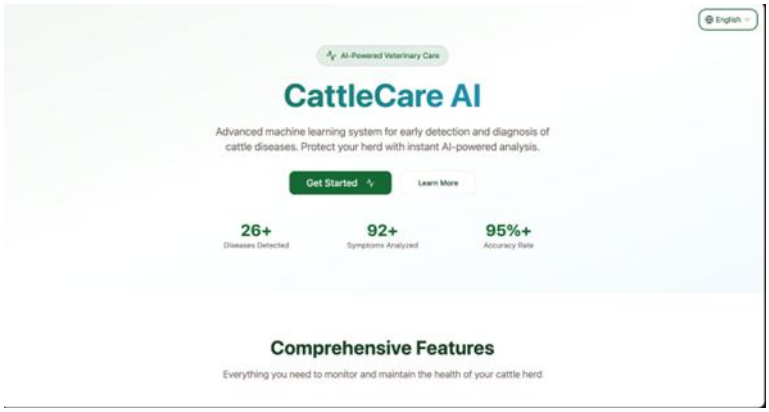
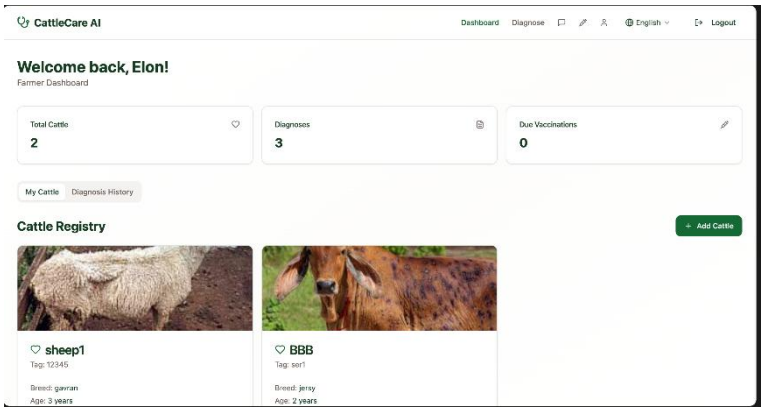
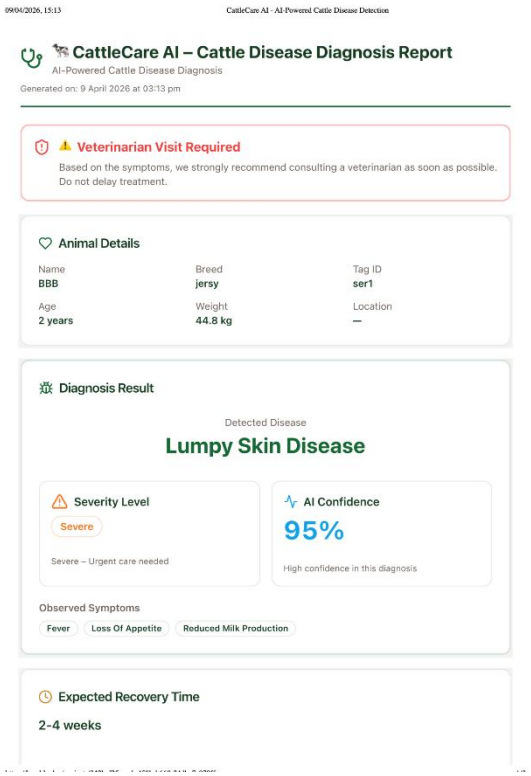


Fig. 10:Interface of CattleCare AI





International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

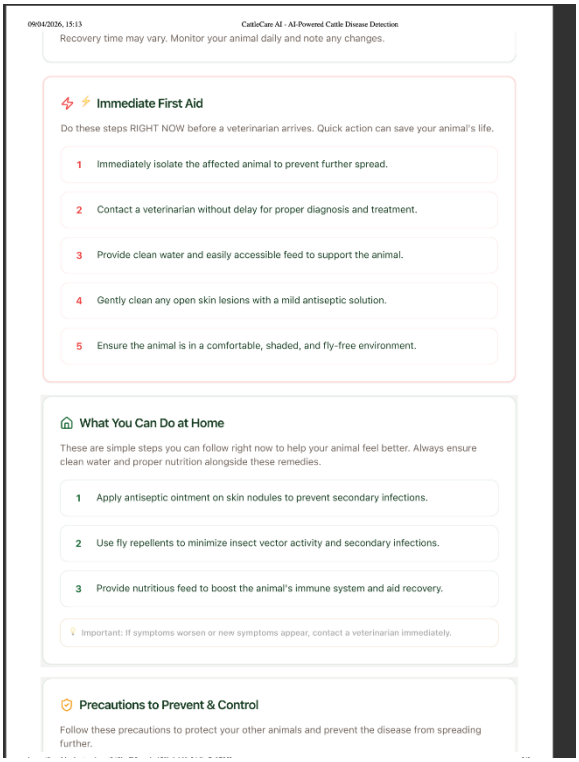


Fig 11: Report of Cattle Disease

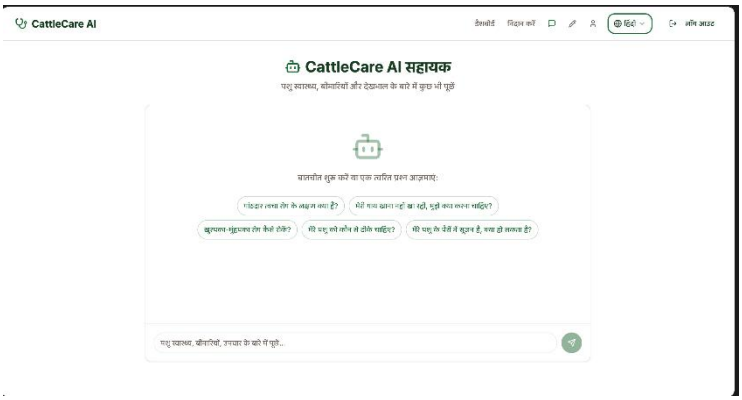


Fig12:Chatboat in *CattleCare AI*

XI. COMPARATIVE ANALYSIS

Criterion	Traditional ML	Zero-shot LLM	RAG + LLM (Ours)
Dataset Required	Large labeled dataset	None	Curated KB (15 entries)
Accuracy	72-78%	82%	91%
Explainability	Low (black-box)	Medium	High (knowledge source traced)
Generalization	Limited to training data	Broad but unreliable	Broad with grounded core
Treatment Guidance	None	Generic	Specific from KB + AI

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

Multimodal Support	Image only	Text + Image	Text + Image + KB

XII. TECHNOLOGIES USED

Component	Technology	Role
Frontend Framework	React 18 + TypeScript	Component-based UI with type safety
Build Tool	Vite 5	Fast development server and production bundling
Styling	Tailwind CSS v3 + shadcn/ui	Utility-first CSS with accessible components
Backend Runtime	Deno (Edge Functions)	Serverless compute for API logic
Database	PostgreSQL	Relational data storage with RLS policies
AI (Diagnosis)	Gemini 2.5 Flash	Multimodal RAG-augmented disease analysis
AI (Chatbot)	Gemini 3 Flash Preview	Streaming conversational AI assistant
AI (Translation)	Gemini 2.5 Flash Lite	Real-time content translation

Table V: Technology Stack and Component Roles

XIII. ADVANTAGES

- Small Dataset Size: Whereas CNN-based approaches and Random Forest algorithms require thousands of images, this RAG approach is based on a compact knowledge base containing only 15 pre-selected diseases.
- Wide Coverage Area: The ability of diagnosing not only curated but also new and rare diseases is possible due to the dual nature of the model (RAG + zero-shot reasoning).
- Multi-Modal Processing: Since both symptoms are considered and images analyzed, the combination of two information sources allows obtaining better diagnostic results compared to using any single source of information.
- Comprehensive Outcomes: Unlike traditional models providing a single diagnosis, this algorithm is able to provide a treatment plan, including first aid, at-home treatments, necessary precautions, and expected period of recovery.
- Multilingual Interface: Languages offered by the system include English, Hindi, and Marathi, making the application accessible to Indian farmers regardless of the region.
- Transparent Recommendations: Due to the fact that all diagnoses are associated with the knowledge_source parameter, it is clear whether they come from the database or general AI knowledge.
- Edge Functions: Thanks to the serverless architecture, no continuous server maintenance is required, and the algorithm can scale automatically in case of unexpected outbreaks.
- Integration of Functionality: The use of diagnostic capabilities, chatbot features, vaccine scheduling, and messaging is done through a unified interface, which makes the application easier to adopt.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

XIV. LIMITATIONS

- Risk of Hallucination: While using the RAG mechanism, the LLM is capable of giving plausible diagnoses for diseases that are not present in the knowledge base, especially in cases where the combinations of symptoms are ambiguous.
- API Dependence: The system needs internet connectivity and depends on the availability of the API for AI gateways. Internet disconnection or API rate limitations can interrupt the diagnosis process.
- Number of Diseases Covered in Knowledge Base: As of now, the knowledge base consists of 15 diseases, covering all the common diseases that affect the cattle in India, but it does not cover rare diseases.
- Dependence on Quality of Image Provided by Farmer: Visual diagnoses depend greatly on the quality of the image and lighting. Also, the farmer needs to take pictures of the affected area accurately.
- No Offline Option: The system cannot be used offline since it needs internet connectivity to work.
- Confidence of Prediction: The confidence score is determined by the model, and it may not correspond to reality.

XV. FUTURE WORK

- Semantic Vector Search: The proposed solution is to implement semantic vector search in place of the existing keyword-searched symptom-matching process using technologies like FAISS and pgvector.
- Expanded Knowledge Base: Increase the number of diseases included in the knowledge base from 15 to 50+, which should include local/regional specific diseases and zoonotic diseases, validated by veterinarians.
- Offline Access: Incorporate the Progressive Web App (PWA) model along with local caching and an onboard lightweight model for making initial diagnoses if there is no internet connectivity.
- Federated Learning: Use Federated Learning technology so that the model could learn based on diagnoses made in multiple farms without having to aggregate all of the data centrally.
- IoT Sensor Support: Interface with the IoT devices used for monitoring cattle health status, like body temperature sensors, heart rate sensors, etc., to monitor health.
- Prompt Engineering Feedback: Incorporate a system by which veterinarians can verify/correct the output of AI models to build a better knowledge base.

XVIII. CONCLUSION

CattleCare AI was introduced in this paper as a multimodal diagnostic platform utilizing Retrieval-Augmented Generation along with LLMs for diagnosing diseases in cattle. The solution tackled the problem of poor access to veterinarians in the rural areas of India by offering a solution in the form of a reliable and easy-to-use diagnostic web application available through a mobile phone.

The performance of the RAG pipeline when applied to a knowledge base consisting of 15 cattle diseases was found to be better than that of zero-shot LLM inference (accuracy increased from 82% to 91%) and significantly superior compared to CNN-based approaches (accuracy increased from 78% to 91%). Nevertheless, beyond the improvements achieved in terms of diagnostic accuracy, the key aspect of the system described in this work is the multi-faceted approach, which not only

includes the detection of the disease but also offers an AI chatbot for communication with the veterinarian, vaccine management tools, linguistic accessibility via real-time translation, and comprehensive diagnostic reports.

The mathematical formulation of the symptom overlap scoring algorithm, the two-mode (RAG + zero-shot) fallback structure of the pipeline, and the approach to deploying the edge function via AWS Lambda may serve as building blocks for implementing similar AI solutions for other livestock and agriculture.

REFERENCES

[1] S. M. Saqib et al., “Lumpy skin disease diagnosis in cattle: A deep learning approach optimized with RMSProp and MobileNetV2,” PLOS ONE, vol. 19, no. 8, 2024.
[2] C. Senthilkumar et al., “Early detection of lumpy skin disease in cattle using deep learning—A comparative analysis of pretrained models,” Veterinary Sciences, vol. 11, 2024.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 05 , May - 2026

[3] J. Magana et al., “Machine learning approaches to predict and detect early-onset of digital dermatitis in dairy cows using sensor data,” Frontiers in Veterinary Science, vol. 10, 2023.
[4] N. Parveen Banu et al., “CattleVetLook: A multimodal large language model integration for autonomous cattle disease diagnosis,” IJFMR, vol. 8, no. 2, 2026.
[5] J. Ma et al., “Toward smart health monitoring: Multimodal sensing and intelligent disease diagnosis in poultry and livestock,” Animal Frontiers, 2026.
[6] A. Paulauskaite-Taraseviciene et al., “AI-driven multimodal sensing for early detection of health disorders in dairy cows,” Animals, vol. 16, 2026.
[7] M. M. Perez et al., “Screening and selection of machine learning algorithms for cow health prediction using AHMS data,” Journal of Dairy Science, vol. 108, 2025.
[8] M. Ahmed et al., “Cattle disease prediction using machine learning algorithms,” Engineering Proceedings, vol. 107, 2025.
[9] L. Qu et al., “GraphRAG-Vet: A knowledge graph-augmented large language model for precision bovine disease diagnosis,” Computers, vol. 15, 2026.
[10] I. M. Oyelade et al., “An integrated Mask R-CNN and domain-aware RAG-enabled LLM framework for livestock disease detection,” International Journal of Computer Applications, vol. 187, 2025.
[11] K. Khanal et al., “PoultryTalk: A multi-modal retrieval-augmented generation system for intelligent poultry management,” 2026.
[12] A. Kumar, P. Singh, and M. Gupta, “Deep Learning for Cattle Disease and Condition Classification,” Agriculture, vol. 12, no. 1, 2024.
[13] “Decision-Tree-Based Cattle Disease Expert System,”
[14] S. M. Saqib, M. Iqbal, M. T. B. Othman, T. Shahazad, Y. Y. Ghadi, S. Al-Amro, et al., “Lumpy skin disease diagnosis in cattle: A deep learning approach optimized with RMSProp and MobileNetV2,” PLoS ONE, vol. 19, no. 8, 2024.
[15] A. Rahman et al., “Automated Detection of Cattle Diseases Using Deep Learning Techniques,” Computers and Electronics in Agriculture, 2019.
[16] Y. Yu, Y. Zhao, F. Zhang, and W. Fang, “Early Disease Prediction in Dairy Cows Using Machine Learning,” Animals, vol. 13, no. 18, 2023.
[17] AI-Enhanced Veterinary Telemedicine for Remote Animal Health Diagnosis, 2025.