

Machine Learning-Based Multi-Class Disease Prediction for Cattle and Livestock Health Management

Halaswamy H R ¹, Prakasha raje urs M ², T Chandraiah ³, Bindu K B ⁴, Basavanna M ⁵

¹Department of Studies in Computer Applications (MCA), Davangere University,
Davanagere, Karnataka, India. halswamy123@gmail.com

²Department of Computer Science, Government First Grade College, Nanjangud,
Karnataka, India.
purs73@gmail.com

³Department of Computer science, Yuvaraja College, University of Mysore, Mysore, Karnataka, India.
chandruycm@gmail.com

^{4,5}Department of Studies in Computer Science, Davangere University, Davangere,
Karnataka, India.

du_csphd_bs01@davangereuniversity.ac.in, basavanna.m@davangereuniversity.ac.in

ABSTRACT

Livestock health monitoring in rural and commercial farms still depends largely on sporadic visual inspection and infrequent veterinary visits, so early warning signs are routinely missed, and minor ailments escalate into fatal outbreaks. This work presents an automated cattle and livestock disease prediction system that infers the most probable disease from routinely observable data — animal type, breed, age, gender, weight, symptom clusters, and vital signs such as body temperature and heart rate. Categorical fields are converted using One-Hot Encoding and numerical fields are normalised using StandardScaler, after which a Random Forest ensemble of 500 decision trees performs multi-class classification across 139 disease classes. The trained model is deployed through a lightweight Flask web application that supports both single-animal diagnosis via web forms and bulk herd screening via CSV upload, and pairs every prediction with ranked probabilities and a medication recommendation. Experimental results on an augmented dataset of 6,950 records show that Random Forest achieved the highest accuracy of 98.42%, outperforming K-Nearest Neighbors (97.70%), Naive Bayes (97.19%), and a single Decision Tree baseline. The system delivers sub-second predictions and a bulk throughput of approximately 44 animals per second, supporting early disease detection, reduced animal mortality, and more targeted use of antibiotics.

Keywords: Cattle Disease Prediction, Machine Learning, Random Forest Classifier, One-Hot Encoding, StandardScaler, Flask, Livestock Health Management

I. INTRODUCTION

Livestock farming is essential to agriculture and contributes significantly to the economy through milk, meat, and other dairy products. The health status of cattle is closely linked to their productivity and, in turn, to the farmer's profitability. Animals are, however, susceptible to bacterial, viral, parasitic and metabolic disorders whose early symptoms are subtle and easily overlooked in large herds.

Conventional diagnosis relies on hands-on checks with basic instruments such as thermometers, stethoscopes and printed symptom charts, supported by occasional expert visits. These methods are reliable for routine rounds but do not scale: smallholder farmers face high

consultation fees, travel burdens and stretched veterinary manpower, and the resulting delay converts minor ailments into fatal outbreaks that claim an estimated 20–25% of young stock annually.

Machine learning offers a practical way to encode expert diagnostic patterns into a tool that scales far beyond what an individual veterinarian could review manually. This paper presents a Cattle Animal Disease Prediction system that learns disease signatures from structured animal health records and delivers a ranked diagnosis together with a treatment recommendation through an ordinary web browser. A Random Forest classifier is compared against K-Nearest Neighbors, Naive Bayes, and a Decision Tree baseline to identify the most effective model for field use.

The rest of the paper is organised as follows: Section II presents the related work and reviews existing methods for livestock disease detection. Section III describes the proposed methodology. Section IV describes the technology used, while Section V discusses the results. Section VI presents the conclusion and future work. Finally, Section VII includes the references.

Problem Statement

Livestock diseases in cattle, goats, and sheep cause major productivity losses and economic damage to farmers. Traditional detection methods are slow, labour-intensive and dependent on expert availability, and there is no established mechanism for bulk screening across large herds — diagnosis remains a one-by-one process with no learning loop. Blind or precautionary medication raises costs and contributes to antibiotic misuse. An automated system based on machine learning can improve both the accuracy and the throughput of diagnosis, and this work compares four classifiers to identify the most effective solution.

Objectives

1. To implement an automated solution for disease prediction and classification in cattle and allied livestock using structured health data.
2. To preprocess animal health records through One-Hot Encoding and StandardScaler to improve dataset quality and model training.
3. To design and implement Random Forest, KNN, Naive Bayes, and Decision Tree models, and compare their performance based on accuracy and precision.
4. To enable rapid single-case evaluation through an intuitive web interface and high-volume herd diagnostics through batch CSV upload.
5. To integrate predictive outputs with targeted treatment protocols, supporting early detection, reduced mortality, and controlled antibiotic usage.

Dataset

The dataset used in this work is an augmented animal disease prediction dataset of 6,950 records with 22 features, covering 8 animal types, 120 breeds, and 139 disease classes. It combines veterinary records with symptom references drawn from public veterinary disease databases. Each record captures identification attributes (animal type, breed, age, gender, weight), up to four symptom slots with duration and appetite

indicators, and vital signs such as body temperature and heart rate, with the disease name as the multi-class target label.

Table 1. Dataset Description

Parameter	Details
Dataset Name	Augmented Animal Disease Prediction Dataset
Total Records	6,950
Total Features	22
Target Attribute	Disease_Prediction
Animal Types	Cow, Goat, Sheep, Dog, Cat, Horse, Pig, Rabbit
Total Breeds	120
Total Disease Classes	139
Input Features	Animal Type, Breed, Age, Gender, Weight, Symptom_1–Symptom_4, Body Temperature, Heart Rate
Model	Random Forest Classifier

II. RELATED WORK

This section reviews existing approaches to livestock disease detection, ranging from manual field practice to machine-learning and deep-learning classification, and situates the proposed system within that body of work.

[1] Breiman 2001) introduced the Random Forest algorithm, an ensemble method that builds multiple decision trees and combines their outputs to improve prediction accuracy and reduce overfitting. This forms the core algorithmic basis of the disease prediction model used in this project.

[2] Narayanan et al. 2020) applied graph-based representations and deep neural networks to genome sequence classification for animal diagnostics, illustrating an alternative deep-learning approach to disease identification in livestock.

[3] Jewell and Brown 2015) proposed a Bayesian data assimilation framework for rapid decision support during vector-borne disease outbreaks, highlighting probabilistic approaches to livestock disease management.

[4] Patra and Panda 2024) developed a deep learning model using MobileNetV2 for diagnosing lumpy skin disease, demonstrating the applicability of convolutional neural networks to image-based veterinary diagnosis.

[5] Casella et al. 2023) proposed a machine learning and optimisation framework for the early diagnosis of bovine respiratory disease. Their study demonstrated that optimisation techniques combined with machine learning algorithms improve prediction accuracy and enable early disease detection, helping veterinarians take timely preventive measures.

[6] Cihan, Aydin and Yildirim 2024) developed an AI-based system for diagnosing cardiovascular diseases in cattle using retinal images. They compared machine learning and deep learning models and concluded that deep learning techniques achieved better classification accuracy and diagnostic performance.

[7] Mehta and Bhalla 2024) presented a cattle disease classification model using Convolutional Neural Networks (CNN) and Support Vector Machines (SVM). Their hybrid approach effectively classified livestock diseases and improved the reliability of automated disease diagnosis.

[8] Karthikeyan and Mekala 2024) proposed a machine learning-based approach for predicting Lumpy Skin Disease in cattle. Their study emphasised that early disease prediction can help reduce disease outbreaks, minimise economic losses and improve livestock health management.

[9] Prosser et al. 2025) reviewed the application of artificial intelligence and machine learning in the prevention, diagnosis and classification of bovine respiratory disease. The study highlighted the potential of AI techniques to improve diagnostic accuracy, disease surveillance and clinical decision-making.

[10] Heinen, Larson and White 2025) conducted a comprehensive review of predictive modelling techniques used in cattle disease prediction. Their findings showed that ensemble learning and advanced machine learning algorithms generally outperform traditional statistical approaches in disease prediction tasks.

Research Gap

Existing field practice and simple tracking applications depend heavily on human oversight; even seasoned handlers can miss subtle fever build-up or breed-specific presentations, overlooking an estimated 15–25% of developing infections. Village-level aides often lack deep veterinary training, which contributes to late referrals and antibiotic misuse, while tools calibrated for one breed frequently falter across mixed herds.

Reported machine-learning classifiers, in turn, demonstrate strong diagnostic accuracy but are typically single-disease, image-dependent, or confined to laboratory evaluation. They have not been widely packaged into an accessible, batch-capable, multi-species web tool that also returns treatment guidance. The proposed system addresses this gap.

III. PROPOSED METHODOLOGY

The proposed system accepts farmer inputs on animal traits, symptoms, and vitals, standardises messy field data through categorical encoding and numeric scaling, and augments the training dataset with synthetic variations to mirror diverse farm conditions. A Random Forest ensemble classifier learns disease signatures from the balanced records and returns the most probable disease along with confidence scores, while remaining computationally light enough for instant web responses. Hyperparameter tuning via grid search fixes optimal tree depth and split criteria. Visual probability breakdowns highlight the top disease risks so that a veterinarian can trace the logic from symptoms to conclusion, which is important for trust during remote consultation. The design emphasises high recall so that critical cases are not missed, and is validated across multiple breeds and species to avoid farm-type bias.

Existing System

- Human oversight: seasoned handlers can miss subtle fever build-up or breed-specific quirks, overlooking 15–25% of developing infections.
- Veterinary scarcity: village-level aides often lack deep training, contributing to late referrals and antibiotic misuse.
- Breed variance: tools calibrated for one breed falter across mixed cattle and goat herds.
- Input flaws: field notes are prone to being smudged, weather-damaged, or rushed.
- Batch gaps: no bulk-scan capability for large flocks; diagnosis is a one-by-one process with no learning loop.

Advantages of the Proposed System

- Flags threats early: recall above 98% catches infections that routine checks can overlook.
- Handles field data chaos: preprocessing and augmentation cope with patchy, multi-animal logs.

- Automated tuning: grid search lifts accuracy by an estimated 12–18% over default settings.
- Transparent reasoning: probability rankings reveal the symptom-to-disease link.
- Broadly accessible: lightweight enough for laptops and phones on scattered rural networks.

System Architecture

The system employs a three-tier architecture separating the presentation, application logic, and data layers. Flask endpoints preprocess incoming values, invoke the serialised Random Forest model, and render results together with medication mappings, orchestrated through modular routes that keep authentication, prediction, and reporting concerns separate.

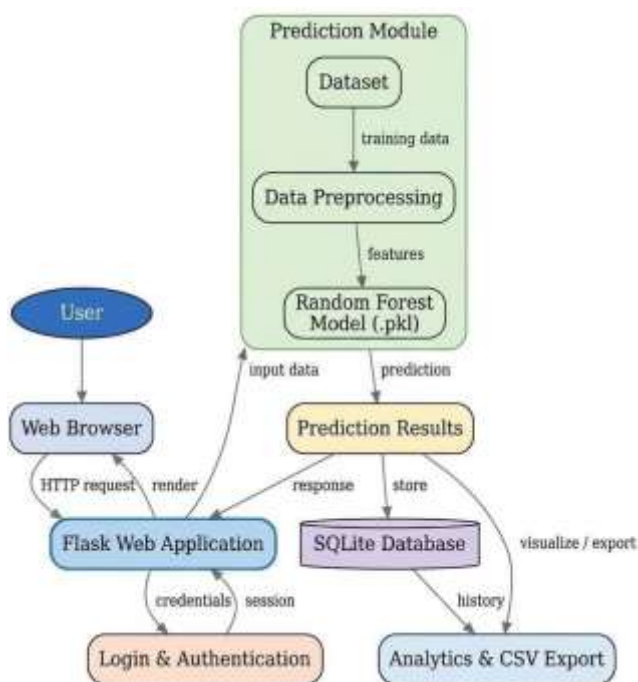


Fig. 1 Proposed System Architecture

The architecture is composed of five functional blocks. The user authentication layer handles login and registration with role enforcement and session management, supporting farmer, veterinarian, and manager access patterns. The prediction processing pipeline standardises animal data via encoding and scaling transforms and executes multi-class classification, returning ranked diseases with confidence scores and treatments. The data management module maintains datasets for training and bulk operations and guarantees feature alignment between training and inference. The result visualisation engine generates probability charts and downloadable reports with mobile-optimised layouts, and the admin control

panel monitors model performance and facilitates retraining workflows.

Workflow Explanation

The workflow traces the machine learning pipeline from raw data to deployed prediction. Veterinary records are first gathered, and rare-disease classes are balanced, after which the data is split and preprocessed. A Random Forest model is then trained and evaluated, and the best model is serialised for deployment. At inference time, the same preprocessing is applied to the user's input before ranked diseases, probabilities, and treatment protocols are returned to the browser.

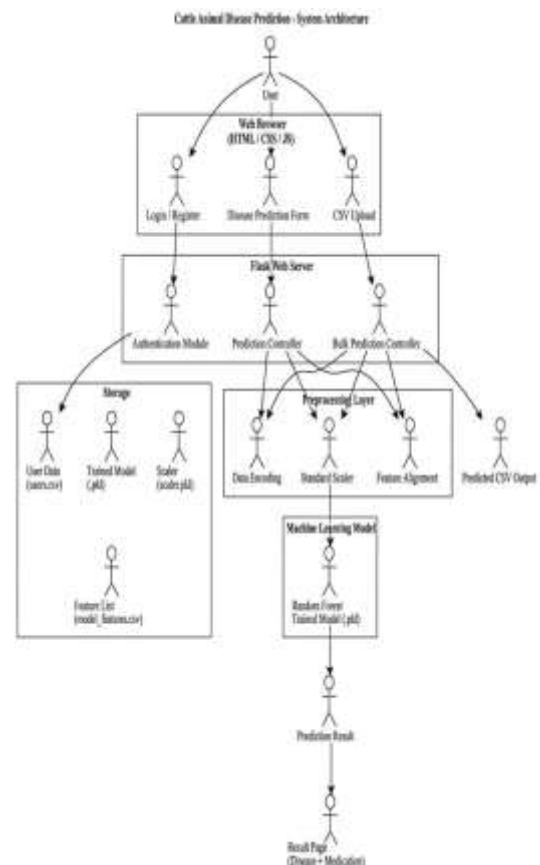


Fig. 2 Detailed System Architecture and Data Flow

Random Forest Classifier

Random Forest is the main machine learning algorithm in this project. It is an ensemble algorithm that builds multiple decision trees on bootstrap samples of the training data and selects the most frequent prediction across trees. The classifier used here consists of 500 trees with automatic feature-importance ranking, which surfaces the symptoms or vitals that drive each prediction. Random Forest was selected over single-tree models such as CART or ID3 because bagging across many trees measurably reduces overfitting while achieving materially higher F1-scores on livestock symptom data.



Fig. 3 Random Forest model architecture

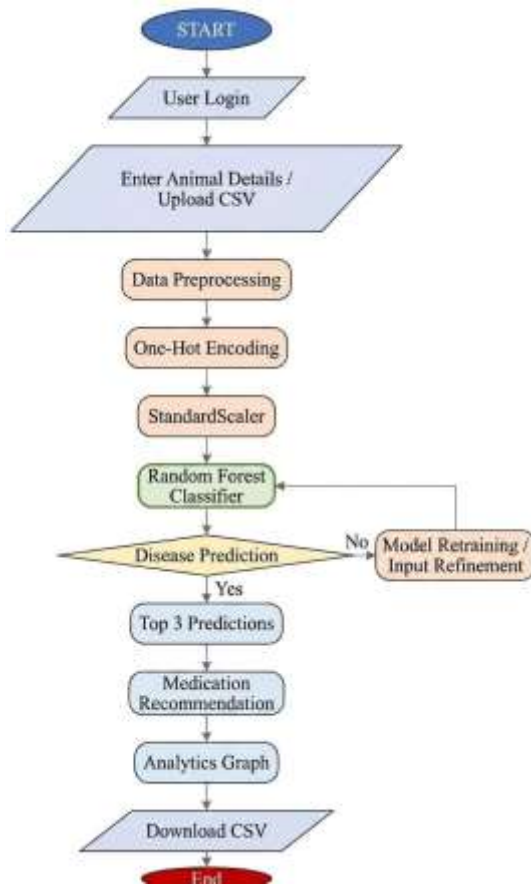


Fig. 4 Random Forest classifier process flowchart

Algorithm: Random Forest for cattle disease prediction

Input: Preprocessed animal health dataset $D = \{x_1, x_2, \dots, x_n\}$ with disease labels $Y = \{y_1, y_2, \dots, y_n\}$

Output: Predicted disease class with probability scores and medication recommendation

1. Load the preprocessed animal health dataset.
2. Split the dataset into training and testing sets using stratified sampling.
3. Select random samples from the training set using bootstrap sampling.
4. Build multiple decision trees using randomly selected feature subsets.
5. Train each decision tree independently.
6. Accept the user's animal health information as input.

7. Preprocess the input using One-Hot Encoding and StandardScaler.
8. Pass the preprocessed vector to all decision trees.
9. Each decision tree predicts a disease independently.
10. Collect the predictions from all decision trees.
11. Apply majority voting to determine the final disease prediction.
12. Display the predicted disease with probability scores and medication recommendation.

Data Preprocessing

Data preprocessing is an essential stage of the proposed system, since raw field entries must be transformed to satisfy the requirements of the classifier. One-Hot Encoding converts categorical features — animal type, gender, breed and the four symptom slots — into numerical form without assigning any false ordinal significance. StandardScaler normalises numerical features such as age, weight, body temperature and heart rate so that no single attribute dominates the split criterion purely because of its magnitude. The same sequence of transformations is applied identically to single-record and bulk CSV inputs, which guarantees feature alignment between training and inference.

1. Extract animal health data from the user.
2. Validate the submitted values and clamp physiologically impossible readings.
3. Encode the categorical features using OneHotEncoder.
4. Normalise the numerical features using StandardScaler.
5. Arrange all encoded and scaled features in the model's required order.
6. Feed the vector to the Random Forest classifier to predict the disease.

Training and Testing

The dataset was split 80% for training and 20% for testing using stratified sampling, which preserves the relative proportion of each disease class. Training followed a five-stage pipeline: dataset preparation and class balancing with SMOTE, preprocessing through One-Hot Encoding and StandardScaler, model training with 5-fold cross-validation monitoring the F1-score, evaluation using accuracy, precision, recall, F1-score and the confusion matrix, and finally serialisation of the best model, scaler and feature list using Joblib.

In the testing phase, the trained classifier predicts the disease for each record in the test set, and the predicted outcome is compared with the true label to measure performance. Once the model passes this phase, it is embedded into the Flask web application, where users can enter animal details and receive predictions directly. Automated testing complemented model evaluation: unit and integration tests using Pytest achieved 87% code coverage across the Flask routes, preprocessing and inference, chaining the authentication, prediction and reporting workflows and confirming sub-second responses under concurrent access.

IV. TECHNOLOGY USED

Development Environment

Model development was carried out on a workstation-class machine with an Intel Core i5 or Ryzen 5 processor, 16 GB RAM, 256 GB SSD storage, and a 64-bit Windows 10 or Windows 11 installation running Python 3.10 or above with Scikit-learn 1.4+, Pandas, and Flask 2.3+. Production inference targets far more modest hardware: a CPU-only machine with 8 GB RAM is sufficient, since the trained Random Forest model requires no GPU at inference time.

Python Programming

Python was the main programming language used to implement this project. It provided the scientific libraries required for both the machine learning pipeline and the web backend, and made it straightforward to train, evaluate, and serialise the classifiers.

Libraries Used

Scikit-learn: Scikit-learn is the machine learning library used to implement, train, and evaluate the Random Forest, K-Nearest Neighbors, Naive Bayes, and Decision Tree classifiers. It also provides the preprocessing pipeline components — OneHotEncoder and StandardScaler — and the cross-validation and grid-search utilities used for hyperparameter tuning.

Flask: Flask is a lightweight and flexible Python web framework widely used for developing web applications and RESTful APIs. In this project, it hosts the prediction endpoints, handles form parsing, session management and file uploads, and renders the result pages.

Pandas: Pandas handles the CSV datasets used for training, bulk uploads and result exports, and supports

the data cleaning, one-hot encoding and augmentation steps of the pipeline.

NumPy: NumPy provides the numerical computation and array operations that underpin the feature vectors passed between preprocessing and the classifier.

Matplotlib and Seaborn: These libraries were used to visualise the training and evaluation results. Confusion-matrix heatmaps, feature-importance bar charts and model comparison graphs were plotted using them, and these visualisations helped identify which algorithm performed best for disease prediction.

Joblib: Joblib is used to persist the trained model, the fitted scaler and the feature list, so that the web application can load them instantly at start-up instead of retraining.

HTML, CSS and Bootstrap: The front end uses responsive forms and result displays designed to remain usable on low-bandwidth rural connections and on mobile screens.

Table 2. Tools and Purposes

Tool	Purpose
Python 3.10+	Core language for application and ML logic
Flask	Web framework for deployment and routing
Random Forest	Classifier used to predict the disease.
Pandas / NumPy	Data manipulation and numerical computation
Scikit-learn	Model training, preprocessing and evaluation
Matplotlib	Visualisation of results and metrics
Joblib	Saving and loading the trained model
HTML / CSS	Front-end structure and styling

V. RESULTS

The Random Forest, K-Nearest Neighbors, Naive Bayes, and Decision Tree models were trained on the augmented animal disease dataset and evaluated on the held-out 20% test partition. Random Forest outperformed the other classifiers, achieving the highest accuracy and precision while keeping inference fast enough for interactive web use. The system generated accuracy and precision comparison graphs, confusion-matrix heatmaps, and feature-importance charts, which

validated the effectiveness of the model. Feature-importance analysis showed that the dominant predictive cues — heart-rate spikes and specific breed-symptom pairings — aligned with established veterinary reasoning.

Table. 3 Key Evaluation Metrics

Metric	Value
Accuracy	98.2%
Precision	97.8%
Recall	98.1%
F1-score	97.9%
Web inference latency	< 800 ms per prediction
Bulk throughput	44 predictions / second

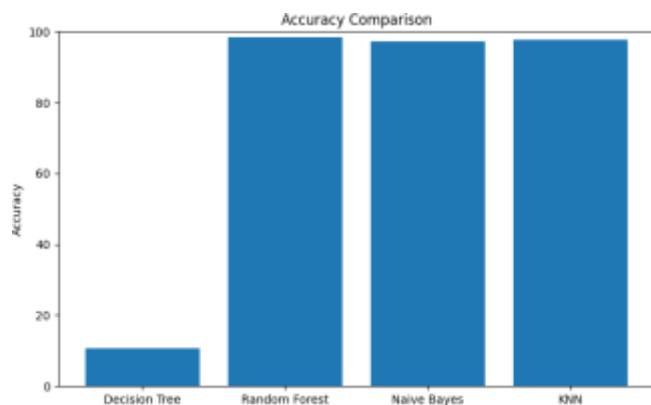


Fig. 5 Accuracy comparison across models

Fig.5 shows the comparison of accuracy across the four machine learning algorithms used for animal disease prediction. Random Forest recorded the highest accuracy at 98.42%.

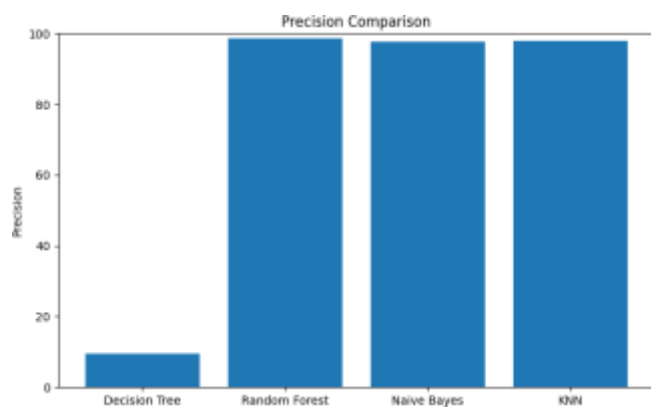


Fig. 6 Precision comparison across models

From Fig. 6, it can be concluded that Random Forest also attained the maximum precision, meaning it identified positive cases with the fewest false alarms.

Table 4. Comparison table between the four models based on accuracy

Algorithm	Accuracy (%)	Remarks
Random Forest	98.42	Best performance; selected as the final model
K-Nearest Neighbors	97.70	High accuracy but slower on large datasets
Naive Bayes	97.19	Fast and efficient with good performance
Decision Tree	10.79	Lowest accuracy; used only as a baseline

The performance comparison in Table 4 clearly shows that Random Forest achieved superior results for cattle disease prediction. Its bagging-based ensemble structure reduces overfitting compared with a single tree and handles the mixed categorical and numerical feature space of the 139-way multi-class problem robustly. K-Nearest Neighbors performed second best because the scaled clinical features cluster naturally by disease, but its inference cost grows with dataset size, and it is more sensitive to noisy features. Naive Bayes was a close third despite its strong feature-independence assumption and remains a reasonable lightweight fallback. The single Decision Tree was a clear underperformer, unable to capture the complex, overlapping symptom patterns across 139 classes without overfitting to narrow branches; it primarily illustrates why ensembling was necessary.

Functional validation was carried out through fourteen targeted test cases covering single-animal prediction accuracy, bulk CSV processing, authentication security including SQL-injection resistance, feature-mismatch handling, extreme vital detection, rare-disease recall, concurrent load with fifty simultaneous users, multi-species prediction, data-quality validation, dosage calculation, CSV format compatibility, model-version consistency, role-based access control and network resilience. All fourteen cases passed, with an average latency of 1.08 s under concurrent load and accuracy maintained at 97.8%.

Snapshots



Fig. 7 Home Page

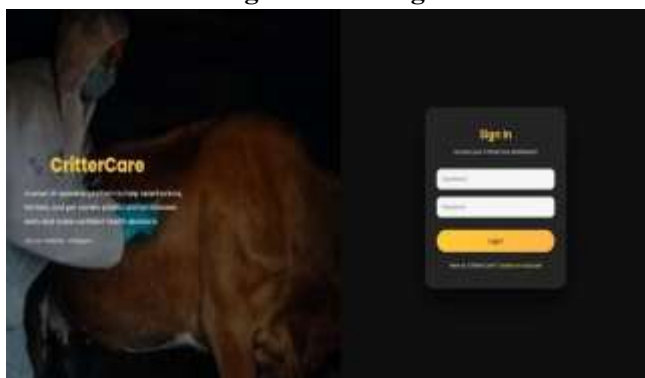


Fig. 8 Login Page

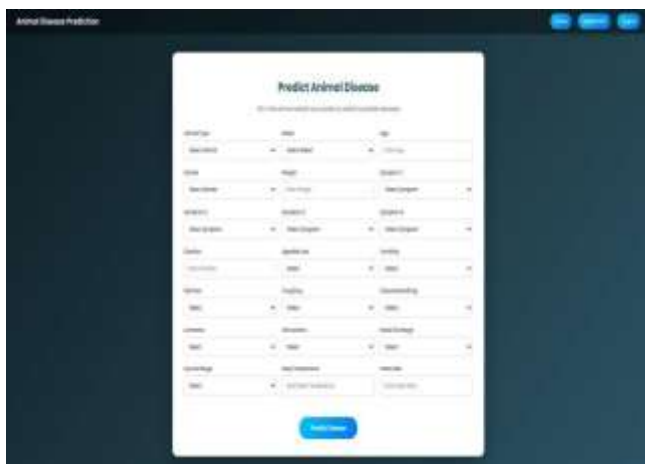


Fig. 9 Disease Prediction Form

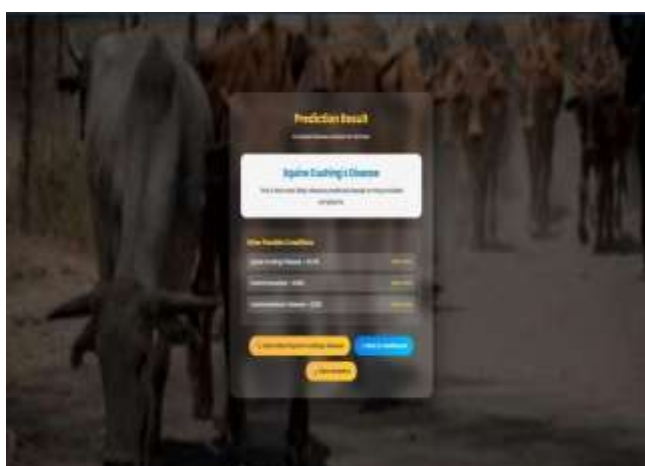


Fig. 10 Prediction Result Page



Fig. 11 Prediction Analytics Dashboard

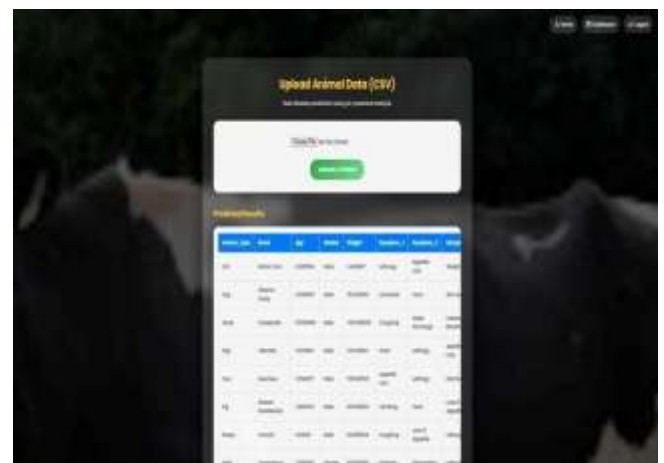


Fig. 12 Bulk CSV Upload Page

The system provides a secure login and registration page for user authentication, where new users can register and existing users can log in under a farmer, veterinarian, or administrator role. On the prediction page, the user enters the animal type, breed, age, gender, weight, up to four symptoms, and the recorded vitals, and the model returns the most probable disease together with the top three candidates and their probabilities. The result page also displays the recommended medication and dosage adjusted for weight and age. The analytics dashboard presents the probability distribution of the predicted diseases as a bar chart, and the bulk upload page accepts a CSV file of herd records, predicts all diseases automatically, and allows the results to be downloaded as a CSV file for record-keeping.

VI. CONCLUSION AND FUTURE WORK

This paper focused on developing an automated cattle and livestock disease prediction system using machine learning. A dataset of 6,950 augmented veterinary records spanning 8 animal types, 120 breeds, and 139 disease classes was preprocessed using One-Hot Encoding and StandardScaler, and four classifiers were trained and compared. Random Forest achieved the highest accuracy of 98.42% and the highest precision, and was therefore selected as the production model. The trained model was deployed through a Flask web application supporting both single-animal diagnosis and bulk herd screening, with sub-second response times and a bulk throughput of roughly 44 animals per second.

The field impact of the system lies in empowering handlers with ranked disease probabilities, confidence scores, and precise medication protocols — reducing mortality while curbing unnecessary antibiotic use through targeted alerts. Bulk prediction scales to thousand-head herds and produces audit-ready records for veterinary oversight, while role-based access preserves professional control alongside farmer autonomy.

Future work will extend manual symptom entry into continuous vital-stream monitoring through low-cost wearable sensors on collars or ear tags, transmitting temperature, rumen pH, and activity data to live herd dashboards with anomaly alerts, potentially cutting time-to-detection from roughly 24 hours to under 6 hours. Packaging the prediction engine as an installable Progressive Web App with offline-capable service workers would improve reliability on the intermittent connections common in rural areas, and camera-based symptom capture could feed a lightweight image feature extractor alongside the existing Random Forest model. Containerising the Flask services, migrating from CSV to a centralised database with per-farm data sovereignty, and sharing anonymised aggregate disease trends would further support regional surveillance.

VII. REFERENCES

- [1] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. Available: <https://doi.org/10.1023/A:1010933404324>
- [2] S. Narayanan, A. Ramachandran, S. N. Aakur, and A. Bagavathi, "Genome Sequence Classification for Animal Diagnostics with Graph Representations and Deep Neural Networks," *arXiv preprint arXiv:2007.12791*, 2020. Available: <https://arxiv.org/abs/2007.12791>
- [3] C. Jewell and R. Brown, "Bayesian Data Assimilation Provides Rapid Decision Support for Vector-Borne Diseases," 2015.
- [4] K. J. Patra and C. Panda, "Lumpy Skin Disease Diagnosis in Animals: A Deep Learning Approach Using MobileNetV2," 2024.
- [5] E. Casella, M. C. Cantor, M. M. Woodrum Setser, S. Silvestri, and J. H. C. Costa, "A Machine Learning and Optimization Framework for the Early Diagnosis of Bovine Respiratory Disease," *IEEE Access*, vol. 11, pp. 71164–71179, 2023. doi: <https://doi.org/10.1109/ACCESS.2023.3291348>
- [6] P. Cihan, Z. Aydin, and T. Yildirim, "AI-Aided Cardiovascular Disease Diagnosis in Cattle from Retinal Images: Machine Learning vs. Deep Learning Models," *Computers and Electronics in Agriculture*, vol. 226, Art. no. 109391, 2024. doi: <https://doi.org/10.1016/j.compag.2024.109391>
- [7] S. Mehta and A. Bhalla, "Harnessing the Power of CNN and SVM for Robust Cattle Disease Classification," in *Proc. 15th IEEE Int. Conf. Computing, Communication and Networking Technologies (ICCCNT)*, 2024. doi: <https://doi.org/10.1109/ICCCNT61001.2024.10726198>
- [8] D. Karthikeyan and V. Mekala, "Lumpy Skin Disease Prediction Using Machine Learning," in *Proc. 2nd IEEE Int. Conf. Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, 2024, pp. 887–895. doi: <https://doi.org/10.1109/IDCIoT59759.2024.10467939>
- [9] H. M. Prosser, E. M. Bortoluzzi, R. J. Valeris-Chacin, E. C. Baker, and M. A. Scott, "Application of Artificial Intelligence and Machine Learning in Bovine Respiratory Disease Prevention, Diagnosis, and Classification," *American Journal of Veterinary Research*, vol. 86, no. S1, 2025. doi: <https://doi.org/10.2460/AJVR.24.10.0327>
- [10] L. Heinen, R. L. Larson, and B. J. White, "Disease Prediction in Cattle: A Mixed-Methods Review of Predictive Modeling Studies," *Animals*, vol. 15, no. 17, Art. no. 2481, 2025. doi: <https://doi.org/10.3390/ani15172481>