Software Requirements Specification

for

Predicting Lumpy Skin Disease in Cattle through Machine Learning

Version 1.0 approved

Prepared by:

Akanksha Bhosale, Suyash Ahiwale, Ankit Kalyani, Riya Khuspe

Sinhgad College of Engineering, Pune

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this system is to provide an efficient tool for predicting LSD outbreaks in cattle, thereby enhancing disease management and minimizing economic losses in the livestock sector**.**

1.2 Document Conventions

- **1. Title Page**
- **Project Title:** Predicting Lumpy Skin Disease in Cattle Using Machine Learning
- **Date:**
- **Version:** 1.0
- **Authors:** Akanksha Bhosale, Suyash Ahiwale, Ankit Kalyani, Riya Khuspe
- **Organization:** Sinhgad College of Engineering, Pune
- **2. Table of Contents**
- Include headings and subheadings with page numbers for easy navigation.
- **3. Formatting**
- **Font:** Use a clear, professional font (e.g., Arial, Times New Roman).
- **Font Size:** 11 or 12 pt for body text, 14-16 pt for headings.
- **Line Spacing:** 1.5 or double spacing for readability.
- **Margins:** Standard 1-inch margins on all sides.
- **Sections:** Use numbered headings (e.g., 1. Introduction) for organization.
- **4. Section Headings**
- Use consistent formatting for headings:
    - **Main Sections:** Bold and larger font (e.g., 14 pt)
    - **Subsections:** Bold and slightly smaller (e.g., 12 pt)
    - **Sub-subsections:** Times New Roman font, (e.g., 12 pt)
- **5. Visuals and Tables**

- **Figures and Tables:** Numbered and titled (e.g., Figure 1: Data Collection Process).
- **Captions:** Provide clear explanations for visuals.
- **Charts and Graphs:** Use appropriate colors and labels to enhance clarity.
- **6. Code and Algorithms**
- **Code Blocks:** Use a monospaced font (e.g., Courier New) for any code snippets.
- **Comments:** Include comments in code to explain logic and functionality.
- **7. References**
- Use a consistent citation style (e.g., APA, IEEE) for any references to literature or existing systems.
- Include a "References" section at the end of the document.
- **8. Revision History**
- Maintain a section at the start of the document to track changes, including dates and descriptions of updates.

.

1.3 Intended Audience and Reading Suggestions

Intended Audience

1. Farmers and Livestock Owners-
- Needs: Early detection of LSD to mitigate losses, manage herd health, and improve productivity.
- Suggestions:
- Develop a user-friendly interface for data input and prediction results.
- Provide educational resources on LSD symptoms and prevention.

2. Veterinarians
- Needs: Accurate prediction tools to assist in diagnosis and treatment plans for affected cattle.
- Suggestions:
- Include detailed reports on predictions with recommendations for action.
- Allow for integration with veterinary management systems for seamless workflow.

3. Agricultural Researchers

- Needs: Insights into disease patterns, risk factors, and the efficacy of ML models in agricultural settings.
- Suggestions:
- Share data and model performance metrics to facilitate further research.
- Publish findings in relevant journals or conferences.

4. Government and Regulatory Bodies

- Needs: Tools to monitor and control outbreaks, ensuring public health and livestock safety.
- Suggestions:
- Provide access to aggregated data and trends for policy-making and response strategies.
- Ensure compliance with veterinary health regulations in the application of the model.

Suggestions for the Project

1. Data Collection and Management

- Ensure the data used for training the model is diverse, high-quality, and representative of different regions and cattle breeds.

2. Model Transparency

- Make the model's workings transparent to build trust among users. Explain how predictions are made and the significance of various input features.

3. User Feedback Mechanism

- Incorporate a system for users to provide feedback on predictions, which can help refine and improve the model over time.

4. Scalability

- Design the system to be scalable, allowing it to accommodate larger datasets or additional features in the future.

5. Continuous Learning

- Implement mechanisms for the model to learn from new data, ensuring its predictions remain relevant and accurate as conditions change.

1.4 Product Scope

-Target Users: Farmers, veterinarians, agricultural scientists.

- Functionality: Data collection, preprocessing, model training, prediction, and reporting.

1.5 References

[1] E. Chandralekha, et al. (2023). "Predicting and Analyzing Lumpy Skin Disease Using Ensemble of Machine Learning Models." 2023 Global Conference on Information Technologies and Communications (GCITC). DOI: 10.1109/GCITC60406.2023.10425950

[2] Vidur Sharma, et al. (2023). "Lumpy Skin Disease Detector." 2023 Seventh International Conference on Image Information Processing (ICIIP). DOI: 10.1109/ICIIP61524.2023.10537770

[3] A. Thakallapelli, S. Ghosh, S. Kamalanayan (2016). "Real-time frequency based reduced order modeling of large power grid." IEEE Power and Energy Society General Meeting. DOI: 10.1109/PESGM.2016.7741877

[4] W.H. Ahmed, et al. (2022). "Development of a machine learning model for the diagnosis of lumpy skin disease using clinical signs and symptoms." IEEE Access. DOI: 10.1109/ACCESS.2019.2923963

[5] Singh, et al. (2023). "Use of machine learning for the prediction of lumpy skin disease outbreaks." 2023 IEEE Conference on Data Analytics for Business and Industry (ICDABI). DOI: 10.1109/ICDABI56818.2023.9746456

2. Description

2.1 Product Perspective

Problem Statement

- Lumpy Skin Disease is a viral infection affecting cattle, leading to economic losses due to reduced productivity, increased veterinary costs, and trade restrictions. Early detection and prediction can help mitigate these impacts.

Target Users

- Farmers: To monitor cattle health and make informed decisions.
- Veterinarians: To provide timely interventions and treatments.
- Agricultural Authorities: For disease surveillance and control measures.

Product Goals

- Early Detection: Develop a system that predicts the likelihood of LSD outbreaks in cattle.
- User-Friendly Interface: Create an intuitive dashboard for users to input data and receive predictions.
- Real-Time Alerts: Implement notifications for farmers and vets regarding potential outbreaks.

Features

- Data Collection: Gather data on cattle health, environmental conditions, and historical outbreak patterns.
- Predictive Analytics: Utilize machine learning algorithms to analyze data and predict disease occurrences.
- Visualization Tools: Provide graphical representations of data trends, outbreak forecasts, and risk levels.
- Mobile Compatibility: Ensure accessibility through mobile devices for on-the-go monitoring.

Technology Stack

-Data Sources: IoT devices, health records, weather data, and satellite imagery.

- Machine Learning Frameworks: TensorFlow, Scikit-learn, or PyTorch for model development.
- Cloud Services: AWS or Azure for data storage and processing.
- Front-End Development: React or Angular for building the user interface.

Data Requirements
- Historical health data of cattle.
- Weather patterns and climate data.
- Geographic distribution of past outbreaks.
- Socio-economic factors affecting farming practices.

Market Analysis
- Assess the demand for cattle health monitoring solutions.
- Identify competitors and analyze their offerings.
- Explore potential partnerships with agricultural organizations and veterinary services.

Implementation Plan
-Phase 1: Data collection and preliminary research.
- Phase 2: Model development and testing.
- Phase 3: User testing and feedback loop.
- Phase 4: Launch and marketing strategies.

Metrics for Success
- Accuracy of predictions (measured against actual outbreak occurrences).
- User engagement metrics (number of active users, feedback ratings).
- Reduction in LSD incidence on farms using the solution.

Long-Term Vision
- Expand the platform to include predictions for other livestock diseases.
- Integrate with broader agricultural management systems for comprehensive farm health monitoring.

2.2 Product Functions

1. Data Collection

- User Input:

- Allow farmers and veterinarians to input data on cattle health, demographics (age, breed), and environmental conditions (temperature, humidity).

- Automated Data Integration:

- Integrate with existing farm management systems to automatically collect relevant data.

- Data Sources:

- Aggregate historical data on LSD outbreaks, veterinary records, and environmental factors from public databases.

2. Data Preprocessing

- Data Cleaning:

- Identify and rectify inaccuracies or missing values in the dataset.

- Normalization:

- Standardize data formats and scales to ensure uniformity across inputs.

- Feature Selection:

- Analyze and select the most relevant features that impact LSD prediction, enhancing model performance.

3. Machine Learning Model Development

- Model Training:

- Train various machine learning models (e.g., decision trees, random forests, neural networks) using historical data to identify patterns associated with LSD.

- Model Evaluation:

- Evaluate model performance using metrics such as accuracy, precision, recall, and F1 score to ensure reliability.

- Hyperparameter Tuning:

- Optimize model parameters to improve prediction accuracy through techniques like grid search or random search.

4. Prediction Generation

- Real-time Predictions:
- Generate predictions based on user inputs and provide a risk assessment for LSD outbreaks.
- Confidence Scores:
- Present confidence levels for each prediction, helping users gauge the reliability of the results.

5. User Notifications and Alerts

- Alert System:
- Notify users of potential LSD outbreaks based on predictions, allowing for timely intervention.
- Health Recommendations:
- Provide actionable recommendations based on predictions (e.g., vaccination, isolation of affected cattle).

6. Visualization and Reporting

- Dashboard:
- Offer a user-friendly dashboard displaying key metrics, prediction results, and health alerts in real-time.
- Historical Trends:
- Visualize historical data trends related to LSD outbreaks, helping users understand patterns over time.

- Exportable Reports:
  - Generate reports summarizing predictions and health recommendations, which can be shared with stakeholders.

8. Feedback Mechanism

- User Feedback Collection:
  - Enable users to provide feedback on predictions, which can be used to refine and improve the model.
- Continuous Learning:

- Incorporate feedback to enhance the accuracy and relevance of predictions over time.

9. Compliance and Security

- Data Privacy:
  - Ensure user data is stored securely and complies with relevant data protection regulations.
- Audit Trails:
  - Maintain logs of data inputs and predictions for accountability and regulatory compliance.

2.3 User Classes and Characteristics

1. Farmers and Livestock Owners

- Characteristics:
- Typically have hands-on experience with cattle management.
- Varying levels of technical expertise; may have limited knowledge of machine learning or data analytics.
- Focused on maximizing herd health and productivity while minimizing costs.

2. Veterinarians

- Characteristics:
- Professionals with a medical background in animal health.
- Familiar with livestock diseases, including LSD, and their implications.
- Often involved in making treatment and management decisions based on health data.

3. Agricultural Researchers

- Characteristics:
- Experts in veterinary science, agriculture, or data science.
- Engaged in studying livestock diseases, agricultural practices, or machine learning applications in farming.
- Interested in analyzing patterns, validating models, and publishing findings.

4. Government and Regulatory Bodies

- Characteristics:

- Officials and analysts focused on livestock health and disease control at local, regional, or national levels.

- Involved in setting policies, regulations, and guidelines for disease management.

5. Tech Developers and Data Scientists

- Characteristics:

- Professionals responsible for building and maintaining the machine learning models and software infrastructure.

- Skilled in programming, data analysis, and understanding of machine learning algorithms.

2.4 Operating Environment

1. Hardware Requirements

- User Devices:

- Farmers and Veterinarians:

- Smartphones or tablets (iOS and Android) for mobile applications.

- Desktop or laptop computers with internet access for web-based interfaces.

- Developers and Researchers:

- High-performance computers or servers for model training and data processing.

- Server Infrastructure:

- Cloud-based servers (e.g., AWS, Google Cloud, Azure) for data storage, model training, and deployment.

- Optionally, local servers for organizations preferring to manage data on-premises.

2. Software Requirements

- Frontend Technologies:

- Web applications developed using frameworks like React, Angular, or Vue.js.

- Mobile applications built using frameworks like Flutter or React Native for cross-platform compatibility.

- Backend Technologies:

- Server-side languages (e.g., Python, Node.js) to handle data processing and model inference.

- Databases (e.g., PostgreSQL, MongoDB) for storing user data, predictions, and historical records.

- Machine Learning Frameworks:

- Libraries such as TensorFlow, scikit-learn, or PyTorch for developing and training predictive models.

- Data processing libraries like Pandas and NumPy for handling data cleaning and preprocessing.

3. Network Requirements

- Internet Connectivity:

- Reliable internet connection for users to access cloud-based features, submit data, and receive predictions.

- Options for offline data entry, which can be synced when internet access is available.

4. Security Requirements

- Data Protection:

- Implementation of secure data transmission protocols (e.g., HTTPS) to safeguard sensitive information.

- Authentication and authorization mechanisms (e.g., OAuth, JWT) to ensure only authorized users can access the system.

- Compliance:

- Adherence to data protection regulations (e.g., GDPR, HIPAA) to ensure user privacy and data security.

5. Operating Systems

- User Devices:

- Compatibility with major operating systems, including Windows, macOS, iOS, and Android.

- Server Environments:

- Use of Linux-based operating systems (e.g., Ubuntu, CentOS) for server deployments due to their stability and support for various software tools.

2.5 Design and Implementation Constraints

1. Data Constraints

- Data Availability:

- Limited access to high-quality, historical data on LSD outbreaks and relevant environmental factors may hinder model training and validation.

- Data Quality:

- Inconsistent data formats, missing values, and inaccuracies in the collected data can impact model performance.

- Data Privacy:

- Compliance with data protection regulations (e.g., GDPR) is essential when handling sensitive user and animal health data.

2. Technical Constraints

- Computational Resources:

- The need for significant computational power for training complex machine learning models may require cloud-based solutions, which can incur additional costs.

- Integration Challenges:

- Integrating the system with existing farm management software and data sources may pose compatibility issues, requiring additional development effort.

- Model Complexity:
- Balancing the complexity of the machine learning models with the need for interpretability is crucial, especially for users without technical backgrounds.

3. User Interface Constraints

- User Experience:
- Designing an intuitive and user-friendly interface that caters to users with varying levels of technical expertise can be challenging.

- Mobile vs. Desktop:
- Ensuring that the application performs well on both mobile devices and desktop environments may require responsive design and testing.

4. Environmental Constraints

- Connectivity Issues:
- Limited internet access in rural areas could impact data submission and real-time prediction capabilities. Solutions for offline data entry may need to be considered.

- Hardware Limitations:
- Users may have older devices that could affect the performance of mobile and web applications, necessitating optimizations for lower specifications.

5. Regulatory Constraints

- Compliance Requirements:
- The system must adhere to veterinary regulations and agricultural standards, which may influence the design and implementation of features related to disease management.

- Approval Processes:

- Gaining approvals from relevant authorities for data collection and use may slow down the project timeline.

6. Budget and Time Constraints

- Resource Allocation:

- Budget limitations may restrict the ability to hire additional personnel or invest in advanced technologies necessary for model development and deployment.

- Timeline Pressures:

- Tight project timelines can lead to compromises in testing and validation, which could affect the reliability and effectiveness of the final product.

2.6 User Documentation

Introduction

This system is designed to predict the occurrence and spread of Lumpy Skin Disease (LSD) in cattle using machine learning models. The tool aids farmers, veterinarians, and livestock management personnel in taking timely preventive measures to protect cattle and mitigate the spread of the disease.

System Overview

The software utilizes machine learning classifiers like Random Forest, LightGBM (LGBM), and Artificial Neural Network (ANN) to analyze historical and environmental data. The system predicts the likelihood of LSD outbreaks in specific regions and provides an assessment of potential risk levels.

Installation and Setup

1. System Requirements

Operating System: Windows, macOS, or Linux

RAM: At least 4 GB

Disk Space: Minimum 2 GB free space

Python version: 3.8 or higher

Required Libraries: scikit-learn, pandas, NumPy, matplotlib, keras, OpenCV, and any other necessary packages

2. Installation Steps

Install Python: Python Downloads

Install necessary libraries using the command:

pip install scikit-learn pandas NumPy matplotlib seaborn

Download the project files from the provided repository.

Using the Application

1. Launching the Application

Navigate to the project directory.

Run the main script using the command:

python main.py

2. Input Data Requirements

Data should include features such as cattle population density, previous disease history, weather conditions (temperature, humidity), vaccination status, and any other relevant factors.

3. Predicting Disease Outbreak

Once the application is running, upload the dataset in CSV format.

Choose the desired machine learning model for prediction (e.g., Random Forest, LGBM, ANN).

Configure model parameters if needed, or use default settings.

Click on the "Predict" button to generate the results.

The prediction output will show the probability of an LSD outbreak in the given area.

4. Understanding the Results

Probability Score: Displays the likelihood of an LSD outbreak.

Risk Assessment: Provides risk levels (Low, Medium, High) based on the prediction.

Model Accuracy Metrics: Includes performance evaluation metrics like accuracy, precision, recall, and F1-score.

System Features

Interactive Dashboard: Visualizes prediction results, historical trends, and risk assessment.

Model Selection: Allows users to choose from different machine learning models.

Parameter Tuning: Users can modify model parameters to optimize prediction performance.

Data Visualization: Displays graphs for data trends, feature importance, and prediction results.

Troubleshooting

1. Prediction Errors

Ensure the input data is in the correct format (CSV) and includes all necessary features.

Verify that all required Python libraries are installed.

2. Application Crashes

Check for compatibility issues with the operating system and Python version.

Ensure adequate system memory is available.

2.7 Assumptions and Dependencies

Assumptions

1.Data Availability:

- Sufficient historical data on lumpy skin disease incidents, environmental conditions, and cattle demographics will be accessible for model training and validation.

2. User Engagement:

- Farmers and veterinarians will actively engage with the system, providing timely and accurate data inputs to ensure reliable predictions.

3. Technical Proficiency:

- Users possess a basic level of technical proficiency, enabling them to navigate the system and interpret the outputs effectively.

4. Infrastructure Stability:

- Users will have access to reliable internet connectivity to utilize the web and mobile applications for data input and receiving predictions.

5. Regulatory Compliance:

- The project will comply with all relevant veterinary and data protection regulations, allowing for the lawful collection and use of data.

6. Model Reliability:

- The machine learning models developed will be able to accurately predict the likelihood of LSD outbreaks based on the data provided.

Dependencies

1. Data Sources:

- The system will depend on access to external datasets, including:
  - Historical veterinary records.
  - Environmental data (e.g., weather patterns, humidity, temperature).
  - Local and regional disease outbreak reports.

2. Technological Frameworks:

- The project relies on specific software and frameworks, including:
- Machine learning libraries (e.g., TensorFlow, scikit-learn).
- Frontend and backend technologies (e.g., React for frontend, Python for backend).

3. Cloud Infrastructure:

- The deployment of the application may depend on cloud service providers (e.g., AWS, Google Cloud) for data storage, model training, and application hosting.

4. User Feedback Mechanisms:

- Continuous improvement of the predictive model will depend on the collection of user feedback and real-world outcomes to refine the model's accuracy.

5. Collaboration with Experts:

- The project may depend on collaboration with veterinary experts and agricultural researchers for validation of the model and insights into cattle health management.

3. External Interface Requirements

3.1 User Interfaces

-Web Application: Accessible via standard web browsers.
- Mobile Compatibility: Responsive design for mobile devices.

3.2 Hardware Interfaces

-Server: Minimum specifications for hosting the application.
- User Devices: Basic requirements for accessing the system.

3.3 Software Interfaces

- Programming Languages: Python, JavaScript.
- Frameworks: Flask/Django for backend, React/Vue for frontend.
- Databases: MySQL/PostgreSQL for data storage.

3.4 Communications Interfaces

1. User Interface (UI)
- Web Application:
- Frontend Frameworks: Developed using frameworks like React or Angular to create a responsive and user-friendly interface.
- Dashboard: Displays key metrics, prediction results, and alerts in an intuitive layout.
- Forms for Data Entry: Simple forms for users to input data regarding cattle health, demographics, and environmental conditions.

- Mobile Application:
- Cross-Platform Compatibility: Developed using frameworks like Flutter or React Native to ensure usability on both iOS and Android devices.

- Push Notifications: Alerts users about potential LSD outbreaks and important updates.

2. Backend Interface

-API (Application Programming Interface):

- RESTful API: Facilitates communication between the frontend and backend, enabling data submission and retrieval of predictions.

- Endpoints for Data Handling:

- Data Submission: Endpoint for receiving user input data.

- Prediction Requests: Endpoint for generating and retrieving predictions based on user inputs.

- User Management: Endpoints for user authentication, registration, and profile management.

3. Data Integration Interfaces

- External Data Sources:

- APIs for Data Sources: Integrate with external databases and services to pull in historical data, environmental factors, and veterinary reports.

- CSV/Excel Uploads: Allow users to upload data files in common formats for bulk data entry.

- SMS Notifications:

- Optional SMS alerts for critical updates or predictions, ensuring that users are informed even without internet access.

5. Support and Feedback Interface

- Help Center:

- An integrated help section within the application providing FAQs, tutorials, and troubleshooting guides.

- Feedback Forms:

- Simple forms for users to submit feedback, report issues, or suggest enhancements, which can be reviewed by the development team.

4. System Features

1. User Registration and Management

- Account Creation:
  - Users can create accounts with secure authentication (email/password, social media login).

- Role-Based Access:
  - Different access levels for farmers, veterinarians, researchers, and administrators, ensuring tailored functionality for each user group.

2. Data Input and Management

- Manual Data Entry:
  - User-friendly forms for entering data related to cattle health, demographics, and environmental conditions.

- Automated Data Import:
  - Integration with existing farm management systems to automatically import relevant data (via APIs or file uploads).

- Data Validation:
  - Real-time checks to ensure data integrity and accuracy during input.

3. Prediction Engine

- Machine Learning Models:
  - Implementation of various machine learning algorithms (e.g., decision trees, neural networks) to predict the likelihood of LSD outbreaks.

- Real-Time Predictions:
  - Generate predictions based on user inputs, providing timely insights for decision-making.

- Confidence Scoring:

- Each prediction includes a confidence score indicating the model's reliability.

4. Alerts and Notifications

- Customizable Alerts:

- Users can set preferences for receiving alerts about potential LSD outbreaks, with options for frequency and delivery method (email, SMS).

- Health Recommendations:

- Automated suggestions for preventive measures and treatment options based on predictions.

5. Reporting and Visualization

- Dashboard:

- A central dashboard displaying key metrics, recent predictions, alerts, and historical trends.

- Data Visualization:

- Graphical representations of prediction outcomes, risk assessments, and historical outbreak data for better user comprehension.

- Exportable Reports:

- Ability to generate and download detailed reports summarizing findings, recommendations, and historical data in PDF or Excel formats.

6. User Feedback Mechanism

- Feedback Forms:

- Users can easily provide feedback on predictions and overall system performance.

- Continuous Learning:

- Incorporation of user feedback to refine models and improve prediction accuracy over time.

7. Support and Resources

- Help Center:

- Comprehensive documentation, tutorials, and FAQs available within the application for user support.

- Contact Support:

- Direct access to customer support via chat or email for resolving issues and inquiries.

8. Compliance and Security

- Data Privacy:

- Implementation of security measures to protect user and cattle health data, including encryption and secure data storage.

- Regulatory Compliance:

- Adherence to relevant data protection regulations (e.g., GDPR) to ensure lawful data handling.

Data Preprocessing

1. Data Collection

- Sources:

- Historical health records of cattle.

- Environmental data (temperature, humidity, etc.).

- Geographic data (location of farms, proximity to outbreaks).

- Veterinary reports and treatment outcomes.

- Livestock management data (feeding practices, vaccination records).

2. Data Cleaning

- Handling Missing Values:

- Identify and fill missing data using techniques like mean/mode imputation, interpolation, or removing incomplete records.

- Removing Duplicates:
- Ensure no duplicate records exist to maintain data integrity.
- Outlier Detection:
- Use statistical methods (e.g., Z-score, IQR) to identify and handle outliers that may skew the model.

3. Data Transformation
- Normalization/Standardization:
- Scale numerical features to a uniform range (e.g., Min-Max scaling or Z-score normalization) to improve model performance.
- Categorical Encoding:
- Convert categorical variables (e.g., breed, location) into numerical format using techniques like one-hot encoding or label encoding.
- Feature Engineering:
- Create new features that might improve model performance, such as:
- Weather indices (combining temperature and humidity).
- Time variables (e.g., season, month of the year).
- Historical infection rates in the area.

4. Data Splitting
- Training, Validation, and Test Sets:
- Split the dataset into training (e.g., 70%), validation (e.g., 15%), and test sets (e.g., 15%) to ensure unbiased model evaluation.
- Stratified Sampling:
- If the dataset is imbalanced (e.g., fewer cases of LSD), use stratified sampling to maintain the proportion of classes in each subset.

5. Exploratory Data Analysis (EDA)
- Visualization:
- Use plots (histograms, box plots, scatter plots) to understand distributions and relationships between features.

- Correlation Analysis:
- Calculate correlation coefficients to identify relationships between features and the target variable (LSD occurrence).

6. Feature Selection
- Importance Ranking:
- Use methods like Recursive Feature Elimination (RFE) or feature importance from tree-based models (e.g., Random Forest) to identify significant predictors.
- Dimensionality Reduction:
- Consider techniques like PCA (Principal Component Analysis) if the feature space is too large, helping to reduce complexity.

7. Data Augmentation
- If the dataset is limited, consider techniques such as:
- Synthetic data generation using SMOTE (Synthetic Minority Over-sampling Technique) to create more examples of minority classes.
- Data augmentation techniques (e.g., rotating images of cattle, if image data is included).

8. Data Pipeline Development
- Automation:
- Build an automated data pipeline using tools like Apache Airflow or Python scripts to streamline the process from data collection to model input.
- Monitoring:
- Implement monitoring to ensure data quality and integrity over time.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

-Response Time: Predictions should be generated within seconds.
- Scalability: System must handle increasing data volume efficiently.

5.2 Safety Requirements

-Data Privacy: Ensure user data is encrypted and securely stored.
- Access Control: User authentication and role-based access.

5.3 Security Requirements

1. Data Protection

-Data Encryption:
- All sensitive data, both in transit and at rest, should be encrypted using industry-standard protocols (e.g., AES-256 for storage, TLS for data transmission).

- Secure Data Storage:
- Use secure cloud services or on-premises solutions that comply with relevant data protection regulations to store user and cattle health data.

2. User Authentication and Authorization

- Multi-Factor Authentication (MFA):
- Implement MFA for user accounts to enhance security during login and prevent unauthorized access.

- Role-Based Access Control (RBAC):

- Establish RBAC to ensure that users can only access the data and functionalities relevant to their roles (e.g., farmers, veterinarians).

- Password Security:
- Enforce strong password policies, requiring users to create complex passwords and regularly update them.

3. Access Logging and Monitoring

- Audit Trails:
- Maintain detailed logs of user activities within the system, including data access, modifications, and predictions generated.

- Monitoring and Alerts:
- Implement monitoring tools to detect unauthorized access attempts or unusual activities, triggering alerts for immediate investigation.

4. Compliance with Regulations

- Data Privacy Compliance:
- Ensure adherence to relevant regulations, such as GDPR or HIPAA, which govern the handling of personal and sensitive data.

- User Consent Management:
- Obtain explicit user consent for data collection, processing, and sharing, and provide options for users to withdraw consent.

5. Application Security

- Vulnerability Assessment:
- Regularly conduct security assessments and penetration testing to identify and mitigate vulnerabilities in the application.

- Input Validation:
- Implement robust input validation and sanitization to protect against common threats such as SQL injection and cross-site scripting (XSS).

- Secure Coding Practices:
- Follow secure coding standards during development to minimize security flaws and ensure code quality.

5.4 Software Quality Attributes

1. Functionality
- Accuracy: The predictive model should provide reliable and precise predictions of LSD outbreaks based on the data inputs.
- Usability: The system must be user-friendly, allowing users of varying technical expertise to navigate easily, input data, and interpret predictions.

2. Reliability
- Availability: The system should be accessible 24/7, with minimal downtime for maintenance or updates.
- Fault Tolerance: The application should gracefully handle errors and exceptions, ensuring that users receive meaningful feedback and can continue using the system without significant disruption.

3. Performance
- Response Time: Predictions should be generated quickly, ideally within a few seconds, to facilitate timely decision-making.

- Scalability: The system should efficiently handle increasing amounts of data and users without a degradation in performance.

4. Security

- Data Protection: Robust security measures must be in place to protect sensitive user and animal health data from unauthorized access and breaches.
- User Authentication: Strong authentication mechanisms (e.g., multi-factor authentication) should be implemented to ensure that only authorized users can access the system.

5. Maintainability

- Modularity: The system should be designed in a modular fashion, allowing for easier updates and maintenance of individual components (e.g., the prediction engine, user interface).
- Documentation: Comprehensive documentation for both users and developers should be provided to facilitate easier maintenance and enhancements.

8. Usability

- User Experience (UX): The interface should be intuitive and visually appealing, enhancing user engagement and satisfaction.
- Help and Support: Easy access to help resources, tutorials, and customer support should be available to assist users in navigating the system.

5.5 Business Rules

1. Data Entry Rules

- Mandatory Fields: Users must provide specific mandatory data fields (e.g., cattle ID, health status, environmental conditions) to generate predictions.
- Data Format Validation: All data inputs must conform to specified formats (e.g., date formats, numerical ranges) to ensure accuracy in processing.

2. Prediction Generation Rules

- Model Activation**:** Predictions can only be generated if sufficient data has been entered (e.g., minimum number of observations).
- Prediction Frequency: Users can request predictions no more than a specified number of times per day to manage server load and maintain performance.

3. Alert and Notification Rules

- - Alert Thresholds: Automated alerts for potential LSD outbreaks are triggered when the predicted probability exceeds a predefined threshold (e.g., 70% likelihood).
- - Notification Preferences: Users can customize their alert preferences regarding the method (e.g., email, SMS) and frequency of notifications.

4. User Management Rules

- - Role Assignment: Users must be assigned specific roles (e.g., farmer, veterinarian) to determine their access levels and permissions within the system.
- - Account Verification: New user accounts must undergo an email verification process before access is granted.

5. Data Privacy and Compliance Rules

- - Consent Management: User consent must be obtained for collecting and processing personal and animal health data, with options for users to withdraw consent at any time.
- - Data Retention Policies: Historical data must be retained for a specified period (e.g., 5 years) for compliance and model training purposes, after which it can be archived or deleted.

6. Reporting and Analysis Rules

- - Report Generation: Users can generate reports on-demand, summarizing predictions, alerts, and user data over a specified time period.
- - Data Visualization Guidelines: Visualizations must adhere to guidelines for clarity and accessibility, ensuring they effectively communicate insights.

6. Other Requirements

1. Project Management Requirements

- Development Timeline:

- A clear timeline for project phases (requirements gathering, development, testing, deployment) must be established and monitored to ensure timely delivery.

- Resource Allocation:

- Sufficient resources (personnel, technology, budget) must be allocated to each phase of the project to meet objectives.

2. Training and Documentation Requirements

- User Training:

- Provide comprehensive training sessions for end-users (farmers, veterinarians) to ensure they understand how to effectively use the system.

- Technical Documentation:

- Detailed documentation for developers and system administrators must be created to facilitate ongoing maintenance and future enhancements.

3. Regulatory and Compliance Requirements

- Compliance with Veterinary Regulations:

- The system must adhere to local and international veterinary regulations regarding animal health data management and reporting.

- Data Protection Compliance:

- Ensure compliance with data protection laws (e.g., GDPR, HIPAA) for handling personal and health-related data.

4. System Performance Requirements

- Load Handling:

- The system must be capable of handling concurrent user requests efficiently, with performance benchmarks established (e.g., response time under load).

- Data Backup and Recovery:

- Implement regular data backup procedures and a recovery plan to minimize data loss in case of system failures.

5. Environmental Requirements

- Operating Conditions:

- The system should be designed to operate in various environmental conditions, especially for users in rural or remote areas with limited connectivity.

- User Accessibility:

- Ensure that the tool is accessible to users with varying levels of technical proficiency, including options for multiple languages if necessary.

Appendix A: Glossary

Definitions, Acronyms, and Abbreviations

- LSD: Lumpy Skin Disease
- ML: Machine Learning
- API: Application Programming Interface
- CSV: Comma-Separated Values
- UI: User Interface

Appendix B: Analysis Models

1. Descriptive Analysis Models

-Statistical Summaries:

- Use basic statistical techniques (mean, median, mode, standard deviation) to summarize historical data on LSD occurrences and related factors.

- Correlation Analysis:

- Analyze the relationships between different variables (e.g., environmental conditions, cattle demographics) to identify potential predictors of LSD.

2. Predictive Analysis Models

- Logistic Regression:

- A statistical method used for binary classification problems. It predicts the probability of an event (e.g., outbreak of LSD) based on one or more predictor variables.

- Decision Trees:

- A tree-like model that splits data into subsets based on the value of input features. It helps in making decisions based on various conditions and is interpretable.

- Random Forest:

- An ensemble learning method that constructs multiple decision trees and combines their outputs to improve prediction accuracy and control overfitting.

- Support Vector Machines (SVM):

- A supervised learning model that finds the hyperplane that best separates different classes in the feature space, effective for both linear and non-linear data.

- Neural Networks:

- Deep learning models that can learn complex patterns in data through multiple layers of interconnected nodes. Suitable for handling large datasets with non-linear relationships.

- Gradient Boosting Machines (GBM):

- An ensemble technique that builds models sequentially, where each new model corrects errors made by the previous ones, providing high predictive accuracy**.**

3. Evaluation Models

- Cross-Validation:

- A technique to assess the performance of predictive models by partitioning the data into subsets, training on some, and validating on others to ensure generalization**.**

- Confusion Matrix:

- A table used to evaluate the performance of a classification model, providing insights into true positives, false positives, true negatives, and false negatives.

- Performance Metrics:

- Key metrics to evaluate model performance, including:

- Accuracy: The proportion of correctly predicted instances.

- Precision: The ratio of true positive predictions to the total predicted positives.

- Recall (Sensitivity): The ratio of true positive predictions to the actual positives.

- F1 Score: The harmonic mean of precision and recall, balancing both metrics.

- ROC-AUC Score: The area under the receiver operating characteristic curve, assessing the model's ability to distinguish between classes.

4. Visualization Models

- Heatmaps:

- Visual representation of data correlations and interactions between different variables related to LSD outbreaks.

- Box Plots:

- Used to visualize the distribution of data points, identifying outliers and understanding the spread of variables influencing LSD.

- Time Series Analysis:

- Analyze historical data trends over time to identify patterns related to LSD outbreaks, seasonality, and other temporal factors.

Appendix C: To Be Determined List

1. Data Sources

- Identification of Data Sources:

- Finalize the sources of historical and real-time data for training and validating the machine learning models (e.g., veterinary reports, environmental data, farm management systems).

- Data Access Agreements:

- Determine the necessary agreements or permissions needed to access external data sources.

2. Model Selection

- Model Evaluation Criteria:

- Establish clear criteria for selecting machine learning models based on performance metrics, interpretability, and resource requirements.

- Final Model Choices:

- Confirm which specific algorithms will be implemented in the initial development phase versus future iterations.

3. User Requirements

- User Feedback Mechanisms:

- Determine the most effective ways to gather user feedback on system usability and predictive accuracy during testing phases.

- User Roles and Permissions:

- Finalize the specific roles and permissions for different types of users (e.g., farmers, veterinarians, researchers).

4. Regulatory Compliance

- Data Protection Regulations:

- Confirm applicable data protection and privacy regulations that the system must adhere to, especially concerning animal and user data.

- Veterinary Regulations:

- Identify any veterinary regulations or standards that must be considered in the handling and reporting of cattle health data.

5. System Performance

- Performance Benchmarks:

- Determine specific performance benchmarks for model accuracy, response time, and system load capacity.

- Scalability Requirements:

- Establish scalability requirements based on expected user growth and data volume.