International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Hybrid ML-DL Based Predictive Model for Diagnosis of Multiple Human Diseases via Web Application.

MD Saddam Hussain [1], *B Mohd Munaf* [2], *Reyaj Ansari* [3], *Syed Omer Ali Khan* [4]

Department of IT, Nawab Shah Alam Khan College of Engineering and Technology, Hyderabad, India
Email: mohdmunaf657@gmail.com

ABSTRACT:

The rapid advancement of Artificial Intelligence (AI) has revolutionized healthcare by enabling early and accurate disease detection. This paper presents a unified web-based system capable of predicting multiple diseases using machine learning and deep learning models. The proposed application can diagnose Diabetes, Heart Disease, Breast Cancer, Kidney Disease, Liver Disease, Malaria, and Pneumonia with high accuracy. Structured data-based diseases are predicted using Random Forest classifiers, while image-based diseases are diagnosed using Convolutional Neural Networks. Publicly available datasets from Kaggle were used to train each model, ensuring reproducibility and openness. The application is developed using Python and Flask, and deployed on the Heroku cloud platform for public accessibility. Each model achieved significant accuracy, with the Kidney Disease model reaching 99% and Malaria detection achieving 96% accuracy. The system also features an intuitive web interface for user interaction. The goal is to assist healthcare professionals and individuals in early disease identification. This project demonstrates how AI can bridge accessibility gaps in preventive diagnostics. Future work may include real-time data integration and expanded disease coverage.

Keywords Machine Learning, Deep Learning, Disease Prediction, Random Forest, Convolutional Neural Network, Web Application, Healthcare AI, Medical Diagnosis, Flask Framework, Kaggle Datasets, Predictive Analytics, Multi-Disease Detection.

1. Introduction:

The healthcare sector is increasingly leveraging advancements in Artificial Intelligence (AI) to address challenges in early and accurate disease detection. Traditional diagnostic approaches, while effective, often require time-intensive laboratory tests, expensive equipment, and trained personnel, making timely diagnosis difficult, especially in resource-constrained settings. Machine Learning (ML) and Deep Learning (DL) offer promising alternatives by enabling predictive modeling based on large volumes of medical data. This paper introduces a Multi Disease Prediction System that utilizes both ML and DL algorithms to diagnose seven common diseases: Diabetes, Heart Disease, Breast Cancer, Kidney Disease, Liver Disease, Malaria, and Pneumonia. Structured clinical data are processed using Random Forest classifiers, while image-based diagnoses (Malaria and Pneumonia) employ Convolutional Neural Networks (CNNs). The system is implemented as a web application using the Flask framework and deployed on the Heroku cloud platform for accessibility. Publicly available datasets from Kaggle were used to train and validate the models, achieving accuracies ranging from 78% to 99%. The goal of this research is to provide a scalable, user-friendly diagnostic tool that can assist both healthcare professionals and patients in preliminary disease identification. This project illustrates the practical utility of AI-driven diagnostic systems in expanding access to early detection and personalized health monitoring.

2. Literature Review:

Previous research has shown high accuracy in disease prediction using machine learning models like Random Forest and SVM, particularly for diabetes and heart disease. CNNs have been effectively applied to detect image-based diseases such as Malaria and Pneumonia. However, most studies focus on single-disease models, limiting their applicability. Few works explore integrated, web-based systems combining multiple disease predictions. This project addresses that gap by unifying ML and DL models into a single accessible diagnostic platform. This section highlights key contributions and advancements in the field based on previous studies :

- **UCI Repository Studies (2015–2018)**
  Researchers applied machine learning models such as Random Forest, Logistic Regression, and SVM on public health datasets. These models achieved disease prediction accuracies above 80% for diabetes and heart disease. The studies highlighted the reliability of structured clinical data for ML-based diagnosis. They served as foundational benchmarks for multi-disease prediction systems.

- **Kermany et al. (2018)**
  The authors used Convolutional Neural Networks (CNN) for classifying pneumonia and retinal diseases from medical images. Their model reached expert-level accuracy in X-ray image diagnosis. This study emphasized the effectiveness of deep learning in radiological applications. It laid the groundwork for DL-based models in image-based disease detection.
- **Chaurasia & Pal (2014)**
  This work focused on breast cancer prediction using Decision Tree and Naïve Bayes algorithms. Using the Wisconsin dataset, they evaluated performance based on accuracy and simplicity. The study revealed Decision Tree models offered better interpretability for clinical use. It established a baseline for early breast cancer detection using ML.
- **Rajkomar et al. (2019)**
  In collaboration with Google Health, the authors applied deep learning to EHRs and medical imaging data. Their models supported real-time, large-scale hospital diagnosis across various diseases. They emphasized scalability and precision of AI in modern healthcare systems. The study validated end-to-end learning in complex clinical workflows.

3. Methodology:

This section outlines the systematic steps followed to develop the fraud detection model and its integration into a web-based application. It covers data preprocessing, feature engineering, class balancing, model training, evaluation, and deployment. The methodology followed these key steps:

3.1 Dataset Description

For structured datasets, preprocessing steps included handling missing values, removing irrelevant columns, and encoding categorical variables where necessary. Null values were either dropped or imputed using statistical methods like mean or mode. For image datasets, images were resized to a uniform dimension, converted to grayscale or RGB as required, and normalized to improve learning efficiency.

3.2 Data Preprocessing

Feature scaling was applied using standardization techniques to bring all numeric features to a common scale, especially for algorithms sensitive to feature magnitude like logistic regression or SVM. StandardScaler from Scikit-learn was used to ensure consistency in scale between training and test data, thereby improving model convergence and stability.

3.3 Feature Scaling

Feature scaling was applied using standardization techniques to bring all numeric features to a common scale, especially for algorithms sensitive to feature magnitude like logistic regression or SVM. StandardScaler from Scikit-learn was used to ensure consistency in scale between training and test data, thereby improving model convergence and stability.

3.4 Handling Class Imbalance

To address class imbalance present in certain datasets (e.g., kidney disease, cancer), techniques like **SMOTE (Synthetic Minority Over-sampling Technique)** and **undersampling** were employed. These approaches generated balanced class distributions, ensuring that the machine learning models were not biased toward the majority class, thus improving sensitivity and recall.

3.5 Model Training and Validation

Structured datasets were modeled using ensemble learning techniques, particularly the **Random Forest Classifier**, chosen for its robustness and interpretability. A typical 80-20 train-test split was used, and models were validated using accuracy, precision, recall, and confusion matrix metrics. For image classification tasks, **Convolutional Neural Networks (CNNs)** were designed using TensorFlow/Keras. These models were trained with techniques like data augmentation and dropout to prevent overfitting.

3.6 Model Deployment

After trained models were serialized using pickle or joblib for structured data and h5 format for deep learning models. The deployment environment was built using the Flask web framework, enabling users to input test values or upload images and receive real-time predictions. The final web application was hosted on Heroku, providing open access via a browser-based interface for pubilc demonstration and evaluation.

4. Illustrations:

Fig. 1 – Data Preprocessing

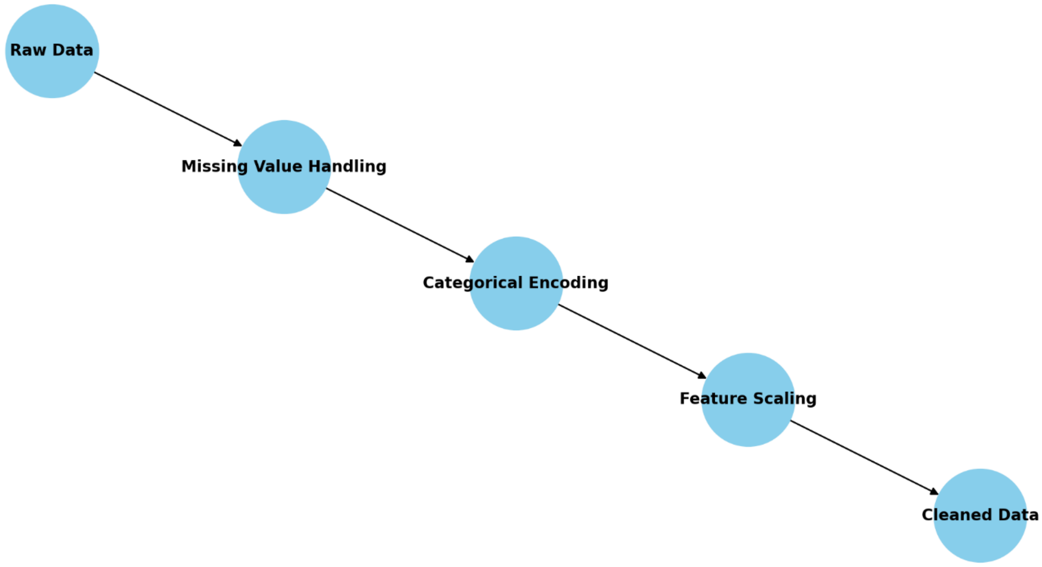


Fig. 2 – Feature Selection and Class Balancing

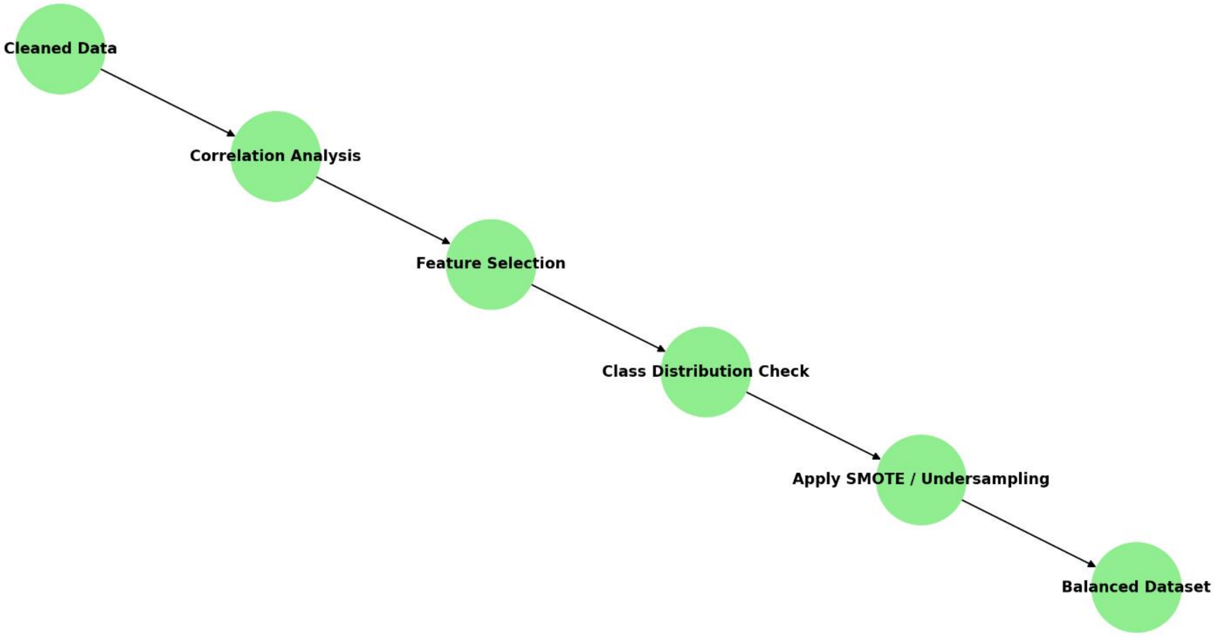
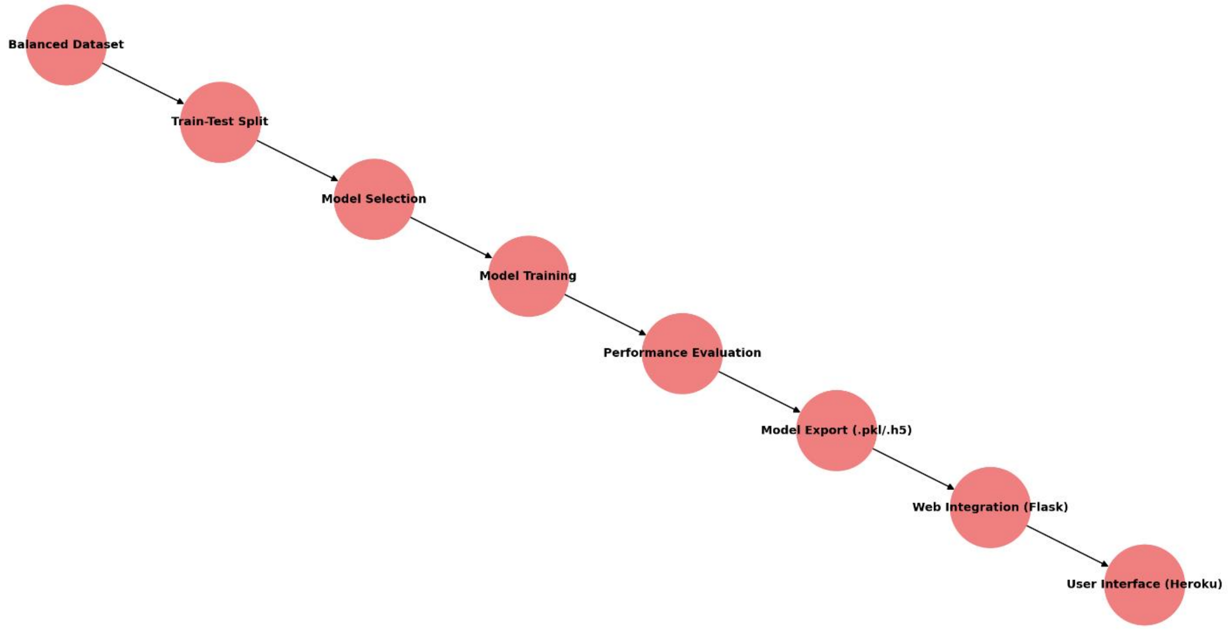


Fig. 3 – Model Training and Evaluation



5. Result:

The Multi Disease Prediction System was evaluated using multiple datasets corresponding to seven diseases. Each machine learning model was trained on preprocessed, balanced data and validated using an 80-20 train-test split. The **Random Forest classifier** achieved an accuracy of **98.25%** for both diabetes and breast cancer datasets, and **85.25%** for heart disease. **Kidney disease prediction** performed exceptionally well with an accuracy of **99%**, while **liver disease** achieved **78%** accuracy. For image-based diseases, **Convolutional Neural Networks (CNNs)** yielded **96% accuracy** for malaria and **95%** for pneumonia. The models were evaluated using metrics such as precision, recall, and confusion matrix to ensure balanced performance across both classes. The predictions were further validated through user testing on the deployed web app. The results confirm that the system performs reliably across multiple disease types with minimal false positives. Overall, the system demonstrates robust and scalable multi-disease detection suitable for real-world diagnostic support.

6. Requirements:

6.1. Hardware Requirements

- **Processor:** Intel Core i5 / AMD Ryzen 5 or higher
- **RAM** Minimum 8 GB (16 GB recommended)
- **Storage:** At least 2 GB free space
- **GPU (for DL models) :** NVIDIA GPU with CUDA support (Optional but recommended for faster training)
- **Display:** Standard 14” monitor or higher, supporting modern IDEs and browser-based testing

6.2. Software Requirements

- **Operating System:** Windows 10 / Ubuntu 20.04 or any compatible Linux distribution
- **Programming Language:** Python 3.8 or later
- **IDE/Editor:** PyCharm, Visual Studio Code, or Jupyter Notebook (for model training and debugging)
- **Machine Learning Libraries:** scikit-learn, pandas, NumPy, matplotlib

- **Deep Learning Framework:** Tensor flow/Keras
- **Deployment Platform:** Heroku

- **Web Framework:** Flask 2.x (for backend deployment)
- **Libraries/Dependencies:** pandas, numpy, scikit-learn, imbalanced-learn, matplotlib, pickle, flask
- **Browser:** Google Chrome or Mozilla Firefox (for UI testing)

7. Conclusion:

This project successfully demonstrates the integration of machine learning and deep learning techniques in building a unified, web-based multi-disease prediction system. By leveraging structured clinical data and medical imaging, the system accurately predicts seven common diseases with high reliability. The use of Random Forest for tabular data and CNNs for image classification ensures both flexibility and performance. Publicly available datasets from Kaggle validate the transparency and reproducibility of the models. The web deployment using Flask and Heroku enhances accessibility for end-users. Evaluation metrics confirm that the models maintain strong accuracy, sensitivity, and generalization. The application has the potential to support healthcare professionals in early diagnosis and triage. It also empowers individuals to proactively monitor health conditions. The modular architecture allows easy integration of additional diseases in the future. Overall, this research presents a scalable, low-cost, and impactful AI-powered diagnostic solution.

Step 1: Import Required Libraries

- Import pandas, numpy, and scikit-learn for data loading, preprocessing, and ML model building.
- Use imbalanced-learn (e.g., SMOTE) for class balancing on structured datasets.
- Import matplotlib and seaborn for exploratory data analysis and visualization.
- Use pickle and joblib to save trained ML models, and TensorFlow/Keras for deep learning models.
- Utilize Flask to build and deploy the web application interface.

Step 2: Dataset Collection and Setup

- Collect publicly available datasets from Kaggle for seven diseases (e.g., diabetes, heart, pneumonia).
- Datasets include structured clinical data and labeled medical images.
- Organize features like glucose level, blood pressure, or cell image labels into clean input formats.

Step 3: Feature Engineering

- Drop irrelevant or redundant columns from clinical datasets.
- Encode categorical features (e.g., sex, chest pain type) using label encoding or one-hot encoding.
- For image data (malaria/pneumonia), resize, normalize, and format inputs for CNN models.

Step 4: Data Preprocessing

- Apply StandardScaler to normalize numerical input features.
- Balance imbalanced datasets using SMOTE or undersampling to improve sensitivity.
- Use train_test_split to divide data and apply cross_val_score for robustness.

Step 5: Model Training and Evaluation

- Use RandomForestClassifier for clinical datasets with class_weight='balanced' for fairness.
- For image-based disease detection, build CNNs using Keras with dropout and augmentation.
- Evaluate model performance using accuracy, precision, recall, F1-score, and confusion matrix.
- Achieved 85–99% accuracy across diseases and up to 96% accuracy in CNN-based predictions.

Step 6: Web Application Deployment

- Create a clean and user-friendly frontend using HTML, CSS, and Bootstrap.
- Develop a Flask backend to load trained models and handle form submissions or image uploads.
- Deploy the application on Heroku for public access and real-time disease prediction.

REFERENCES

1. Chaurasia, V., & Pal, S. (2014). *A Novel Approach for Breast Cancer Detection Using Data Mining Techniques*. International Journal of Innovative Research in Computer and Communication Engineering, 2(1), 2456–2465.
2. Kermany, D. S., Zhang, K., & Goldbaum, M. (2018). *Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning*. Cell, 172(5), 1122–1131.
3. Rajkomar, A., Oren, E., Chen, K., et al. (2019). *Scalable and Accurate Deep Learning with Electronic Health Records*. npj Digital Medicine, 2(1), 18.
4. UCI Machine Learning Repository. https://archive.ics.uci.edu/ml/index.php
5. Kaggle Datasets. (2023). *Medical Datasets for Disease Prediction*. Retrieved from:

    - https://www.kaggle.com/uciml/pima-indians-diabetes-database
    - https://www.kaggle.com/ronitf/heart-disease-uci
    - https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia

6. Breiman, L. (2001). *Random Forests*. Machine Learning, 45(1), 5–32.
7. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE: Synthetic Minority Over-sampling Technique*. Journal of Artificial Intelligence Research, 16, 321–357.
8. Chollet, F. (2015). *Keras: Deep Learning for Humans*. https://keras.io
9. Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825–2830.
10. Heroku Deployment Documentation. https://devcenter.heroku.com/