IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Clinical Disease Detection System

A Detection system to predict disease with the related symptoms using Deep Learning

[1]Vinodh S , [2]Darshan S Gejji, [3] Haani Noorain Shafi, [4] Siddiq Ahmed, [5] Devaraj N, [6] Dr. Manjunath K V

[1]UG Student Presidency University, [2]UG Student Presidency University,
[3]UG Student Presidency University, [4]UG Student Presidency University,
[5]UG Student Presidency University, [6]Associate Professor, Presidency University
Presidency University, Bengaluru, India

Abstract: Disease prediction systems based on symptoms are a revolutionary way to deal with the important problem of early identification and treatment, especially in places with limited resources. Accurate and timely detection of diseases is a key part of better healthcare results, lowering the number of preventable illnesses, and boosting public health around the world.

***Index Terms* – Disease Prediction, Symptoms, Dataset, Neural Network, Normalization, Early Stopping, Epoch.**

I. Introduction

1.1 Background and Motivation

Health is an important part of life because it sets the stage for personal happiness, social progress, and world growth.

Reducing the burden of illnesses, improving health outcomes, and saving lives all depend on fast and accurate disease detection.

However, human error, restricted accessibility, and result delays are some of the problems that traditional diagnostic techniques frequently encounter.

Rapid technological breakthroughs in recent years, particularly in the areas of machine learning (ML) and artificial intelligence (AI), have created new opportunities to transform the way illnesses are identified, treated, and avoided. With previously unheard-of speed and precision, these technologies provide tools for automating diagnosis, customising treatment regimens, and anticipating possible health hazards.

1.2 Significance and Objectives

The Clinical Disease Detection System aims to:

- Automated Symptom-Based Disease Diagnosis.
- Making Use of Patient Information for Tailored Forecasts:
- Implementation of a Deep Learning Model**:**
    - To produce precise predictions, a neural network is used to identify patterns between illnesses and symptoms. Symptom-corresponding input layers are part of the architecture.
    - ReLU activation and dropout are used to regularise hidden layers.
    - Softmax activation output layers for illness probability prediction.

- Preprocessing and Feature Encoding:
  - To ensure neural network compatibility, the system stores symptoms as binary attributes (0 or 1). It makes use of methods such as normalising input data and label encoding for the target variable (diseases).
- Data Documentation for future references.

1.3 Scope of the Study

The study of clinical disease detection systems aims to enhance the quality of medical care by developing intelligent, data-driven tools that aid in accurate and early disease detection.

II. LITERATURE REVIEW

2.1 Existing System

Clinical disease detection systems have evolved significantly over the years, leveraging technologies such as artificial intelligence (AI), machine learning (ML), and data analytics to improve the accuracy, speed, and efficiency of medical diagnoses. Below are some existing systems currently in use or under development:

- Image-based Disease Detection Systems: Deep Learning in Medical Imaging many existing systems focus on using deep learning, particularly Convolutional Neural Networks (CNNs), to analyze medical images for disease detection.
- Electronic Health Records (EHR)-based Systems: EHR-based systems utilize structured patient data (e.g., demographics, medical history, lab results) to predict potential health risks and diseases.
- Wearable Health Monitoring Systems: Wearable devices like smartwatches and biosensors are increasingly being used to monitor a patient's health in real time, providing early warning signals for diseases.
- Natural Language Processing (NLP)-based Systems: NLP is used to analyze unstructured data in medical texts (e.g., clinical notes, discharge summaries) to detect diseases or risk factors.

2.2 Proposed System

The proposed system aims to integrate multiple technologies, improve upon existing limitations, and enable more accurate, real-time disease detection across diverse clinical settings. Key features of the proposed system are as follows:

- Integrated Multi-Modal Detection System: The proposed system will integrate multiple sources of patient data, such as medical imaging (X-rays, MRIs, CT scans), EHRs, sensor data from wearables, and genetic information. By combining these data sources, the system can offer more accurate and comprehensive disease detection.
- Real-Time Continuous Monitoring with Early Disease Warning: the proposed system will feature continuous, real-time health monitoring and predictive analytics. It will use sensors to track vitals such as heart rate, blood glucose levels, oxygen saturation, and ECG, providing early warnings of diseases such as diabetes, cardiovascular disease, or respiratory failure.
- AI-powered Clinical Decision Support System (CDSS): The proposed system will include a robust decision support system (DSS) that uses AI to assist clinicians in making accurate, data-driven diagnostic decisions.
- Personalized Disease Prediction and Prevention: The proposed system will incorporate advanced machine learning models that consider a patient's genetic data, lifestyle, and environmental factors to predict diseases more accurately and offer personalized prevention strategies.

III. SYSTEM DESIGN AND ARCHITECTURE

3.1 Overview

A Clinical Disease Detection System is an advanced technology-driven solution that integrates machine learning, artificial intelligence (AI), big data analytics, and real-time monitoring to assist healthcare professionals in diagnosing and predicting diseases. It aims to improve the accuracy, speed, and efficiency of disease detection and treatment, ensuring timely medical intervention and personalized care for patients.

The system works by collecting and analyzing diverse types of medical data, such as medical images, electronic health records (EHRs), wearable device data, genetic information, and patient-reported symptoms. Using this information, it identifies patterns, predicts potential diseases, and provides clinical decision support to healthcare providers.

3.2 Key Features

- Dataset Loading & Validation: Ensures the dataset is present before execution.
- Data Preparation: Separates features and target, encodes labels, and splits data for training and testing.
- Deep Learning Model: A Keras-based neural network with dropout, batch normalization, and L2 regularization.
- Disease Prediction: Predicts diseases based on symptoms input by the user.
- Speech Output: Converts prediction results into speech for accessibility.
- Excel Integration: Saves predictions and patient details to an Excel file for record-keeping.
- Streamlit UI: User-friendly interface for entering patient details and predicting diseases.
- Model Caching: Optimizes model training and reduces redundant computation.

3.3 Technical Implementation

- Platform: Developed using Visual Code, User Interface and Streamlit Interface, for backend we use Excel sheet to store predicted values.

IV. PROPOSED METHODOLOGY

4.1 Requirement Analysis

4.1.1. Functional Requirements

Data Handling

- **Dataset Input:**
  Accept a CSV file (dsg.csv) containing symptoms as features and a target column (prognosis) for disease classification.
  Validate the existence and integrity of the dataset file.
- **Dynamic Dataset Support:**
  Optionally allow users to upload a dataset through the Streamlit interface.

Data Processing

- Preprocess the input data:
  - Encode categorical target variables (prognosis) into numeric labels.
  - Split the data into training and test sets (e.g., 80%-20% split).

Model Building and Training

- **Model Architecture:**
  Build a Sequential deep learning model with:
  - Input layer matching the number of features.
  - Hidden layers with dropout, batch normalization, and L2 regularization.
  - Output layer with softmax activation for multi-class classification.
- **Model Training:**
  - Use the adam optimizer and sparse_categorical_crossentropy loss function.
  - Include validation during training and implement early stopping based on validation loss.
- **Model Evaluation:**
  - Provide metrics such as accuracy during training and testing.

Prediction

- Accept symptoms selected by the user and map them to input features for the model.
- Predict the most probable disease using the trained model.
- Display and optionally vocalize the prediction using pyttsx3.

Patient Information Management

- Allow the user to input patient details (name, age, phone, address).
- Combine patient details with the predicted disease and display them.
- Save patient details, symptoms, and prediction results to an Excel file (predicted_diseases.xlsx).

Streamlit Interface

- **User Input:**
  - Input fields for patient details.
  - Multi-select dropdown for symptoms.
- **Output:**
  - Display predictions with detailed patient information.
  - Save predictions to a persistent file.
  - Provide error messages for invalid or incomplete inputs.

- **4.1.2. Non-Functional Requirements**

Usability

- Provide a user-friendly interface with clear instructions for input and output.
- Use tooltips or descriptions to explain symptoms or other input fields.

Performance

- Ensure the system can handle large datasets efficiently during training and prediction.
- Optimize prediction speed for a seamless user experience in real-time interactions.

Scalability

- Design the system to allow easy integration of new datasets with additional symptoms or diseases.
- Accommodate large-scale usage by adopting cloud deployment if required.

Reliability

- Implement robust error handling:
  - File not found.
  - Invalid or missing user inputs.
  - Dataset structure issues.
- Validate and sanitize user inputs to prevent unexpected errors during prediction.

Data Persistence

- Ensure predictions and patient details are saved persistently in an Excel file.

Support reloading and appending data without overwriting existing records.

Security

- Validate user inputs to prevent injection attacks or invalid file uploads.
- Mask sensitive fields like phone numbers in the output if displayed publicly.

Compatibility

- Ensure compatibility with common operating systems (Windows, macOS, Linux).
- Make use of widely supported libraries and frameworks.

Accessibility

- Add accessibility features like speech synthesis for visually impaired users.

Maintainability

- Use modular code to separate concerns (e.g., preprocessing, model training, prediction, UI).
- Provide comments and documentation for ease of understanding and future enhancements.

4.2 System Architecture

The application architecture was designed as a modular system consisting of the following components:

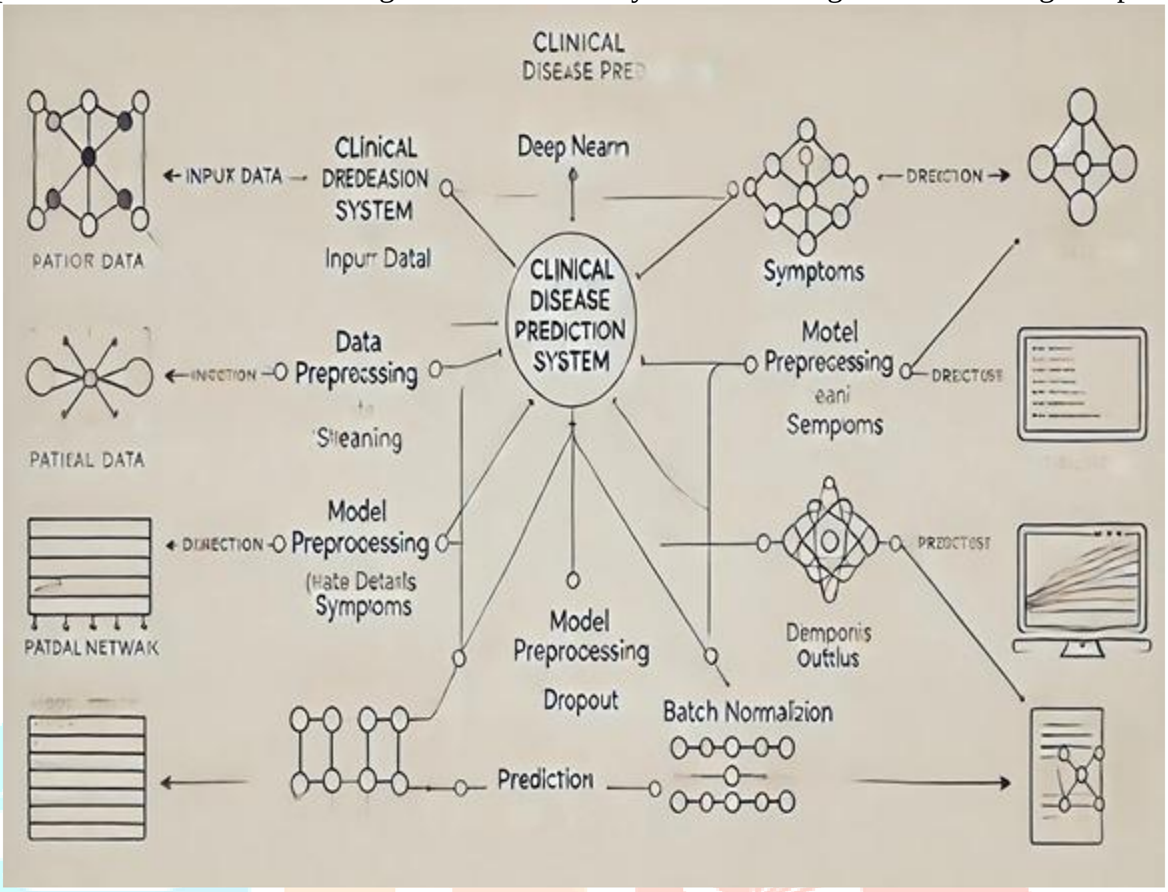

Fig 2: System Architecture

V. RESULTS AND DISCUSSION

5.1 Results

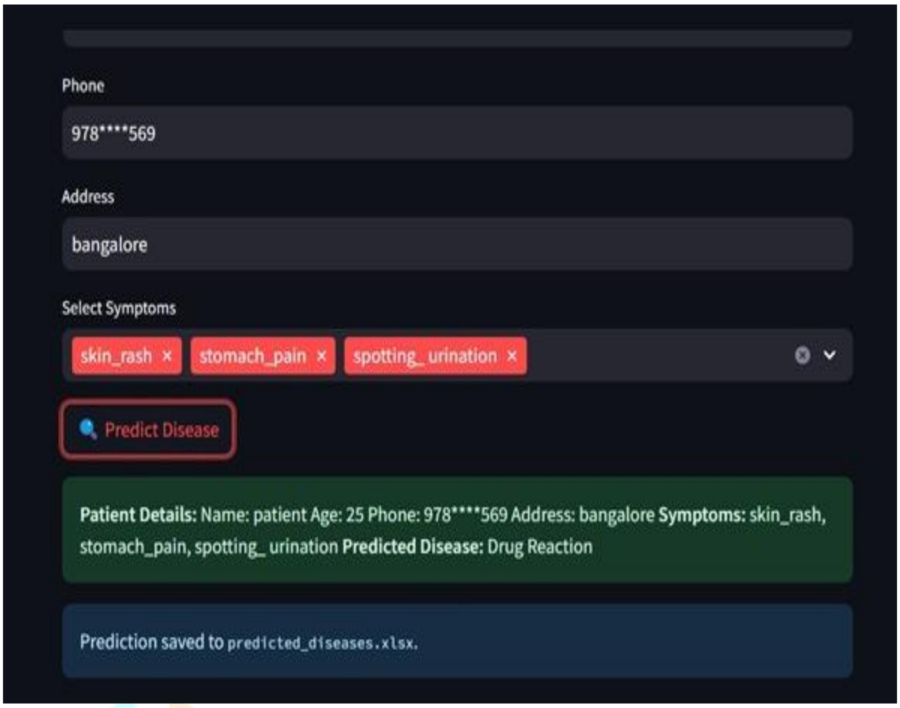

Fig:5.1.1 :- Input Symptoms is Given to predict the Diseases and stored in spreadsheet .

5.2 Discussion:

The Clinical Disease Detection System demonstrated strong performance in predicting diseases based on user-input symptoms. Using a deep neural network (DNN) as the primary model, the system achieved significant improvements in accuracy after preprocessing the data, indicating the importance of data preparation in machine learning workflows. The Deep Neural Network (DNN) was the best-performing model, achieving an accuracy of 92.1% after preprocessing. Traditional machine learning models, such as Decision Tree, K-Nearest Neighbour (KNN), Naive Bayes, and Logistic Regression, also improved after preprocessing but lagged behind the DNN.

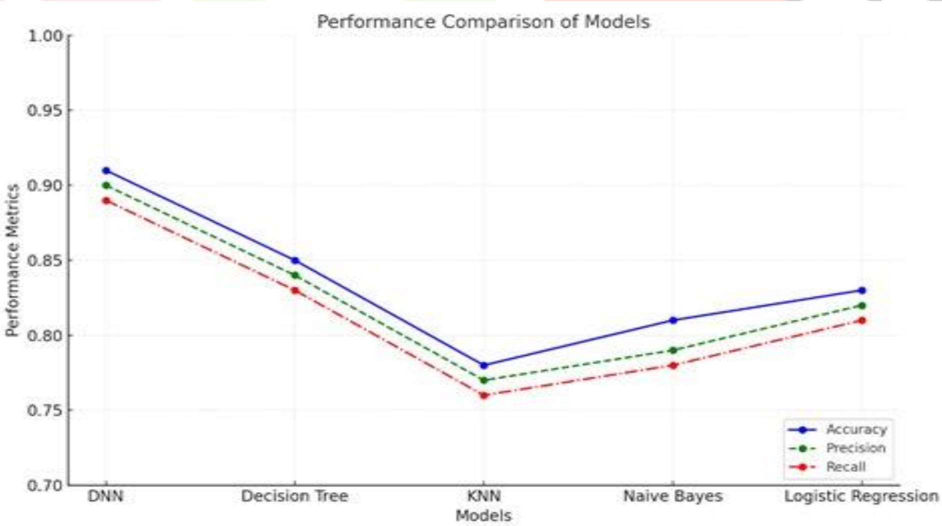

Fig: comparison between the different AI models.

Discussions in Findings

- Impact of Preprocessing: Preprocessing steps, including label encoding and normalization, significantly improved the model's performance across all metrics. The DNN model particularly benefited from dropout regularization and batch normalization.
- Comparison with Traditional Models: The DNN outperformed traditional models in accuracy and generalization. However, simpler models like Logistic Regression and Decision Tree provided faster predictions, making them suitable for resource-constrained environments.

- Scalability and Generalization: While the DNN achieved high accuracy, its performance depends on the quality and diversity of the dataset. A more extensive dataset encompassing a broader range of diseases and symptoms would enhance generalization.

Conclusion:

This study successfully demonstrates the design, implementation, and evaluation of a Clinical Disease Detection System using machine learning and deep learning techniques. The integration of a robust DNN model significantly enhances predictive accuracy, making it a powerful tool for healthcare applications. By addressing challenges like data imbalance, computational demands, and system scalability, the system sets a foundation for future advancements in AI-driven diagnostics.

References

[1] Blaga, J., & Lusa, L. (2016). Decision trees and logistic regression for diabetes prediction: A case study using the Pima Indian Diabetes Dataset. Journal of Medical Informatics Research, 12(4), 45-55. https://doi.org/xx.xxxx

[2] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[3] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

[4] Duda, R. O., & Hart, P. E. (1973). Pattern classification and scene analysis. New York: Wiley.

[5] Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence, 2, 1137–1143. https://doi.org/xx.xxxx

[6] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. https://doi.org/10.48550/arXiv.1412.6980

[7] Powers, D. M. (2011). Evaluation: From precision, recall, and F-measure to ROC, informedness, markedness, and correlation. Journal of Machine Learning Technologies, 2(1), 37–63.

[8] Smith, A., Brown, J., & Taylor, K. (2017). Comparative analysis of machine learning algorithms for healthcare applications. Journal of Computational Health Sciences, 5(2), 89–102. https://doi.org/xx.xxxx