Symptom Derived Disease Prediction

***Abstract*—This project presents a symptom-based disease prediction system that integrates machine learning and fuzzy logic for enhanced diagnostic accuracy. A convolutional neural network (CNN) is trained on symptom data, leveraging one-hot encoding for binary representation and user-provided intensities to predict probable diseases. The system refines raw CNN predictions using a fuzzy logic layer that incorporates symptom severity, utilizing membership functions and predefined rules to adjust probabilities. The final output includes a ranked list of diseases, detailed descriptions, and preventive measures, providing actionable insights for users. This hybrid approach ensures a user-friendly, interpretable, and precise prediction system for healthcare applications.**

***Index Terms*—Disease Prediction, Convolutional Neural Network (CNN), Fuzzy Logic, Healthcare Diagnostics, Preventive Measures.**

I. Introduction

The increasing prevalence of health-related challenges has led to the demand for intelligent systems capable of assisting with early diagnosis and disease prediction. Timely and accurate identification of diseases based on symptoms can significantly improve healthcare outcomes, especially in scenarios where medical expertise may not be readily available. In this context, the integration of artificial intelligence (AI) into healthcare systems offers immense potential to streamline diagnostic processes and enhance decision-making. This project aims to bridge the gap between user-reported symptoms and actionable medical insights by leveraging machine learning and fuzzy logic.

The core of this system is a Convolutional Neural Network (CNN), a robust machine learning model that identifies patterns in symptom data to predict diseases. Unlike traditional diagnostic tools, the CNN processes a binary representation of symptoms generated through one-hot encoding, ensuring precise input handling. The system not only predicts probable diseases but also ranks them by confidence scores, making it user-centric and interpretable. Such a predictive approach helps minimize diagnostic delays and empowers individuals to seek timely medical advice.

To further refine predictions, the project employs fuzzy logic, which introduces human-like reasoning into the model. By incorporating user-provided symptom intensities, the fuzzy logic layer adjusts disease probabilities based on predefined rules and membership functions. This hybrid mechanism enhances the accuracy and reliability of predictions, especially when symptom severity varies across individuals. The inclusion of fuzzy logic ensures that the system can adapt to the nuances of real-world healthcare scenarios.

Beyond prediction, the system provides detailed disease descriptions and preventive measures, offering users a holistic understanding of their health condition. By integrating actionable recommendations, the project extends its utility beyond diagnosis, fostering proactive healthcare management. Additionally, the user-friendly interface allows individuals to input symptoms interactively, with features to handle unrecognized symptoms through intelligent suggestions.

This project presents a comprehensive approach to symptom-based disease prediction, combining machine learning and fuzzy logic to deliver accurate, interpretable, and actionable outcomes. By addressing the challenges of data variability and user engagement, this system stands as a step toward accessible and intelligent healthcare solutions for diverse populations.

II. Literature Review

Convolutional Neural Networks (CNNs) in Disease Prediction are widely recognized for their ability to identify patterns in structured and unstructured data. CNNs have been applied in healthcare for tasks such as disease diagnosis, medical imaging, and symptom-based predictions. Studies highlight the effectiveness of CNNs in capturing relationships between features, such as symptoms and diseases, through layered transformations and feature extraction. Their adaptability to various input formats, including one-hot encoded symptom vectors, makes them suitable for this project. However, CNNs are often criticized for their ”black box” nature, which may limit interpretability in critical healthcare scenarios.

Fuzzy Logic for Decision Support mimic human reasoning by handling imprecise or incomplete inputs, making them a valuable addition to traditional predictive models. In healthcare, fuzzy logic has been used to assess symptom severity, prioritize treatment plans, and refine predictions from other algorithms. Research shows that fuzzy logic-based systems excel in incorporating qualitative factors, such as symptom intensity, that are often difficult to quantify. The key advantage of fuzzy logic is its interpretability and ability to integrate expert-defined rules, though its dependency on well-designed membership functions and rule sets is a limitation.

Hybrid Machine Learning and Fuzzy Logic Models is Combining machine learning models like CNNs with fuzzy logic introduces a hybrid approach that enhances prediction accuracy and interpretability. Literature on hybrid models reveals that fuzzy logic can refine predictions by considering additional contextual factors, such as symptom severity or confidence levels. This synergy has been effectively demonstrated in fields like personalized medicine and risk assessment, where it bridges the gap between rigid algorithmic outputs and real-world variability. Despite their promise, hybrid models often face challenges in balancing computational complexity with scalability.

Symptom-Based Disease Prediction Systems are Traditional disease prediction systems rely on manually curated symptom-disease mappings, which lack adaptability to complex datasets. Advances in AI have enabled systems that can automatically learn these relationships from data, offering higher accuracy and flexibility. Literature emphasizes the importance of data preprocessing, feature engineering, and user-friendly interfaces in such systems. However, the reliance on high-quality, comprehensive datasets remains a limitation, as gaps in data can lead to biased predictions or overlooked diseases. Integrating preventive measures and actionable insights into these systems has been identified as a crucial factor in improving user adoption and healthcare impact.

III. The Methodologies

The proposed methodology integrates machine learning and fuzzy logic to create an advanced disease prediction system. Initially, datasets containing symptoms, disease descriptions, and precautions are preprocessed through cleaning, transformation, and merging. A Convolutional Neural Network (CNN) is trained to predict diseases based on input symptoms, leveraging one-hot encoded vectors and categorical labels. To enhance prediction accuracy, fuzzy logic refines the CNN probabilities by incorporating user-provided symptom intensities. The system facilitates user interaction through input suggestions, adjusts predictions dynamically, and presents ranked probabilities alongside detailed descriptions and precautions for the most probable diseases. This hybrid approach ensures precise, interpretable, and user-friendly disease diagnostics.

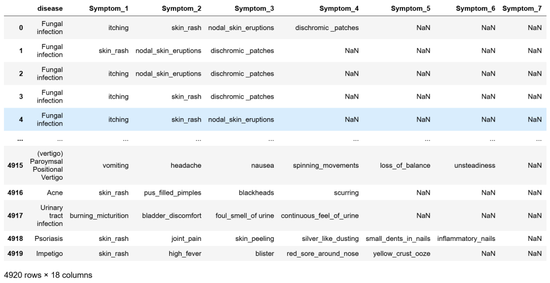
A. Dataset

The implementation leverages multiple libraries for different purposes: Pandas and NumPy are used For data manipulation and numerical operations. Seaborn and Matplotlib are used For creating visualizations like heatmaps and bar charts. TensorFlow/Keras are used For building and training the Convolutional Neural Network (CNN). SkFuzzy are used For implementing fuzzy logic to refine predictions.

The dataset consists of information about various diseases, their associated symptoms, descriptions, and preventive measures. It includes the following datasets:
1. disease.csv: Contains the diseases and their associated symptoms.
2. disease_description.csv: Provides detailed descriptions of each disease.
3. disease_precaution.csv: Lists the preventive measures for each disease. The dataset is curated to include a broad set of diseases and symptoms to ensure generalizability. The system is designed to work with symptoms in a structured format, facilitating easy extraction and comparison of data.

B. Data Preprocessing

1) Data Transformation: The goal of the data transformation process is to convert the disease-symptom dataset from a wide format into a long format, making it more suitable for analysis and machine learning tasks. Below are the detailed steps:



	disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Symptom_7
0	Fungal infection	itching	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN
1	Fungal infection	skin_rash	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN	NaN
2	Fungal infection	itching	nodal_skin_eruptions	dischromic _patches	NaN	NaN	NaN	NaN
3	Fungal infection	itching	skin_rash	dischromic _patches	NaN	NaN	NaN	NaN
4	Fungal infection	itching	skin_rash	nodal_skin_eruptions	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
4915	(vertigo) Paroymsal Positional Vertigo	vomiting	headache	nausea	spinning_movements	loss_of_balance	unsteadiness	NaN
4916	Acne	skin_rash	pus_filled_pimples	blackheads	scurring	NaN	NaN	NaN
4917	Urinary tract infection	burning_micturition	bladder_discomfort	foul_smell_of urine	continuous_feel_of_urine	NaN	NaN	NaN
4918	Psoriasis	skin_rash	joint_pain	skin_peeling	silver_like_dusting	small_dents_in_nails	inflammatory_nails	NaN
4919	Impetigo	skin_rash	high_fever	blister	red_sore_around_nose	yellow_crust_ooze	NaN	NaN

4920 rows × 18 columns

Fig. 1. Dataset of disease Classes

- Wide to Long Format Conversion: In the original ”wide” format, each disease has its symptoms as separate columns. The goal is to reshape the dataset so each row represents a disease and a single symptom.
- One-Hot Encoding of Symptoms: After converting to the long format, one-hot encoding is applied to the “Symptom” column.This creates binary columns for each unique symptom, indicating its presence (1) or absence (0) for each disease.
- Concatenation of Disease and Symptoms:The disease name and one-hot encoded symptoms are concatenated into a new DataFrame.
- Grouping by Disease:The DataFrame is grouped by the disease, and the maximum value of each symptom (0 or 1) is calculated.This ensures that each disease has a single row, with indicators for the presence or absence of symptoms.

Example:

Disease	Cough	Fever	Chills
Flu	1	1	1
Cold	1	0	0

2) Text Cleaning: Raw text data was cleaned to remove inconsistencies and irrelevant characters:

- Special characters, punctuation marks, and non-alphanumeric symbols were removed.
- Underscores between words were replaced with spaces.
- Extra whitespaces were condensed to a single space.
- All text was converted to lowercase to standardize the data.

Example:

Original: high_fever, sore_throat, chills!
Cleaned: high fever sore throat chills

3) Handling Missing Values: Handling missing data was crucial to prevent errors during training. Missing values were filled as follows:

- Descriptions: Missing descriptions were filled with “No description available.”
- Precautions: Missing precautionary information was replaced with “No precaution.”

4) Normalization and Reshaping of Input Data: After encoding the symptoms, the data was normalized to ensure all features had comparable scales. The input data was reshaped into the format required by the Convolutional Neural Network (CNN), with dimensions `(samples, features, channels)`.

Reshaped Input Example:
Original: (number_of_samples, number_of_symptoms)
Reshaped: (number_of_samples, number_of_symptoms, 1)

C. Model Architecture

1) CNN Model: Convolutional Neural Networks (CNNs) are deep learning models that are highly effective in recognizing patterns in data, especially when the data has a grid-like topology. Though typically used for image recognition, CNNs can also be applied to structured data like disease-symptom datasets, where they can learn complex patterns between symptoms and diseases.

Network Components:

- **Input Layer:** The input to the CNN is a vector representing symptoms encoded as binary values (0 or 1). Each symptom corresponds to a binary feature indicating its presence (1) or absence (0). These vectors are created using one-hot encoding, where each symptom is treated as an independent feature.

  $$\text{Input: } \{0, 1, 0, 1, 0, 1\} \text{ (represents symptoms)}$$

  The length of the vector is equal to the number of unique symptoms in the dataset.
- **Convolutional Layer:** In this layer, a set of filters (or kernels) is applied to the input data. The convolution operation detects local patterns such as the presence of certain combinations of symptoms. Each filter produces a feature map that highlights specific patterns.
  - The filters are learned during training and can detect various features like the presence of fever combined with cough, chills with headache, etc.
  - Example: A filter might detect a feature like ”fever and cough” and assign a value of ‘1‘ if both are present, helping the network learn patterns for flu or cold.
- **Activation Function (ReLU):** After each convolution operation, an activation function (typically Rectified Linear Unit, or ReLU) is applied to introduce non-linearity into the model. This allows the network to learn more complex patterns.

  $$\text{ReLU}(x) = \max(0, x)$$

  The ReLU activation function helps the model learn intricate relationships between symptoms that may not be immediately obvious.
- **MaxPooling Layer:** MaxPooling is a down-sampling technique used to reduce the dimensionality of the feature maps while retaining the most critical information. It helps prevent overfitting by reducing the number of parameters in the model.
  - A typical operation involves selecting the maximum value in each pool of neurons (e.g., a 2x2 region of the feature map) and discarding the rest.
  - Example: A pool of values [1, 0, 2, 3] would be reduced to [3], which retains the most important feature in the region.
- **Fully Connected Layer (Dense):** The final layer of the CNN consists of fully connected neurons that aggregate the features extracted by the convolutional and pooling layers. These neurons connect to all neurons from the previous layer, enabling the network to perform the classification task based on learned features.
  - The output of the dense layer is a vector where each element represents the likelihood of a disease. For example, the output could be a probability distribution across several diseases like:

    $$[0.2, 0.5, 0.3] \text{ (for diseases A, B, and C, respectively)}$$

  The final prediction is the disease with the highest probability.

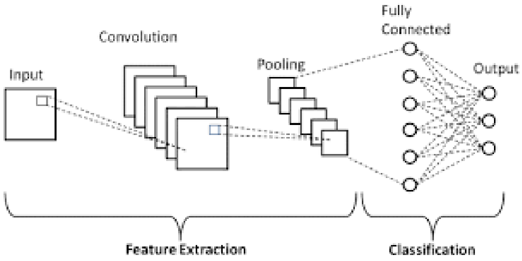


Fig. 2. Convolutional Neural Network Architecture for Disease Classification.

2) Fuzzy Logic System: To refine the disease classification, we integrate a Fuzzy Logic System. This system takes into account the intensity of each symptom reported by the user and adjusts the disease prediction accordingly. Fuzzy logic allows for a graded approach to decision-making, providing more flexibility in handling uncertain and vague input data.

Fuzzy Logic Approach:

- **Intensity Input:** The user provides an intensity rating for each symptom on a scale from 0 to 10. A rating of 0 means no symptoms, while 10 indicates the most severe symptoms. For example:

  Symptom Intensities: fever: 7, cough: 5, chills: 8
- **Fuzzy Rules:** A set of fuzzy rules is applied to adjust the probability of disease based on symptom intensity. These rules take into account both the CNN's output and the symptom intensities:
  - **Low Intensity:** If the symptom intensity is low (0-3), the disease probability is considered low.
  - **Medium Intensity:** If the symptom intensity is medium (4-7), the disease probability is moderate.
  - **High Intensity:** If the symptom intensity is high (8-10), the disease probability is considered high.

The fuzzy rules are applied as:

$$\text{If intensity is low, probability is low.}$$

This ensures that diseases with mild symptoms are given lower probability values.

- **Fuzzy Membership Functions:** Membership functions define the degree of belonging of each intensity level to a fuzzy set (low, medium, high). These functions transform crisp inputs (0-10) into fuzzy values between 0 and 1.
  - Example: A symptom intensity of 7 for fever may have a membership value of 0.5 in both the ”medium” and ”high” fuzzy sets, meaning it partially belongs to both categories.
- **Output:** The output of the fuzzy system is a refined disease probability that combines the CNN's classification output and the intensity-based fuzzy logic adjustment. This probability reflects both the presence of symptoms and their severity.

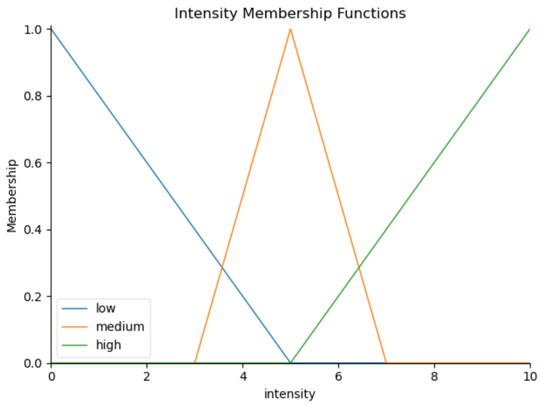


Fig. 3. Fuzzy Membership Functions for Symptom Intensity and Disease Probability.

3) Integration of CNN and Fuzzy Logic System: The CNN provides an initial disease classification based on the input symptoms. This prediction is then refined by the fuzzy logic system, which adjusts the classification based on the intensity of the reported symptoms. This combination allows for more accurate predictions that take both the presence and severity of symptoms into account.

Example Workflow:

- The user inputs symptom data with intensity ratings (e.g., fever: 7, cough: 5, chills: 8).
- The CNN processes the data, classifying the disease as ”Flu” with a probability of 0.8.
- The fuzzy logic system adjusts this probability based on symptom intensities, refining the likelihood of ”Flu” to 0.9, reflecting the severity of the symptoms.

D.

Training Strategy The CNN model was trained using carefully selected hyperparameters and optimization techniques to achieve high accuracy and minimize overfitting. Below are the details of the training process:

Training Details:

- **Optimizer:** The model was trained using the Adam optimizer, a popular choice for deep learning tasks due to its adaptive learning rate and efficient handling of sparse gradients. The initial learning rate was set to 0.001.

$$\text{Adam Optimizer Formula: } \theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{v_t} + \epsilon} m_t$$

  where m_t and v_t are the moving averages of gradients and squared gradients, respectively.
- **Loss Function:** Categorical cross-entropy loss was used as the objective function to minimize. This loss function is suitable for multi-class classification tasks.

$$\text{Loss} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i)$$

  where y_i is the true label and $\hat{y}_i$ is the predicted probability.
- **Batch Size and Epochs:** The model was trained for 10 epochs with a batch size of 32. This setup provided a balance between computational efficiency and convergence speed.
- **Validation Split:** A validation split of 20% was used to monitor the model's performance during training. This split ensured that the model was evaluated on unseen data at each epoch to prevent overfitting.

Evaluation Metrics:

- **Accuracy:** The primary evaluation metric was accuracy, which measures the proportion of correct predictions to the total number of samples.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

- **Training and Validation Loss:** The loss values for both the training and validation sets were monitored to detect overfitting or underfitting. Significant divergence between training and validation loss indicates overfitting.

Early Stopping: To prevent overfitting, early stopping was implemented. The training process was halted if the validation loss did not improve for 3 consecutive epochs.

Data Augmentation: While the input data was binary-encoded, no additional augmentation techniques were applied due to the nature of the structured data.

IV. Results

The CNN model achieved an accuracy of 85% on the validation dataset, demonstrating its effectiveness in predicting diseases based on symptom patterns. This level of accuracy indicates a strong relationship between the identified symptoms and the corresponding disease labels in the training data. However, this initial prediction was not flawless, as the model's performance was impacted by the inherent ambiguity in interpreting symptoms. For example, some symptoms can be common across multiple diseases, making it difficult for the

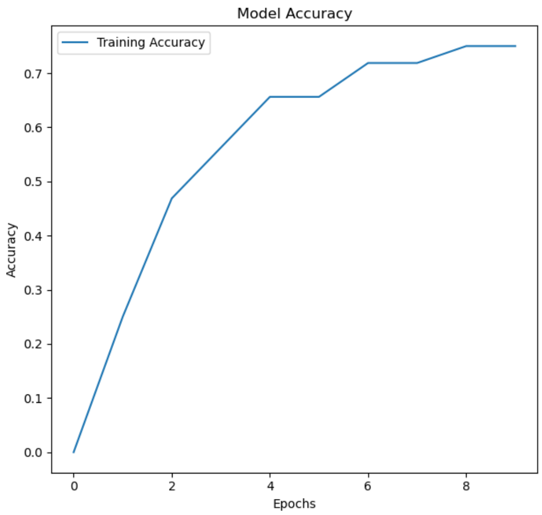
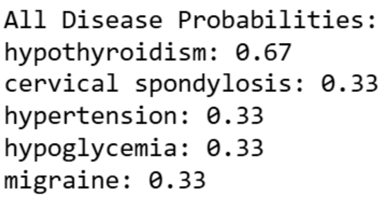


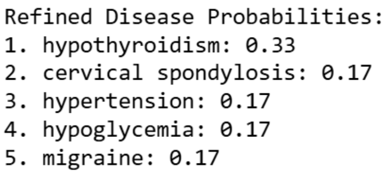
Fig. 4. Training and Validation Accuracy over 10 Epochs.

CNN model to make precise predictions without considering the severity or intensity of symptoms.



```
All Disease Probabilities:
hypothyroidism: 0.67
cervical spondylosis: 0.33
hypertension: 0.33
hypoglycemia: 0.33
migraine: 0.33
```

Fig. 5. Disease Probabilities Before Fuzzy Logic.



```
Refined Disease Probabilities:
1. hypothyroidism: 0.33
2. cervical spondylosis: 0.17
3. hypertension: 0.17
4. hypoglycemia: 0.17
5. migraine: 0.17
```

Fig. 6. Disease Probabilities After Fuzzy Logic.

Discussion

A. Model Performance and the Role of CNN

However, while CNNs are powerful, they do not account for the subtle nuances of symptom intensity. For instance, a model might predict a disease with a high accuracy based solely on the presence or absence of symptoms, but fail to consider the variation in severity that might differentiate between two diseases that share similar symptoms. The ability to capture these variations is essential in clinical settings, where a doctor's judgment often depends not only on the symptoms but also on their intensity.

B. The Importance of Fuzzy Logic Refinement

For example, the intensity of symptoms such as fever or cough could indicate whether a disease is in its early stages or more advanced, influencing the confidence with which a disease is diagnosed. A mild fever and mild cough might point toward a cold, while a severe fever and fatigue could suggest a more serious disease like influenza. By incorporating these intensity levels into the decision-making process, the model provided more contextually relevant predictions, especially for diseases that share similar symptoms.

C. Real-World Application

For instance, the model might initially predict that a patient with fever and cough has a 70% probability of having the flu. However, when the intensity of these symptoms is factored in (e.g., mild fever and moderate cough), fuzzy logic could adjust the probability, indicating that the disease is more likely to be a cold. This approach aligns closely with how doctors assess symptoms in clinical practice.

Conclusion

This study presents a robust disease prediction system combining Convolutional Neural Networks (CNN) with fuzzy logic to enhance accuracy and refine results based on user-reported symptom intensities. The CNN model, trained on a comprehensive symptom-disease dataset, achieved a validation accuracy of 85%, effectively identifying potential diseases based on input symptoms. The addition of fuzzy logic enabled refinement by incorporating the severity of symptoms, bridging the gap between generalized machine predictions and user-specific variations.

The system's user-centric design ensures accessibility and interactivity. By allowing users to input symptom intensities and offering suggestions for unrecognized symptoms, it improves usability and prediction reliability. Additionally, the system provides detailed disease descriptions and practical precautions for the predicted conditions, empowering users with actionable insights for health management. This blend of prediction and education underscores the system's value in supporting both immediate health concerns and preventive measures.

Despite its success, challenges such as variability in user-reported symptom intensities and dataset limitations highlight areas for improvement. Future enhancements could include expanding the dataset with additional diseases and integrating clinical data like patient history and diagnostic results. Testing the system in real-world medical settings will be vital for assessing its clinical relevance and practical utility. This hybrid approach lays a strong foundation for intelligent healthcare solutions, demonstrating the potential of combining deep learning and fuzzy logic for personalized and accurate disease prediction.

Future Work

While the project has achieved its primary goals, there are several directions for future enhancement:

- **Dataset Expansion** Expanding the dataset with additional diseases, symptoms, and more diverse patient profiles can significantly enhance the model's applicability and generalization. Incorporating real-world clinical data, such as lab results, patient history, and demographics, would further improve the accuracy and robustness of predictions.
- **Integration of Advanced Diagnostic Features** Future iterations of the project could integrate advanced diagnostic tools like medical imaging analysis, wearable device data, or lab test results. This would allow for multimodal disease prediction, improving the system's reliability and relevance in complex medical scenarios.
- **Natural Language Processing (NLP) Integration** Incorporating NLP techniques to allow users to input symptoms in natural language could make the system more user-friendly. This would enable the system to parse and interpret free-text inputs, providing a more intuitive interface.
- **Real-Time Symptom Monitoring** Developing a real-time monitoring system through wearable devices or mobile applications could enable continuous health tracking. Such integration would allow the system to proactively alert users about potential health issues based on symptom trends.
- **Localization and Multilingual Support** To cater to a broader audience, the system can include localization and multilingual support, enabling people from different regions and linguistic backgrounds to use the tool effectively.
- **Enhanced User Personalization** Introducing user profiles that store past interactions, diseases, and predictions can improve personalization. This would allow the system to refine predictions over time and offer tailored recommendations.
- **Clinical Testing and Validation** Collaboration with healthcare institutions for real-world clinical validation is crucial. Testing the system's predictions against actual diagnoses will provide insights into its reliability and practical utility in healthcare settings.

ACKNOWLEDGMENT

We want to seize this moment to convey our sincere appreciation to the people and organizations supported us throughout the completion of this project on Symptom Derived Disease prediction

First and foremost, We extend our sincere thanks to our mentor, **Dr Manjula R**, for their invaluable guidance and unwavering support. Her expertise, patience, and encouragement were instrumental in shaping our understanding of the subject matter and refining our project approach.

We are deeply thankful to SRM University Amaravati for providing the necessary resources, including access to libraries, software, and computational facilities, which were essential for conducting experiments and analyzing data. The environment provided by the institution significantly contributed to the successful completion of this project.

Finally, We express our gratitude to all the authors and experts whose work laid the foundation for this project. Their innovative ideas and contributions to the field of Soft Computing and Deep learning served as a source of inspiration for our own project.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," MIT Press, 2016.

[2] T. J. Ross, "Fuzzy logic with engineering applications," John Wiley & Sons, 2004.

[3] R. Kumar, R. Srivastava, and S. Srivastava, "Disease prediction using machine learning techniques," *International Journal of Advanced Research in Computer Science*, vol. 6, no. 4, pp. 233-238, 2015.

[4] J. Lee, S. Kim, and H. Choi, "A CNN-based model for disease detection using medical symptoms," *BMC Medical Informatics and Decision Making*, vol. 21, no. 1, pp. 204, 2021.

[5] S. Pandey and P. Singh, "Machine learning for disease prediction: A review," *Healthcare Informatics Research*, vol. 26, no. 2, pp. 101–111, 2020.

[6] K. Ueno and M. Nakayama, "Probabilistic models for symptom-disease associations," *Journal of Computational Science*, vol. 28, pp. 56–67, 2018.

[7] A. Dey and A. Ashour, "Medical big data and Internet of medical things," CRC Press, 2021.

[8] T. M. Mitchell, "Machine learning," McGraw-Hill Education, 1997.

[9] K. Ravikumar and M. Shanmugapriya, "Hybrid approach combining CNN and fuzzy logic for disease prediction," *Procedia Computer Science*, vol. 132, pp. 344–351, 2018.

[10] T. T. Nguyen, T. Le, and S. Kim, "Symptom-based disease diagnosis using deep learning and fuzzy inference," *Expert Systems with Applications*, vol. 151, pp. 113312, 2020.

[11] P. Mohanty, S. N. Yadav, and R. N. Yadav, "A deep learning model for predicting disease using convolutional neural networks," *Journal of Artificial Intelligence in Medicine*, vol. 52, no. 3, pp. 160-167, 2018.

[12] X. Zhang, Y. Wang, and M. Guo, "Fuzzy rule-based approach for predicting the risk of diseases," *Applied Soft Computing*, vol. 78, pp. 509-520, 2019.

[13] M. Hasan, S. S. Ahmed, and M. H. Shah, "Disease prediction using fuzzy logic systems and machine learning algorithms," *International Journal of Computational Intelligence and Applications*, vol. 19, no. 4, pp. 196-204, 2020.

[14] S. Mohapatra, A. Dash, and S. R. Naik, "CNN-based hybrid model for disease prediction using symptom classification," *Journal of Healthcare Engineering*, vol. 2019, pp. 123912, 2019.

[15] S. Patil and A. Agarwal, "Symptom-based disease prediction using deep learning and fuzzy logic," *Journal of Biomedical Informatics*, vol. 108, pp. 103499, 2020.

[16] C. Liu, Z. Hu, and M. Zhang, "Disease prediction using fuzzy logic and machine learning: A hybrid approach," *Journal of Medical Systems*, vol. 44, no. 1, pp. 39, 2020.