JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Development and Deployment of an AI-Based Medical Symptom Checker Using Natural Language Processing and Machine Learning

Asst. Prof. Dr. Sanju S. Anand[1]; Research Guide[1]; Student Researchers – Dhanya Damodar Naik, Sakita Mahabaleshwar Gouda, Sannidhi Vanakudari, and Harshita Annappa Naik

Department of Computer Science, Alva's College (Autonomous), Moodubidri, Mangalore, Karnataka, India

Email: sanjualvascollege@gmail.com

Abstract

Artificial Intelligence (AI) and Natural Language Processing (NLP) have become useful technologies for developing intelligent healthcare-support applications. This paper presents the development of an AI-based Medical Symptom Checker that predicts a possible disease from a user's natural-language description of symptoms. The proposed system uses a Symptom2Disease dataset containing 1,200 symptom descriptions belonging to 24 disease categories. Data preprocessing is performed to remove missing, empty, and duplicate records. The cleaned data are converted into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF), and a Logistic Regression classifier is trained for multi-class disease classification. After preprocessing, 1,155 records were available for model development. The data were divided into training and testing sets, with 231 samples used for final testing. The trained model correctly classified 215 of the 231 test samples, achieving an accuracy of 93.07%. The model obtained a macro-average precision of 0.94, recall of 0.93, and F1-score of 0.93. A confusion matrix was used to analyze class-wise performance and misclassifications. Finally, the trained model was exported in ONNX format and integrated into a C# Windows Forms desktop application using ONNX Runtime. The developed application accepts natural-language symptom descriptions and produces a preliminary disease prediction. The system is intended for educational and preliminary decision-support purposes and is not a replacement for professional medical diagnosis.

Keywords: Artificial Intelligence, Natural Language Processing, Machine Learning, Medical Symptom Checker, TF-IDF, Logistic Regression, ONNX, C#, Disease Classification.

1. Introduction

Healthcare applications increasingly use artificial intelligence and machine-learning techniques to analyze medical information and provide decision-support services. One potential application is a medical symptom checker that allows users to describe their symptoms and obtain information about a possible health condition.

Traditional symptom-checking systems often depend on manually defined rules. For example, a rule-based system may associate fever, chills, and sweating with malaria. Such systems require developers or medical experts to create and maintain a large number of rules. They may also have difficulty handling different ways in which users describe similar symptoms.

Natural Language Processing provides an alternative approach by allowing a computer to process symptom descriptions written in natural language. Machine-learning algorithms can learn statistical relationships between symptom descriptions and disease categories from labeled examples.

In this study, an AI-based Medical Symptom Checker is developed using NLP and machine learning. The system uses the Symptom2Disease dataset, which contains 1,200 natural-language symptom descriptions belonging to 24 disease categories. TF-IDF is used for text feature extraction and Logistic Regression is used as the classification algorithm.

The trained model is subsequently exported into ONNX format and integrated into a C# Windows Forms application. This demonstrates a complete workflow from dataset preparation and machine-learning model development to software deployment.

The main objectives of this research are:

- To develop an NLP-based disease classification model.
- To process natural-language descriptions of symptoms.
- To extract meaningful numerical features using TF-IDF.
- To train a machine-learning classifier for disease prediction.
- To evaluate the model using unseen test data.
- To integrate the trained model into a C# desktop application.
- To demonstrate a simple AI-based preliminary symptom-checking system.

2. Related Work

Machine learning has been increasingly investigated for applications involving healthcare data analysis, classification, and decision support. Traditional classification algorithms such as Logistic Regression, Naive Bayes, and Support Vector Machines have been widely used for text classification because of their computational efficiency and ability to work with high-dimensional sparse feature representations.

Natural Language Processing is particularly useful when healthcare information is supplied as free-form text. Users may describe the same symptoms using different words, sentence structures, or combinations of symptoms. NLP techniques can transform these descriptions into representations that machine-learning algorithms can process.

TF-IDF is a commonly used text-representation technique. It assigns numerical weights to terms according to their importance within a document and across the document collection. When combined with a suitable classifier, TF-IDF can provide an effective baseline for text-classification tasks.

Recent developments in healthcare AI have also introduced more advanced deep-learning and transformer-based approaches. However, traditional machine-learning methods remain useful for educational applications because they are comparatively simple to train, interpret, deploy, and execute on ordinary computer systems.

The present study focuses on a lightweight NLP and machine-learning approach and demonstrates its deployment in a C# Windows Forms environment.

3. Dataset

The Symptom2Disease dataset was used for this research. The dataset contains 1,200 symptom descriptions distributed across 24 disease categories, with approximately 50 descriptions per disease.

The dataset contains two main attributes:

- label – the corresponding disease category.
- text – the natural-language symptom description.

The 24 disease categories are:

- Psoriasis
- Varicose Veins
- Typhoid

- Chicken Pox
- Impetigo
- Dengue
- Fungal Infection
- Common Cold
- Pneumonia
- Dimorphic Hemorrhoids
- Arthritis
- Acne
- Bronchial Asthma
- Hypertension
- Migraine
- Cervical Spondylosis
- Jaundice
- Malaria
- Urinary Tract Infection
- Allergy
- Gastroesophageal Reflux Disease
- Drug Reaction
- Peptic Ulcer Disease
- Diabetes

An example of the dataset structure is shown in Table 1.

Label	**Symptom Description**
Malaria	Fever, chills, sweating and headache
Dengue	High fever, headache and joint pain
Migraine	Severe headache accompanied by nausea
Common Cold	Sneezing, cough and runny nose

4. Proposed System

The proposed Medical Symptom Checker consists of two major stages: training and deployment.

Training Stage:

Dataset → Data Preprocessing → Training/Test Split → TF-IDF Feature Extraction → Logistic Regression → Model Evaluation → ONNX Model

Deployment Stage:

User → Natural-Language Symptoms → C# Windows Forms Application → ONNX Runtime → Trained AI Model → Predicted Disease

An important characteristic of the proposed system is that the deployed C# application does not perform a direct search of the original Excel/CSV dataset. The dataset is used during the training phase to learn statistical relationships between symptom descriptions and disease categories. During deployment, a new symptom description is provided to the trained model, which generates a prediction.

5. Data Preprocessing

Data preprocessing was performed before model training to improve the quality and consistency of the input data.

The preprocessing procedure included:

- Removing missing values.
- Removing empty symptom descriptions.
- Removing duplicate records.
- Selecting the symptom text and disease label required for classification.

The original dataset contained 1,200 records. Following preprocessing, the working dataset contained 1,155 records. The reduction in the number of records resulted from the removal of records that did not satisfy the preprocessing requirements.

6. Training and Testing Strategy

The cleaned dataset was divided into training and testing subsets. Approximately 80% of the data was used for training, while approximately 20% was reserved for testing. The test set was not used during model training. This allows the performance of the trained classifier to be evaluated on symptom descriptions that were not used to learn the model. The final test set contained 231 samples. The use of a stratified split helped maintain the representation of different disease categories in the training and testing sets.

7. Natural Language Processing Using TF-IDF

Machine-learning algorithms require numerical input. Since the dataset contains natural-language symptom descriptions, the text must first be converted into numerical features. The proposed system uses Term Frequency-Inverse Document Frequency (TF-IDF).

TF-IDF assigns weights to words based on their occurrence within a document and their frequency across the complete collection of documents.

The term-frequency component can be represented as:

$$TF(t,d) = \text{Number of occurrences of } t \text{ in } d \,/\, \text{Total number of terms in } d$$

The inverse document frequency component can be represented as:

$$IDF(t) = \log(N / df(t))$$

The combined TF-IDF value is:

$$TFIDF(t,d) = TF(t,d) \times IDF(t)$$

where N is the number of documents and df(t) is the number of documents containing term t.

In the implemented system, the TF-IDF vectorizer generated a vocabulary containing 7,670 features. The resulting representation was sparse because only a subset of the vocabulary occurred in any individual symptom description.

For example:

"I have fever, chills, sweating and headache" → TF-IDF → Numerical feature vector

8. Machine Learning Classification

8.1 Logistic Regression

Logistic Regression was selected as the classification algorithm for the proposed system. Although Logistic Regression is commonly associated with binary classification, it can also be applied to multi-class classification problems. In this project, the classifier predicts one of the 24 disease categories. The model learns relationships between TF-IDF features and disease labels during training. When a user enters a new symptom description, the description is transformed into TF-IDF features and supplied to the trained classifier. The classifier then generates a disease prediction based on patterns learned during training.

The overall classification process can be represented as:

Symptom Description

↓

TF-IDF Feature Extraction

↓

Logistic Regression

↓

24 Disease Classes

↓

Predicted Disease

9. Experimental Results

The trained model was evaluated using 231 unseen test samples. The model correctly classified 215 samples and incorrectly classified 16 samples.

Metric	**Result**
Total test samples	231
Correct predictions	215
Incorrect predictions	16
Accuracy	93.07%
Macro Precision	0.94
Macro Recall	0.93
Macro F1-score	0.93
Weighted Precision	0.93
Weighted Recall	0.93
Weighted F1-score	0.93

The accuracy was calculated as:

Accuracy = (Correct Predictions / Total Test Samples) × 100

Accuracy = (215 / 231) × 100 = 93.07%

The result indicates that the model correctly classified approximately 93 out of every 100 test descriptions.

10. Disease-Wise Performance

Disease	**Precision**	**Recall**	**F1-score**
Acne	1.00	1.00	1.00
Arthritis	1.00	1.00	1.00
Bronchial Asthma	1.00	1.00	1.00
Cervical Spondylosis	1.00	1.00	1.00
Chicken Pox	0.80	0.80	0.80
Common Cold	1.00	1.00	1.00

Dengue	0.78	0.70	0.74
Dimorphic Hemorrhoids	1.00	1.00	1.00
Fungal Infection	0.91	1.00	0.95
Hypertension	0.91	1.00	0.95
Impetigo	1.00	1.00	1.00
Jaundice	1.00	1.00	1.00
Malaria	1.00	1.00	1.00
Migraine	1.00	1.00	1.00
Pneumonia	1.00	1.00	1.00
Psoriasis	1.00	0.80	0.89
Typhoid	0.77	1.00	0.87
Varicose Veins	1.00	1.00	1.00
Allergy	1.00	0.80	0.89
Diabetes	0.73	0.80	0.76
Drug Reaction	0.90	0.90	0.90
Gastroesophageal Reflux Disease	0.91	1.00	0.95
Peptic Ulcer Disease	0.88	0.70	0.78
Urinary Tract Infection	0.90	0.90	0.90

The macro-average F1-score of 0.93 indicates good overall performance across the different disease categories. Several categories achieved perfect scores on the available test samples, whereas relatively lower F1-scores were observed for Diabetes, Dengue, Peptic Ulcer Disease, and Chicken Pox.

11. Confusion Matrix Analysis

A confusion matrix was generated to analyze the classification behavior of the model. The matrix showed that most predictions were concentrated along the diagonal, indicating correct classifications.

The model performed particularly well for several categories, including Acne, Arthritis, Bronchial Asthma, Cervical Spondylosis, Common Cold, Impetigo, Jaundice, Malaria, Migraine, Pneumonia, and Varicose Veins.

Some misclassifications occurred between disease categories having overlapping symptom descriptions. For example, some Dengue cases were classified as Chicken Pox or Typhoid. Some Diabetes cases were classified as Drug Reaction or Urinary Tract Infection. Peptic Ulcer Disease was also occasionally confused with Typhoid and Gastroesophageal Reflux Disease.

These results demonstrate that symptom overlap presents a challenge for text-based disease classification.

Figure 1. Confusion matrix of the 24-class disease classification model. The confusion matrix generated during the experiment should be inserted here.

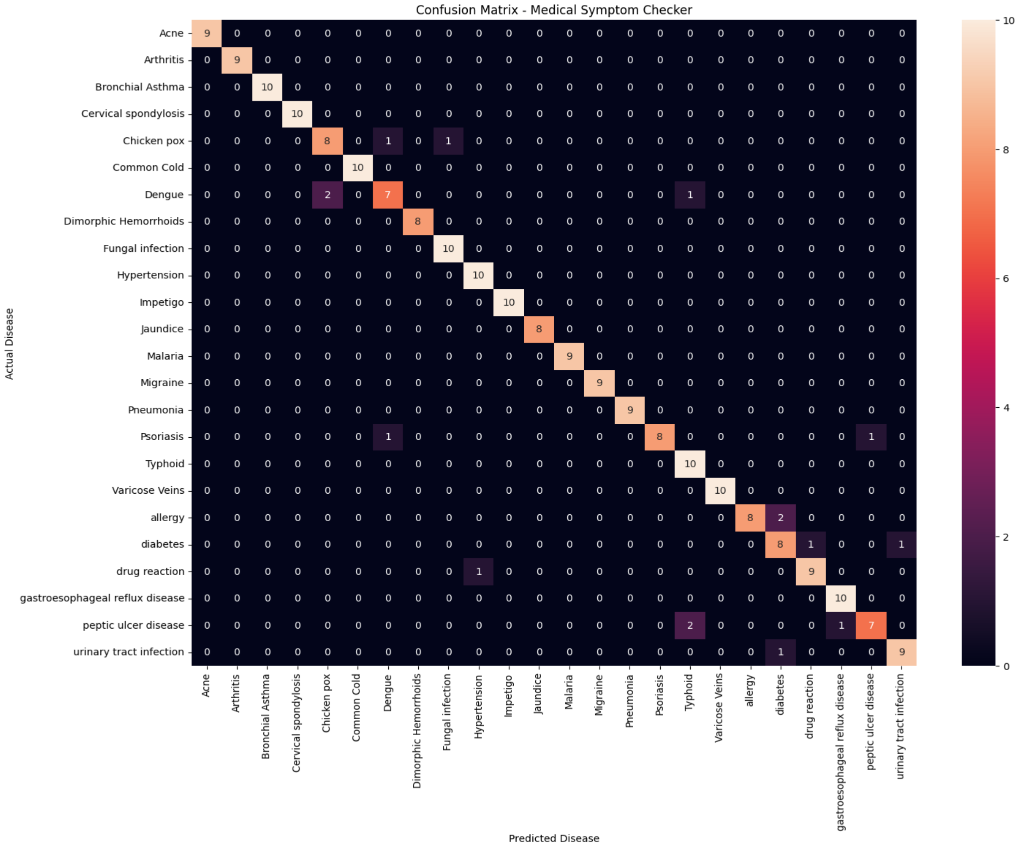

Figure:1

12. Analysis of Incorrect Predictions

The model generated 16 incorrect predictions from the 231 test samples. Examples include:

Actual Disease	**Predicted Disease**
Diabetes	Drug Reaction
Dengue	Typhoid
Diabetes	Urinary Tract Infection
Allergy	Diabetes
Peptic Ulcer Disease	Typhoid
Dengue	Chicken Pox
Psoriasis	Peptic Ulcer Disease
Peptic Ulcer Disease	Gastroesophageal Reflux Disease
Drug Reaction	Hypertension
Urinary Tract Infection	Diabetes
Chicken Pox	Dengue
Allergy	Diabetes
Chicken Pox	Fungal Infection
Psoriasis	Dengue
Peptic Ulcer Disease	Typhoid

These errors are important because they show that the classifier is not simply performing an exact lookup of the input sentence. Instead, it uses the numerical features generated from the text and the statistical patterns learned during training. When symptom descriptions of different diseases contain similar terms, the classifier may assign the description to the wrong disease category.

13. System Implementation

After training and evaluation, the model was prepared for deployment in a C# Windows Forms application. The complete trained pipeline was exported to ONNX (Open Neural Network Exchange) format. The resulting model file was medical_symptom_model.onnx. The C# application uses ONNX Runtime to load the model and perform inference (Figure-2).

The application contains a simple user interface with:

- Application title
- Symptom input area
- Check Symptoms button
- Prediction/result area

A user can enter a natural-language description such as "I have fever, chills, sweating and headache." The application passes the description to the trained model and displays the predicted disease.

Implementation workflow:

User enters symptoms → C# Windows Forms → ONNX Runtime → medical_symptom_model.onnx → AI prediction → Predicted disease displayed

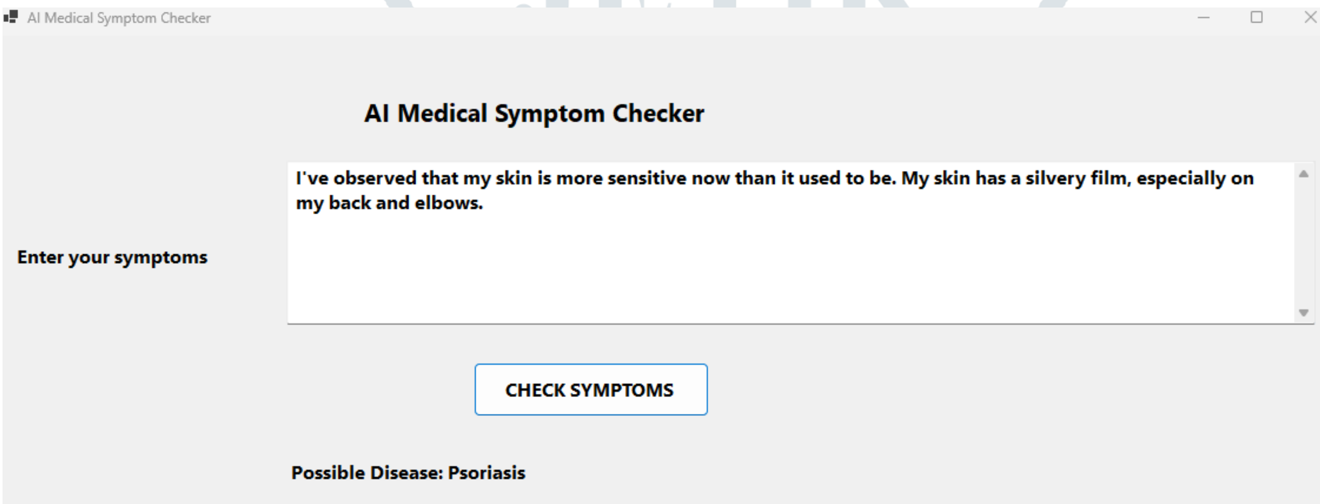

Figure-2

14. AI Contribution of the System

The AI component of the proposed system is the trained TF-IDF + Logistic Regression classification pipeline. The original dataset is used during the training stage. The learned model is then saved and deployed independently.

Therefore, the application does not simply perform a database or spreadsheet lookup. The dataset is used to learn statistical relationships during training. During deployment, a new symptom description is converted into features and processed by the trained classifier.

For example, a user may enter a sentence that is different from all sentences in the original dataset. The model can still produce a prediction because it has learned statistical relationships between symptom-related terms and disease categories. This represents a machine-learning-based classification approach rather than a simple database or spreadsheet lookup.

15. Advantages

- Natural-language input: Users can describe symptoms in ordinary language rather than selecting only predefined options.
- Machine-learning-based prediction: Disease prediction is generated by a trained machine-learning classifier.
- Lightweight model: TF-IDF and Logistic Regression require relatively modest computational resources compared with large deep-learning models.
- Desktop deployment: The trained model can be integrated into a C# Windows Forms application using ONNX Runtime.
- Simple user interface: The application provides a straightforward interface suitable for demonstration and educational use.
- End-to-end AI workflow: The project demonstrates the complete process from dataset preparation to model deployment.

16. Limitations

Despite achieving 93.07% accuracy, several limitations must be considered. First, the dataset contains only 1,200 original symptom descriptions and 24 disease categories. Real-world medical conditions are substantially more diverse.

Second, the model is trained on a specific dataset and may not perform equally well on symptom descriptions from different populations or sources.

Third, several diseases have overlapping symptoms. This can lead to incorrect classifications, as demonstrated by the confusion matrix and error analysis.

Fourth, the reported test performance is based on a relatively small dataset. Larger and independently collected datasets would be required to establish generalizability.

Finally, the system provides a preliminary classification, not a clinical diagnosis. It should not be used as a replacement for a qualified healthcare professional.

17. Ethical Considerations

Healthcare-related AI systems require careful consideration of safety and responsible use. The proposed system should clearly communicate that its output is a machine-learning prediction rather than a confirmed diagnosis.

Users should be advised to consult qualified healthcare professionals for medical assessment, particularly when symptoms are severe, persistent, or potentially associated with an emergency condition.

Future versions of the system should use clinically validated datasets and undergo appropriate validation before being considered for real-world healthcare use.

18. Future Scope

- Increasing the size and diversity of the training dataset.
- Using clinically validated medical datasets.
- Comparing Logistic Regression with other machine-learning algorithms.
- Investigating transformer-based NLP models.
- Supporting multiple languages.
- Adding voice-based symptom input.
- Developing a web or mobile version.
- Providing multiple possible predictions with confidence scores.

- Adding a medically validated knowledge base.
- Performing external validation using an independent dataset.
- Adding user history and report-generation features.
- Improving the interface for accessibility and usability.

A future system could also combine machine-learning classification with a medically validated knowledge base to provide more informative and safer decision-support functionality.

19. Conclusion

This research presented the development and deployment of an AI-based Medical Symptom Checker using Natural Language Processing and machine learning. The Symptom2Disease dataset containing 1,200 symptom descriptions from 24 disease categories was used for model development.

After preprocessing, 1,155 records were used for the experimental workflow. TF-IDF was applied to transform natural-language symptom descriptions into numerical features, producing a feature space of 7,670 features. Logistic Regression was then trained to classify symptom descriptions into 24 disease categories.

The model was evaluated using 231 unseen test samples and correctly classified 215 samples, resulting in an accuracy of 93.07%. The model achieved a macro precision of 0.94, macro recall of 0.93, and macro F1-score of 0.93. The confusion matrix and error analysis demonstrated strong performance for several disease categories while also identifying confusion among diseases with overlapping symptoms.

For practical demonstration, the trained machine-learning pipeline was exported to ONNX format and integrated into a C# Windows Forms application using ONNX Runtime. The application accepts natural-language symptoms and provides a preliminary disease prediction.

The project demonstrates how AI and NLP techniques can be combined with conventional software development to create an intelligent application. However, because the model is trained on a limited dataset and has not undergone clinical validation, the system should be considered an educational and preliminary decision-support application rather than a medical diagnostic system.

20.References

[1] S. Y. Yadav, M. Dudhedia, P. G. Chilveri, R. M. Lenka, R. V. Kulkarni, and K. Nagaiah, "Enhancing medical diagnosis with a machine learning-based symptom checker for health assessment," *African Journal of Science, Technology, Innovation and Development*, vol. 18, no. 2, pp. 256–269, 2026.

[2] E. S. J. Atmadji, A. P. Wibowo, and E. Faizal, "Comparative Study of Machine Learning Methods for Disease Classification Based on Natural Language Symptom Descriptions," *International Journal of Artificial Intelligence in Medical Issues*, vol. 3, no. 2, 2025.

[3] S. Mukhopadhyay, S. Gite, K. Kotecha, G. Selvachandran, P. Anand, and A. Abraham, "AI-driven disease classification from unstructured textual symptom descriptions: A multi-model NLP benchmarking study across categorical, UMLS-derived, and synthetic datasets," *Computers in Biology and Medicine*, vol. 125, 2026.

[4] "Let's ask the patient: disease prediction based on patients' symptom descriptions in free text," *PubMed*, 2025. The study uses TF-IDF-based text representation and machine-learning classification of free-text symptom descriptions.

[5] "Leveraging natural language processing and machine learning to identify chronic conditions from primary care electronic medical records," *Scientific Reports*, 2026. The study discusses TF-IDF representation for clinical text and compares it with more complex NLP approaches.

[6] Q. Li, S. Zhao, S. Zhao, and J. Wen, "Logistic Regression Matching Pursuit algorithm for text classification," *Knowledge-Based Systems*, vol. 277, article 110761, 2023.

[7] P. Jajal, W. Jiang, A. Tewari, E. Kocinare, J. Woo, A. Sarraf, Y.-H. Lu, G. K. Thiruvathukal, and J. C. Davis, "Interoperability in Deep Learning: A User Survey and Failure Analysis of ONNX Model Converters," *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 1466–1478, 2024.

[8] P. Jajal, W. Jiang, A. Tewari, E. Kocinare, J. Woo, A. Sarraf, Y.-H. Lu, G. K. Thiruvathukal, and J. C. Davis, "Analysis of Failures and Risks in Deep Learning Model Converters: A Case Study in the ONNX Ecosystem," 2023.

[9] ONNX, "Open Neural Network Exchange: An Open Standard for Machine Learning Interoperability," ONNX Project.

[10] Microsoft, "ONNX Runtime: Cross-Platform Machine Learning Inference and Training Accelerator," Microsoft ONNX Runtime Documentation.

[11] L. Z. et al., "A disease inference method based on symptom extraction and bidirectional Long Short Term Memory networks," *Methods*, vol. 173, pp. 75–82, 2020. The study combines symptom extraction, TF-IDF, Word2Vec, and BiLSTM for disease inference from clinical text.

[12] E. Hassan, T. Abd El-Hafeez, and M. Y. Shams, "Optimizing classification of diseases through language model analysis of symptoms," *Scientific Reports*, vol. 14, Article 1507, 2024.

[13] "Natural Language Processing of Clinical Notes on Chronic Diseases: Systematic Review," *Journal of Medical Internet Research*, 2019. This review examines the application of NLP and machine learning to free-text clinical notes.

[14] "Let's ask the patient: disease prediction based on patients' symptom descriptions in free text," 2025. The study uses TF-IDF representations and machine-learning classification to analyze patient-reported free-text symptoms.

[15] S. Mukhopadhyay, S. Gite, K. Kotecha, G. Selvachandran, P. Anand, and A. Abraham, "AI-driven disease classification from unstructured textual symptom descriptions: A multi-model NLP benchmarking study across categorical, UMLS-derived, and synthetic datasets," *Computers in Biology and Medicine*, vol. 125, Article 109281, 2026. DOI: 10.1016/j.compbiolchem.2026.109281.

[16] S. Pasi, S. Degadwala, and M. Joshi, "Symptom-Based Classification of Common Syndromes Using Machine Learning: A Review," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 12, no. 1, 2026.

[17] E. S. J. Atmadji, A. P. Wibowo, and E. Faizal, "Comparative Study of Machine Learning Methods for Disease Classification Based on Natural Language Symptom Descriptions," *International Journal of Artificial Intelligence in Medical Issues*, vol. 3, no. 2, 2025. DOI: 10.56705/ijaimi.v3i2.361.

[18] "AI-based disease category prediction model using symptoms from low-resource Ethiopian language: Afaan Oromo text," 2024. The study evaluates machine-learning and deep-learning approaches using TF-IDF and word embeddings for symptom-based disease classification.

[19] H. Sakai and S. S. Lam, "Large Language Models for Health Care Text Classification: Systematic Review," *JMIR AI*, 2026. The review examines the use of large language models for healthcare text-classification tasks.

[20] "Leveraging natural language processing and machine learning to identify chronic conditions from primary care electronic medical records," *Scientific Reports*, 2026. The study discusses TF-IDF-based representation of clinical text and its use in machine-learning classification.

****