A

Mini Project Report on

**Disease diagnosis using machine learning**

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

IN

### Computer Science & Engineering

### Artificial Intelligence & Machine Learning

by

Om Panchal (22106025)

Ashutosh Pandey(22106121)

Ayush Gupta (22106074)

Under the guidance of

## Prof. Poonam Tiware



### Department of Computer Science & Engineering

### (Artificial Intelligence & Machine Learning)

**A. P. Shah Institute of Technology**

**G. B. Road, Kasarvadavali, Thane (W)-400615**

**University Of Mumbai**

**2024-2025**

## 

## A. P. SHAH INSTITUTE OF TECHNOLOGY

## CERTIFICATE

This is to certify that the project entitled “**Disease diagnosis using machine learning”** is a bonafide work of Om Panchal (22106025), Aashuthosh Pandey (22106121), Ayush Gupta (22106074) submitted to the University of Mumbai in partial fulfillment of the requirement for the award of **Bachelor of Engineering** in **Computer Science & Engineering (Artificial Intelligence & Machine Learning).**

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Prof. Poonam Tiware | Dr. Jaya Gupta |
| Mini Project Guide | Head of Department |

## 

## A. P. SHAH INSTITUTE OF TECHNOLOGY

## Project Report Approval

This Mini project report entitled“**Disease diagnosis using machine learning*”*** by **Om Panchal, Ayush gupta , and Aashuthosh pandey**is approved for the degree of ***Bachelor of Engineering*** in ***Computer Science &Engineering***, (AIML) ***2024-25***.

##### External Examiner:

##### Internal Examiner:

Place: APSIT, Thane

Date:

**Declaration**

##### We declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

|  |  |  |
| --- | --- | --- |
| Om Panchal | Aashuthosh Pandey | Ayush Gupta |
| (22106025) | (21203024) | (21203026) |

#### ABSTRACT

A new diagnostic tool has been developed to assist in identifying diseases based on patient symptoms. This tool leverages machine learning techniques, specifically Support Vector Classification (SVC), to predict potential diseases and provide relevant dietary recommendations. The system, implemented using Flask, allows users to input their symptoms, which are then processed to generate a disease prediction. The predicted disease is matched with a dataset of dietary recommendations tailored to manage the identified condition. This approach aims to enhance the efficiency of healthcare diagnostics by integrating symptom analysis with personalized dietary guidance, offering a practical solution for improving disease management and patient care. The tool’s design addresses the need for accurate, accessible diagnostic support, particularly beneficial in settings with limited resources.

Keywords:

healthcare diagnostics, disease prediction, dietary recommendations, machine learning, Flask, Support Vector Classification (SVC)

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| Index | | | Page no. |
| Chapter-1 | | |  |
|  | Introduction | |  |
|  |  |  |  |
| Chapter-2 | | |  |
|  | Literature Survey | |  |
|  | 2.1 | History |  |
|  | 2.1 | Review |  |
|  |  |  |  |
| Chapter-3 | | |  |
|  | Problem Statement | |  |
|  |  |  |  |
| Chapter-4 | | |  |
|  | Experimental Setup | |  |
|  | 4.1 | Hardware setup |  |
|  | 4.2 | Software Setup |  |
|  |  |  |  |
| Chapter-5 | | |  |
|  | Proposed system and Implementation | |  |
|  | 5.1 | Block Diagram of proposed system |  |
|  | 5.2 | Description of Block diagram |  |
|  | 5.3 | Implementation |  |
|  |  |  |  |
| Chapter-6 | | |  |
|  | Conclusion | |  |
|  |  |  |  |
| References | | |  |
|  |  |  |  |

# CHAPTER 1 INTRODUCTION

### INTRODUCTION

In the realm of modern healthcare, accurate and timely diagnosis is fundamental to effective disease management and optimal patient care. With the increasing complexity and variety of medical conditions, coupled with the extensive range of symptoms that may be associated with various diseases, the demand for sophisticated diagnostic tools has never been higher. Traditional diagnostic methods, while invaluable, often rely heavily on clinical expertise and can be constrained by limitations in both resources and accessibility. To address these challenges, a new diagnostic system has been introduced, designed to streamline and enhance the diagnostic process through the integration of advanced machine learning techniques and practical healthcare applications.

This innovative system employs Support Vector Classification (SVC) algorithms to process and analyze patient-reported symptoms, offering predictions about potential diseases with notable accuracy. By utilizing a comprehensive dataset, the system not only identifies the most probable conditions but also provides personalized dietary recommendations tailored to the specific needs of each patient. This dual-functionality not only aids in disease identification but also supports effective management through dietary adjustments, making it a valuable tool for both patients and healthcare providers.

Developed using Flask, the system features an intuitive user interface that simplifies the process of inputting symptoms and receiving both diagnostic predictions and dietary suggestions. Its design is particularly advantageous in resource-limited settings, where access to specialized medical care and diagnostic facilities may be limited. By offering a practical and accessible solution, this tool empowers users to take a proactive role in their health management and assists healthcare providers in delivering more targeted and personalized care.

In addition to its immediate applications in individual patient care, this system represents a broader advancement in the integration of technology and healthcare. It demonstrates the potential of machine learning to transform diagnostic practices, offering a scalable and adaptable solution that can be applied across various medical contexts. As healthcare continues to evolve, such innovations promise to play a pivotal role in enhancing diagnostic accuracy, optimizing treatment plans, and ultimately improving patient outcomes on a global scale.

# CHAPTER 2 LITERATURE SURVEY

#### LITERATURE SURVEY

###### 2.1-HISTORY

The journey of integrating technology into healthcare diagnostics began in earnest during the mid-20th century with the advent of early computing systems. In the 1960s, the development of the first computer-based diagnostic tools marked the initial foray into using technology to aid medical decision-making. Early systems, such as the MYCIN program developed in the 1970s, utilized rule-based algorithms to assist in diagnosing bacterial infections and recommending antibiotics. These pioneering efforts laid the groundwork for more sophisticated systems by demonstrating the potential of computers to enhance clinical decision support. However, these early systems were limited by their reliance on predefined rules and lacked the capacity for handling the complex, unstructured data typical in medical practice.

The 1980s and 1990s ushered in a new era of diagnostic technology with the emergence of more advanced computational methods and the introduction of artificial intelligence. During this period, the development of knowledge-based systems and expert systems became prominent. Systems like INTERNIST-I, developed in the 1980s, were designed to assist physicians in diagnosing a wide range of conditions by utilizing a vast repository of medical knowledge. This era also saw the rise of decision support systems that combined clinical guidelines with computational algorithms, providing a more dynamic approach to diagnostics. Despite these advancements, the systems of this era were still constrained by limited computational power and the complexity of integrating diverse data sources.

The early 2000s marked a significant turning point with the advent of machine learning and the expansion of electronic health records (EHRs). The availability of large datasets and improved computational capabilities enabled the development of more sophisticated diagnostic models. Machine learning algorithms, such as neural networks and support vector machines, began to gain prominence in medical diagnostics, offering the ability to analyze complex patterns and predict disease outcomes with greater accuracy. This period also saw the rise of data-driven approaches to healthcare, where the focus shifted from rule-based systems to models capable of learning from vast amounts of medical data. The integration of EHRs provided a rich source of patient information, further enhancing the capabilities of diagnostic systems.

In the 2010s and beyond, the convergence of big data, advanced algorithms, and cloud computing has revolutionized the field of healthcare diagnostics. The proliferation of wearable health devices and mobile health applications has generated an unprecedented volume of health data, driving innovations in real-time monitoring and personalized medicine. AI and deep learning technologies have further advanced diagnostic capabilities, enabling systems to process and analyze high-dimensional data with remarkable precision. These advancements have led to the development of diagnostic tools that offer not only predictions but also actionable insights and personalized treatment recommendations. As technology continues to evolve, the future of healthcare diagnostics promises even greater integration of AI, enhanced data analytics, and improved accessibility for patients and healthcare providers alike.

#### 2.2-LITERATURE REVIEW

**2.2.1. Diagnostic Systems and Machine Learning**

The application of machine learning in diagnostic systems represents a significant advancement in healthcare technology. Support Vector Machines (SVMs) and other machine learning algorithms, such as Random Forests and Neural Networks, have been extensively researched and applied to disease prediction. Lippi et al. (2018) provided a comprehensive review of how SVMs have been used to diagnose various conditions, including cancer and cardiovascular diseases. These studies highlight the models' ability to handle complex, multidimensional data and provide accurate predictions based on patient symptoms and clinical history. Rajkomar et al. (2019) further demonstrated the efficacy of deep learning techniques in predicting patient outcomes, emphasizing their potential to revolutionize diagnostics through high-dimensional data analysis and pattern recognition.

**2.2.2 Disease Prediction and Management**

Predictive modeling in disease management has garnered substantial attention in recent research. Deo (2015) explored the role of machine learning in predicting the onset of diseases such as diabetes and heart disease, noting the potential for early intervention and personalized treatment plans. Jha et al. (2020) expanded on these findings by incorporating longitudinal data to enhance prediction accuracy, thus improving early detection and patient management strategies. These advancements in predictive analytics enable healthcare providers to identify high-risk individuals and tailor interventions more effectively, leading to better health outcomes and more efficient resource utilization.

**2.2.3. Dietary Recommendations and Disease Management**

The integration of dietary recommendations into disease management is an emerging field that complements traditional medical treatments. Heshmat et al. (2018) investigated the impact of personalized dietary interventions on chronic conditions such as hypertension and diabetes, demonstrating that tailored dietary plans can significantly improve patient health outcomes. Yao et al. (2021) further explored this concept by developing dietary guidelines based on machine learning models that account for individual health profiles and dietary preferences. The inclusion of dietary recommendations within diagnostic systems represents a holistic approach to healthcare, addressing both medical and lifestyle factors to enhance overall patient well-being.

**2.2.4. Technological Integration and Accessibility**

Technological integration into accessible platforms has been a key factor in the widespread adoption of diagnostic tools. Reddy et al. (2017) discussed the advantages of web-based diagnostic applications, highlighting their ability to provide real-time support and facilitate user-friendly interactions. Gupta et al. (2020) expanded on this by examining the impact of mobile and web applications on healthcare accessibility, particularly in underserved areas. These studies underscore the importance of making diagnostic tools accessible to a broader population, improving healthcare delivery in both urban and rural settings. The shift towards digital platforms has enabled more widespread use of advanced diagnostic technologies, making them available to users who might otherwise lack access to specialized medical care.

# CHAPTER 3

# Problem Statement

**Problem statement**

The rapid advancement of technology in healthcare has introduced new opportunities for improving diagnostic accuracy and patient care. However, despite these advancements, significant challenges remain in the effective application of diagnostic tools, particularly in the realm of machine learning and artificial intelligence. One of the primary issues is the gap between the development of sophisticated diagnostic algorithms and their practical implementation in real-world clinical settings. While machine learning models can achieve high levels of accuracy in controlled environments, translating these results into effective, everyday use in diverse healthcare contexts remains a major hurdle. This disconnect often results in underutilization of these advanced tools, limiting their potential benefits for patient care.

Another critical issue is the integration of diagnostic tools with existing healthcare infrastructure. Many current systems struggle to seamlessly incorporate advanced diagnostic models into established workflows, leading to inefficiencies and potential disruptions in clinical practice. This challenge is compounded by the variability in healthcare settings, where resources and technical capabilities can vary significantly. As a result, there is a need for diagnostic tools that are not only technologically advanced but also adaptable and easy to integrate into diverse healthcare environments. Ensuring that these tools can be effectively deployed and used across different settings is essential for maximizing their impact on patient outcomes.

Additionally, data privacy and security concerns pose significant challenges to the adoption of machine learning-based diagnostic tools. The use of personal health data to train and operate these models raises important questions about data protection and patient consent. Ensuring that diagnostic systems adhere to stringent privacy regulations and maintain the confidentiality of patient information is crucial for gaining trust and acceptance from both healthcare providers and patients. Addressing these concerns requires a balanced approach that safeguards data while enabling the effective use of diagnostic technologies.

# CHAPTER 4

# Experimental Setup

#### 4.Experimental Setup

#### 4.1 Hardware Setup

1. **Processor (CPU):**
   * A dual-core processor, such as an Intel i3, is the minimum requirement to ensure that the application runs smoothly. This will allow for basic operations without significant lag. However, for more demanding tasks, such as model predictions and data processing, a quad-core processor (like an Intel i5) is recommended for optimal performance.
2. **Memory (RAM):**
   * A minimum of 4 GB of RAM is necessary to run the Flask application alongside other applications on the system. This amount of memory will support basic data processing and user interactions. For a more responsive experience, especially when handling larger datasets or running multiple applications, 8 GB or more is recommended.
3. **Storage:**
   * Users should have at least 10 GB of available hard disk space to accommodate the application files, datasets, and model files. This storage will ensure that the system can efficiently manage the data and support future updates. An SSD (Solid State Drive) is recommended to improve data access speed and overall system performance.
4. **Graphics Processing Unit (GPU):**
   * While a dedicated GPU is not strictly required for running the application, having one can enhance performance, especially during model training and complex computations. A mid-range GPU can significantly speed up processing times for large datasets. If users plan to perform advanced machine learning tasks, investing in a capable GPU is advisable.
5. **Network:**
   * A stable internet connection is essential for downloading necessary dependencies and any updates for the software. This will also facilitate online resources, such as documentation and community support. For applications that may require cloud hosting or additional data access, a reliable connection is crucial.

### Software Requirements

1. **Operating System:**
   * The application is compatible with Windows 10 or later, macOS Mojave (10.14) or later, and recent Linux distributions. Each operating system should be updated to ensure compatibility with the latest software. Students should ensure their OS supports Python and related packages.
2. **Python:**
   * Python 3.6 or later is required to run the application efficiently, with Python 3.8 or higher recommended for better library compatibility. Users should install Python from the official website or through a package manager. Ensuring that Python is correctly set up is crucial for running the Flask app and its dependencies.
3. **Flask Framework:**
   * Flask is the primary web framework used to build the application. It can be easily installed using pip, Python’s package manager, with a simple command. This lightweight framework allows for rapid development and deployment of web applications, making it suitable for the project.
4. **Data Processing Libraries:**
   * Essential libraries like NumPy, pandas, and scikit-learn must be installed to facilitate data handling, analysis, and machine learning tasks. These libraries provide powerful tools for numerical computations and data manipulation. Users can install them easily via pip to enhance the application’s functionality.
5. **Model Handling:**
   * The pickle library, included in Python's standard library, is used for loading the trained machine learning model. This allows for efficient storage and retrieval of the model, ensuring it can be used for predictions. Users should be familiar with how to utilize pickle to save and load Python objects effectively.
6. **Web Browser:**
   * A modern web browser, such as Google Chrome, Mozilla Firefox, or Safari, is necessary to access the Flask application via localhost. These browsers support the latest web standards and ensure a seamless user experience. Users should ensure their browser is up to date for optimal performance.
7. **Text Editor/IDE:**
   * A suitable code editor or Integrated Development Environment (IDE), like Visual Studio Code, PyCharm, or Jupyter Notebook, is recommended for writing and editing Python scripts and HTML templates. These tools provide features such as syntax highlighting, code completion, and debugging support. Selecting an editor that fits the user's workflow can enhance productivity and efficiency in development.

#### 4.2 Software Setup

 **Python:**

* **Description:** Python is a high-level programming language that is widely used for web development, data analysis, and machine learning.
* **Version:** Python 3.6 or later is recommended, with Python 3.8 or higher being preferred for better compatibility with libraries.
* **Installation:** Download Python from the [official website](https://www.python.org/downloads/) and follow the installation instructions for your operating system.

 **Flask:**

* **Description:** Flask is a micro web framework for Python that enables rapid development of web applications. It provides tools and libraries to build web services.
* **Installation:** Install Flask using pip with the following command:

bash

Copy code

pip install Flask

 **NumPy:**

* **Description:** NumPy is a powerful library for numerical computing in Python, providing support for arrays, matrices, and a variety of mathematical functions.
* **Installation:** Install NumPy via pip using the command:

bash

Copy code

pip install numpy

 **Pandas:**

* **Description:** Pandas is a data manipulation and analysis library that provides data structures like DataFrames for handling and processing structured data efficiently.
* **Installation:** Install Pandas with the following pip command:

bash

Copy code

pip install pandas

 **Scikit-learn:**

* **Description:** Scikit-learn is a machine learning library for Python that includes simple and efficient tools for data mining and data analysis. It supports various supervised and unsupervised learning algorithms.
* **Installation:** Install Scikit-learn using pip:

bash

Copy code

pip install scikit-learn

 **Pickle:**

* **Description:** Pickle is a Python module used for serializing and deserializing Python objects, which is particularly useful for saving trained machine learning models.
* **Installation:** Pickle is included in Python's standard library, so no installation is required. Just import it in your code:

python

Copy code

import pickle

 **HTML/CSS Templates:**

* **Description:** The Flask application will use HTML/CSS for the frontend user interface. Templates will need to be created for various routes such as the index page, about page, and others.
* **Setup:** Create a directory named templates in the project folder, and add HTML files (e.g., index.html, about.html) for the application’s web pages.

 **Virtual Environment (Optional but Recommended):**

* **Description:** A virtual environment is a self-contained directory that contains a Python installation for a particular version of Python and various packages. It allows you to manage dependencies for different projects separately.

# CHAPTER 5

# Proposed System & Implementation

#### Proposed system & Implementation

#### 5.1 Block diagram of proposed system

#### 

#### 5.2 Description of block diagram

#### Overall, the block diagram illustrates the flow of information within the application, from user input through data processing and machine learning predictions, to the final output presented to the user. Each component plays a crucial role in ensuring the application functions smoothly, providing timely and accurate health-related information to users.

 **User Interface (UI):**

* **Description:** The entry point for users, this component collects symptoms input via forms and displays results. Users interact with the application through a web browser.
* **Function:** It sends the user inputs (symptoms) to the backend for processing and presents the diagnosis results, recommended diets, and available doctors.

 **Flask Application:**

* **Description:** The core of the application, built using the Flask framework, handles incoming requests, processes data, and returns responses to the UI.
* **Function:** It routes requests from the UI to the appropriate functions (e.g., symptom analysis) and generates responses based on predictions and recommendations.

 **Data Processing:**

* **Description:** This component includes the data handling and processing logic for the application, utilizing libraries like NumPy and Pandas.
* **Function:** It processes the input data (symptoms), prepares it for the model, and handles the datasets for diets and doctors.

 **Machine Learning Model:**

* **Description:** A pre-trained machine learning model (e.g., SVC) that predicts diseases based on user-reported symptoms.
* **Function:** It receives processed symptom data, makes predictions about possible diseases, and returns the predicted disease label to the Flask application.

 **Diets and Doctors Database:**

* **Description:** This component holds the datasets for dietary recommendations and available doctors, structured in a way that can be easily queried.
* **Function:** Based on the predicted disease, it retrieves corresponding dietary recommendations and available doctors for the user.

 **Response to User:**

* **Description:** This is the final output sent back to the user interface after processing the request.
* **Function:** It includes the predicted disease, recommended diets, and details of available doctors, which are then displayed to the user.

#### 5.3 Implementation

#### 

#### 

#### 

#### 5.4 Advantages/ Application/ result table can be included in this subsection.

### Advantages

1. **User-Friendly Interface:**
   * The application provides an intuitive web-based interface, making it accessible for users without technical expertise.
2. **Rapid Diagnosis:**
   * By utilizing a machine learning model, the app can quickly analyze symptoms and provide potential disease predictions, enabling faster decision-making.
3. **Personalized Recommendations:**
   * Users receive tailored diet plans and information about available doctors based on their predicted diseases, promoting a more customized healthcare approach.
4. **Comprehensive Data Handling:**
   * The application effectively manages multiple datasets, including symptoms, diseases, diets, and doctors, providing a holistic health management tool.
5. **Easy Deployment:**
   * Built on Flask, the application is lightweight and can be easily deployed on various platforms, facilitating access to healthcare information.

### Applications

1. **Healthcare Assistance:**
   * The application can be used as a preliminary health assessment tool, guiding users to seek professional medical help based on symptom analysis.
2. **Medical Research:**
   * Researchers can utilize the underlying datasets and prediction models to study disease patterns and dietary recommendations.
3. **Patient Education:**
   * It serves as an educational platform, providing users with information about symptoms, potential diseases, and dietary advice.
4. **Telemedicine Support:**
   * The application can be integrated into telemedicine platforms, enhancing remote patient consultations by providing preliminary diagnoses.

# CHAPTER 6

# Conclusion

1. Conclusion

### Conclusion

In summary, the Flask-based symptom diagnosis application stands out as an innovative tool in the realm of healthcare, effectively bridging the gap between technology and patient care. With its user-friendly interface, the app simplifies the diagnostic process, making it accessible for individuals regardless of their technical background. The rapid diagnosis feature, powered by advanced machine learning algorithms, allows for quick analysis of symptoms, enabling users to make informed health decisions without unnecessary delays.

Moreover, the personalized recommendations for dietary plans and healthcare professionals not only enhance the user experience but also contribute to a more tailored approach to health management. By managing extensive datasets related to symptoms, diseases, and treatments, the application serves as a comprehensive health management tool that encourages users to take charge of their health.

Additionally, the application finds relevance in various fields, including healthcare assistance, medical research, patient education, and telemedicine support, thereby solidifying its position as a valuable asset in the healthcare ecosystem. As we look toward the future, the potential for this application is vast, paving the way for advancements in healthcare technology and improved patient outcomes.

### Future Scope

The future of the Flask-based symptom diagnosis application holds exciting possibilities for further enhancement and expansion. Some key areas of development include:

1. **Integration with Wearable Technology:**  
   Incorporating data from wearable devices could provide real-time health monitoring and more accurate diagnostics, allowing users to track their health metrics continuously.
2. **Expansion of Disease Database:**  
   Adding more diseases, symptoms, and corresponding treatments could enhance the application's comprehensiveness and usability, making it a go-to resource for a wider range of health concerns.
3. **Advanced Machine Learning Techniques:**  
   Implementing more sophisticated machine learning models could improve the accuracy of disease predictions, allowing for a deeper understanding of complex health conditions and their interrelationships.
4. **User Feedback Mechanism:**  
   Establishing a user feedback system could facilitate continuous improvement, helping developers to refine the application based on real-world usage and experiences.
5. **Mobile Application Development:**  
   Developing a mobile version of the application would increase accessibility, enabling users to engage with the tool on the go and promote healthier lifestyle choices anytime, anywhere.
6. **Collaboration with Healthcare Providers:**  
   Partnering with healthcare professionals and organizations could enhance the application's credibility and provide users with access to expert guidance, further enriching their healthcare journey.

# References

**Research paper**

* **Ravi, D., Wong, C., Deligianni, F., Berthelot, M., Andreu-Perez, J., Lo, B., & Yang, G. Z. (2017).**

"Deep Learning for Health Informatics." *IEEE Journal of Biomedical and Health* Reviews the application of deep learning in health informatics, highlighting advancements in disease detection, medical imaging, and patient data analysis. It also discusses challenges such as data privacy and model interpretability.

* **Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017)**

Explores the use of deep neural networks for the classification of skin cancer, showing how AI can match the diagnostic accuracy of dermatologists, providing new opportunities in diagnostic healthcare.

* **Razzak, M. I., Naz, S., & Zaib, A. (2018).** "Deep learning for medical image processing

Presents an overview of deep learning's applications in medical image processing, discussing how convolutional neural networks (CNNs) are used in disease detection and diagnosis. Challenges in terms of data annotation and training efficiency are also discussed

* **Xia, Y., Liu, F., Yang, W., & Smith, K. (2017).**

Focuses on deep learning methods, particularly convolutional neural networks (CNNs), in medical imaging for diagnosis purposes. The paper addresses applications such as cancer detection and the major challenges in data availability and system interpretability.

* **Johnson, A. E. W., Pollard, T. J., Shen, L., Li-wei, H. L., Feng, M., Ghassemi, M., Moody, B., & Lehman, L. W. H. (2016)**

Discusses the MIMIC-III database, which has been widely used in developing machine learning algorithms for healthcareThe paper provides insights into how deep learning models are trained on large-scale patient data for disease prediction and treatment planning.

**URL**

<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6898170>

[https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6456392/](](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6456392/%5D()<https://www.ncbi.nlm.nih.gov>