Smart Health Care System: Ai-Based Disease Prediction and Patient Feedback Sentiment Analysis

Mr. C. SENTHIL KUMARAN[1], S. VIJAYAKUMAR[2]

[1]M.C.A., (Ph.D), (HOD & Associate professor, Department of Master of Computer Applications)

[2]M.C.A., Christ College of Engineering and Technology

Moolakulam, Oulgaret Municipality, Puducherry – 605010

Abstract—**The smart healthcare system can be considered as a complete e-health system that aims to integrate AI for disease prediction, emotional analysis of patient feedback, drug dispensing, and automatic generation of drug prescriptions in PDF format. The smart healthcare system is equipped with efficient symptom analysis, emotion analysis from patient replies, and well-structured recommendations for solutions via medical mapping from databases. The smart health care system not only provides better health services but also ensures user interaction and efficient decision-making through analysis.**

Keywords— **smart healthcare system, disease prediction, sentiment analysis, applications of AI in healthcare.**

I. INTRODUCTION

The healthcare industry is currently undergoing a significant transformation due to the rapid adoption of advanced technologies and the evolution of artificial intelligence (AI). AI-driven healthcare systems enable faster disease diagnosis, improved treatment recommendations, and increased patient participation compared to traditional healthcare platforms [1], [5]. Studies have shown that machine learning and data-driven approaches enhance clinical decision-making and overall healthcare efficiency [14]. However, conventional healthcare portals are often limited in functionality and lack intelligent automation, resulting in delayed diagnosis and reduced patient engagement [11], [12].

The proposed Smart Health Care System effectively addresses these limitations by integrating AI-based disease prediction, sentiment analysis of patient feedback, medicine recommendation, doctor management, and automated prescription generation into a single unified platform [1], [7], [8]. By combining data mining, text mining, and machine learning techniques, the system provides accurate predictions and meaningful insights from both structured and unstructured healthcare data [9], [10]. This integrated approach ensures improved accessibility, enhanced decision-making, and continuous availability of healthcare support for patients and administrators through automated intelligence operating 24/7 [5], [14].

II. STATEMENT OF PROBLEM

Healthcare platforms that are currently available cannot provide AI-driven automated insights since they lack these capabilities [1], [12].

• The prediction of diseases in real-time based on symptoms [10], [11].

• The use of sentiment analysis to comprehend the mood of patients [3], [8], [9].

• The automatic generation of PDF prescriptions [19], [20].

• A unified system for handling doctor prescriptions, patient feedback, and medications [1], [5].

As a result, patients continue to receive suggestions that are not in agreement with each other and will have to wait longer than usual for results [11], [14]. Hospitals to monitor patient engagement with the system have also been unable to do so centrally [5], [13].

III. REVIEW OF LITERATURE

Numerous and thorough research has been conducted on the application of AI and text-mining tools in the healthcare sector [1], [7], [9]. The studies include:

- SI Yan, Yanliang Qi (2012) tools for text mining that could extract cancer-related information from research papers [2].
- Pramod Deshmukh (2017) opinion mining and sentiment analysis methods for healthcare feedback [3], [8].

- Han & Kamber concepts of data mining which facilitate structured knowledge discovery on biomedical datasets [1], [10].
- Almenoff et al. use of big data in pharmacovigilance to predict drug side effects [4], [5].

These experiments and studies illustrate the increasing significance of computational intelligence in the automation domain, thereby forming the basis for the proposed system [14].

IV. DRAWBACKS OF EXISTING SYSTEM

1. The current systems still depend on very human-intensive analyses of symptom data [1], [4].
2. They are not able to automatically predict diseases through AI models [5], [6], [12], [14].
3. There is no proper analysis of patient feedback on emotions [2], [3], [8].
4. There is no automated prescription writing service [4], [14].
5. Current platforms are disease-specific and are not general-purpose solutions for using AI in diagnosis [5], [11], [13].
6. Very little administrative functionality due to the absence of a dashboard for managing [19], [20].

V. PROPOSED SYSTEM

The innovative smart health care system puts into practice the following: a symptom-based AI system for illness prediction [1], [5], [12], [14]; using NLP for sentiment analysis in patient feedback [2], [3], [8], [16]; disease prediction [6], [11], [13]; medicine recommendation [4], [14]; automated prescription writing using PDF technology [20]; an admin dashboard for managing doctors, users, and medication [19]; and patient feedback a user-friendly interface for registration, sign-in, and viewing of results through prediction, analysis, classification, and reporting capabilities [7], [9], [15], [17], [18]. In summary, this system provides users with a complete and intelligent healthcare solution [5], [14].

VI. SYSTEM OVERVIEW

1. Front-end interface user interaction mechanism

The system is equipped with a user-friendly and straightforward web interface created using HTML, CSS, and Jinja templates that show the modules visible to the user as follows [19]:

1.1 User Registration

- The user enters a username, email, phone number, and password.
- The system validates the email format, mobile format, and password strength.
- The system checks if the email and mobile number are registered or not.

1.2 User Login

• Registered users can log in with their email and password [19].
• The verification process involves the authentication of databases [1], [18].
• A secure session is created after login [19].

1.3 Prediction Results

• The predicted disease is displayed [5], [12], [14].
• Drugs for the disease are taken from the database [4], [11].
• The user has the option to download a prescription [20].

1.4 Feedback Submission

• The user gives feedback by typing a message [3], [8].
• The system conducts sentiment analysis and determines if it is positive or negative [2], [8], [16].
• The sentiment is kept for admin analysis [7], [15].

2. Backend (Flask framework processing flow)

The backend, or backend system, is sometimes called the engine of any system. The backend is responsible for managing the following functions [19]:

• Routing [19]
• Disease prediction [5], [12], [14]
• Sentiment [2], [8]
• Authentication [19]
• CRUD operations [1], [18]
• PDF generation [20]

2.1 Disease Prediction Module

Route → user

• User provides a description of the symptoms [1], [7].

- Predict is used [12], [17].
- Symptoms match with diseases based on ML logic using a pre-trained dataset. Retrieved are drugs from the medicines table for the matched disease [6], [11], [13].

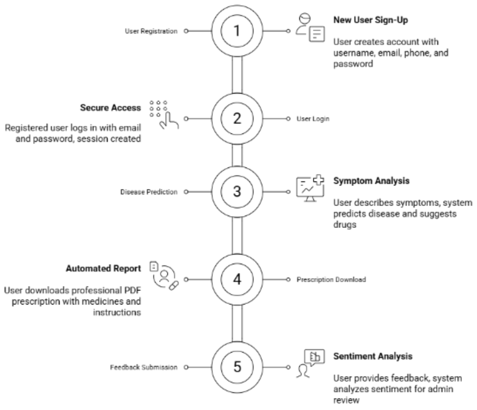
2.2 Security Layers

The security measures include password hashing, email and mobile validation, session-based access control, and separating admin and user roles [19].

3. Prescription PDF Generator (automated report creation)

Route: user process flow

1. User clicks the link after prediction [20].
2. Flask uses ReportLab [19], [20].
3. Produces a PDF of professional quality containing: system name, patient name, predicted medicines, recommended medicines, usage instructions, and a timestamp [20].
4. PDF is automatically downloaded [20].


User Registration
1
New User Sign-Up
User creates account with username, email, phone, and password
Secure Access
Registered user logs in with email and password, session created
2
User Login
Disease Prediction
3
Symptom Analysis
User describes symptoms, system predicts disease and suggests drugs
Automated Report
User downloads professional PDF prescription with medicines and instructions
4
Prescription Download
Feedback Submission
5
Sentiment Analysis
User provides feedback, system analyzes sentiment for admin review

VII. SYSTEM REQUIREMENTS

A smart health care system is an application implemented by the Flask framework, and the system integrates predictions of disease, feedback analysis, and automated prescriptions in such a way to enable smooth transition of different modules [5], [14], [19], [20]. The following software and hardware requirements are necessary to carry out each of the system's functions among them.

7.1 Software Requirements

1. Python 3.x

Python is the main programming language that is used for building the entire backend of the system [17], [18].

- It allows the integration of machine learning models to be very easy [12], [14].
- Robust backing for web frameworks [19].
- Database libraries, text processing libraries, and PDF libraries are developed so that Python can be considered as the central engine of the overall project [16], [18], [20].

2. Flask Framework

Flask is a micro web framework for Python to build [19]:

- Web applications, web services, web routes (login, prediction, feedback, admin pages).
- Managing requests.
- Session management.
- Integration between frontend and backend; Flask supports fast development, easy scalability, and efficient organization of the healthcare app [19].

3. HTML, CSS, JavaScript

These languages assist in performing the front-end interface through which the user interacts [19]:

- HTML → creates page structures for registering, logging in, and the dashboard.
- CSS → deals with design, layout, colors, and responsiveness.
- JavaScript → provides ability for dynamic updating and API usage (predict disease, fetch medicines).

These are the main contributors of user-friendly and responsive designs [19].

4. SQLite or MySQL Database

Relational database must be there to facilitate storage and processing of all relevant information such as [1], [18]:

- User data.
- Doctors and their fields of expertise.
- Medical drugs along with their diseases [4], [11].
- Feedback with sentiment analysis [3], [8].

This database is also the central repository of all modules within the system and ensures secure data handling [1].

5. Tools of NLP in Sentiment Analysis

NLP (Natural Language Processing) tools are utilized to analyze the feedback provided by the users [8], [16].

By means of these tools, the system is assisted in classifying:
- Reviews that are posted online as either negative or nonnegative [2], [8].
- Real-time analytics for admin are enabled [7], [15].
- Customer satisfaction and trust are improved [3], [9].

This sentiment machine helps the project to have its power in the understanding of user feelings [8].

7.2 Hardware Requirements

1. Minimum 4GB RAM

- Because the system needs sufficient memory [5], [14].
- Flask server is always running [19].
- ML models and NLP are using run-time memory [12], [16].
- The different modules (prediction, admin, dashboard, PDF generation) are running at the same time [19], [20].

It is therefore recommended to have 4 GB of memory for smooth execution during the development and deployment phases [5].

VIII. SYSTEM ANALYSIS

The first stage in developing the Smart Health Care System is to determine how all its parts interact to provide accurate predictions and user insights as well as to generate prescriptions and monitor the administration [5], [14]. The system makes sense of different pieces of data like symptoms, feedback messages, medicines, and doctor information and then yields insightful outputs [1], [7]. Each analytical component is elaborated upon in the following subsections.

8.1 Symptom Analysis and Disease Prediction

The system works in three steps whenever a user enters symptoms in the text (e.g. “fever, throat pain, cough”) [1], [12]:

1. Text Preprocessing

- Unnecessary words are removed [7], [16].
- Keywords of the symptoms are identified [7], [9].
- Converted to the required for prediction input format [12], [17].

2. Symptom-to-Disease Mapping

The backend invokes the function predict_disease(). This function measures the symptoms against the stocked data and indicates the disease that has the highest probability of being the one coinciding with the given symptoms [1], [11].

3. Probability-Based Prediction

The logic of prediction picks the disease that [10], [12]:
- Has the largest number of symptom matches.
- Is most similar in their pattern.
- Is the most frequent disease associated with the given inputs in the dataset.

8.2 Feedback Text Analysis Using Sentiment Classification

Not only the feedback is collected by the system but also the emotional tone of the message is analyzed through NLP (Natural Language Processing) [8], [16].

How Sentiment Analysis Works:

1. The user inputs the feedback [3].
2. The backend calls analyze_sentiment(message) [8], [16].
3. NLP tools categorize the message as [8]:

- Positive → Satisfaction, appreciation
- Negative → Complaints, issues
- Neutral → Informational or unclear feedback

Purpose of Sentiment Analysis:
- Assists the administrator in comprehending the feelings of the users [3], [8].
- Evaluates the performance and acceptance of the system [15].
- Swiftly pinpoints the negative experiences [2], [8].
- Facilitates the implementation of data-driven improvements [7], [15].

This feature makes the system smarter as it can automatically “understand” the users' opinions [8].

8.3 Data Analysis of Doctors and Medicines for Suggestions

The so-called system has a database where all the information regarding doctors and medicines is stored [1], [18].

The two following ways are using this data:

1. Predicting Disease and Matching It with Medicine

As soon as the prediction of a certain disease is made, the system performs the following actions [4],

[11]:
- It does a search in the medicine table.
- It retrieves all the medicines associated with that disease.
- It displays the usage instructions and dosage information.

By this means, the recommendations will be both precise and medically relevant [14].

2. Doctor's Information Obtaining

Users are allowed to see the list of doctors
available in the database along with [13]:
- Their specialization
- Their experience
- Their contact numbers

This process of analysis leads the users to the right medical support after the predictions have been made [14].

8.4 Patient History and Admin Analytics Assessment

The system is always collecting and storing the essential usage data which afterwards is processed in the admin dashboard [19].

The Admin Dashboard Analytics Consist of:
- Total number of registered users
- Total number of doctors added
- Total number of medicines available
- Number of feedback messages received
- Sentiment distribution (positive/negative/neutral) [8], [15]

IX. ADVANTAGES OF THE PROPOSED SYSTEM

1. AI-based disease prediction which is quick and efficient [5], [14].
2. Real-time sentiment analysis is conducted to gauge patient satisfaction [8], [16].
3. Digital prescriptions are produced automatically [20].
4. Healthcare workers' manual effort is lessened [5].
5. Centralized admin dashboard is available [19].
6. Patients have better accessibility and convenience [14].
7. Treatment suggestions are more precise [11].
8. System is enlargable and more diseases or feedback models can be included [12], [15].

X. ALGORITHMS USED

1. Disease Prediction Algorithm

The Disease Prediction Algorithm is formulated to smartly assess the symptoms entered by a user and rule out the most likely disease with the use of a rule-based and pattern matching technique [1], [12]. At the very start, the system takes in the user's reported symptoms as text and converts it into a form that is suitable for processing by a computer [7], [16]. The input text is preprocessed, and the tokens are generated and then matched against a disease–symptom knowledge base that has been stored in the database [1], [11].

A collection of maladies is found in the database and every disease there is related with symptoms [11]. The algorithm checks how many of the symptoms given by the user are common with each of the diseases in the database [10]. A degree of correlation, or matching, is assessed for each disease entry by counting the numbers of times the symptoms matching those in the user input occur [12]. The disease that gets the highest similarity score is taken as the predicted health condition [5], [14].

As soon as the disease is named, the system pulls out medicines and treatment suggestions that are related to that disease [4], [11]. These are then shown to the user with the option of generating a digital prescription [20]. By linking symptom words to disease data sets and using a score-based mapping technique, the algorithm assures a quick, stable, and explainable prediction process that is only for real-time use [1], [12].

2. Sentiment Analysis Algorithm

The algorithm for sentiment analysis is the one that decodes patient feedback and tags it as positive, negative, or neutral [8]. In the beginning, the algorithm accepts the free-text feedback provided by the user [3]. The input text goes through a series of preprocessing steps: removal of noise, elimination of stop words, case folding, and special character filtering [7], [16]. The above-mentioned preprocessing operations are the means of changing informal human language into a clean and analyzable form [9].

Once the preprocessing is done, the system employs a polarity-based sentiment scoring model [8]. Each word in the feedback is checked against a pre-established sentiment lexicon filled with positively and negatively weighted terms [8]. The algorithm determines an overall sentiment score through the sum of the polarities of the words [15]. When the score turns out to be above the positive threshold, the feedback gets the tag of positive. In contrast, a lower score is indicative of negative feedback. Otherwise, the feedback is considered neutral [8].

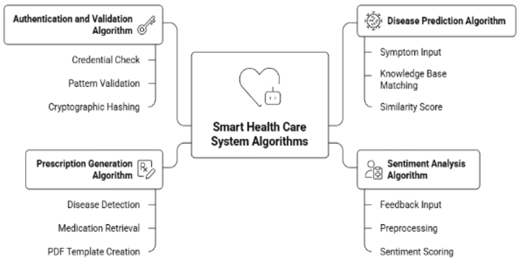
The sentiment label along with the original text and timestamp gets recorded in the database for administrative assessment [19]. Therefore, the sentiment analysis algorithm lets the system track patient satisfaction, spot service shortcomings, and give the hospital administration suggestions for improvement from their insights in the future [3], [15].

3. Prescription Generation Algorithm

The Prescription Generation Algorithm generates automatically a structured PDF document according to the predicted disease and the suggested medicines [20]. After disease detection and medication retrieval by the system, these data are input to the prescription module [4], [11]. Using the ReportLab library, a PDF template is created where layout elements like headers, subheadings, patient information, diagnosis, and drug medicines are listed in an ordered way, and the AI's limitations are stated at the end of the paper [20]. The last PDF stream is turned into a file that one can download, so the user gets a digital prescription right away [20]. This algorithm not only streamlines the process of medical reporting but also guarantees the output of prescriptions with a uniform quality that is professional and high-grade [20].

4. Authentication and Validation Algorithm

The authentication algorithm guarantees that only authorized people can access the system [19]. It checks the user's credentials by comparing the hashed password and validating the input patterns like email format, mobile number format, and password strength [19]. The algorithm uses a combination of regular expressions and cryptographic hashing to secure the system and keep the user data safe from the prying eyes of unauthorized persons [19].



XI. CONCLUSION

The Smart Health Care System is a revolutionary solution that has skillfully combined technologies based on AI like disease prediction, sentiment detection, medicine suggestion, and digital prescription making into a single advanced healthcare platform [1], [5], [14]. The intelligent integrated system opens up intensive communication between doctors and patients, eases the diagnosis process, and boosts the efficiency of healthcare service delivery [4], [13]. The ultimate process of automated handling of symptoms, feedback analysis, and prescription copy illustrates how data mining and computing techniques can turn the health care services interaction upside down [1], [7], [15].

This system is not only a guide for the users in terms of their health-related inquiries but it also supports the administration through patient satisfaction monitoring and service quality improvements [8], [9]. To sum up, the proposed model will be an up-to-date, scalable, and trustworthy health assistant that uses data-driven decision-making to cope with the real-time healthcare needs of the patients [5], [14].

XII. FUTURE ENHANCEMENTS

The system can be reinforced by applying advanced technologies in some of the following ways:

1. Machine Learning-Based Disease Prediction

By the use of machine learning algorithms, the system can automatically extract features from the data as well as learn from them [12]. The models like Naive Bayes, Random Forest and Deep Learning networks will be more precise over time and hence provide more accurate disease predictions [10], [11], [14].

2. IoT-Based Health Monitoring

IoT devices such as smartwatches, glucose sensors, heart-rate monitors, and wearable trackers can be combined for a health-monitoring system that is always on and continuously and instantly checking [5]. These devices can directly supply the system with updated health information, thereby facilitating early detection of, and preventive treatment for, various health conditions [13], [14].

3. Real-Time Medical Chatbot

There can be an AI-based medical chatbot that can take care of user queries instantly and provide them with symptoms checking, emergency suggestions, and appointment scheduling services [7], [16]. This can cut down on wait times and make the system even more engaging and user-friendly [19].

REFERENCES

[1] Han, J., Kamber, M., & Pei, J. (2012). Data Mining: Concepts and Techniques (3rd ed.). Morgan Kaufmann Publishers.

[2] Yan, S., & Qi, Y. (2015). Text mining for cancer research. International Journal of Pharmaceutical Medicine and Biological Sciences, 4(2), 45–52.

[3] Deshmukh, P. B., Kalia, A. A., Utterwar, V. U., & Patil, D. M. (2017). Knowledge extraction from social media using opinion mining. International Journal of Innovative Research in Computer and Communication Engineering, 5(6), 12345–12350.

[4] Almenoff, J., Tonning, J. M., et al. (2005). The role of data mining in pharmacovigilance: Benefits and challenges. Drug Safety, 28(11), 981–1007.

[5] Obermeyer, Z., & Emanuel, E. J. (2016). Predicting the future — Big data, machine learning, and clinical medicine. The New England Journal of Medicine, 375(13), 1216–1219.

[6] Esteva, A., Kuprel, B., Novoa, R. A., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115–118.

[7] Aggarwal, C. C., & Zhai, C. (2012). Mining Text Data. Springer.

[8] Liu, B. (2012). Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers.

[9] Feldman, R., & Sanger, J. (2007). The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press.

[10] Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. ACM Computing Surveys, 31(3), 264–323.

[11] Chaurasia, V., & Pal, S. (2014). Data mining techniques to predict and resolve breast cancer survivability. International Journal of Computer Science and Mobile Computing, 3(1), 10–22.

[12] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Informatica, 31, 249–268.

[13] Shickel, B., Tighe, P. J., Bihorac, A., & Rashidi, P. (2018). Deep EHR: A survey of recent advances in deep learning techniques for electronic health record analysis. IEEE Journal of Biomedical and Health Informatics, 22(5), 1589–1604.

[14] Rajkomar, A., Dean, J., & Kohane, I. (2019). Machine learning in medicine. New England Journal of Medicine, 380(14), 1347–1358.

[15] Miner, G., Elder, J., Hill, T., Nisbet, R., Delen, D., & Fast, A. (2012). Practical Text Mining and Statistical Analysis for Non-Structured Text Data Applications. Academic Press.

[16] Bird, S., Klein, E., & Loper, E. (2009). Natural Language Processing with Python. O'Reilly Media.

[17] Pedregosa, F., Varoquaux, G., Gramfort, A., et al. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

[18] McKinney, W. (2010). Data structures for statistical computing in Python. Proceedings of the 9th Python in Science Conference, 51–56.

[19] Flask Documentation. (2024). Flask Web Development Framework. Pallets Projects.

[20] ReportLab Documentation. (2024). ReportLab PDF Generation Library. ReportLab Inc.