



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 5, May 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

 9940 572 462

 6381 907 438

 ijirset@gmail.com

 www.ijirset.com

Early Disease Prediction based on Symptoms Using Machine Learning

Mr. K. Lakshmi Kanth Reddy¹, P. Harshini², L. Deepak³, R.Sri Nihal Reddy⁴

Assistant Professor, Department of Artificial Intelligence, Anurag University, Hyderabad, India¹

Student, Department of Artificial Intelligence, Anurag University, Hyderabad, India²³⁴

ABSTRACT: Healthcare prediction has a critical role in saving lives. Intelligent systems are used to assess complex data relationships and convert them into valuable knowledge. They are increasing rapidly in the healthcare industry. As a result, systems based on deep learning and machine learning are becoming increasingly significant in the development of processes for disease diagnosis and prediction. These systems, which might be based on photographs or clinical data, provide extensive clinical support by simulating human vision and even diagnosing disorders that human intelligence cannot detect. Predictive analytics are vital in the healthcare industry. It can have a significant impact on the accuracy of sickness prediction, which, in the case of an accurate and timely prognosis, may save patient lives; conversely, an inaccurate prediction could risk patient lives. As an outcome, diseases need to be accurately analyzed and projected. Thus, reliable and effective methods of predictive analysis in healthcare are critical. Furthermore, lifestyle disorders affect a large portion of the population today—not just in India, but globally as well. The daily routines and general way of life that an individual leads are the main causes of these disorders. Heart disease and hypertension are two examples of these conditions, which are very common. We frequently come across people battling similar illnesses in our own lives, and in certain cases, these conditions may even result in deaths. Unfortunately, receiving treatment is frequently postponed or nonexistent when a person is unaware of their health situation. Inspired by this worry, we have started working on a portal that will evaluate data supplied by users and offer forecasts for possible lifestyle-related illnesses. This platform provides preventive methods to fight against common lifestyle disorders in addition to forecasting prospective health hazards. The system additionally offers management options in the event of moderate symptoms. The main goal of this system is to raise personal health awareness, which will allow for prompt medical attention when needed and maybe save many lives.

KEYWORD: Prediction, Diseases, Machine Learning.

I. INTRODUCTION

People live hectic lives in today's fast-paced environment, frequently leaving little time for crucial responsibilities like making regular appointments for medical examinations. But there is a solution that works perfectly with our contemporary lifestyles: a smartphone app for health monitoring. Considering how commonplace technology is these days, this method is really convenient. Imagine being able to stay informed about your health status whether you're in line at the grocery store or on your daily commute. This implies that you won't have to schedule a separate time for a medical appointment or endure long lines at a clinic. Instead, with the aid of a straightforward online application, you may make effective use of such little periods of time.

II. OBJECTIVES

User-friendly web application: The creation of an intuitive web application that utilizes machine learning methods to forecast lifestyle diseases according to user symptoms is a noteworthy progression in customized healthcare. Through the seamless integration of predictive algorithms into an intuitive interface, people may enter their symptoms and obtain customized risk assessments for diseases associated with their lifestyles. The program facilitates proactive health management by enabling users to have a better understanding of their health condition and potential dangers through its user-friendly design and clear results. By utilizing machine learning models that have been trained on vast datasets, the program is able to offer precise forecasts and practical insights that are customized to the individual profiles of each user. The platform might also provide preventive advice, instructional materials, and ideas for changing one's lifestyle to lower the chance of illness. Through the integration of machine learning capabilities with user-centered design principles, this web application is a useful resource for preventing diseases in their early stages and enhancing general health.

Health Analysis: Proactive healthcare management is encouraged when patients are empowered to monitor their own health through cutting-edge technologies.

Patients can obtain important daily insights into their well-being by giving them the skills and resources to track a variety of health indicators, including their levels of physical activity, sleep patterns, dietary habits, and vital signs. These monitoring systems frequently take the form of wearable technology or easily navigable smartphone apps that have sensors built in to gather data in real-time. Through the utilization of these technologies, patients can acquire a more profound comprehension of their health condition, see patterns or trends, and make well-informed judgments to enhance their lifestyle choices.

Doctors appointments: The proposed application will function as an all-inclusive healthcare platform, enabling patients to conveniently and easily take charge of their medical care. Users will be able to peruse a wide variety of healthcare providers based on specializations, areas, and availability by using an easy-to-use interface that provides them with access to a directory of accessible doctors. The app will make it easy for patients to book appointments when they've found a qualified physician, saving them the trouble of making traditional phone calls or waiting around.

Using Symptoms to Predict Early Lifestyle Disease with Machine Learning, patients can choose convenient appointment times that fit their schedules by directly integrating scheduling features into the platform, which encourages accessibility and flexibility in the delivery of healthcare. In order to improve user convenience and engagement, the application may also include features like real-time availability updates, appointment reminders, and the opportunity to postpone or cancel appointments as necessary. All things considered, this program makes it easier to locate and schedule appointments with healthcare professionals, which eventually improves access to high-quality care and encourages proactive healthcare management.

Order Medicines: Patients can browse a comprehensive inventory of pharmaceuticals, including over-the-counter and prescription treatments, using an easy-to-use interface. The program facilitates the ordering process for patients by utilizing integrations with nearby pharmacies or online pharmaceutical delivery providers. This allows patients to conveniently place orders from their computers or cell phones. With just a few clicks, patients can quickly search for specific prescriptions, examine comprehensive product information, and add them to their virtual shopping cart. Patients can choose their preferred delivery method, dose, and amount through a simplified ordering process. In order to help patients adhere to their treatment programs, the app may also include functions like prescription management, refill alerts, and medication reminders. The application gives consumers the ability to take charge of their healthcare journey by providing a one-stop shop for scheduling doctor's appointments and managing medications. This encourages convenience, accessibility, and treatment adherence.

III. ARCHITECTURE

Data Collection and Preprocessing:

Data collection: Symptoms and multiple diseases are collected from various sources like Kaggle and Google.

Data Preprocessing: Preprocessing procedures such as cleaning, normalization, and feature engineering are applied to the gathered data. To maintain consistency throughout the dataset, features are scaled, and missing values are handled.

Feature Selection: The features in the dataset are the symptoms and the diseases. All these features are taken as input; each disease is classified based on the type of feature status (1 or 0), and the disease name is used as a label.

Chatbot: For the chatbot application, a text file data set is taken as input, which has features as questions and labels as answers.

Data Splitting: The training dataset is used to train the algorithm, while the testing subset of the dataset is kept separate. This training dataset is made up of matching labels (train_y) and features (train_x), where labels indicate the related disease outcomes and features indicate the symptoms or traits connected to each case. The fit function of the algorithm receives input from the pre-processed dataset, which includes pertinent health information. The decision tree model is able to make predictions based on the input features because it is able to identify patterns and correlations within the data through this approach. In order to reduce prediction mistakes and increase accuracy, the algorithm can adjust and optimize its internal parameters during the training process, which makes it crucial.

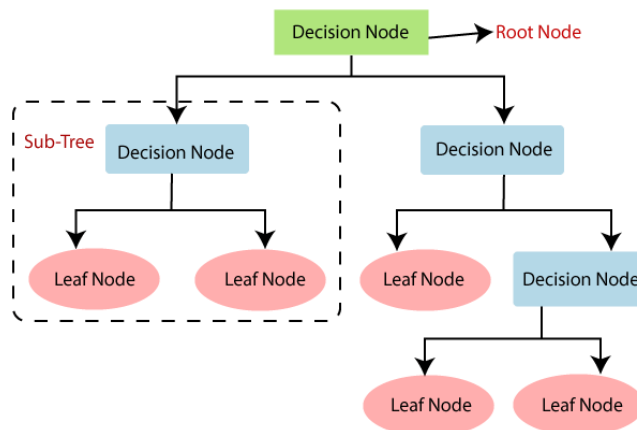
Apply the ML decision tree algorithm: pre-processed datasets with essential health data, including disease outcomes, medical history, and symptoms, are created. For training and prediction, this dataset is fed into the machine learning model.

Model accuracy: The model's performance on unseen data is evaluated following the training phase, which involves training the machine learning (ML) model on the training dataset using the decision tree technique. The test dataset, which was withheld during the training phase, is used for this purpose. By contrasting the expected disease labels with the actual labels in the test dataset, the model's accuracy is determined. The accuracy statistic specifically calculates the percentage of all test dataset instances with correctly predicted labels.

Decision Tree Algorithm

Decision Trees are supervised learning techniques that are frequently used to solve classification and regression problems, however they are best suited for classification problems. A decision tree consists of two nodes: the decision node and the leaf node. While leaf nodes indicate the outcome of decisions and have no more branches, decision nodes are used to make any type of decision and have many branches. The features of the submitted dataset are utilized to make judgments or conduct tests. It is a graphical tool that displays all of the possibilities for solving a problem or making a choice based on specific criteria. It is called a decision tree because, like a tree, it starts at the root node and extends through succeeding branches to produce a tree-like structure. A decision tree simply asks a question and then divides the tree into subtrees based on the answer (Yes/No).

The below diagram views the basic structure of a decision tree:



IV. SOFTWARE REQUIREMENTS

Software requirements refer to the comprehensive specifications of the features, functionalities, capabilities, and limitations that the program must follow in order to satisfy its users' and stakeholders' needs and expectations. The software development lifecycle is guided by these criteria, which provide the basis for the software solution's design, development, testing, and deployment. Functional requirements outline the precise functions and operations that the software must carry out, including data processing, recommendation engines, user registration, and prediction algorithms. The characteristics and features that the program must have, such as performance, scalability, reliability, security, usability, and maintainability, are covered by non-functional requirements. Additionally, software requirements include any external interfaces or dependencies that the system must work with, such as APIs for accessing nutritional data or machine learning libraries for predictive modeling. By painstakingly defining and adhering to software requirements, development teams can ensure that the finished product meets user expectations, meets industry standards, and provides a smooth and effective user experience.

Software requirements are:

- Python: A programming language for system development
- Flask: web application framework for front-end development that uses python
- MySQL: A database stores data in separate tables
- DataSet: Disease Dataset, Google.

V. HARDWARE REQUIREMENTS

Hardware requirements specify the physical components and language structure required to support the software solution efficiently. These criteria not only establish hardware configurations and specifications, but also ensure the system's optimal performance, scalability, and dependability. Typically, hardware requirements include a variety of components like servers, workstations, networking equipment, storage devices, and other peripherals needed for deployment and operation. Servers, for example, may be required to fulfill specified CPU, RAM, and storage capacity specifications in order to meet the computational needs and data storage requirements of the software. Networking devices, such as routers and switches, must be capable of efficiently managing network traffic in order to permit communication between system components and external interfaces.

Moreover, specialized peripherals, such as barcode scanners, printers, or sensors, may be required for seamless integration with the program, allowing for functions such as data entry, output, or monitoring.

Hardware requirements are:

- CPU/ GPU: Computational resources for model training Minimum 4GB
- Memory: RAM for processing large datasets and model training 16GB And Above
- Storage: Solid-state drive (SSD) for storing historical data, trained model weights, and application file

VI. TECHNOLOGY USED

Python: Python was chosen as the major programming language because of its simplicity, readability, and large ecosystem of libraries and frameworks. It is widely used in the data science and machine learning communities due to its ease of use and flexibility. Python's rich library ecosystem enables developers to efficiently implement various aspects of the stock price prediction system, from data preprocessing to model training and deployment.

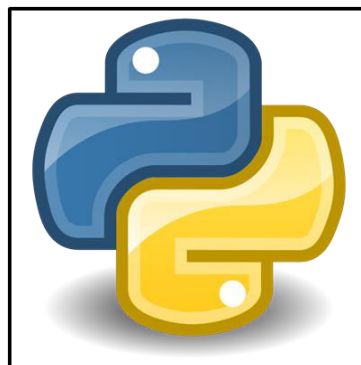


Fig: Python

Pandas: Pandas is a versatile data manipulation library in Python, widely used for handling structured data and time series data structures. It supports strong data structures such as DataFrame and Series, as well as functions for data cleaning, transformation, indexing, and aggregation. In the stock price prediction system, Pandas facilitates the preprocessing of raw financial data, handling missing values, scaling features, and organizing data into suitable formats.



Fig: pandas

VSCode: Visual Studio Code (VSCode) is a popular code editor created by Microsoft that is recognized for being lightweight, versatile, and feature-rich for software development. It offers built-in support for Python development, along with a wide range of extensions for debugging, version control, and project management. With its intuitive user interface and powerful features, VS-Code serves as the integrated development environment (IDE) for developing the stock price prediction system, providing developers with a productive and customizable environment for coding, testing, and debugging.

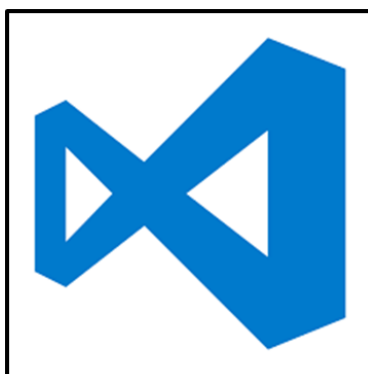


Fig: VSCode

Flask: Flask is a web framework, and Python module, that allows you to easily create web applications. It has a tiny and easy-to-extend core: it's a microframework without an ORM (Object Relational Manager) or similar functionality. It is a Python-based microweb framework. It is characterized as a microframework because it does not require any specific tools or libraries. It lacks a database abstraction layer, form validation, and other components that rely on pre-existing third-party libraries.



Fig: Flask

NumPy: NumPy can be used to perform many different mathematical operations on arrays. It enhances Python with sophisticated data structures that provide efficient computations with arrays and matrices, as well as a comprehensive set of high-level mathematical functions that work with these arrays and matrices.



Fig: Numpy

VII. PROCEDURES

Project Setup:

To set up a project for lifestyle disease detection with a Decision Tree algorithm, the first step is project planning. This includes establishing the project scope, objectives, and target audience, as well as outlining the necessary features and functionalities. With a clear plan in place, the environment is set up by selecting appropriate programming languages and frameworks, along with establishing a development environment either locally or on cloud platforms. Following this, relevant datasets containing lifestyle factors and disease outcomes are collected and pre-processed to handle missing values and ensure data consistency. The project progresses with the implementation of the Decision Tree model using machine learning libraries like sci-kit-learn, where the model is trained and adapted based on the training dataset. Evaluation measures such as accuracy and precision are calculated to determine the model's performance on the testing dataset. Subsequently, a user-friendly interface is developed, allowing users to input their lifestyle factors and obtain disease predictions in real time. The application and machine learning model are then deployed on chosen platforms, and thorough testing ensures functionality and reliability. Comprehensive documentation is created to guide users and developers, and ongoing maintenance and updates are planned to address evolving requirements and ensure the system's longevity and effectiveness.

Data Acquisition:

This involves gathering datasets containing information about disease and symptoms. Data acquisition may involve accessing public health databases, collaborating with healthcare institutions for anonymized patient records, or collecting survey data from individuals. The collected data should be comprehensive, representative, and sufficiently large to capture the variability and complexity of lifestyle-related diseases. Additionally, ensuring data integrity, privacy compliance, and ethical considerations are paramount throughout the acquisition process.

Model Development:

This involves utilizing the collected data on diseases and corresponding symptoms to build a Decision Tree model. The model is designed to analyze patterns and relationships between symptoms and disease, allowing it to predict the likelihood of diseases based on user inputs.

During development, attention is given to fine-tuning the model's parameters and optimizing its performance to ensure accurate predictions. Once trained, the model becomes capable of providing personalized risk assessments for various lifestyle-related diseases, empowering individuals to make informed decisions about their health and well-being. During model training, the Decision Tree algorithm learns to discern patterns and relationships between lifestyle factors and disease outcomes.

Evaluation: Evaluating the Decision Tree model for the early lifestyle disease detection system entails several essential phases to determine its effectiveness and reliability in forecasting disease risks based on user-provided symptoms. The dataset is initially divided into training and testing subsets, allowing the model training on one and evaluation on the other.

Front-end development using Flask: Developing a front-end application with Flask serves as a crucial interface through which users can interact with the system conveniently. By integrating Flask's backend functionality with a frontend interface, individuals can input their symptoms and receive personalized predictions about their disease effectively. Through intuitive design and seamless navigation, the frontend application enhances user engagement and accessibility, enabling individuals to make informed decisions about their well-being and potentially preventing the onset of lifestyle-related diseases. Moreover, the user can consult a doctor and order medicines.

Integration and Testing: Integrating all system modules, including data pretreatment, model building, assessment, and frontend interface, guarantees that components communicate and share data seamlessly. This process entails linking individual modules, managing dependencies, and confirming the system's overall functionality.

Thorough testing, which includes unit tests, integration tests, and end-to-end tests, detects and resolves any problems, failures, or inconsistencies inside the system. Testing ensures that each module executes as planned and that the system functions consistently under a variety of settings and scenarios.

VIII. TESTING

Test case: 1

Test case: input valid parameter values in the specified order. 2. mount the valid dataset

Test case scenario: Collecting input data.

Test case: 2

Test case: Descriptive diagram of the relationship between various types of information in the dataset is to be created

Test case scenario: Creating a Model ML Algorithm (Decision Tree)

Test case: 3

Test case: By analyzing the dataset, the data model is to be created i.e Accuracy score is provided

Test case scenario: Training

Test case: 4

Test case: Testing the model whether the predicted result is correct or not.

Test case scenario: Testing

Test case: 5

Test case: By considering the user inputs, the disease is predicted

Test case scenario: Prediction

Test case: 6

Test case: checking if the model whether predicted result is correct or not.

Test case scenario: Display the result in the web application

IX. RESULTS

```
from sklearn import tree

clf3 = tree.DecisionTreeClassifier() # empty model of the decision tree
clf3 = clf3.fit(X,y)
print("Accuracy on test: ", clf3.score(X_test,y_test)*100)
```

Accuracy on test: 95.1219512195122

```
from sklearn.metrics import accuracy_score
y_pred=clf3.predict(X_test)
print(accuracy_score(y_test, y_pred)*100)

dacc=accuracy_score(y_test, y_pred)*100
```

95.1219512195122

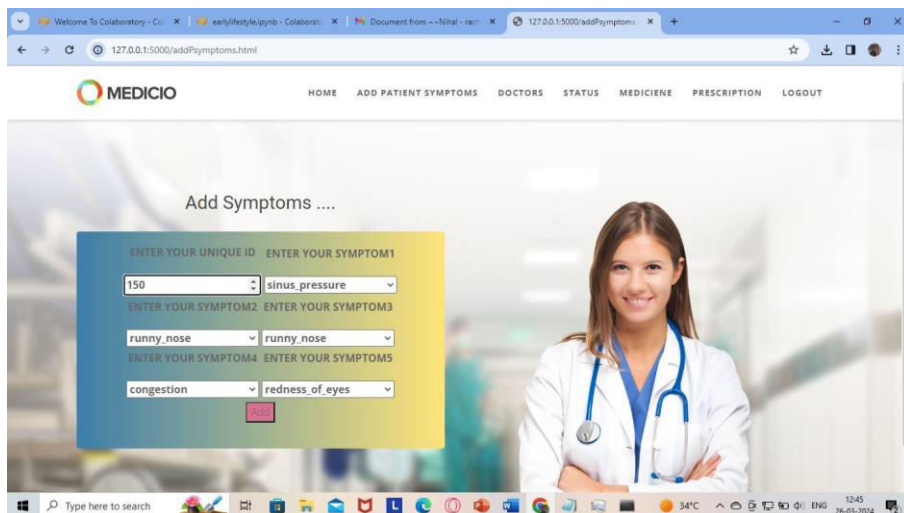


Fig : Patient Symptoms Log

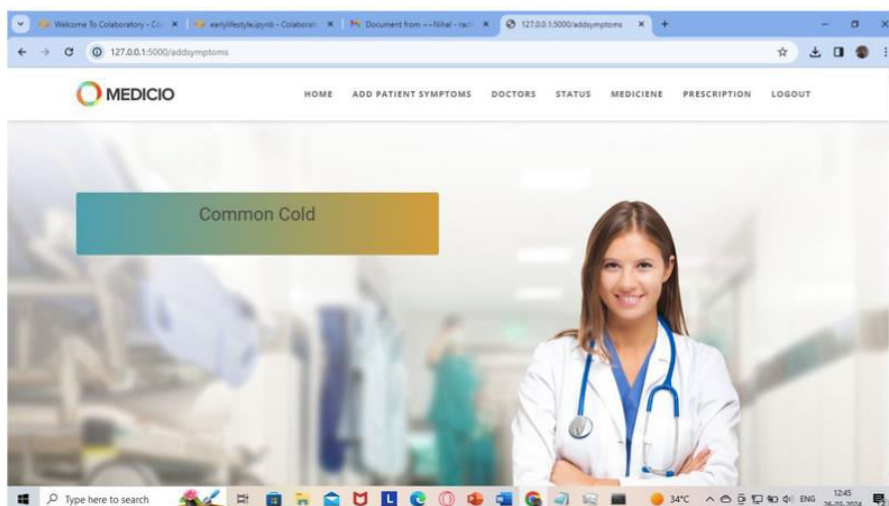


Fig : Result Page

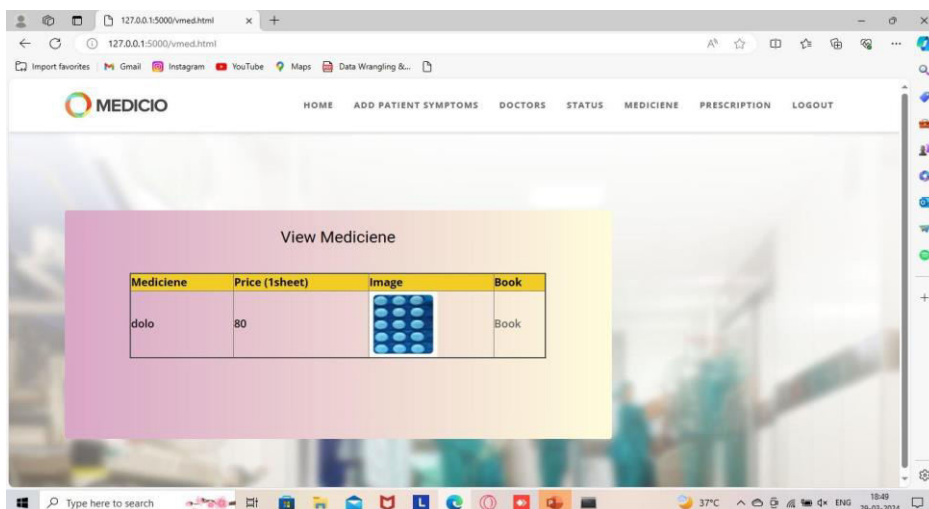


Fig: Medicine ordering Page

X. CONCLUSION

In conclusion, the development of an early lifestyle disease detection system using machine learning, particularly with the Decision Tree algorithm, represents a significant advancement in proactive healthcare management. After a lot of trial and error, we have successfully designed and implemented a robust system capable of leveraging user-provided symptoms to accurately predict disease. By integrating Flask for the frontend interface, users can conveniently input their symptoms and promptly receive personalized disease predictions. Meanwhile, the backend, powered by the Decision Tree algorithm, has undergone rigorous testing and validation to ensure its accuracy and reliability in disease detection based on symptoms. Moreover, we have gone beyond mere prediction by incorporating additional functionalities into our system. Notably, we have integrated doctor appointment scheduling and medicine ordering services, enriching the user experience and facilitating comprehensive healthcare management. With these enhancements, users not only receive disease risk predictions but also gain access to medical consultations and necessary medications, further empowering them to take proper measures to safeguard their health.

REFERENCES

- [1]. PEIYING ZHANG, XINGZHE HUANG, AND MAOZHEN LI. Disease Prediction and LifeStyle Intervention System Based on Symptom Similarity Analysis IEEE ACCESS 2019.
- [2]. M. Chen, Y. Hao, K. Hwang, L. Wang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities", IEEE Access, vol. 5, no. 1, pp. 8869–8879, 2017.
- [3]. IM. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, and C. Youn, "Wearable 2.0: Enable human-cloud integration in next generation healthcare system," IEEE Commun. , vol. 55, no. 1, pp. 54–61, Jan. 2017.
- [4]. Y. Zhang, M. Qiu, C.-W. Tsai, M. M. Hassan, and A. Alamri, "HealthCPS: Healthcare cyber-physical system assisted by cloud and big data," IEEE Syst. J., vol. 11, no. 1, pp. 88-95, Mar. 2017.
- [5]. R.Udayakumar, R.Sugumar, T.Devi, R.Elankavi,(2024). A Hybrid Modified Artificial Bee Colony (ABC)-Based Artificial Neural Network Model for Power Management Controller and Hybrid Energy System for Energy Source Integration, MDPI Engineering Proceedings, Volume 59, Issue 35, pp.1-12, December 2023, <https://doi.org/10.3390/engproc2023059035>.
- [6]. Ajinkya Kunjir, Harshal Sawant, NuzhatE. Shaikh, "Data Mining and Visualization for prediction of Multiple Diseases in Healthcare," in IEEE big data analytics and computational intelligence, Oct 2017 pp.2325
- [7]. Amin, S.U.; Agarwal, K.; Beg, R., "Genetic neural network based data mining in prediction of heart disease using risk factors", IEEE Conference on Information & Communication Technologies (ICT), vol., no,pp, 1227-31,11-12 April 2013.
- [8]. K. Oh, D. Lee, B. Ko and H. Choi, "A Chatbot for Psychiatric Counseling in Mental Healthcare Service Based on Emotional Dialogue Analysis and Sentence Generation", 2017 18th IEEE International Conference on Mobile Data Management (MDM), pp. 371-375, 2017.
- [9]. Sarioguz, O., & Miser, E. (2024). Artificial intelligence and participatory leadership: The role of technological transformation in business management and its impact on employee participation. International Research Journal of Modernization in Engineering, Technology and Science, 6(2), Article 1618. <https://www.doi.org/10.56726/IRJMETS49539>
- [10]. S.J. Du Preez, Manoj Lall and S. Sinha, An intelligent web-based voice chat bot, vol. 38, 2009.
- [11].S. Sri Gowthem and K.P. Kaliyamurthie, "Smart Appointment Reservation System", International Journal of Innovative Research in Science Engineering and Technology, vol. 4, no. 6, 2015.
- [12].Y. Jiang, H. Zhan and Z. Huang, "Design and Implementation of A General Web-based Course Teaching Management System", International Journal of Education and Management Engineering, vol. 2, no. 11, pp. 1-7, 2012.
- [13]. N. Liu, S. Ziya and V. Kulkarni, "Dynamic Scheduling of Outpatient Appointments Under Patient No-Shows and Cancellations", Manufacturing & Service Operations Management, vol. 12, no. 2, pp. 347-364, 2010.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details