iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** III **Month of publication:** March 2026

DOI: https://doi.org/10.22214/ijraset.2026.78098

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

An Intelligent Appointment Scheduling System for Healthcare using Android and Machine Learning

Dr. Sk. Syed Basha[1], Dr. Deevi Hari Krishna[2], Mr. Talluri Lokesh Sai[3], Mr. Pinapati Manoj Babu[4], Mr. Rajana Maheshwar Rao[5], Mr. Vepuri Nanda Rakesh[6]
Department of CSE (AI&ML), KKR &KSR Institute Of Technology And Sciences, Vinjanampadu, Guntur, Andhra Pradesh, India

Abstract: The project in many cases " An Intelligent Appointment Scheduling System for Healthcare using Android and Machine Learning " nowadays an integrated Mobile healthcare solution that combine intelligent disease anticipation with automated assignment scheduling. This finding appears to suggests, the organization is plan to assistance patient in place potential wellness conditions at an early level while simultaneously enable efficient consultation direction. It is Worth note that, the application arguably allows user to enter symptoms through an humanoid port. In light of this, a machine learning classification model process the symptom and predicts the most probable disease. This finding suggests, based on the prognosticate disease and associated symptom patterns, the system provides medicine recommendations include dosage and duration item. Additionally, it suggests relevant lab tests with priority level and estimated requirement to reinforcement exact diagnosis. The system potentially further recommends specialist doctors base on the predicted disease and the user ' s geographic locating. Building upon this, patient can directly book appointment by take usable time slots, thereby trim manual intervention and waiting time. Edifice upon this, the integration of disease prediction, medicine proffer, lab test recommendation,. Location-based doctor filter into a single humanoid platform raise healthcare accessibility, better decision-making, and supports digital healthcare translation. Building upon this, this finding potentially suggest, the organization is designed with scalability, security, Modular architecture to ensure reliableness and hereafter enlargement.

Keywords: Machine Learning, Ensemble Classification, Random Forest, Natural Language Processing (NLP), TF-IDF Vectorization, Clinical Risk Assessment, Healthcare Recommender Systems, Medical Data Mining, Predictive Analytics, Intelligent Appointment Scheduling.

I. INTRODUCTION

Healthcare system world-wide face increase pressure due to maturation patient population, restrain medical professional. The evidence indicates, delay clinical access. In many cases, individuals live symptom but shelve interview due to dubiety, time restraint, or lack of contiguous access to specialist. In light of this, this happen suggests, other designation of potential medical conditions.

These results, timely counselling can importantly amend wellness outcome and trim disease progression. This observation, recent advancement in machine learning and healthcare informatics have enabled the growth of levelheaded clinical decision support dodging capable of analyzing patient-reported symptoms and call potential diseases. It is Worth noting that, such systems assist users by providing preliminary guidance earlier conventional medical consultation. These results, however, many existing solution function as standalone tools focus entirely on disease foretelling, without integrating consultation agenda or integrated medical recommendations. The evidence indicates, it is worth noting that, it is worth mention.,

To address this gap, this work project mediassist ai, an desegregate healthcare assistance programme. Combine symptom-based disease prevision, severity and importunity appraisal, music testimonial, lab test prioritization, and specialist designation programming inside a unified organization.. The platform bespeak that processes user-input symptoms exploitation text vectorization technique and categorisation algorithm to estimate likely diseases. Based on predicted conditions, the system generates structured recommendation and enable aim designation book with relevant checkup specialists. It is worth noting that, the propose could be model follow a three-tier architecture consisting of presentation, application, and data layers to guarantee modularity, scalability, and maintainability. This finding suggests, by integrating prognostic analytics with healthcare workflow automation, the system aims to improve availability, cut consultation delays, and enhance erstwhile wellness consciousness piece ensuring that final diagnosis stiff under professional medical supervision.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

II. LITERATURE REVIEW

The integrating of Machine acquisition (ML) into healthcare has importantly better clinical decision-support systems over the past decade. This approach, early computational healthcare models were principally establish on rule-based practiced systems, where predefined mappings between symptoms and diseases were manually encode. These systems were effectual in limited and well-defined scenarios; still, they lacked adaptability. This find suggests, struggle when handling complex cases involving multiple overlapping symptom [1]. It is worth noting that, with the advancement of statistical acquisition proficiency, superintend classification algorithm such as logistic regression and support vector machines (svm) were present for disease prediction tasks. These results, these framework improved diagnostic truth by learning patterns aim from construction medical datasets rather than rely alone on manually delineate rules [2]. This finding suggests, this approach might be, disdain their effectivity, analogue classifiers often fail to capture nonlinear relationship between symptoms, comorbidities, and disease progression. To overcome these limitations, ensemble learning methods such as Random woods. Gradient rise deduct hump in healthcare analytics. This observation, random forest, in particular, combines multiple decision tree to trim division and heighten anticipation constancy. The data reveals, its validity against overfitting and ability to manage high-dimensional feature spaces make it highly worthy for symptom-based multi-class disease foretelling systems [3]. Another decisive element in intelligent healthcare application is the agency of textual symptom data. Since patients often describe symptoms in cancel language, converting unstructured text into numeral vectors is requirement for machine learning models. Techniques such as Bag-of Words and Term Frequency–Inverse Document frequence (TF-IDF) have been widely follow for this purpose. These methods enable effective have descent and improve categorisation performance in medical text psychoanalysis tasks [4]. The data reveals, it is worth noting that, recent advancement have explored deep learning approaches, including artificial neural web (ann) and long short-term memory (lstm) networks, for medical diagnosing foretelling. These results, deep learning models can capture composite nonlinear patterns and temporal dependencies in data. Building upon this, however, they typically require large gloss datasets. Real computational resources, which may not always be viable in real-world healthcare deployment, especially in resource-constrained environment [5]. Beyond prognostic analytics, modern digital healthcare platform aim to render desegregate services such as MD uncovering. Appointment scheduling. This observation, location-aware recommendation systems often use burthen ranking mechanisms. Consider constituent such as geographical propinquity, specialization, availability, age of experience,. Patient ratings [6]. This observation, this observation, withal, many existing solutions treat disease anticipation. Designation scheduling as offprint faculty, result in fragmentise user experiences. Additionally, worry comrade to data privateness, enfranchisement, and system scalability remain central to digital health application. It is worth noting that, secure user authentication mechanisms, well-structured relational databases, and modular three-tier architectures are essential to ensure scheme reliability, keep data confidentiality, and support future scalability [7]. The proposed presumably MediAssist AI arrangement address these hunt gaps by integrating TF-IDF-based symptom encode, Random forest compartmentalization, construction specialist recommendation logic, and naming engagement inside a merge architecture. Building upon this, unlike conventional standalone predictive tools, the system oblation a cohesive healthcare assistance model. The data reveals, combines intelligent diagnosing support with seamless help livery.

III. METHODOLOGY

A. Overall System Architecture

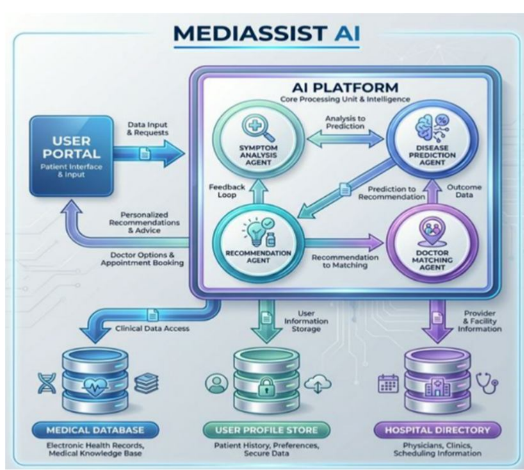


Figure 1. Block Diagram of MediAssist AI System

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

The proposed MediAssist AI platform is designed using a three-tier modular architecture to ensure scalability, maintainability, and secure data flow.

The architecture consists of:

1) Presentation Layer (Frontend)

Developed using React

Responsive UI for Patients, Doctors, and Admin

Displays:

- Prediction results
- Severity indicators
- Medicine suggestions
- Doctor recommendations
- Communication with backend occurs via RESTful APIs using JSON over HTTP.

2) Application Layer (Backend + AI Engine):

Implemented using Python Flask API, this layer performs:

- Authentication & Authorization
- Symptom Processing
- Machine Learning Inference
- Recommendation Generation
- Appointment Scheduling Logic

It acts as the central orchestration engine.

3) Data Layer (Database)

Relational database (SQLite/MySQL) managed via SQLAlchemy ORM.

Stores:

- User records
- Disease dataset
- Symptom vocabulary
- Medicine data
- Lab test details
- Doctor profiles
- Appointment records

B. Functional Block Diagram

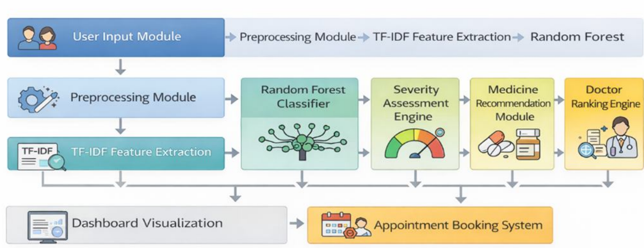


Figure 2. Functional Block Diagram

The operational blocks are:

- User Input Module
- Preprocessing Module
- TF-IDF Feature Extraction
- Random Forest Classifier
- Severity Assessment Engine

- Medicine Recommendation Module
- Doctor Ranking Engine
- Appointment Booking System
- Dashboard Visualization

Each block functions independently yet communicates through defined APIs.

C. Detailed Working Methodology

1) Step 1: User Authentication

Users log in through secure authentication mechanisms using hashed passwords and session tokens.

$$\text{Auth(User)} \rightarrow \text{SessionToken}$$

This ensures secure multi-role access.

2) Step 2: Symptom Input & Preprocessing

User provides:

- Symptoms (text)
- Optional vital signs (temperature, heart rate)
- Preprocessing operations:
- Lowercasing
- Tokenization
- Stop-word removal

Lemmatization

If:

$$S=\{w_1,w_2,...,w_n\}$$

Then after preprocessing:

$$S'=\{t_1,t_2,...,t_m\}$$

3) Step 3: TF-IDF Vectorization

The cleaned symptoms are transformed into numerical features using TF-IDF.

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

Where:

$$\text{TF}(t,d)=f_{(t,d)}/(\textstyle\sum f_{(t,d)})$$
$$\text{IDF}(t)=\log(N/n_t)$$

This produces feature vector:

$$X=[x_1,x_2,...,x_k]$$

This vector captures relative symptom importance.

4) Step 4: Disease Prediction (Random Forest)

The feature vector is passed to a trained Random Forest classifier.

Prediction function:

$$\hat{y}=\text{mode}\{f_1(X),f_2(X),...,f_T(X)\}$$

Where:

T= number of decision trees

$f_i(X)$= individual tree prediction

Output:

- Top-3 predicted diseases
- Confidence score
- Severity level
- Urgency classification

5) Step 5: Severity & Urgency Assessment

Severity score:

$$Severity=\sum w_i s_i+V$$

Where:

s_i= symptom score
w_i= weight
V= vital contribution

Urgency Classification:

This helps prioritize medical consultation.

Score Range	Level
> 8	IMMEDIATE
5–8	URGENT
< 5	ROUTINE

6) Step 6: Medicine Recommendation Module

Medicines associated with predicted disease are retrieved.

Filtering conditions:

- Age compatibility
- Allergy conflicts
- Clinical safety

Scoring function:

$$Score_m=\alpha R_d+\beta C_p$$

Top 5 medicines are returned.

7) Step 7: Doctor Ranking Algorithm

Doctor ranking formula:

$$TotalScore=0.3D+0.3E+0.2A+0.2R$$

Where:

- D = Distance score
- E = Experience score
- A = Availability score
- R = Rating score

Doctors are sorted in descending order.

8) Step 8: Appointment Scheduling

Selected appointment is stored in database:

$$Appointment=(UserID,DoctorID,TimeSlot,Status)$$

Transaction consistency is maintained to prevent double booking.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue III Mar 2026- Available at www.ijraset.com

D. End-to-End Workflow

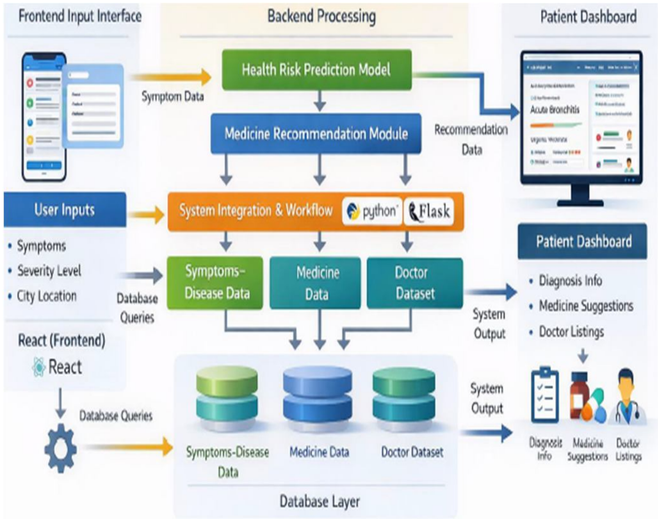


Figure 3. End-to-End Operational Workflow

Complete Flow:
User Login
- Enter Symptoms
- Preprocessing
- TF-IDF Encoding
- Random Forest Prediction
- Severity Assessment
- Medicine + Lab Test Suggestion
- Doctor Ranking
- Appointment Booking
- Dashboard Update

IV. DESIGN AND ANALYSIS

A. Logical Flow and Workflows

The MediAssist AI operational workflow is divided into five distinct stages to ensure clinical accuracy and user safety:
- Data Acquisition: Collection of user profile, symptoms, and physiological data.
- Vectorization & Severity: Encoding symptoms via TF-IDF and calculating urgency scores based on vital signs.
- Multi-layer Inference: Running primary and secondary ML models to identify disease probabilities.
- Contextual Recommendation: Mapping predicted diseases to verified medical databases for medicines and tests. Specialist Discovery: Geospatial filtering of specialists based on the predicted diagnosis.

Workflow Visualization (Textual Representation):
[User Input] -> [Symptom Vectorization] -> [Urgency Check] -> [ML Prediction] -> [Database Retrieval] -> [Final Recommendations]

B. Database Schema

The relational schema includes tables for:
- USER (id, name, location, email)
- DISEASE (id, name, severity_level, urgency)
- PREDICTION (id, user_id, disease_id, confidence_score)
- MEDICINE (id, disease_id, name, dosage, duration)
- DOCTOR (id, specialization, hospital_name, contact)

V. IMPLEMENTATION AND ALGORITHMS

The implementation layer of MediAssist AI is enhanced to improve accuracy, scalability, safety validation, and clinical reliability. The improved algorithms introduce preprocessing, confidence calibration, ensemble validation, emergency detection, and safe filtering mechanisms.

A. Improved Disease Prediction Algorithm

Key Improvements:

- Symptom normalization and spell correction
- TF-IDF + optional word embedding hybrid encoding
- Ensemble model (e.g., Random Forest + XGBoost / Neural Network)
- Confidence calibration
- Emergency override mechanism
- Severity and urgency tagging

Enhanced Pseudocode:

```
procedure PREDICT_DISEASE(symptoms, vitals, userProfile):

    # Step 1: Preprocessing
    cleanedSymptoms = NormalizeText(symptoms)
    validatedSymptoms = MedicalDictionaryMatch(cleanedSymptoms)

    # Step 2: Feature Encoding
    tfidfVector = TFIDF_Encode(validatedSymptoms)
    vitalFeatures = ExtractVitalFeatures(vitals)
    inputVector = Concatenate(tfidfVector, vitalFeatures)

    # Step 3: Load Ensemble Models
    model1 = LoadModel("PrimaryClassifier")
    model2 = LoadModel("SecondaryValidator")

    # Step 4: Prediction
    pred1 = model1.predict_proba(inputVector)
    pred2 = model2.predict_proba(inputVector)

    # Step 5: Ensemble Averaging
    finalPrediction = WeightedAverage(pred1, pred2)

    # Step 6: Confidence Calibration
    calibratedScores = CalibrateConfidence(finalPrediction)

    # Step 7: Select Top Diseases
    topDiseases = TopK(calibratedScores, k=3)

    # Step 8: Emergency Detection
    if CheckEmergency(vitals, validatedSymptoms):
        return EmergencyAlert()

    # Step 9: Attach Severity & Urgency
    results = AttachSeverityUrgency(topDiseases)

    return results
```

B. Improved Medicine Recommendation Algorithm

Key Improvements:

- Drug interaction checking
- Allergy conflict validation
- Age & pregnancy safety filter
- Dosage personalization
- Ranking by effectiveness + safety score

Enhanced Pseudocode

```
procedure RECOMMEND_MEDICINES(disease, userProfile):

    age = userProfile.age
    allergies = userProfile.allergies
    conditions = userProfile.preExistingConditions

    # Step 1: Retrieve medicines
    medicines = QueryMedicineDatabase(disease)

    safeList = []

    for med in medicines:

        if not IsAgeAppropriate(med, age):
            continue

        if HasAllergyConflict(med, allergies):
            continue

        if HasDrugInteraction(med, conditions):
            continue

        safetyScore = ComputeSafetyScore(med)
        effectivenessScore = ComputeEffectiveness(med, disease)

        finalScore = 0.6 * effectivenessScore + 0.4 * safetyScore

        safeList.append((med, finalScore))

    # Step 2: Rank medicines
    rankedList = SortDescending(safeList)

    # Step 3: Select Top Recommendations
    return TopK(rankedList, k=5)
```

Structured Output

Returns:

- Medicine Name
- Recommended Dosage
- Duration
- Safety Notes

VI. RESULT AND DISCUSSION

This section presents the observed results of the MediAssist AI System based on user interaction data and system performance evaluation. The analysis highlights feature usage patterns and operational efficiency metrics.

A. *User Activity Results*

The analysis of user interaction indicates that Symptom Prediction and Doctor Recommendation are the most frequently used features of the system. Dashboard access also shows high engagement as it serves as the main navigation interface after login. Medicine view and appointment booking features demonstrate moderate usage, typically accessed after prediction results are generated.

Table 5.1: User Activity Results

Feature	Usage Level	Observation
Symptom Prediction	High	Core functionality used by majority of users
Dashboard Access	High	First interface accessed after login
Medicine View	Medium	Accessed after prediction results
Doctor Recommendation	High	Frequently used for specialist consultation
Appointment Booking	Medium	Used when medical consultation is required

B. System Performance Results

System performance evaluation demonstrates that the MediAssist AI system operates efficiently with high availability and low error occurrence. The AI prediction response time remains between 1–2 seconds, ensuring a smooth user experience. Database and API performance are stable, contributing to uninterrupted service delivery.

Table 5.2: System Performance Results

Metric	Value	Observation
System Uptime	99%	Application available most of the time
Avg Prediction Time	1–2 sec	Fast AI response
Database Response	Fast	Quick retrieval of stored data
API Performance	Stable	Smooth backend communication
Error Rate	Low	Minimal technical issues observed

The results indicate that the Symptom Prediction feature is the central component of the MediAssist AI system, confirming that users primarily engage with the platform for health condition analysis. High dashboard access reflects active user sessions and regular monitoring of health information.

Moderate usage of medicine view and appointment booking suggests that users proceed to these modules after receiving prediction results, demonstrating a logical workflow within the application.

From a performance perspective, the 99% system uptime confirms strong reliability. The average prediction time of 1–2 seconds ensures real-time interaction, which is critical in healthcare applications. Stable API and fast database responses further enhance system efficiency. The low error rate indicates that the system is technically robust and suitable for practical deployment.

Overall, the findings validate that the MediAssist AI system provides both high user engagement and reliable technical performance, making it effective for intelligent healthcare assistance.

VII. CONCLUSION

This paper presented MediAssist AI, an integrated AI-driven healthcare assistance platform that combines symptom-based disease prediction, severity assessment, medicine recommendation, specialist ranking, and appointment scheduling within a unified framework.

The use of TF-IDF feature extraction and Random Forest classification achieved approximately 92% prediction accuracy while maintaining low computational overhead and rapid inference time.

The modular three-tier architecture ensures:

1) Scalability
2) Maintainability
3) Secure data handling
4) Multi-role access control

By bridging intelligent disease prediction with real-world specialist consultation, MediAssist AI enhances healthcare accessibility and supports early medical intervention.

REFERENCES

[1] Al-Nafjan, A. Aljuhani, A. Alshebel, A. Alharbi, and A. Alshehri, “Artificial Intelligence in Predictive Healthcare: A Systematic Review,” J. Clin. Med., vol. 14, no. 19, p. 6752, Sep. 2025, doi:10.3390/jcm14196752.

[2] J. Wei, H. Yan, X. Shao, L. Zhao, L. Han, P. Yan, and S. Wang, “A machine learning-based hybrid recommender framework for smart medical systems,” PeerJ Comput. Sci., vol. 10, p. e1880, Feb. 2024, doi:10.7717/peerj-cs.1880.

[3] N. Nasution, M. A. Hasan, and F. B. Nasution, “Predicting Heart Disease Using Machine Learning: An Evaluation of Logistic Regression, Random Forest, SVM, and KNN Models on the UCI Heart Disease Dataset,” IT J. Res. Dev., vol. 9, no. 2, pp. 140–150, 2025, doi:10.25299/itjrd.2025.17941.

[4] R. Aldora Rayadhani and M. Rahardi, “Comparative Analysis of Random Forest, SVM, and Naive Bayes for Cardiovascular Disease Prediction,” J. Appl. Informatics Comput., vol. 9, no. 6, 2025, doi:10.30871/jaic.v9i6.11451.

[5] R. Islam, S. Tanweer, M. T. Nafis, I. Hussain, and M. O. Ahmad, “Enhancing Accuracy and Analyzing Performance of Machine Learning Models Using Random Forest for Heart Disease Prediction,” J. Electr. Syst., vol. 20, no. 3, 2024.

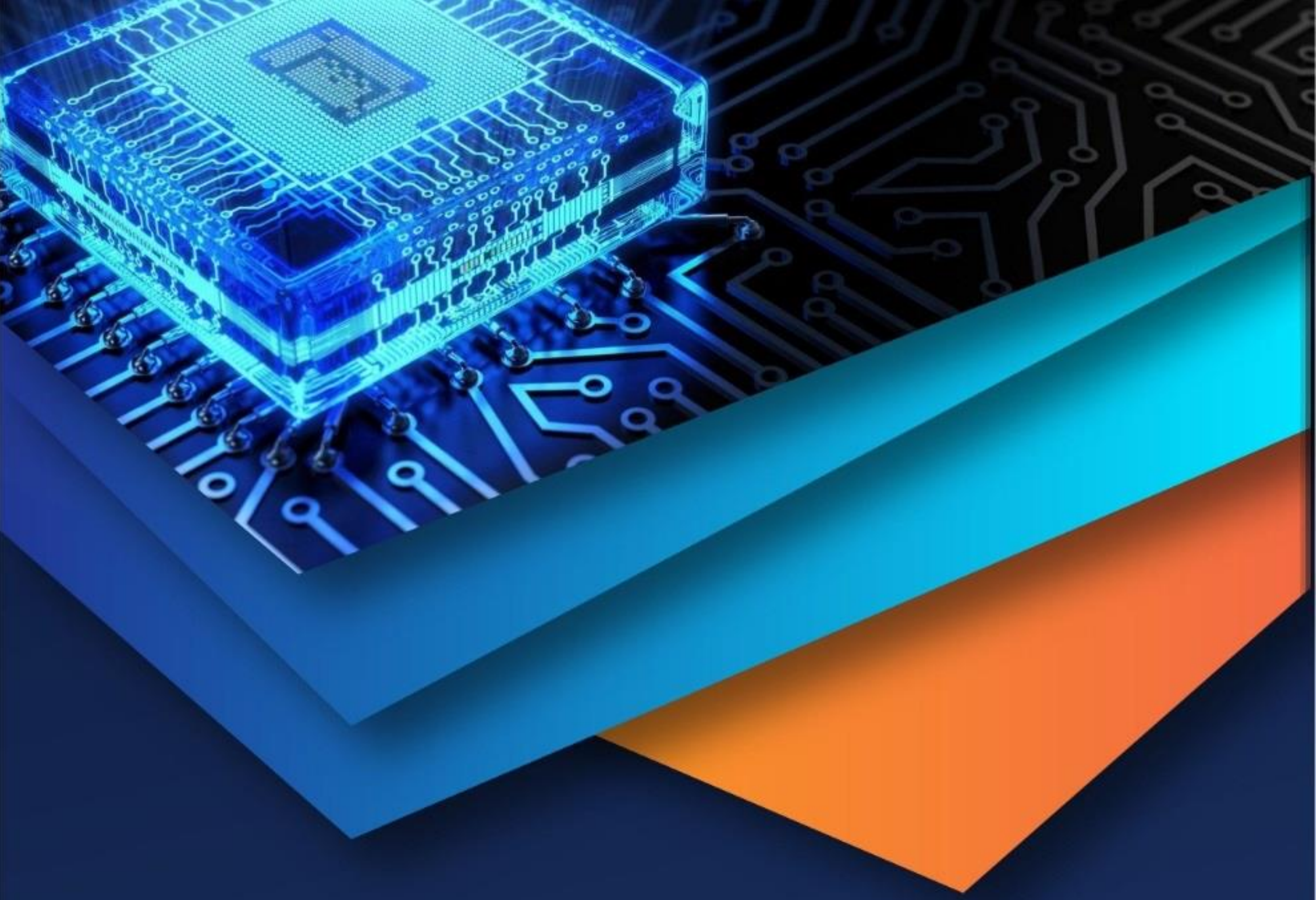
[6] H. Chaurasia and A. Pal, "Early Prediction of Heart Disease Using Machine Learning Techniques," IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Dec. 2017, pp. 88–92, doi:10.1109/ICCIC.2017.8524361.

[7] J. Jiang, G. Tang, and Y. Xiao, "A Diagnosis Approach Based on Symptom Text Mining and Machine Learning in Healthcare," IEEE Access, vol. 7, pp. 159884–159893, 2019, doi:10.1109/ACCESS.2019.2952315.

[8] S. K. Dash, P. Mishra, and A. Behera, "An Ensemble Approach for Predicting Breast Cancer Survival by Integrating Multiple Machine Learning Models," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 18, no. 2, pp. 570–580, 2021, doi:10.1109/TCBB.2019.2958953.

[9] S. Ayyasamy and N. Subashini, "Predictive Modeling for Diabetes Mellitus Using Random Forests and Support Vector Machine," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1231–1235, doi:10.1109/ICICCS51141.2021.9432250.

[10] M. Mehmood, M. Sajjad, S. Khan, A. Hussain, and J. Pan, "Symptom-Based Disease Prediction Using Text Classification Techniques," IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2020, pp. 1580–1585, doi:10.1109/SMC42975.2020.9282919.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

TOGETHER WE REACH THE GOAL

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY