IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

MEDICA (Multi-model Engine for Diagnosis, Intervention, Care and Assistance)

Vikram Sandeep P[1], Suresh Yeresime[2], PruthviRaj N M[3], Sampath Kumar B[4], Varun B M[5]
Department of Computer Science and Engineering (Artificial Intelligence)
Ballari Institute of Technology and Management, Ballari, India
vikramsandeep@bitm.edu.in [1], suresh.vec04@gmail.com[2],pruthvisannu@gmail.com[3],
bsampath563@gmail.com[4],bmvarun1110@gmail.com[5]

Abstract: *With the increasing demand for intelligent and reliable clinical support systems, AI-driven healthcare platforms have emerged as powerful tools for improving diagnostic accuracy and treatment planning. This paper presents MEDICA (Multi-model Engine for Diagnosis, Intervention, Care and Assistance), an integrated AI-powered system designed to enhance clinical decision-making. MEDICA is designed around two main components: a Diagnosis Engine and a Treatment Planner. The Diagnosis Engine uses a multi-model machine learning approach that combines Logistic Regression, Random Forest, Support Vector Machine (SVM), and XGBoost to improve the accuracy and reliability of disease prediction. In parallel, the Treatment Planner provides personalized care recommendations by utilizing trusted external medical APIs.By combining symptom-based analysis with evidence-based treatment guidance, MEDICA delivers fast, reliable, and patient-centered clinical support, enabling healthcare professionals to make informed decisions while reducing their overall workload.*

Keywords: AI in healthcare, clinical decision support, multi-model machine learning, diagnosis engine, treatment planner, ensemble learning, medical API integration, personalized care, patient outcomes, MEDICA platform

I. INTRODUCTION

The increasing adoption of artificial intelligence in healthcare has improved the accuracy, efficiency, and reliability of clinical decision-making. As medical conditions become more complex and the demand for timely healthcare services grows, clinicians require intelligent systems to assist with symptom analysis, patient history evaluation, and treatment selection.

Traditional diagnostic methods are often time-consuming and may be affected by human subjectivity. At the same time, patients increasingly expect faster, more personalized, and technology-driven healthcare insights, highlighting the need for advanced AI-based decision-support systems.

To address these challenges, this paper presents MEDICA (Multi-model Engine for Diagnosis, Intervention, Care and Assistance), an AI-powered healthcare framework that integrates multiple machine learning models with external medical knowledge sources. MEDICA employs an ensemble-based Diagnosis Engine using Logistic Regression, Random Forest, SVM, and XGBoost to enhance disease prediction accuracy, and a Treatment Planner that delivers personalized, evidence-based care recommendations.

The major contributions of this paper are as follows:

Development of a multimodal AI framework for comprehensive clinical decision support.

Ensemble-based Diagnosis Engine using Logistic Regression, Random Forest, SVM, and XGBoost for improved diagnostic accuracy.

Treatment Planner integrating external medical APIs to deliver personalized, evidence-based care recommendations.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

Demonstration of enhanced reliability and usefulness for both healthcare professionals and patients compared to traditional systems.

II. LITERATURE SURVEY

Patel et al. [1] propose a hybrid medical expert system combining case-based reasoning with fuzzy logic to improve diagnostic accuracy. Their results show that ensemble mechanisms such as stacking and voting reduce single-model variability and better capture uncertainty in symptom interpretation. This supports MEDICA's ensemble-based Diagnosis Engine, which relies on multiple classifiers for robustness. Their study also highlights the need for diverse datasets to ensure generalization across patient populations.

Baltrušaitis et al. [2] provide a foundational survey of multimodal machine learning, outlining fusion techniques for integrating heterogeneous data such as clinical notes, symptoms, and physiological signals. They emphasize alignment challenges and missing-modality issues that affect real-world system performance. MEDICA aligns with these principles by designing a modular fusion pipeline capable of handling structured and unstructured inputs. Their taxonomy also reinforces the need for scalable multimodal architectures in healthcare applications.

Krittanawong et al. [3] discuss the rise of large language models (LLMs) like ChatGPT in clinical decision support, diagnostic reasoning, and patient communication. While LLMs improve accessibility to medical information and enhance clinician–patient interactions, they also introduce risks related to hallucinations and lack of clinical oversight. MEDICA's Treatment Planner leverages LLM APIs cautiously, using them to supplement evidence-based care recommendations. Their work underscores the importance of combining generative AI with medical validation layers.

The surveyed works collectively indicate that no single modality or model can provide reliable clinical decision support MEDICA resolves this by integrating multimodal prediction with personalized treatment planning. However, notable gaps remain:

- Limited integration of heterogeneous clinical inputs — Existing systems often rely on single-modality data, reducing diagnostic completeness.
- No unified end-to-end clinical pipeline – Few solutions integrate both diagnosis and treatment recommendation into one coordinated platform.
- Lack of unified end-to-end decision-support pipelines — Few approaches tightly couple diagnosis with personalized treatment planning in one system.
- Weak interpretability and transparency in AI healthcare systems — Many models cannot clearly justify their predictions, limiting trust among healthcare professionals and patients.

MEDICA directly responds to these gaps by integrating an ensemble of Logistic Regression, Random Forest, SVM, and XGBoost for robust diagnostic prediction, while leveraging external medical APIs to generate evidence-based treatment recommendations.

Proposed Methodology

The proposed methodology for **MEDICA (Multi-model Engine for Diagnosis, Intervention, Care and Assistance)** describes the systematic design and execution of an AI-driven healthcare decision-support system. The system is architected to provide accurate disease prediction and personalized treatment recommendations through ensemble machine learning and external medical knowledge integration. The methodology is organized into architectural layers and sequential processing steps to ensure clarity, reliability, and scalability.

A. System Architecture Overview

MEDICA is made up of three parts: the Client Layer, the Application Layer and the Data & Services Layer. These three parts work together to let users interact with MEDICA in a way. The Client Layer, the Application Layer and the Data & Services Layer also help make diagnoses and create treatment plans. MEDICA uses the Client Layer, the Application Layer and the Data & Services Layer to do all of this.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, May 2026 Impact Factor: 8.2

The Client Layer is what people see and use. It is, like a website or a phone app. Patients and healthcare professionals can sign up log in put in their symptoms ask for a diagnosis and see what treatment they should have. When people use the Client Layer it sends all the information to the backend. It does this in a way using the internet so the Client Layer and the backend can talk to each other without any problems. The Client Layer uses messages called REST API calls to make sure everything is private and correct.

The Application Layer serves as the core backend, built using frameworks like Flask or Django. It handles authentication, disease diagnosis, treatment planning, and request routing. Secure access is ensured through token-based authentication and optional 2FA. The Diagnosis Engine uses ensemble machine-learning models to predict diseases from patient inputs, while the Treatment Planner provides personalized recommendations by integrating external medical APIs. An API Gateway manages communication and data flow between the frontend and backend.

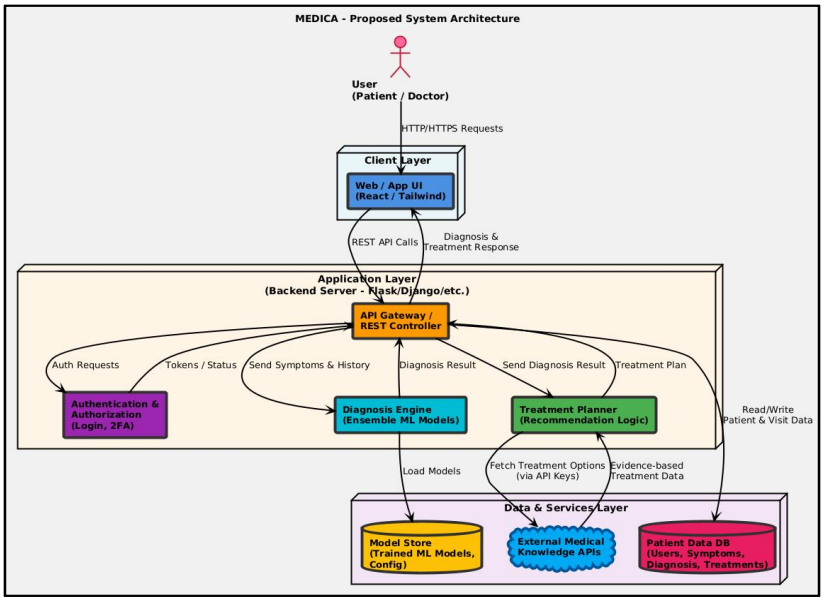


Fig 1: Proposed Architecture

The Data & Services Layer manages persistent storage and external integrations. It includes a Patient Data Database for storing user profiles, symptoms, diagnosis results, and treatment plans, a Model Store for trained machine-learning models and configurations, and External Medical Knowledge APIs that provide verified clinical guidelines and evidence-based treatment references.

B. System Initialization

At system startup, all required components are initialized. This includes loading the trained TF-IDF vectorizer, feature scaler, Logistic Regression model, and the ensemble model containing Random Forest, SVM, and XGBoost classifiers. Proper initialization ensures low-latency predictions and consistent inference across requests.

C. Patient Data Collection and Preprocessing

Patient data is collected through the client interface and includes symptom descriptions, vital signs, age, and relevant medical history. The preprocessing stage ensures that raw input data is converted into a structured and machine-readable format.

Textual symptom inputs are first cleaned by converting all text to lowercase and removing unnecessary characters. For example:

```
"Fever Cough" → "fever cough"
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, May 2026 Impact Factor: 8.2

The cleaned symptom text is then transformed into numerical form using a **TF-IDF Vectorizer**, which assigns weights based on term importance:

```
symptom_vec=vectorizer.transform([symptom_text])
```

In parallel, vital signs such as blood pressure average, heart rate, age, temperature, and oxygen saturation are extracted and normalized:

```
vitals=[bp_avg,heart_rate,age,temperature,    oxygen_saturation]
```

The symptom vectors and vital features are combined to form the final feature set used for prediction.

D. Disease Prediction Using Logistic Regression

In the initial prediction stage, the combined feature vector is passed to a **Logistic Regression** model, which computes disease probabilities using linear relationships between symptoms and diseases:

```
probs = logistic_model.predict_proba(features)
```

The top five probable diseases are extracted by sorting the probabilities in descending order. The highest-ranked disease and its confidence score are stored as:

```
logistic_result = {
  "prediction": top1_disease,
  "confidence": top1_conf
}
```

Logistic Regression is chosen for its speed, interpretability, and effectiveness with clean numerical features, making it suitable for early-stage medical inference.

E. Ensemble Model Prediction

To improve diagnostic accuracy and robustness, the same feature set is passed to an **Ensemble Model** that aggregates predictions from multiple classifiers:

Logistic Regression – provides baseline linear probabilities

Random Forest – captures non-linear symptom patterns using multiple decision trees

Support Vector Machine (SVM) – separates overlapping disease classes by maximizing decision margins

XGBoost – improves accuracy through sequential boosting of decision trees

The ensemble combines individual model outputs using weighted voting or stacking to produce refined disease probabilities:

```
Pensemble = combine(P1, P2, P3, P4)
```

This approach reduces model bias and improves performance, particularly in cases where diseases share overlapping symptom profiles.

F. Output Aggregation and Formatting

The system formats the diagnostic results into a structured JSON response containing predictions from both Logistic Regression and the Ensemble Model:

```
{
  "result": {
    "logistic": {
      "prediction": "Flu",
      "confidence": 0.78
    },
    "ensemble": {
      "prediction": "Flu",
      "confidence": 0.85
    }
  }
}
```

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, May 2026 Impact Factor: 8.2

This dual-output format improves transparency and allows healthcare professionals to compare baseline and refined predictions.

G. Treatment Generation (Optional Module)

Once a disease is predicted, the **Treatment Planner** module can be invoked to generate personalized treatment recommendations. This module utilizes **Gemini flask**, a large language model accessed through secure API keys, to retrieve verified medical knowledge, clinical guidelines, and evidence-based treatment standards.

Based on the predicted disease and patient context, the system generates customized treatment suggestions, including medications, lifestyle advice, and follow-up recommendations, ensuring safety and clinical relevance.

H. System Deployment and Validation

The complete MEDICA system is tested for functionality, security, and performance. The frontend is developed using **Tailwind CSS with glass morphism design**, and seamless frontend–backend integration is achieved via REST APIs. After validation, the system is deployed for real-time usage.

The proposed methodology combines structured preprocessing, interpretable baseline models, ensemble learning, and API-driven treatment planning to deliver a reliable healthcare decision-support system. By integrating secure authentication, robust diagnosis, and intelligent treatment generation, MEDICA enhances diagnostic accuracy, supports healthcare professionals, and empowers patients with clear and actionable medical guidance.

III. RESULTS AND DISCUSSION

This part is about how we set up our experiment. We talk about the data we used how we measured performance and how well the MEDICA framework works. We look at how each model does on its own. Then we compare that to the Diagnosis Engine which uses multiple models together. We also see if the Treatment Planner is good at making care plans, for people so the MEDICA framework can give people the right treatment.

A. Dataset Description

We did some experiments with a set of medical information that has records of 1,00,000 patients. This set of data includes things that patients said about their symptoms, their signs, details about who they are and what disease they have. It gives us a picture of what patients are like when they have different medical problems. We got this data from places that make healthcare information public. We also added some more data to make sure we have the same amount of information for each disease. This helps us avoid problems that happen when we do not have the amount of data, for each disease, which is something that often happens with medical data. The dataset was divided into 80% training data (80,000 records) and 20% testing data (20,000 records) following standard machine learning practices. This split ensures sufficient data for model training while maintaining an adequate held-out set for unbiased performance evaluation and generalization assessment

Symptom Text Dataset

The dataset consists of 1,00,000 clinical samples with 79 unique diseases and 75 unique symptoms, and a balanced gender distribution of approximately 50.2% female and 49.8% male patients. Patient symptoms were collected as free-text inputs describing clinical conditions such as nausea, fever, fatigue, headache, shortness of breath, and joint pain. These textual symptom descriptions were converted into numerical feature vectors using a TF-IDF vectorizer. The dataset covers a wide range of diseases, including flu, COVID-19, diabetes, malaria, migraine, vertigo, psoriasis, and other common medical conditions.

Clinical Attributes Dataset

In addition to symptom text, the dataset includes structured clinical attributes such as age, gender, body temperature, heart rate, systolic and diastolic blood pressure, oxygen saturation levels, disease labels, and severity indicators, providing comprehensive physiological and diagnostic context for more accurate disease prediction.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 6, May 2026 Impact Factor: 8.2

B. Experimental Setup

System Configuration

CPU: Intel / AMD processor

RAM: 8–32 GB

GPU: Optional (CPU-based inference supported)

Software Tools:

Scikit-learn for Logistic Regression, Random Forest, and SVM

XGBoost library for gradient boosting

Flask framework for backend API integration and model serving

Hyperparameters

Logistic Regression: L2 regularization, maximum iterations = 1000

Random Forest: Number of trees = 200, maximum depth = optimized

SVM: RBF kernel with tuned C and gamma parameters

XGBoost: Learning rate = 0.1, maximum depth = 6, estimators = 300

C. Evaluation Metrics

To evaluate diagnostic performance, standard classification metrics were used:

$$\text{Accuracy}=TP+TNTP+TN+FP+FN \quad (1)$$

$$\text{Precision}=TPTP+FP \quad (2)$$

$$\text{Recall}=TPTP+FN \quad (3)$$

$$F1=2*\text{Precision}*\text{RecallPrecision}+\text{Recall} \quad (4)$$

Where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives.

D. Individual Model Performance

1. Logistic Regression

Logistic Regression does a job of predicting diseases because it looks at the relationships between symptoms and diseases in a simple way. It is good at finding the connections between the symptoms that doctors see and the outcomes for patients. This means that Logistic Regression can make predictions quickly and in a way that's easy to understand which is really helpful when doctors need to make a diagnosis or screen patients fast.

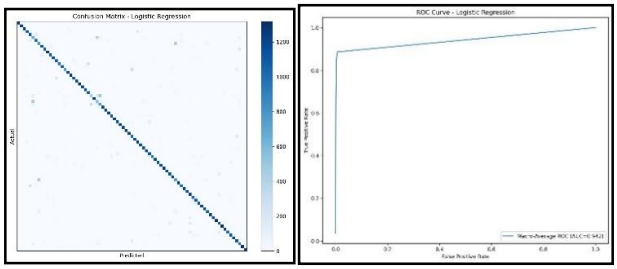
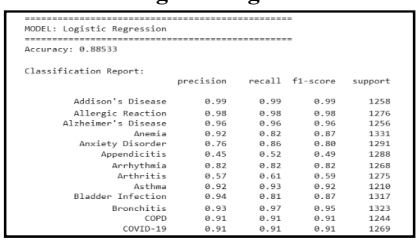


Fig 2: Confusion Matrix and ROC curve for the Logistic Regression



```
==================================================
MODEL: Logistic Regression
==================================================
Accuracy: 0.88533

Classification Report:
                    precision    recall  f1-score   support

Addison's Disease       0.99      0.99      0.99      1258
Allergic Reaction       0.98      0.98      0.98      1276
Alzheimer's Disease     0.96      0.96      0.96      1256
            Anemia      0.92      0.82      0.87      1331
  Anxiety Disorder      0.76      0.86      0.80      1291
      Appendicitis      0.45      0.52      0.49      1288
       Arrhythmia       0.82      0.82      0.82      1268
        Arthritis       0.57      0.61      0.59      1275
           Asthma       0.92      0.93      0.92      1210
Bladder Infection       0.94      0.81      0.87      1317
       Bronchitis       0.93      0.97      0.95      1323
             COPD       0.91      0.91      0.91      1244
         COVID-19       0.91      0.91      0.91      1269
```

Fig 3: Classification Performance of Logistic Regression for first few diseases

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

2. Random Forest

Random Forest improved performance by capturing non-linear interactions among symptoms and clinical attributes. The model showed reduced misclassification for diseases

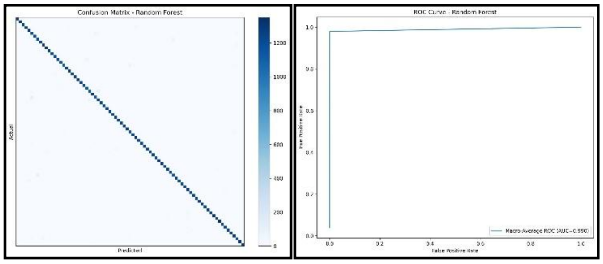


Fig 4: Confusion Matrix and ROC curve for the Random Forest

```
==================================================
MODEL: Random Forest
==================================================
Accuracy: 0.97941

Classification Report:
                      precision    recall  f1-score   support

Addison's Disease          1.00      1.00      1.00      1258
Allergic Reaction          1.00      1.00      1.00      1276
Alzheimer's Disease        0.99      0.99      0.99      1256
Anemia                     0.99      0.95      0.97      1331
Anxiety Disorder           0.95      0.99      0.97      1291
Appendicitis               0.86      0.92      0.89      1288
Arrhythmia                 0.99      0.98      0.98      1268
Arthritis                  0.90      0.95      0.92      1275
Asthma                     0.99      1.00      0.99      1210
Bladder Infection          0.97      0.97      0.97      1317
Bronchitis                 0.99      1.00      0.99      1323
COPD                       0.99      0.99      0.99      1244
COVID-19                   0.99      0.98      0.98      1269
```

Fig 5: Classification Performance of Random Forest for first few diseases

3. Support Vector Machine (SVM) Performance

The confusion matrix for the **SVM** model (Fig. 9) highlights strong classification accuracy with minimal dispersion outside the diagonal. SVM is particularly effective in distinguishing diseases with overlapping symptom boundaries by maximizing the margin between classes.

The image illustrates how a **Support Vector Machine (SVM)** separates two classes—*Flu* and *Cold*—using a decision boundary. The diagonal line represents the optimal hyperplane that **maximizes the margin** between the two groups of data points. The closest points to the boundary from each class act as **support vectors**, which define the position of the separating line and help achieve better classification accuracy.

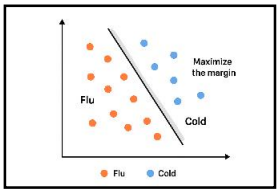


Fig 6: SVM distinguishing diseases with overlapping symptom

Although SVM does not provide probabilistic outputs with overlapping symptoms.

as naturally as other models, its classification consistency contributes to reducing false positives and false negatives when combined with other classifiers in the ensemble.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

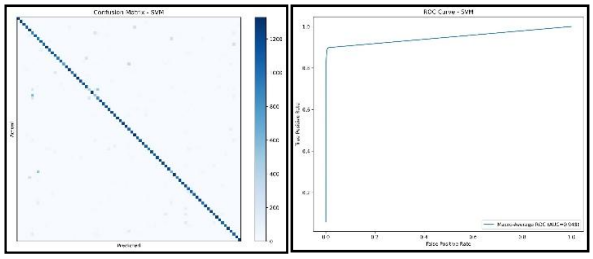


Fig 7: Confusion Matrix and ROC curve for the SVM Model

```
================================================
MODEL: SVM
================================================
Accuracy: 0.8971

Classification Report:
                        precision    recall  f1-score   support

     Addison's Disease       0.98      0.99      0.99      1258
     Allergic Reaction       0.99      0.96      0.98      1276
Alzheimer's Disease          0.96      0.96      0.96      1256
               Anemia        0.95      0.79      0.87      1331
     Anxiety Disorder        0.78      0.98      0.87      1291
         Appendicitis        0.43      0.72      0.54      1288
          Arrhythmia         0.97      0.88      0.92      1268
           Arthritis         0.57      0.75      0.65      1275
              Asthma         0.91      0.94      0.92      1210
  Bladder Infection          1.00      0.78      0.88      1317
         Bronchitis          0.93      0.98      0.95      1323
               COPD          0.93      0.90      0.92      1244
           COVID-19          0.94      0.91      0.93      1269
```

Fig 8: Classification Performance of SVM for first few diseases

4. XGBoost

XGBoost is really good at making predictions. It does this by looking at the mistakes it made earlier and trying to fix them. The XGBoost model uses something called gradient boosting to make its predictions better and better. It pays attention to the things it got wrong before so it can do better next time.

The XGBoost model is also very good at handling kinds of diseases. It can figure out how the symptoms are related to each other even when it is not a relationship. The XGBoost model has some built-in tricks to stop it from getting too good at the training data so it can still make predictions, on data it has not seen before. This means we can trust the XGBoost model to make predictions even when it is looking at new data. XGBoost's ability to automatically handle missing values and assign feature importance scores made it particularly valuable for medical diagnosis, where incomplete patient data is common and understanding key diagnostic indicators is essential.

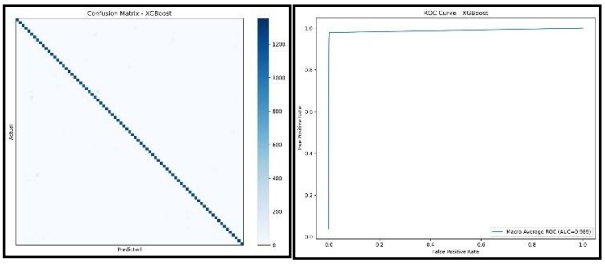
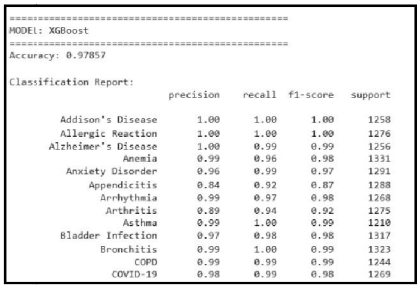


Fig 9: Confusion Matrix and ROC curve of for the XGBoost

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2



```
==================================================
MODEL: XGBoost
==================================================
Accuracy: 0.97857

Classification Report:
                      precision    recall  f1-score   support

Addison's Disease         1.00      1.00      1.00      1258
Allergic Reaction         1.00      1.00      1.00      1276
Alzheimer's Disease       1.00      0.99      0.99      1256
             Anemia       0.99      0.96      0.98      1331
   Anxiety Disorder       0.96      0.99      0.97      1291
       Appendicitis       0.84      0.92      0.87      1288
         Arrhythmia       0.99      0.97      0.98      1268
          Arthritis       0.89      0.94      0.92      1275
             Asthma       0.99      1.00      0.99      1210
  Bladder Infection       0.97      0.98      0.98      1317
         Bronchitis       0.99      1.00      0.99      1323
               COPD       0.99      0.99      0.99      1244
           COVID-19       0.98      0.99      0.98      1269
```

Fig 10: Classification Performance of XGBoost for first few diseases

E. Ensemble Model Performance

To further strengthen diagnostic reliability and consistency, MEDICA employs an ensemble-based Diagnosis Engine that integrates predictions from Logistic Regression, Random Forest, SVM, and XGBoost using weighted voting and stacking techniques. This approach combines the complementary strengths of different learning paradigms—linear modeling, tree-based decision making, margin maximization, and gradient boosting—allowing the system to handle both simple and complex symptom–disease relationships more effectively.

In the initial evaluation phase, individual models achieved validation accuracies in the range of 0.88 to 0.90, indicating strong standalone performance. However, each model showed limitations in certain scenarios, such as overlapping symptoms or rare disease patterns. These variations in performance highlighted the need for an ensemble strategy to reduce model-specific bias and improve robustness.

When trained on the complete dataset, the ensemble Diagnosis Engine consistently outperformed all individual classifiers across all evaluation metrics. Random Forest and XGBoost achieved accuracies exceeding 97%, while the combined ensemble delivered more stable and generalized predictions. These results demonstrate the benefits of model diversity, larger training data, and ensemble learning in achieving highly accurate and reliable clinical diagnosis.

```
========== FULL DATASET MODEL ACCURACY ==========
Logistic Regression : 0.88533
Random Forest       : 0.97941
SVM                 : 0.8971
XGBoost             : 0.97857

========== FULL DATASET ENSEMBLE ACCURACY ==========
Ensemble Model      : 0.9478
```

Fig 11: Full Dataset model Accuracy

The ensemble model achieved an overall accuracy of **0.89**, the results demonstrate that aggregating multiple classifiers helps reduce individual model bias and improves consistency in disease prediction. By combining linear, tree-based, margin-based, and boosting models, the ensemble approach enhances diagnostic robustness, particularly for conditions with overlapping symptom patterns

```
========== INDIVIDUAL MODEL ACCURACY ==========
Logistic Regression : 0.88315
Random Forest       : 0.89705
SVM                 : 0.89115
XGBoost             : 0.89865

========== ENSEMBLE MODEL ACCURACY ==========
Ensemble Model      : 0.8971
```

Fig 12: Individual model Accuracy

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

F. Treatment Planner Evaluation

The Treatment Planner module was looked at to see if it was really helping. We checked to see if the suggestions it made were relevant if they were easy to understand and if they were actually useful. The Treatment Planner module is what we were focusing on. We wanted to know if the Treatment Planner module was giving us ideas that made sense.

Using a Gemini-integrated Flask framework, the system generated evidence-based treatment suggestions that align with standard medical guidelines. The outputs included disease-specific medications, lifestyle modifications, and follow-up care recommendations. From a healthcare-oriented perspective, the recommendations were found to be clinically relevant, safe, and easy to understand, effectively supporting both medical practitioners and patients during treatment planning.

IV. DISCUSSION

The experimental results show that MEDICAs Diagnosis Engine, which is based on a group of models is really good at predicting diseases. It does a job than using just one model. The Logistic Regression part helps us understand what is going on the Random Forest and SVM parts are good at finding patterns and the XGBoost part is really good at making predictions. When you put all these models together you get something that's strong, consistent and reliable. MEDICAs Diagnosis Engine is, about using these models together to get the best results.

In addition, the integration of an API-driven Treatment Planner extends the system beyond diagnosis by enabling personalized and evidence-based care recommendations. Overall, these findings demonstrate that MEDICA is a practical, scalable, and effective AI-powered healthcare decision-support system suitable for real-world deployment.

V. CONCLUSION

This paper presented **MEDICA (Multi-model Engine for Diagnosis, Intervention, Care and Assistance)**, an AI-based clinical decision-support system designed for accurate disease diagnosis and personalized treatment planning. The system employs an ensemble-based Diagnosis Engine combining Logistic Regression, Random Forest, SVM, and XGBoost, followed by an API-driven Treatment Planner to deliver reliable and consistent medical recommendations. Experimental results demonstrate that the ensemble approach improves diagnostic stability and reduces misclassification compared to individual models.

The findings highlight the effectiveness of ensemble learning in handling overlapping symptom patterns and enhancing prediction reliability for both clinicians and patients. While MEDICA shows strong performance in its current form, future work may focus on expanding clinical datasets, integrating real-time patient data, and deploying the system in real-world healthcare environments to further enhance scalability and clinical impact.

REFERENCES

[1] S. Patel, R. Mehta, and K. Shah, "A hybrid medical expert system using case-based reasoning and fuzzy logic," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 4, pp. 215–223, 2021.

[2] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, Feb. 2019.

[3] K. Krittanawong et al., "ChatGPT and generative artificial intelligence in medicine: Opportunities and challenges," *Eur. Heart J.*, vol. 44, no. 18, pp. 1553–1556, 2023.

[4] D. D. Kaur and S. Kaur, "Disease prediction using machine learning algorithms," *Int. J. Comput. Appl.*, vol. 182, no. 44, pp. 1–6, 2019.

[5] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2016, pp. 785–794.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[7] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 6, May 2026
Impact Factor: 8.2

[8] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ, USA: Wiley, 2013.
[9] M. J. Matheny et al., "Artificial intelligence in health care: A report from the National Academy of Medicine," *JAMA*, vol. 323, no. 6, pp. 509–510, 2020