© JUN 2026 | IRE Journals | Volume 9 Issue 12 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I12-1719110

Tumor Detection Using ML

SUYASH DHUMNE[1], SHRIPAD PAWAR[2], DATTA RAHEGAONKAR[3], CHAITANYA HINGMIRE[4]
[1,2,3,4]*Department of CS SKNCOE(BK), Pune Maharashtra, India*

Abstract- ***Breast cancer is one of the most prevalent and life-threatening diseases affecting women worldwide. Early detection and accurate diagnosis play a crucial role in improving survival rates and enabling timely medical intervention. Traditional diagnostic methods often rely on manual examination and expert interpretation, which can be time-consuming and susceptible to human error. The rapid advancement of Machine Learning (ML) technologies has created new opportunities for developing intelligent systems capable of supporting healthcare professionals in disease diagnosis and clinical decision-making. This paper presents the implementation and deployment of a Web-Based Breast Tumor Detection System that utilizes the Random Forest machine learning algorithm to classify tumors as either benign or malignant. The proposed system is developed using the Wisconsin Breast Cancer Diagnostic Dataset, which contains various tumor-related clinical features such as radius, texture, perimeter, area, smoothness, and symmetry. The collected data undergoes preprocessing steps including feature validation, normalization, and label encoding to improve prediction performance and model reliability.***

Keywords— ***Breast Cancer Detection, Tumor Classification, Machine Learning, Random Forest Classifier, Wisconsin Breast Cancer Dataset, Flask API, Next.js, Healthcare Analytics, Predictive Modeling, Clinical Decision Support System, Data Preprocessing, Web-Based Healthcare Application, Early Cancer Diagnosis, Artificial Intelligence in Healthcare.***

I. INTRODUCTION

Breast cancer is one of the most common and life-threatening cancers affecting women worldwide. According to global health reports, millions of new breast cancer cases are diagnosed every year, making it a major public health concern.

Traditionally, breast cancer diagnosis is performed through clinical examinations, mammography, biopsy reports, and laboratory analysis conducted by medical professionals. Although these methods are effective, they often require considerable time, expertise, and resources. In addition, manual interpretation of medical data may sometimes lead to inconsistencies and diagnostic errors, particularly when dealing with large volumes of patient information. As a result, there is an increasing demand for intelligent systems that can assist healthcare professionals in making faster and more accurate diagnostic decisions.

Recent advancements in Artificial Intelligence (AI) and Machine Learning (ML) have transformed various domains of healthcare by enabling automated analysis of complex medical data. Machine learning algorithms can identify hidden patterns, relationships, and trends within clinical datasets that may not be easily recognized through conventional methods. These capabilities have made machine learning an effective tool for disease prediction, medical image analysis, patient monitoring, and clinical decision support systems.

1.2 Problem Statement

1. Dependence on Manual Diagnosis — Traditional breast cancer diagnosis relies heavily on medical expertise, laboratory examinations, and manual interpretation of clinical data, which can be time-consuming and susceptible to human error.
2. Lack of Automated Prediction Systems — Many healthcare facilities, particularly in resource-constrained regions, lack intelligent systems capable of providing quick and accurate tumor classification based on patient diagnostic features.
3. Delayed Diagnostic Decision-Making — The conventional diagnostic process often involves multiple stages of testing and evaluation, leading to delays in identifying malignant tumors and initiating timely treatment.
4. Limited Accessibility of Advanced Diagnostic Tools — High-end diagnostic technologies and specialist consultations may not be readily available in rural or remote areas, limiting access to early breast cancer detection services.

1.3 Objectives

1. Accurate Tumor Classification — To develop a machine learning-based system capable of accurately classifying breast tumors as benign or malignant using clinical diagnostic features.
2. Early Cancer Detection Support — To assist healthcare professionals in identifying potential breast cancer cases at an early stage, improving treatment outcomes and patient survival rates.
3. Automated Prediction Process — To reduce dependency on manual diagnostic procedures by providing automated and data-driven tumor prediction capabilities.
4. Development of a Web-Based Platform — To design and implement a user-friendly web application that enables users to enter medical parameters and receive real-time prediction results.
5. Integration of Machine Learning and Healthcare — To utilize the Random Forest algorithm for enhancing diagnostic accuracy and supporting intelligent healthcare decision-making.

II. SYSTEM ARCHITECHTURE

2.1 Technology Stack

The proposed Breast Tumor Detection System is developed using a combination of modern web technologies, machine learning frameworks, and cloud deployment platforms to ensure accurate prediction, scalability, and ease of access. The technology stack used in the implementation is described below:

Frontend Technologies:

- Next.js – Used for developing a responsive and interactive user interface.
- React.js – Provides component-based architecture for efficient frontend development.
- HTML5, CSS3, and JavaScript – Used for designing and implementing the web interface.

Backend Technologies:

- Python 3.8+ – Primary programming language used for machine learning model development and backend processing.
- Flask Framework – Used to create RESTful APIs and handle communication between the frontend and machine learning model.

Machine Learning Technologies:

- Scikit-learn – Utilized for implementing the Random Forest classification algorithm and model evaluation.
- Pandas – Used for data manipulation, cleaning, and preprocessing.
- NumPy – Provides numerical computation support for machine learning operations.

2.2 Architecture Overview

The proposed Breast Tumor Detection System follows a three-tier architecture that integrates a web-based frontend, a backend processing layer, and a machine learning prediction model to provide accurate and real-time tumor classification.

1. Presentation Layer (Frontend) – Developed using Next.js, this layer provides a user-friendly interface where users can enter tumor-related clinical features and view prediction results.
2. Application Layer (Backend API) – Implemented using Flask, this layer handles user requests, validates input data, performs preprocessing operations, and communicates with the machine learning model.
3. Machine Learning Layer – Consists of a trained Random Forest classifier responsible for analyzing input features and classifying tumors as benign or malignant.
4. Data Preprocessing Module – Performs feature validation, normalization, encoding, and transformation of user-provided data before sending it to the prediction model.
5. Dataset Management Layer – Utilizes the Wisconsin Breast Cancer Dataset for model training, testing, and validation to ensure reliable classification performance.
6. Prediction Engine – Generates tumor classification results along with confidence scores based on patterns learned during model training.
7. Communication Layer – Facilitates secure data exchange between the frontend and backend through RESTful APIs using JSON data format over HTTPS.

8. Deployment Layer – The frontend application is hosted on Vercel, while the Flask backend API and trained machine learning model are deployed on AWS EC2 to ensure scalability and accessibility.

III. PROPOSED METHODOLOGY

The proposed Breast Tumor Detection System utilizes machine learning techniques to classify breast tumors as benign or malignant based on clinical diagnostic features. The methodology consists of data collection, preprocessing, model training, prediction generation, and deployment through a web-based platform. The complete workflow is designed to provide accurate, fast, and reliable tumor classification while ensuring ease of use for healthcare professionals and end users.

3.1 Data Collection

The system utilizes the Wisconsin Breast Cancer Dataset, which contains diagnostic measurements extracted from breast tumor samples. The dataset consists of 569 patient records with multiple numerical features such as radius, texture, perimeter, area, smoothness, compactness, concavity, symmetry, and fractal dimension. These features serve as input parameters for training the machine learning model.

3.2 Data Preprocessing

Before training the model, the dataset undergoes several preprocessing operations to improve prediction accuracy and model performance.

1. Missing values are identified and handled appropriately.
2. Irrelevant attributes are removed from the dataset.
3. Diagnosis labels are encoded into numerical values.
4. Feature scaling and normalization are applied to ensure consistency among input variables.
5. The processed dataset is divided into training and testing subsets for model evaluation.

3.3 Model Training

The Random Forest algorithm is selected as the primary classification model due to its high accuracy, robustness, and ability to handle complex medical datasets.

1. Multiple decision trees are generated using random subsets of training data.
2. Each tree learns patterns from different combinations of features.
3. The model is trained using the processed breast cancer dataset.
4. Hyperparameters are optimized to improve prediction performance and reduce overfitting.

3.4 Prediction Process

After successful training, the model is integrated into the prediction system.

1. Users enter tumor-related diagnostic features through the web interface.
2. The frontend sends the input data to the Flask backend API.
3. The backend performs validation and preprocessing of the received data.
4. The processed input is passed to the trained Random Forest model.
5. The model predicts whether the tumor is benign or malignant.

3.5 Web Application Integration

To improve accessibility and usability, the machine learning model is integrated into a web-based platform.

1. The frontend is developed using Next.js.
2. The backend API is implemented using Flask.
3. RESTful APIs facilitate communication between frontend and backend services.
4. Results are displayed in real time through an interactive user interface.

IV. MATHEMATICAL MODEL

4.1 System Set Theory

Let the system be represented as:

$S = \{U, F, B, M, D, O\}$

Where:

- U = User / Healthcare Professional
- F = Frontend Interface (Next.js)
- B = Backend API (Flask)
- M = Machine Learning Model (Random Forest)
- D = Breast Cancer Dataset
- O = Prediction Output

The overall system can be represented as:
S = Input → Processing → Classification → Output

4.2 Input Feature Vector
The user provides tumor-related diagnostic features through the web interface. These features are represented as an input vector:

$X = \{x_1, x_2, x_3, ..., x_n\}$

Where:
- x_1 = Radius
- x_2 = Texture
- x_3 = Perimeter
- x_4 = Area
- ...
- x_n = Other diagnostic features

The feature vector is passed to the preprocessing module for validation and normalization.

4.3 Preprocessing Function
The preprocessing operation transforms the raw input data into a format suitable for prediction.

$P(X) = X'$

Where:
- P(X) = Preprocessing Function
- X = Original Input Features
- X′ = Normalized and Processed Feature Vector

4.4 Random Forest Classification Model
The Random Forest model consists of multiple decision trees:

$RF = \{T_1, T_2, T_3, ..., T_k\}$

Where:
- RF = Random Forest Classifier
- T_1 to T_k = Individual Decision Trees
- k = Total Number of Trees

Each tree independently predicts the tumor class.

4.5 Majority Voting Mechanism

The final prediction is generated using majority voting among all decision trees.

$Y = Mode(T_1(X'), T_2(X'), ..., T_k(X'))$

Where:
- Y = Final Prediction
- Mode() = Majority Voting Function
- Y = 0 → Benign Tumor
- Y = 1 → Malignant Tumor

V. IMPLEMENTATION

The implementation of the proposed Breast Tumor Detection System focuses on integrating machine learning techniques with modern web technologies to provide an efficient, accurate, and user-friendly platform for breast cancer prediction. The system is developed using a combination of Python, Flask, Scikit-learn, and Next.js, enabling seamless interaction between users and the machine learning model.

5.1 Dataset Implementation
The system utilizes the Wisconsin Breast Cancer Dataset, which contains 569 patient records and multiple diagnostic features related to breast tumors. The dataset includes attributes such as radius, texture, perimeter, area, smoothness, compactness, concavity, symmetry, and fractal dimension. The diagnosis attribute serves as the target variable, indicating whether a tumor is benign or malignant.

5.2 Data Preprocessing Implementation
To improve model performance and ensure data consistency, several preprocessing operations are performed before training the machine learning model.

1. Dataset validation is conducted to identify missing or inconsistent values.
2. Unnecessary attributes are removed from the dataset.

3. Diagnosis labels are converted into numerical values using label encoding techniques.
4. Feature normalization and scaling are applied to standardize input values.
5. The dataset is divided into training and testing sets for model evaluation and validation.

These preprocessing steps improve the quality of input data and enhance prediction accuracy.

5.3 Random Forest Model Implementation

The core functionality of the system is implemented using the Random Forest classification algorithm. The model is trained using the processed dataset and learns patterns associated with benign and malignant tumors.

The implementation process includes:

1. Initializing the Random Forest classifier using Scikit-learn.
2. Training the model on historical breast cancer data.
3. Generating multiple decision trees using random subsets of data and features.
4. Applying majority voting among trees to determine the final classification result.
5. Evaluating model performance using test data.

The Random Forest algorithm was selected because of its high accuracy, robustness, and ability to reduce overfitting compared to individual decision trees.

5.4 Backend API Implementation

The backend component is developed using the Flask framework. It acts as an intermediary layer between the frontend application and the machine learning model.

The backend performs the following operations:

1. Receives user input from the frontend through REST API requests.
2. Validates and preprocesses incoming feature values.
3. Loads the trained Random Forest model stored in serialized format (.pkl file).
4. Sends processed data to the prediction engine.
5. Generates classification results and confidence scores.
6. Returns prediction responses in JSON format.

VI. RESULTS

The performance of the proposed Breast Tumor Detection System was evaluated using the Wisconsin Breast Cancer Dataset and the Random Forest classification algorithm. The evaluation focused on prediction accuracy, system performance, usability, and the effectiveness of integrating machine learning with a web-based platform. The experimental results demonstrate that the system can accurately classify tumors as benign or malignant while providing real-time prediction support.

6.1 Classification Performance

The Random Forest classifier exhibited strong classification capabilities by effectively learning patterns from the breast cancer dataset. The model successfully distinguished between benign and malignant tumors based on multiple diagnostic features such as radius, texture, perimeter, area, and smoothness. The ensemble learning approach improved prediction reliability and reduced the risk of overfitting.

6.2 Prediction Accuracy

The trained model achieved high prediction accuracy during testing and validation phases. Experimental observations indicate that the Random Forest algorithm provides superior performance compared to many traditional classification techniques due to its ability to combine predictions from multiple decision trees. The model consistently produced reliable tumor classifications, making it suitable for healthcare support applications.

6.3 Real-Time Prediction Results

The integration of the machine learning model with the Flask backend and Next.js frontend enabled real-time prediction generation. Users were able to enter diagnostic parameters through the web interface and receive classification results within a short response time. This real-time capability enhances the practical usability of the system in clinical and research environments.

VII. TESTING METHODOLOGIES

The proposed Breast Tumor Detection System was subjected to various testing methodologies to ensure accuracy, reliability, usability, and overall system performance. Testing was performed on individual components as well as the integrated system to verify that the application functions correctly under different conditions.

7.1 Unit Testing

Unit testing was conducted on individual modules of the system, including data preprocessing functions, machine learning prediction modules, and backend API components. Each module was tested independently to ensure that it produced the expected outputs for given inputs and handled invalid data appropriately.

7.2 Model Validation Testing

The Random Forest classifier was evaluated using training and testing datasets obtained from the Wisconsin Breast Cancer Dataset. Model validation was performed to assess the classifier's ability to correctly identify benign and malignant tumors. Performance was analyzed using classification results, prediction accuracy, and consistency of outputs across different test samples.

7.3 API Testing

The Flask backend API was tested to verify proper communication between the frontend application and the machine learning model. Various API requests were executed to ensure successful data transmission, response generation, input validation, and error handling. The API consistently returned accurate prediction results and confidence scores within acceptable response times.

7.4 Integration Testing

Integration testing was performed to validate the interaction between the Next.js frontend, Flask backend, and Random Forest prediction model. The testing process ensured smooth data flow across all system components and verified that prediction results generated by the machine learning model were correctly displayed on the user interface.

7.5 User Interface Testing

The web-based interface was tested to evaluate usability, responsiveness, and functionality. Different input scenarios were examined to ensure that users could easily enter tumor-related features and receive prediction results without encountering interface-related issues. Compatibility testing was also performed across multiple web browsers and devices.

VIII. ADVANTAGES & LIMITATIONS

8.1 Advantages

1. High Prediction Accuracy – The Random Forest algorithm provides reliable and accurate classification of breast tumors as benign or malignant based on diagnostic features.
2. Early Cancer Detection Support – The system assists healthcare professionals in identifying potential cancer cases at an early stage, improving treatment planning and patient outcomes.
3. Reduced Manual Effort – Automated prediction minimizes the dependency on manual analysis and reduces the workload of medical practitioners.
4. Real-Time Prediction – Users receive instant classification results and confidence scores through the web-based platform, enabling faster decision-making.
5. User-Friendly Interface – The Next.js-based frontend provides an intuitive and responsive interface that simplifies data entry and result interpretation.

8.2 Limitations

1. Dependence on Dataset Quality – The performance of the machine learning model is directly influenced by the quality, diversity, and size of the training dataset.
2. Limited to Structured Data – The current implementation only supports numerical clinical features and does not process medical images such as mammograms or MRI scans.
3. No Direct Medical Diagnosis – The system provides prediction support and should not be considered a replacement for professional medical consultation or diagnosis.

4. Internet Dependency – Since the application is web-based, a stable internet connection is required for accessing prediction services.

IX. FUTURE SCOPE

The proposed Breast Tumor Detection System demonstrates the potential of machine learning in supporting early breast cancer diagnosis. Although the current implementation provides accurate tumor classification using clinical diagnostic features, several enhancements can be incorporated in future versions to improve system capabilities, accuracy, and real-world applicability.

1. Integration of Medical Imaging Data – Future versions of the system can incorporate mammogram, ultrasound, MRI, and histopathology images to enable image-based tumor detection using deep learning techniques such as Convolutional Neural Networks (CNNs).
2. Multi-Cancer Detection Support – The system can be extended to classify and predict other types of cancers, including lung cancer, brain tumors, skin cancer, and cervical cancer, thereby increasing its healthcare applications.
3. Explainable Artificial Intelligence (XAI) – Explainable AI techniques such as SHAP and LIME can be integrated to provide detailed explanations of prediction results, helping healthcare professionals understand the factors influencing model decisions.
4. Integration with Hospital Information Systems – The application can be connected with Electronic Health Records (EHRs), Laboratory Information Systems (LIS), and Hospital Management Systems (HMS) to streamline clinical workflows and improve data accessibility.
5. Real-Time Clinical Monitoring – Future implementations may include patient monitoring features that continuously analyze diagnostic data and generate alerts for high-risk cases.

X. CONCLUSION

The proposed Breast Tumor Detection System successfully demonstrates the application of machine learning techniques for the early detection and classification of breast cancer tumors. The system utilizes the Random Forest algorithm to analyze clinical diagnostic features and accurately classify tumors as benign or malignant. By leveraging the Wisconsin Breast Cancer Dataset, the model is capable of identifying complex patterns within medical data and generating reliable prediction results that can support preliminary diagnostic decision-making.

The integration of a Next.js frontend, Flask-based backend API, and machine learning prediction model provides a complete web-based solution that enables users to access tumor classification services through an interactive and user-friendly interface. The implemented architecture ensures efficient communication between system components while delivering real-time prediction results along with confidence scores. Cloud deployment further enhances accessibility, scalability, and availability of the application.

Experimental evaluation demonstrates that the Random Forest classifier provides strong classification performance, making it a suitable choice for healthcare prediction applications. The system reduces manual effort, improves prediction speed, and offers a cost-effective approach for supporting breast cancer diagnosis. Furthermore, the web-based implementation allows healthcare professionals and researchers to access predictive insights conveniently without requiring specialized software or hardware resources.

REFERENCES

[1] W. H. Wolberg, W. N. Street, and O. L. Mangasarian, “Machine Learning Techniques to Diagnose Breast Cancer from Fine-Needle Aspirates,” Cancer Letters, vol. 77, no. 2–3, pp. 163–171, 1994.

[2] O. L. Mangasarian, W. N. Street, and W. H. Wolberg, “Breast Cancer Diagnosis and Prognosis via Linear Programming,” Operations Research, vol. 43, no. 4, pp. 570–577, 1995.

[3] L. Breiman, "Random Forests," Machine Learning Journal, vol. 45, no. 1, pp. 5–32, 2001.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[5] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Burlington, MA, USA: Morgan Kaufmann Publishers, 2012.

[6] T. M. Mitchell, Machine Learning. New York, NY, USA: McGraw-Hill Education, 1997.

[7] K. P. Murphy, Machine Learning: A Probabilistic Perspective. Cambridge, MA, USA: MIT Press, 2012.

[8] S. Raschka and V. Mirjalili, Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow. Birmingham, UK: Packt Publishing, 2017.

[9] A. Jafari, M. Alhajj, and A. Rokne, "Machine-Learning Methods in Detecting Breast Cancer and Related Conditions: A Survey," Taylor & Francis Healthcare Technology Journal, vol. 18, no. 4, pp. 215–228, 2024.

[10] A. La Moglia, G. Romano, and F. Marino, "Breast Cancer Prediction Using Machine Learning Classification Techniques," ScienceDirect Healthcare Analytics, vol. 7, pp. 100–112, 2025.

[11] UCI Machine Learning Repository, "Breast Cancer Wisconsin (Diagnostic) Dataset," University of California, Irvine, CA, USA, 2025. Available: https://archive.ics.uci.edu

[12] A. Khalid, M. Hassan, and S. Ahmed, "Breast Cancer Detection and Prevention Using Machine Learning," PMC Medical Informatics Journal, vol. 15, no. 2, pp. 85–97, 2023.

[13] R. Hickman, J. Parker, and D. Lewis, "Deep Learning Algorithms for Breast Cancer Detection and Classification," International Journal of Medical AI Research, vol. 11, no. 3, pp. 145–159, 2024.

[14] P. Boddu, S. Rao, and K. Kumar, "Machine Learning Algorithms for Breast Cancer Detection: A Systematic Review," Journal of Healthcare Analytics, vol. 9, no. 1, pp. 45–63, 2025.