A Machine Learning-Based Screening Framework for Vitamin Deficiency Risk Prediction Using Random Forest and XGBoost

Imrana Siddiqua S[1], Kamaleshware P[2], Dr. S Mohana[3], Dr. V Punitha[4]
[1,2]*Department of Computer Science and Engineering, Saranathan College of Engineering, India*
[3,4]*Guided by M.E., Ph.D., Professor, Department of Computer Science and Engineering*

***Abstract*—Vitamin deficiencies may remain unrecognized when symptoms are nonspecific or access to laboratory testing is limited. This work presents a machine learning-based screening framework derived from a final-year project for predicting possible vitamin deficiency patterns from demographic, dietary, lifestyle, and symptom-related inputs. The project implements Random Forest and XGBoost classifiers and also includes an application-layer rule-based scoring module for Vitamin D, Vitamin B12, and iron risk indicators. The user-facing system is implemented as a web application with Flask-based backend services, authentication, dashboard, prediction, and history modules. The source project reports approximate accuracies of 92% for Random Forest and 95% for XGBoost, together with successful functional test cases. However, the supplied training code fits both models on the complete dataset and does not show a held-out train-test split or independent evaluation procedure; therefore, the reported accuracy values are treated here as project-reported results rather than independently validated generalization performance. The proposed framework demonstrates how structured health-related inputs can be transformed into machine-learning-ready features and integrated into an accessible screening-oriented application.**

***Index Terms*—Data preprocessing, healthcare, machine learning, Random Forest, vitamin deficiency, XGBoost.**

I. INTRODUCTION

Vitamin deficiency is a recurring nutritional health concern because inadequate intake or absorption of essential nutrients can affect immunity, metabolism, bone health, blood formation, and neurological function. The project report identifies Vitamin D, Vitamin B12, and iron as key deficiency categories and notes that fatigue, anemia, bone-related problems, and neurological symptoms may accompany nutritional deficiencies.

Conventional identification of deficiencies commonly depends on laboratory testing and clinical interpretation. The project report highlights cost, time, access to facilities, and limited awareness as practical barriers, particularly where healthcare resources are constrained. These limitations motivate the use of computational screening tools that can analyze readily available self-reported information and provide an initial risk indication.

Machine learning methods are increasingly used for structured healthcare data because they can model nonlinear relationships among demographic, behavioral, dietary, and symptom variables [6], [9]. Within the project, Random Forest is used as a bagging-based ensemble baseline [2], while XGBoost is used as a boosting-based comparison model [3].

The contribution of this work is not a replacement for medical diagnosis, but an application-oriented framework that combines structured feature engineering, two tree-based classifiers, rule-based nutrient scoring, and a web interface. The paper consolidates the final-year project into a research-paper format and explicitly distinguishes experimentally reported results from details that are not fully documented in the supplied implementation.

A. Objectives and Scope

The project objectives are to preprocess vitamin-related data, engineer numerical features from user inputs, train Random Forest and XGBoost models, expose prediction functionality through a Flask backend, store prediction records, and provide a user-friendly interface. The intended use is early screening and awareness based on symptoms, diet, and lifestyle information rather than laboratory confirmation.

II. RELATED WORK

Prior work reviewed in the project report establishes three relevant strands. First, machine learning has been applied to healthcare prediction, risk assessment, and decision support using structured patient data [6], [9]. Second, Random Forest improves stability by averaging predictions from multiple decision trees and introducing randomness in sampling and feature selection [2]. Third, XGBoost constructs boosted trees sequentially, using gradient-based optimization and regularization to reduce prediction error [3].

The report also cites nutrition-focused work that applies machine learning to deficiency prediction from diet, lifestyle, and symptoms [7], while a WHO source is used in the project to emphasize the public-health relevance of micronutrient deficiencies [10]. These studies motivate a system that uses accessible attributes as inputs while retaining the need for later clinical confirmation.

The main gap identified from the project context is practical integration. The reviewed approaches are summarized at the algorithm or healthcare-analysis level, whereas the proposed project connects data preprocessing, prediction logic, persistent history, and a web interface within a single application.

III. PROPOSED METHODOLOGY

A. Input Data and Feature Representation

The project dataset is stored as a CSV file, with each row representing a user record and a target field representing the deficiency category used by the supervised learning experiment. The dataset description explicitly documents age, gender, diet, milk intake, egg intake, fish intake, greens intake, fruits intake, sunlight exposure, activity level, fatigue, bone pain, muscle weakness, pale skin, and dizziness. The implementation section additionally shows a broader feature vector containing skin tone, sunlight timing, work type, smoking, hair fall, anemia, thyroid, and digestion indicators.

The dataset description explicitly documents 15 core fields: age, gender, diet, milk, eggs, fish, greens, fruits, sunlight exposure, activity level, fatigue, bone pain, muscle weakness, pale skin, and dizziness. The implementation section additionally shows skin tone, sunlight timing, work type, smoking, hair fall, anemia, thyroid, and digestion in its feature vector. This difference is retained rather than silently merging the two descriptions.

B. Data Preprocessing and Feature Engineering

Categorical fields are converted into numerical representations before modeling. The implementation maps gender and diet categories to integer values and encodes multi-level attributes such as sunlight exposure and activity. Boolean symptoms are converted into binary indicators. The report states that missing values are checked and handled and that feature scaling is not emphasized because both selected models are tree-based and relatively insensitive to scale.

Feature engineering is therefore used to create a consistent numerical feature vector for the models. This transformation also enables the same input structure to be passed from the web application to the prediction layer.

C. Random Forest

Random Forest constructs multiple decision trees using randomized data samples and feature subsets. For a classification problem, the final class can be represented as the mode of the individual tree predictions: $RF(x) = \text{mode}(T1(x), T2(x), \ldots, Tn(x))$. In the supplied training code, the forest is configured with 200 estimators and random_state = 42. The model is used as a baseline because ensemble voting can reduce the instability of a single tree [2].

D. XGBoost

XGBoost builds decision trees sequentially so that later trees focus on residual errors from earlier trees. The project uses XGBClassifier with the evaluation metric set to mlogloss. The method is selected as a comparison model because boosting can capture complex interactions in structured data while incorporating regularization [3].

E. Hybrid Application Logic

The project report describes a hybrid approach. In the supplied implementation, the application-level prediction endpoint uses explicit rules to calculate nutrient risk scores. For example, zero milk intake increments the Vitamin D score, a vegetarian diet increments the Vitamin B12 score, and zero greens intake increments the iron score. Each score is capped at 100. Separately, the training module builds and

saves the Random Forest and XGBoost models as serialized files. The supplied code therefore documents two complementary layers: interpretable rule-based screening outputs and machine-learning model training for classification comparison.

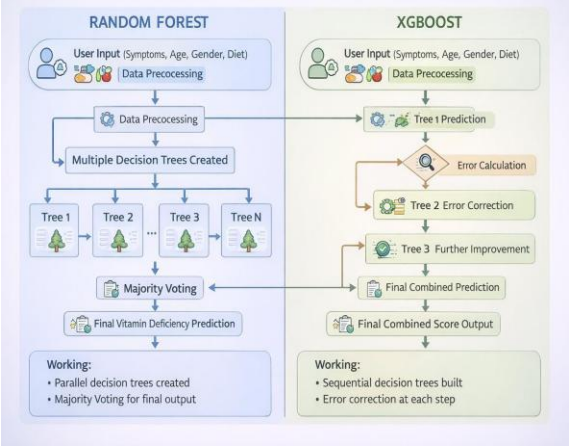

Fig. 1. Project architecture comparing the Random Forest and XGBoost prediction paths.

IV. SYSTEM IMPLEMENTATION

A. Application Components

The web application is organized into Flask blueprints for authentication, prediction, dashboard, assistant, and history functions. CORS is enabled to support frontend-backend communication, and JWT-based authentication is configured for protected access. The dashboard presents recent prediction results, while the history module retrieves stored prediction records for longitudinal viewing.

The implementation separates model training from request-time processing. The training script loads the CSV file, separates the target column from the inputs, trains Random Forest and XGBoost, and persists the fitted models with joblib. This reduces the need to retrain models for every user request.

B. Prediction Output

The application-level prediction response contains normalized values for Vitamin D, Vitamin B12, and iron. The project report illustrates a sample response in which Vitamin D is 75%, Vitamin B12 is 60%, iron is 55%, and the displayed confidence is 0.92. These values are example outputs from the project documentation and should not be interpreted as laboratory measurements.

C. User Interaction Flow

The workflow begins with user entry of demographic, diet, lifestyle, and symptom information. The data is validated and converted to the required numerical representation. The prediction layer produces screening values, after which results are stored and presented through the dashboard. The history module enables earlier records to be reviewed for trend monitoring.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Model Performance

The final-year report states that Random Forest achieved approximately 92% accuracy and XGBoost approximately 95%. These are the central comparative results reported by the project. Because the report does not provide a dataset size, class distribution, train-test split ratio, confusion matrix, precision, recall, F1-score, or confidence intervals, the comparison in this paper is limited to the reported accuracy values.

The Random Forest configuration in the supplied code uses 200 estimators with random_state = 42. The XGBoost configuration instantiates XGBClassifier with eval_metric = mlogloss. The project report records approximately 92% accuracy for Random Forest and 95% for XGBoost.

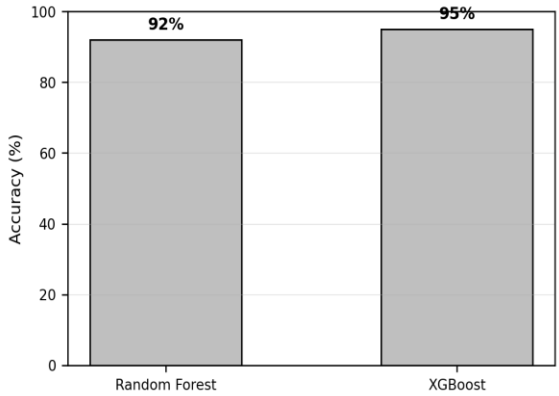

Fig. 2. Project-reported accuracy comparison between Random Forest and XGBoost.

B. Functional Test Cases

The project report lists four functional cases: TC1 uses low sunlight and no milk and expects Vitamin D deficiency; TC2 uses a vegetarian diet and no eggs and expects Vitamin B12 deficiency; TC3 uses no greens and pale skin and expects iron deficiency; and TC4

uses a balanced diet and expects low deficiency. All four cases are marked as passed. These cases validate representative application logic but do not substitute for independent clinical validation or statistically sound model evaluation.

C. Methodological Observations

A close inspection of the supplied implementation reveals an important evaluation limitation. The training code calls fit(X, y) for both models immediately after loading the full dataset, without showing a train-test split or an unseen evaluation set. Consequently, the reported 92% and 95% figures cannot be independently interpreted from the supplied code as held-out test accuracy. A stronger experimental version should define the dataset size and provenance, perform stratified train-test or cross-validation, report class-wise precision, recall and F1-score, and provide a confusion matrix.

There is also a separation between the rule-based prediction endpoint and the model-training module: the endpoint code shown in the report computes nutrient scores through explicit rules and does not show joblib model loading or predict() calls. For a production research prototype, the paper's architecture should therefore describe which output is generated by rules and which output is generated by the trained classifiers, rather than treating them as a single undifferentiated model.

VI. DISCUSSION

The project demonstrates a practical pipeline from user-entered health information to machine-learning-ready features and web-based screening outputs. Its main application-oriented strengths are modular implementation, interpretable rule-based scores, storage of prediction history, and comparison of two widely used tree ensembles. The project also includes Tamil and English interface support and hospital-location assistance, improving usability in the intended regional context.

At the same time, the current project should be positioned as a screening and awareness tool. The dataset description is qualitative and does not document dataset size or class balance, and the model-training code does not expose an independent test protocol. The project therefore provides a useful prototype foundation but requires stronger experimental controls and clinical validation before any diagnostic or treatment claim could be justified.

VII. CONCLUSION AND FUTURE WORK

This paper reformulates the final-year project as a compact research contribution: a screening-oriented vitamin deficiency prediction framework using structured demographic, dietary, lifestyle, and symptom inputs. Random Forest and XGBoost are implemented as comparative classifiers, while the web application adds rule-based nutrient risk scoring, authentication, dashboards, and history management. The project reports approximately 92% accuracy for Random Forest and 95% for XGBoost, with all listed functional test cases marked as passed.

Future work should focus on reproducible evaluation and clinically meaningful validation. The project report proposes larger real-world datasets, additional machine-learning or deep-learning models, mobile deployment, wearable integration, laboratory-data integration, detailed nutrition analysis, voice input, chatbot support, cloud deployment, and stronger security. Before such extensions, a rigorous evaluation protocol should be added so that performance claims are based on unseen data and class-wise metrics rather than accuracy alone.

REFERENCES

[1]. C. M. Bishop, Pattern recognition and machine learning. New York, NY, USA: Springer, 2006.
[2]. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
[3]. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, pp. 785–794, 2016.
[4]. I. Goodfellow, Y. Bengio, and A. Courville, Deep learning. Cambridge, MA, USA: MIT Press, 2016.
[5]. J. Han, J. Kamber, and J. Pei, Data mining: Concepts and techniques. Waltham, MA, USA: Morgan Kaufmann, 2011.
[6]. S. B. Kotsiantis, "Supervised machine learning: A review of classification techniques," Informatica, vol. 31, pp. 249–268, 2007.
[7]. V. Patel and M. Shah, "Predicting nutritional deficiency using machine learning," International

Journal of Engineering Research & Technology, vol. 8, no. 6, 2019.

[8]. J. R. Quinlan, "Induction of decision trees," Machine Learning, vol. 1, pp. 81–106, 1986.

[9]. A. Sharma and P. Mehta, "Application of machine learning in healthcare," International Journal of Computer Applications, vol. 179, no. 1, pp. 10–15, 2018.

[10]. World Health Organization, "Micronutrient deficiencies," 2020. [Online]. Available: https://www.who.int