Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 9, Nomor 2, Agustus 2024, pp 1117-1132
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

Enhancing Medical Diagnostics with Ensemble Machine Learning: A Comparative Study of Gradient Boosting, XGBoost, LightGBM, and Blended Models

Gregorius Airlangga
Universitas Katolik Indonesia Atma Jaya, Indonesia
Email: gregorius.airlangga@atmajaya.ac.id

Abstract

This research investigates the performance of various machine learning models, including Gradient Boosting, AdaBoost, Support Vector Machine (SVM), Logistic Regression, XGBoost, LightGBM, and a Blended Model, in the context of medical diagnostics. The objective of the study is to identify the most accurate and reliable model for predicting outcomes, particularly in cases where the accurate identification of positive instances is critical. The research employs a systematic evaluation using cross-validation and test accuracy metrics to assess each model's performance. Results indicate that ensemble methods, such as Gradient Boosting, XGBoost, and LightGBM, generally outperform simpler models. LightGBM achieved the highest cross-validation accuracy at 89.10%, while the Blended Model demonstrated the potential of combining multiple classifiers, achieving a cross-validation accuracy of 90.19%. However, a common challenge across all models was balancing precision and recall for the positive class, suggesting the need for further optimization. The study concludes that while advanced ensemble methods show promise, enhancing the models' sensitivity to positive cases is crucial for improving their applicability in medical diagnostics. Future research should focus on refining these models to achieve a better balance between precision and recall, ensuring that critical cases are not overlooked.

Keywords: *Machine Learning, Ensemble Methods, Medical Diagnostics, LightGBM, Model Performance*

1. INTRODUCTION

Gestational diabetes mellitus (GDM) is a significant health concern that arises during pregnancy, typically in the second or third trimester [1]–[3]. It is characterized by high blood sugar levels that can lead to severe complications for both the mother and the unborn child, including preeclampsia, macrosomia, and a higher risk of developing type 2 diabetes post-pregnancy [4]. The early diagnosis of GDM is crucial to managing these risks effectively, but traditional diagnostic methods, such as the oral glucose tolerance test (OGTT), are time-consuming, uncomfortable for the patient, and often inaccessible, particularly in low-resource settings [5]. Consequently, there is a growing need for alternative methods that can offer reliable and early detection of GDM, thereby enabling timely intervention [6]. In recent years, the application of machine learning in healthcare has gained considerable traction, offering the potential to revolutionize traditional diagnostic approaches [7], [8]. Machine learning models are capable of analyzing large datasets, uncovering patterns, and making predictions with high accuracy, which is especially valuable in scenarios where human expertise or resources are limited [9]. For GDM, machine learning models can be trained on a variety of clinical and

demographic data to predict the likelihood of its occurrence, providing a scalable and efficient solution for early detection [10].

The urgency of developing effective machine learning models for GDM is further amplified by the increasing prevalence of the condition [11]. The global incidence of GDM is rising, with some estimates suggesting that it affects up to 14% of pregnancies worldwide [12]. In regions like Kurdistan, where this study's data was collected, the challenge is compounded by limited healthcare infrastructure and the lack of widespread screening programs [13]. In such contexts, machine learning models that can be deployed with minimal resources and still deliver high accuracy are of paramount importance [14]. Despite the promising potential of machine learning in this domain, there remains a noticeable gap in the existing literature. Many studies have employed relatively simple models, such as logistic regression, or have focused on individual classifiers like Random Forests or Support Vector Machines. While these models have demonstrated some degree of success, they often fall short in capturing the complex relationships inherent in medical data. Moreover, the reliance on a single model type can lead to overfitting or underfitting, especially when dealing with diverse and imbalanced datasets typical of medical research [15].

The state of the art in machine learning for medical diagnostics has increasingly moved towards the use of ensemble methods, which combine multiple models to improve prediction accuracy and robustness. Among these, stacking models and voting classifiers have shown considerable promise. Stacking involves training several base models and then combining their predictions through a meta-model, typically a more straightforward classifier like logistic regression or another robust model [16]. Voting classifiers, on the other hand, aggregate the predictions of multiple models, typically assigning a weight or ‘vote’ to each model's prediction, to generate a final decision. These techniques help mitigate the weaknesses of individual models and leverage their strengths, leading to improved overall performance. This research contributes to the field by proposing an advanced blended method that integrates multiple stacking models and combines them using a voting classifier [17]. The proposed method capitalizes on the diversity and strengths of different machine learning algorithms, aiming to enhance predictive performance for GDM diagnosis. Specifically, the study introduces two stacking models: one that combines Random Forest, Gradient Boosting, and Support Vector Machine, with XGBoost as the meta-model; and another that integrates Extra Trees, AdaBoost, and Logistic Regression, with LightGBM as the meta-model. These stacking models are then blended using a voting classifier, which combines their predictions along with those from standalone XGBoost and LightGBM models [18], [19]. This blended approach is designed to achieve higher accuracy and robustness compared to conventional methods.

The primary contribution of this study is the development and evaluation of this advanced blended model for GDM prediction. By leveraging the complementary strengths of multiple machine learning algorithms, the proposed model aims to provide a more reliable tool for early diagnosis, particularly in

settings where traditional diagnostic methods are impractical or unavailable. Additionally, this study offers a comprehensive analysis of the performance of various machine learning models, providing valuable insights for researchers and practitioners interested in applying machine learning to medical diagnostics. The remainder of this article is structured as follows. The subsequent section provides a detailed explanation of the research methodology, including data preprocessing, model selection, and evaluation metrics. The results section presents the performance of the proposed blended model in comparison with individual and stacking models, while the discussion section interprets these findings in the context of existing literature, emphasizing the practical implications and potential limitations of the approach. Finally, the article concludes with a summary of the key contributions and directions for future research.

2. RESEARCH METHODOLOGY

The research methodology of this study is meticulously designed to ensure the robustness and reliability of the proposed blended machine learning model for predicting gestational diabetes mellitus (GDM). The methodology comprises three primary components: data preprocessing, model selection, and evaluation metrics. Each of these components is elaborated in detail below, with a focus on advanced techniques and the mathematical foundations underpinning the approaches.

2.1. Data Preprocessing

Data preprocessing is a fundamental stage in the machine learning pipeline, as it profoundly influences the model's performance, interpretability, and generalizability. This study implements a meticulous and multi-faceted preprocessing approach, encompassing feature scaling, addressing class imbalance, and performing feature selection. Each of these steps is underpinned by rigorous mathematical formulations and data-driven techniques, ensuring that the processed data is optimally prepared for subsequent model training and evaluation.

2.1.1. Feature Scaling

The dataset under consideration comprises features with heterogeneous scales, which can lead to suboptimal model performance due to the disproportionate influence of certain features on the learning algorithms. This issue is particularly pronounced in algorithms that rely on distance metrics, such as Support Vector Machines (SVM) and k-Nearest Neighbors (k-NN), as well as gradient-based optimization methods used in neural networks. To address this, MinMax scaling is employed, which rescales each feature to a standardized range, typically [0, 1]. The mathematical formulation for MinMax scaling is defined as presented in the equation 1.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

Where (x) denotes the original feature value, (x') represents the scaled feature value, and $(\min(x))$ and $(\max(x))$ correspond to the minimum and maximum values of the feature across the dataset, respectively. The primary advantage of MinMax scaling lies in its preservation of the original data distribution, albeit within a bounded range. This bounded range ensures that features with larger magnitudes do not disproportionately dominate the gradient descent process during model training, which is critical for achieving faster convergence and avoiding local minima. Moreover, MinMax scaling is particularly suitable for algorithms that do not assume a Gaussian distribution of the data, as it does not alter the underlying shape of the feature distribution, unlike standardization (z-score normalization). In this study, after applying MinMax scaling, the rescaled feature matrix (X') is defined as presented in the equation 2.

$$X' = \left[\frac{x_{ij} - \min(X_j)}{\max(X_j) - \min(X_j)}\right] \quad \text{for} \quad i \in \{1, \dots, n\}, \quad j \in \{1, \dots, m\} \tag{2}$$

where (n) is the number of samples, (m) is the number of features, and (x_{ij}) is the original value of the (j)th feature in the (i)th sample.

2.1.2. Handling Imbalanced Data

Class imbalance is a prevalent issue in many real-world datasets, particularly in medical diagnostics, where the occurrence of certain conditions may be rare. In this study, the dataset exhibits a significant imbalance between the non-GDM (majority) class and the GDM (minority) class. If left unaddressed, this imbalance can lead to biased model predictions, where the classifier disproportionately favors the majority class, thereby compromising the detection of GDM cases. To mitigate this, the Synthetic Minority Over-sampling Technique (SMOTE) is employed. SMOTE is an advanced data augmentation method that synthesizes new samples for the minority class, thereby achieving a more balanced class distribution without simply duplicating existing minority class samples. The synthetic sample generation process in SMOTE is mathematically formulated as presented in the equation 3. Given a minority class sample (x_i), a new synthetic sample (x_{new}) is generated as presented in the equation 3.

$$x_{\text{new}} = x_i + \lambda \times (x_j - x_i) \tag{3}$$

where (x_j) is a randomly selected sample from the (k) −nearest neighbors of (x_i), and (λ) is a random scalar drawn from a uniform distribution $(\lambda \sim \mathcal{U}(0,1))$. The parameter (k) typically determines the number of neighbors to consider, and it influences the diversity of the synthetic samples generated. This interpolation process can be visualized as creating a convex combination of the original sample (x_i) and one of its neighbors (x_j), effectively generating a new sample (x_{new}) that lies along the line segment connecting (x_i) and (x_j) in the feature space. The resulting synthetic dataset is more balanced, which helps the machine learning models to learn decision boundaries that are more sensitive to the minority class, thereby improving the detection rate of GDM cases. Formally, the SMOTE-adjusted

training dataset (X_{SMOTE}) and corresponding labels (y_{SMOTE}) are defined as presented in the equation 4.

$$X_{\text{SMOTE}} = \{x'_1, x'_2, \dots, x'_n\} \text{ and } y_{\text{SMOTE}} = \{y'_1, y'_2, \dots, y'_n\} \quad (4)$$

where (x'_i) represents either an original or synthetic sample, and (y'_i) represents the corresponding class label.

2.2.3. Feature Selection

Feature selection is a critical aspect of data preprocessing, particularly in high-dimensional datasets, where irrelevant or redundant features can introduce noise and degrade model performance. In this study, Recursive Feature Elimination (RFE) with cross-validation is employed to systematically identify the most predictive subset of features. RFE operates by recursively training a machine learning model and ranking features based on their importance scores. In the context of linear models, feature importance is typically derived from the absolute values of the model's weight coefficients ($w = (w_1, w_2, \dots, w_m)$). For ensemble models like Random Forests, feature importance is determined by the Gini importance or the decrease in the impurity criterion attributed to each feature. Mathematically, let ($\mathcal{L}(w, X, y)$) denote the loss function minimized during model training, where (X) is the feature matrix and (y) is the label vector. The importance of feature (j) can be defined as presented in the equation 5.

$$\text{Importance}(j) = \sum_{t=1}^{T} \text{Impurity Reduction}_t(f_j) \quad (5)$$

where ($\text{Impurity Reduction}_t(f_j)$) represents the decrease in impurity (e.g., Gini or entropy) at node (t) due to the splitting on feature (f_j) in a decision tree within the ensemble. RFE proceeds by recursively eliminating the least important features and retraining the model on the remaining features. This iterative process continues until the optimal subset of features is identified, as determined by the cross-validated performance metric (e.g., accuracy, F1-score). The final subset of features is expected to maximize the model's predictive power while minimizing overfitting and computational complexity. The final selected feature set (S) is defined as presented in the equation 6.

$$S = \{f_1, f_2, \dots, f_k\} \quad \text{where} \quad k \leq m \quad (6)$$

Where the selected feature set (S) is used to train the models in the subsequent stages of the machine learning pipeline. This approach ensures that the models focus on the most informative features, thereby enhancing their generalization capabilities and reducing the risk of overfitting to the training data.

2.2. Model Selection

The model selection process in this study is a comprehensive and multifaceted endeavor that involves the careful design, evaluation, and optimization of a suite of advanced machine learning models. The process is grounded in rigorous mathematical principles and leverages the complementary strengths of various algorithms to construct a robust and accurate predictive

model. The models selected for this study include individual classifiers, stacking models, and a blended model that employs a voting classifier. Each model is meticulously chosen based on its theoretical foundations, empirical performance, and its ability to contribute uniquely to the ensemble's overall predictive power.

2.2.1. Individual Classifiers

The individual classifiers chosen for this study represent a broad spectrum of machine learning paradigms, each offering distinct advantages in terms of interpretability, scalability, and performance on different types of data. These classifiers include Gradient Boosting, AdaBoost, Support Vector Machine (SVM), Logistic Regression, XGBoost, and LightGBM. Below, each classifier is described in detail, with a focus on the underlying mathematics and the role it plays within the ensemble framework.

2.2.1.1. Gradient Boosting (GBM)

Gradient Boosting is a sequential ensemble technique that constructs an additive model by fitting weak learners (typically decision trees) to the residuals of the predictions made by the ensemble so far. Each weak learner $(h_m(x))$ is trained to minimize the loss function $(L(y, F_m(x)))$, where $(F_m(x))$ is the current ensemble model. The ensemble model is updated iteratively as presented in the equation 7

$$F_{m+1}(x) = F_m(x) + \nu \cdot h_m(x) \quad (7)$$

Where (ν) is the learning rate, a hyperparameter that controls the contribution of each weak learner. The objective is to solve the optimization problem as presented in the equation 8.

$$\arg\min_{h_m} \sum_{i=1}^{N} \left[\frac{\partial L(y_i, F_m(x_i))}{\partial F_m(x_i)}\right] h_m(x_i) \quad (8)$$

Using gradient descent. The iterative nature of Gradient Boosting allows it to focus on correcting the mistakes made by previous models, resulting in a powerful and flexible method that can model complex relationships in the data.

2.2.1.2. AdaBoost (ABC)

AdaBoost, short for Adaptive Boosting, is a boosting algorithm that improves the performance of weak classifiers by focusing on the samples that are difficult to classify. The algorithm begins by assigning equal weights to all training samples. In each iteration, a weak learner $(h_m(x))$ is trained, and the weights of the misclassified samples are increased so that the next learner focuses more on these hard examples. Formally, the weight update rule for the samples is given by the equation 9.

$$w_i^{(m+1)} = w_i^{(m)} \cdot \exp\left(\alpha_m \cdot \mathbb{1}(y_i \neq h_m(x_i))\right) \quad (9)$$

where (α_m) is the weight assigned to the weak learner $(h_m(x))$ based on its accuracy, and $(\mathbb{1}(\cdot))$ is the indicator function. The final model is a weighted combination of the weak learners as presented in the equation 10.

$$\hat{y} = \text{sign}(\sum_{m=1}^{M} \alpha_m h_m(x)) \quad (10)$$

This approach effectively concentrates the model's capacity on the most challenging parts of the data, improving overall classification accuracy, especially in the presence of noise or complex decision boundaries.

2.2.1.3 Support Vector Machine (SVM)

SVM is a supervised learning model that constructs a hyperplane or a set of hyperplanes in a high-dimensional space to separate different classes. The goal is to find the hyperplane that maximizes the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors. The optimization problem for a linear SVM is formulated as presented in the equation 11.

$$\min_{w,b} \quad \frac{1}{2}|w|^2 \quad \text{subject to} \quad y_i(w \cdot x_i + b) \geq 1 \quad \forall i \quad (11)$$

Where (w) is the weight vector, (b) is the bias term, (x_i) are the feature vectors, and (y_i) are the class labels. For non-linearly separable data, the kernel trick is used to map the data into a higher-dimensional space where a linear separator can be found. The kernel function $(K(x_i, x_j))$ computes the inner product in this transformed space, allowing SVM to handle complex, non-linear decision boundaries.

2.2.1.4. Logistic Regression (LGR)

Logistic Regression is a linear model for binary classification that estimates the probability of the target variable belonging to a particular class using the logistic function. The model predicts the probability that a given input (x) belongs to the positive class as presented in the equation 12.

$$P(y = 1|x) = \frac{1}{1+e^{-(w \cdot x+b)}} \quad (12)$$

The model is trained by maximizing the likelihood of the observed data, which is equivalent to minimizing the cross-entropy loss as presented in the equation 14.

$$L(w, b) = -\sum_{i=1}^{N}[y_i \log(P(y_i = 1|x_i)) + (1 - y_i) \log(1 - P(y_i = 1|x_i))] \quad (13)$$

Logistic Regression is particularly effective when the classes are linearly separable and provides a probabilistic interpretation of the predictions, which can be useful for decision-making in various applications.

2.2.1.5. XGBoost (XGB)

XGBoost is an advanced implementation of gradient boosting that incorporates several innovations to enhance efficiency and accuracy. It introduces regularization into the objective function to control the complexity of the model and prevent overfitting. The objective function for XGBoost is presented in the equation 14.

Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)
Volume 9, Nomor 2, Agustus 2024, pp 1117-1132
Terakreditasi Nomor 204/E/KPT/2022 | ISSN: 2527-5771/EISSN: 2549-7839
https://tunasbangsa.ac.id/ejurnal/index.php/jurasik

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (14)$$

Where $(l(y_i, \hat{y}_i))$ is the loss function (e.g., logistic loss for binary classification), $(\hat{y}_i)$ is the predicted value, and $(\Omega(f_k))$ is the regularization term that penalizes the complexity of the model, which is often defined as presented in the equation 15.

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{d} w_j^2 \quad (15)$$

Here, (T) is the number of leaves in the tree, (λ) is the L2 regularization term on the weights, and (w_j) are the weights associated with the leaf nodes. XGBoost also employs second-order gradient optimization, which uses both the first and second derivatives of the loss function to guide the updates, making the training process more robust and faster.

2.2.1.6. LightGBM (LGB)

LightGBM is a gradient boosting framework that is designed to be highly efficient, both in terms of computation and memory usage. It introduces several key techniques, including Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS selectively retains instances with large gradients, which are more informative, and applies a small weight to instances with small gradients, reducing the computational burden without sacrificing accuracy. The objective function for LightGBM is similar to that of XGBoost but with a focus on efficiency as presented in the equation 16.

$$\mathcal{L}(\theta) = \sum_{i=1}^{N} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (16)$$

where the loss function and regularization terms are defined similarly to XGBoost. LightGBM grows trees leaf-wise rather than level-wise, which allows it to focus on the most significant splits, leading to better accuracy and faster convergence, particularly on large datasets with high-dimensional feature spaces.

2.2.2. Stacking Models

To further enhance the predictive power of the individual classifiers, this study employs advanced stacking models. Stacking is an ensemble learning technique where multiple base models (level-0 models) are trained on the same dataset, and their predictions are used as inputs to a meta-model (level-1 model). The meta-model learns to combine the predictions of the base models in a way that optimally reduces the overall prediction error.

2.2.2.1. Stacking Model 1

The first stacking model in this study combines the Random Forest, Gradient Boosting, and SVM classifiers as base models. The predictions from these base models are then used as features to train an XGBoost meta-model. The stacking process can be expressed mathematically as presented in the equation 17.

$$\widehat{y_{\text{stack1}}} = f_{\text{meta1}}\left(f_{\text{rfc}}(x), f_{\text{gbc}}(x), f_{\text{svc}}(x)\right) \quad (17)$$

where $(f_{\text{rfc}}(x))$, $(f_{\text{gbc}}(x))$, and $(f_{\text{svc}}(x))$ are the outputs of the Random Forest, Gradient Boosting, and SVM base models, respectively, and $(f_{\text{meta1}}(x))$ is the prediction of the XGBoost meta-model. The XGBoost meta-model effectively learns the relationships between the predictions of the base models and the true labels, making it well-suited for combining diverse models that capture different aspects of the data.

2.2.2.2. Stacking Model 2

The second stacking model employs Extra Trees, AdaBoost, and Logistic Regression as base models, with LightGBM serving as the meta-model. The model architecture is described by in the equation 18.

$$\widehat{y_{\text{stack2}}} = f_{\text{meta2}}\left(f_{\text{etc}}(x), f_{\text{abc}}(x), f_{\text{lgr}}(x)\right) \quad (18)$$

where $(f_{\text{etc}}(x))$, $(f_{\text{abc}}(x))$, and $(f_{\text{lgr}}(x))$ represent the predictions of the Extra Trees, AdaBoost, and Logistic Regression base models, respectively, and $(f_{\text{meta2}}(x))$ is the prediction of the LightGBM meta-model. This setup leverages the strengths of both boosting and bagging techniques, as well as the linearity of Logistic Regression, providing a balanced approach that captures both linear and non-linear patterns in the data. The stacking models are trained using cross-validated predictions from the base models to avoid overfitting, ensuring that the meta-model generalizes well to unseen data. The combined use of tree-based models, boosting algorithms, and linear classifiers in a stacking framework allows for the capture of complex interactions among features, leading to a more accurate and robust predictive model.

2.2.3. Blended Model

The final step in the model selection process involves blending the predictions of the stacking models with those of the individual classifiers, particularly XGBoost and LightGBM, using a VotingClassifier. The VotingClassifier aggregates the predictions from each model using a soft voting mechanism, where the final prediction is based on the weighted average of the predicted probabilities from each model. This approach is mathematically formulated as presented in the equation 19.

$$\widehat{y_{\text{blend}}} = \arg\max_k\left(\sum_{i=1}^{n} w_i \cdot P(y = k|x, f_i)\right) \quad (19)$$

Where (w_i) represents the weight assigned to the (i)th model, $(P(y = k|x, f_i))$ denotes the predicted probability for class (k) from the (i)th model, and (n) is the total number of models. The weights (w_i) can be tuned based on the cross-validated performance of each model, allowing the ensemble to place greater emphasis on models that perform better on specific aspects of the data. The blended model capitalizes on the diversity of the individual classifiers and the stacking models, combining their strengths to produce a highly accurate and

generalizable model. The use of soft voting allows the ensemble to make probabilistic predictions that are more nuanced and reflective of the underlying data distribution, rather than relying solely on hard classification decisions. This approach is particularly advantageous in cases where different models excel in different regions of the feature space, as it allows the final prediction to be a consensus that accounts for the varying strengths of each model.

2.3. Evaluation Metrics

Evaluating the performance of machine learning models is a critical aspect of ensuring that the models are not only accurate but also reliable and robust across different datasets and scenarios. In this study, a diverse set of evaluation metrics is utilized to capture various dimensions of model performance, providing a thorough and comprehensive assessment. These metrics include accuracy, precision, recall, F1-score, confusion matrix analysis, and stratified k-fold cross-validation, among others. Each metric is carefully chosen to address the specific challenges posed by the dataset, such as class imbalance, and to ensure that the model's predictions are both correct and meaningful in a real-world context. Accuracy is the most fundamental and widely used metric in classification tasks. It is defined as the proportion of correct predictions—both true positives and true negatives—out of the total number of predictions made. The mathematical formulation for accuracy is given in the equation 20.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (20)$$

Where (TP) (True Positives) represents the number of instances correctly predicted as belonging to the positive class, (TN) (True Negatives) represents the number of instances correctly predicted as belonging to the negative class, (FP) (False Positives) represents the number of instances incorrectly predicted as belonging to the positive class (Type I error), and (FN) (False Negatives) represents the number of instances incorrectly predicted as belonging to the negative class (Type II error). While accuracy provides a straightforward measure of a model's overall performance, it has limitations, especially in the context of imbalanced datasets. In scenarios where one class significantly outnumbers the other, accuracy can be misleading. For example, if a dataset contains 95% negative instances and 5% positive instances, a model that predicts every instance as negative will achieve 95% accuracy, yet fail entirely to identify positive cases. This underscores the need for additional metrics that can provide a more nuanced evaluation of model performance.

To address the limitations of accuracy, particularly in imbalanced datasets, precision, recall, and F1-score are employed. These metrics focus on the model's ability to correctly identify positive instances and manage the trade-offs between different types of classification errors. Precision, also known as the positive predictive value, measures the proportion of true positives among all instances predicted as positive. It is mathematically defined as presented in the equation 21.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (21)$$

Precision is particularly important in scenarios where the cost of false positives is high. For instance, in medical diagnostics, a false positive might lead to unnecessary treatments or further invasive tests, making precision a crucial metric. A model with high precision has a low rate of false positives, indicating that when it predicts a positive outcome, it is likely to be correct. Recall, also referred to as sensitivity or the true positive rate, measures the proportion of true positives among all actual positive instances. It is defined as presented in the equation 22.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (22)$$

Recall is critical in contexts where missing a positive case (false negative) would have severe consequences. In medical diagnostics, for example, failing to identify a disease in an affected patient could result in a lack of treatment, making recall a vital metric. A model with high recall successfully identifies most of the positive instances, but this can sometimes come at the expense of precision, particularly if the model also predicts many false positives. The F1-score is the harmonic mean of precision and recall, combining these two metrics into a single score that balances the trade-off between precision and recall. It is defined as presented in the equation 23.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (23)$$

The F1-score is especially useful when dealing with imbalanced datasets, as it provides a balanced measure that reflects both the accuracy of positive predictions and the ability to capture all positive instances. Unlike the arithmetic mean, the harmonic mean ensures that both precision and recall must be reasonably high for the F1-score to be high. This makes the F1-score a preferred metric in cases where the cost of false positives and false negatives is significant.

3. RESULT AND DISCUSSION

As presented in the table 1, the performance of various machine learning models was assessed based on their ability to accurately classify instances within the dataset. The models considered in this study include Gradient Boosting, AdaBoost, Support Vector Machine (SVM), Logistic Regression, XGBoost, LightGBM, and a Blended Model combining multiple classifiers. The results provide valuable insights into each model's effectiveness, especially in a medical diagnostic context where precision and recall are of utmost importance. The Gradient Boosting model achieved a cross-validation accuracy of 86.68% and a test accuracy of 82.27%. This model exhibited a high precision of 0.90 for the negative class and a recall of 0.87. However, for the positive class, the precision was 0.60 with a recall of 0.65, resulting in an F1-score of 0.63. The model's ability to maintain a balance between precision and recall suggests that it is a reliable option when both types of errors, false positives and false negatives, need to be minimized. The iterative nature of Gradient Boosting, which focuses on correcting errors made by previous models,

helps in reducing the number of false negatives, which is crucial in medical diagnostics.

AdaBoost attained a cross-validation accuracy of 84.58% and a test accuracy of 79.80%. The model's precision for the negative class was 0.84, with a recall of 0.88. For the positive class, the precision was 0.66, with a recall of 0.58, leading to an F1-score of 0.62. While AdaBoost shows reasonable performance, particularly in terms of precision for the positive class, its lower recall indicates a struggle to correctly identify all positive instances. This may be attributed to AdaBoost's sensitivity to noise and outliers, which can lead to overfitting on certain parts of the data, especially when dealing with imbalanced datasets. The SVM model demonstrated a cross-validation accuracy of 81.31% and a test accuracy of 79.80%. The model achieved a precision of 0.85 for the negative class and a recall of 0.88. In the positive class, the precision was 0.64, with a recall of 0.58, resulting in an F1-score of 0.61. SVM is well-known for its effectiveness in maximizing the margin between classes, which is evident in its high precision. However, the relatively lower recall for the positive class suggests that SVM may not fully capture the minority class instances, a common challenge in imbalanced datasets. Implementing class balancing techniques or using different kernel functions could potentially enhance SVM's performance in such contexts.

Logistic Regression yielded a cross-validation accuracy of 81.46% and a test accuracy of 77.34%. The model's precision for the negative class was 0.82, with a recall of 0.87, while the positive class had a precision of 0.64 and a recall of 0.53, resulting in an F1-score of 0.58. Being a linear model, Logistic Regression tends to underperform on complex datasets where non-linear relationships are significant. The lower recall for the positive class indicates that the model misses a substantial number of positive instances, which is a critical limitation in applications such as medical diagnostics. These results suggest that more sophisticated models may be necessary to capture the intricacies of the data effectively. XGBoost achieved a cross-validation accuracy of 88.47% and a test accuracy of 80.79%. The model exhibited a precision of 0.88 for the negative class, with a recall of 0.87. For the positive class, the precision was 0.60, with a recall of 0.61, resulting in an F1-score of 0.61. XGBoost's performance highlights its robustness in handling complex datasets, thanks to its ability to model intricate patterns through boosted decision trees. However, the model's disparity in precision and recall between classes suggests that, while effective, it may require further tuning or the incorporation of ensemble techniques to enhance its performance in imbalanced datasets.

LightGBM produced a cross-validation accuracy of 89.10% and a test accuracy of 81.77%. The model's precision for the negative class was 0.90, with a recall of 0.87, while the positive class had a precision of 0.58 and a recall of 0.64, resulting in an F1-score of 0.61. LightGBM's efficiency in handling large datasets and its capability to model interactions among features effectively make it a strong candidate for scenarios where processing speed and memory efficiency are critical. However, as with the other models, improving recall for the positive class remains a challenge that needs addressing to ensure comprehensive model performance. The Blended Model, combining the strengths of multiple classifiers, achieved a

cross-validation accuracy of 90.19% and a test accuracy of 79.80%. The model exhibited a precision of 0.88 for the negative class and a recall of 0.85. For the positive class, the precision was 0.54, with a recall of 0.60, leading to an F1-score of 0.57. The blended approach illustrates the potential of ensemble methods to capture diverse patterns in the data by leveraging the strengths of different models. However, the slightly lower recall and F1-score for the positive class suggest that further optimization is necessary to enhance the model's ability to accurately identify positive instances.

The results of this study reveal significant differences in the models' ability to handle class imbalance, particularly in identifying positive instances. Gradient Boosting and LightGBM demonstrated high cross-validation accuracy, indicating strong overall performance, but their lower recall for the positive class underscores the common challenge in medical diagnostics: balancing precision and recall to avoid missing critical cases. XGBoost and LightGBM, though slightly lower in overall accuracy, showed a better balance between precision and recall, making them more suitable for scenarios where minimizing both types of errors (false positives and false negatives) is crucial. LightGBM's efficiency and strong performance make it a promising option, particularly when computational resources are a consideration. The relatively lower performance of Logistic Regression and SVM underscores the limitations of linear models and the importance of non-linear techniques in capturing complex relationships in medical data. These results suggest that, while simpler models may offer some advantages in terms of interpretability, more sophisticated methods are necessary to achieve higher accuracy and reliability in critical applications.

Tabel 1. Comparative of Machine Learning Models

Model	Cross-Validation Accuracy	Test Accuracy	Precision (Class 0)	Recall (Class 0)	F1-Score (Class 0)	Precision (Class 1)	Recall (Class 1)	F1-Score (Class 1)
Gradient Boosting	86.68%	82.27%	0.90	0.87	0.88	0.60	0.65	0.63
AdaBoost	84.58%	79.80%	0.84	0.88	0.86	0.66	0.58	0.62
Support Vector Machine	81.31%	79.80%	0.85	0.88	0.86	0.64	0.58	0.61
Logistic Regression	81.46%	77.34%	0.82	0.87	0.84	0.64	0.53	0.58
XGBoost	88.47%	80.79%	0.88	0.87	0.87	0.60	0.61	0.61
LightGBM	89.10%	81.77%	0.90	0.87	0.88	0.58	0.64	0.61
Blended Model	90.19%	79.80%	0.88	0.85	0.87	0.54	0.60	0.57

4. CONCLUSION

This study evaluated the performance of several machine learning models, including Gradient Boosting, AdaBoost, Support Vector Machine (SVM), Logistic Regression, XGBoost, LightGBM, and a Blended Model, in the context of medical diagnostics. The goal was to determine which model or combination of models could provide the most accurate and reliable predictions, particularly in scenarios

where the accurate identification of positive cases is critical. The results indicate that ensemble methods, such as Gradient Boosting, XGBoost, LightGBM, and the Blended Model, generally outperformed simpler models like Logistic Regression and SVM. Among the models tested, LightGBM showed the highest cross-validation accuracy (89.10%) and performed well in test accuracy (81.77%), demonstrating its capability to effectively handle large datasets and model complex interactions between features. XGBoost also exhibited strong performance, with a cross-validation accuracy of 88.47% and a test accuracy of 80.79%, highlighting its robustness and efficiency in classification tasks.

However, a consistent challenge across all models was the imbalance between precision and recall for the positive class, indicating the need for further optimization. Models like Gradient Boosting and LightGBM, while exhibiting high overall accuracy, still showed room for improvement in accurately identifying positive cases without sacrificing precision. The Blended Model, which combined the strengths of multiple classifiers, demonstrated the potential of ensemble approaches in enhancing model performance. Although it achieved the highest cross-validation accuracy (90.19%), its slightly lower test accuracy and F1-score for the positive class suggest that additional tuning is needed to improve its ability to detect positive instances effectively. In conclusion, the study underscores the importance of using advanced ensemble methods and careful evaluation metrics to develop reliable machine learning models for medical diagnostics. While LightGBM and XGBoost emerged as strong contenders, achieving a balance between precision and recall remains crucial for ensuring that critical cases are not overlooked. Future work should focus on refining these models, particularly in improving their sensitivity to the positive class, to further enhance their utility in real-world diagnostic applications.

DAFTAR PUSTAKA

[1] A. A. Choudhury and V. D. Rajeswari, “Gestational diabetes mellitus-A metabolic and reproductive disorder,” Biomed. \& Pharmacother., vol. 143, p. 112183, 2021.

[2] A. Ornoy, M. Becker, L. Weinstein-Fudim, and Z. Ergaz, “Diabetes during pregnancy: a maternal disease complicating the course of pregnancy with long-term deleterious effects on the offspring. a clinical review,” Int. J. Mol. Sci., vol. 22, no. 6, p. 2965, 2021.

[3] R. A. Shqara, Y. N. Francis, S. Or, L. Lowenstein, and M. F. Wolf, “Obstetrical Outcome following Diagnosis of Gestational Diabetes in the Third Trimester (> 29 Weeks) versus Second Trimester (24--28 Weeks): A Retrospective Comparative Study,” Am. J. Perinatol., 2022.

[4] A. Preda et al., “Analysis of maternal and neonatal complications in a group of patients with gestational diabetes mellitus,” Medicina (B. Aires)., vol. 57, no. 11, p. 1170, 2021.

[5] A. Thakur, S. Agrawal, S. Chakole, and B. Wandile, “A Critical Review of Diagnostic Strategies and Maternal Offspring Complications in Gestational Diabetes Mellitus,” Cureus, vol. 15, no. 12, 2023.

[6] M. Zahmatkeshan, S. Zakerabasali, M. Farjam, Y. Gholampour, M. Seraji, and A. Yazdani, “The use of mobile health interventions for gestational diabetes mellitus: a descriptive literature review,” J. Med. Life, vol. 14, no. 2, p. 131, 2021.

[7] M. Javaid, A. Haleem, R. P. Singh, R. Suman, and S. Rab, "Significance of machine learning in healthcare: Features, pillars and applications," Int. J. Intell. Networks, vol. 3, pp. 58–73, 2022.

[8] P. Khan et al., "Machine learning and deep learning approaches for brain disease diagnosis: principles and recent advances," Ieee Access, vol. 9, pp. 37622–37655, 2021.

[9] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," SN Comput. Sci., vol. 2, no. 6, p. 420, 2021.

[10] M. Shehab et al., "Machine learning in medical applications: A review of state-of-the-art methods," Comput. Biol. Med., vol. 145, p. 105458, 2022.

[11] N. Wang et al., "Development and validation of risk prediction models for large for gestational age infants using logistic regression and two machine learning algorithms," J. Diabetes, vol. 15, no. 4, pp. 338–348, 2023.

[12] P. Gyasi-Antwi et al., "Global prevalence of gestational diabetes mellitus: a systematic review and meta-analysis," New Am. J. Med., vol. 1, no. 3, pp. 1–10, 2020.

[13] S. A. Samadi et al., "Screening children for Autism Spectrum Disorders in low-and middle-income countries: Experiences from the Kurdistan region of Iraq," Int. J. Environ. Res. Public Health, vol. 19, no. 8, p. 4581, 2022.

[14] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: a survey of case studies," ACM Comput. Surv., vol. 55, no. 6, pp. 1–29, 2022.

[15] H. Lu, L. Ehwerhemuepha, and C. Rakovski, "A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance," BMC Med. Res. Methodol., vol. 22, no. 1, p. 181, 2022.

[16] F. Arshad, S. Ahmed, A. Amjad, and M. Kabir, "An explainable stacking-based approach for accelerating the prediction of antidiabetic peptides," Anal. Biochem., vol. 691, p. 115546, 2024.

[17] M. R. Indupalli and others, "A Hybrid Blended Stacking Disease Prediction System Based on Symptoms," 2023.

[18] S. Ramasamy, H. C. Kantharaju, N. B. Madhavi, and M. P. Haripriya, "8 Meta-learning through ensemble approach: bagging, boosting, and random forest strategies," Towar. Artif. Gen. Intell. Deep Learn. Neural Networks, Gener. AI, p. 167, 2023.

[19] Y. Gao, H. Hasegawa, Y. Yamaguchi, and H. Shimada, "Gradient Boosting Decision Tree Ensemble Learning for Malware Binary Classification," 2020.