Scientific Research Publishing

Journal of Data Analysis and Information Processing, 2023, 11, 480-511
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

A Hybrid Ensemble Learning Approach Utilizing Light Gradient Boosting Machine and Category Boosting Model for Lifestyle-Based Prediction of Type-II Diabetes Mellitus

Mahadi Nagassou[1], Ronald Waweru Mwangi[2], Euna Nyarige[3]

[1]Institute for Basic Sciences, Technology, and Innovation, Department of Mathematics, Pan African University, Nairobi, Kenya
[2]School of Computing and Information Technology, Department of Computing, Jomo Kenyatta University of Agriculture and Technology, Nairobi, Kenya
[3]Department of Mathematics, Statistics and Actuarial Sciences, Machakos University, Nairobi, Kenya
Email: nagassou.mahadi@students.jkuat.ac.ke, waweru_mwangi@icsit.jkuat.ac.ke, nyarige@mksu.ac.ke

How to cite this paper: Nagassou, M., Mwangi, R.W. and Nyarige, E. (2023) A Hybrid Ensemble Learning Approach Utilizing Light Gradient Boosting Machine and Category Boosting Model for Lifestyle-Based Prediction of Type-II Diabetes Mellitus. *Journal of Data Analysis and Information Processing*, **11**, 480-511.
https://doi.org/10.4236/jdaip.2023.114025

Received: October 14, 2023
Accepted: November 24, 2023
Published: November 27, 2023

Abstract

Addressing classification and prediction challenges, tree ensemble models have gained significant importance. Boosting ensemble techniques are commonly employed for forecasting Type-II diabetes mellitus. Light Gradient Boosting Machine (LightGBM) is a widely used algorithm known for its leaf growth strategy, loss reduction, and enhanced training precision. However, LightGBM is prone to overfitting. In contrast, CatBoost utilizes balanced base predictors known as decision tables, which mitigate overfitting risks and significantly improve testing time efficiency. CatBoost's algorithm structure counteracts gradient boosting biases and incorporates an overfitting detector to stop training early. This study focuses on developing a hybrid model that combines LightGBM and CatBoost to minimize overfitting and improve accuracy by reducing variance. For the purpose of finding the best hyperparameters to use with the underlying learners, the Bayesian hyperparameter optimization method is used. By fine-tuning the regularization parameter values, the hybrid model effectively reduces variance (overfitting). Comparative evaluation against LightGBM, CatBoost, XGBoost, Decision Tree, Random Forest, AdaBoost, and GBM algorithms demonstrates that the hybrid model has the best F1-score (99.37%), recall (99.25%), and accuracy (99.37%). Consequently, the proposed framework holds promise for early diabetes prediction in the healthcare industry and exhibits potential applicability to other datasets sharing similarities with diabetes.

Keywords

Boosting Ensemble Learning, Category Boosting, Light Gradient Boosting Machine

1. Introduction

Type-II diabetes mellitus (T2DM) represents a formidable global health challenge. This chronic metabolic disorder is characterized by high blood glucose levels, resulting from a combination of insulin resistance and inadequate insulin production. The condition is escalating at an alarming rate worldwide, presenting a severe public health crisis due to its long-term complications, including cardiovascular disease, kidney damage, and vision loss, among others. The increasing prevalence and consequential impact of T2DM on global health underscore the urgency for accurate and early prediction models, which are pivotal for preventive measures, timely interventions, and resource allocation in healthcare systems [1].

Identifying individuals at high risk for T2DM has traditionally relied on the assessment of various lifestyle and physiological indicators. Factors such as dietary habits, physical activity levels, and anthropometric measurements play a significant role in determining an individual's risk profile. However, the predictive challenges of T2DM are multifaceted, owing to the complex interplay of these risk factors, necessitating more sophisticated analytical methods capable of capturing the nuanced relationships inherent in patient data.

In this context, machine learning (ML) techniques have emerged as a revolutionary tool in predictive healthcare, offering nuanced insights drawn from large-scale datasets. Tree ensemble models, particularly boosting algorithms, have garnered considerable interest for their superior performance in classification tasks. These algorithms work by iteratively refining weak learners, thereby establishing robust models that can navigate the intricate patterns associated with T2DM risk factors [2].

Specifically, LightGBM and CatBoost, two state-of-the-art boosting algorithms, have marked a significant advancement in this domain. LightGBM optimizes the traditional gradient boosting framework by employing a unique leaf-wise growth strategy, offering an efficient and highly precise model [3]. Despite its benefits, LightGBM can succumb to overfitting, particularly with complex datasets, limiting its practical applicability. Conversely, CatBoost addresses some of these limitations by integrating an advanced system of balanced decision tables and an intrinsic overfitting detector, enhancing model reliability and execution efficiency [4]. Nevertheless, CatBoost requires careful hyperparameter tuning to ensure optimal performance, presenting challenges in model optimization.

Given the individual strengths and limitations of LightGBM and CatBoost, this study introduces a novel hybrid model, synergizing the capabilities of both

algorithms to enhance the accuracy of T2DM predictions. Our approach is designed to consolidate the benefits of both algorithms, mitigating the risk of overfitting by harnessing the strengths of each component while compensating for their respective weaknesses. The model's robustness is further reinforced through Bayesian optimization, a sophisticated hyperparameter tuning technique ensuring optimal performance [5].

This paper details the development and evaluation of our innovative hybrid predictive model, contextualizing its performance against established ML classifiers in T2DM prediction, such as XGBoost, Decision Tree, Random Forest, AdaBoost, and GBM. We adopt a comprehensive set of metrics, including accuracy, precision, recall, F1-score, and log loss, to provide a holistic assessment of model performance.

By enhancing the predictive accuracy of T2DM, our hybrid model serves as a catalyst for proactive healthcare strategies, facilitating early interventions and personalized treatment plans. This advancement not only promises to transform patient outcomes but also contributes significantly to the broader application of ML in predictive healthcare. The ensuing sections of the paper are organized as follows: Section 2 reviews the relevant literature, Section 3 describes the research methodology, Section 4 discusses the findings, and Section 5 concludes the study with insights and implications for future research.

2. Review of Literature

2.1. Methods of Prediction

This section presents previous research conducted on the prediction and detection of Type II Diabetes Mellitus (TIIDM) using machine learning and ensemble learning techniques. The researchers discussed the algorithms, datasets, and methodologies employed in their studies. Experimental methods used in recent scientific studies have shown how important lifestyle, demographic, psycho-social, and genetic risk factors are in the early detection, prevention, and management of diabetes, especially type 2 diabetes [6]-[11].

Zhang L *et al.* [12] developed a framework for TIIDM utilizing machine learning and ensemble learning methods, including Logistic Regression (LR), Classification and Regression Technique (CART), Artificial Neural Network (ANN), Support Vector Machine (SVM), Random Forest (RF), and Gradient Boosting Machine (GBM). They analyzed 36,652 cases and 10 different lifestyle factors from a rural Henan cohort in China. When compared to other classifiers, GBM performs the best.

Ganie SM *et al.* [13] proposed a TIIDM prediction model using machine learning techniques. Their dataset consisted of 1939 records with 11 biological and lifestyle parameters. Various machine learning algorithms such as Bagged Decision Trees, Random Forest, Extra Trees, AdaBoost, Stochastic Gradient Boosting, and Voting (Logistic Regression, Decision Trees, Support Vector Machine) were employed. The greatest rate of accuracy among these classifiers was 99.14%,

which was achieved by Bagged Decision Trees. Kopitar L *et al.* [8] implemented a machine learning system for Type I and Type II Diabetes Mellitus that employs an ensemble learning technique to track glucose levels based on independent features. They used data from 27,050 cases and 111 attributes gathered from patients at 10 different Slovenian healthcare facilities that focused on preventative medicine. For this framework, 59 variables were selected after preprocessing and feature engineering. When compared to other classifiers, LightGBM achieved better results across the board. This included better accuracy, precision, recall, AUC, AUPRC, and RMSE.

Ahmed S *et al.* [9] proposed a machine learning model for the prediction of cardiovascular disease using self-augmented datasets of heart patients and various machine learning models. CatBoost outperformed other models, achieving an accuracy of 87.93%, followed by LightGBM (86.21%), HGBC (84.48%), and XGBOOST (83.78%).

Using a variety of machine-learning classifiers such k-nearest neighbors, decision trees, AdaBoost, naive Bayes, XGBoost, and multi-layer perceptrons, Hasan MK *et al.* [14] created a solid framework for TIIDM. They used EDA to do tasks including outlier detection, missing value completion, data standardization, feature selection, and result validation. With a sensitivity of 0.789, a specificity of 0.934, a false omission rate of 0.092, a diagnostic odds ratio of 66.234, and an AUC of 0.950, the ensembling classifiers AdaBoost and XGBoost performed the best.

Rawat V *et al.* [11] used five machine learning methods for predicting and analyzing patients with diabetes mellitus: AdaBoost, Logic Boost, Robust Boost, Naive Bayes, and Bagging. The PIMA Indian Diabetes Dataset was used, which was found in the UCI machine learning library. Bagging and AdaBoost techniques yielded 79.69 and 81.77 percent accuracy in classification, respectively.

As can be seen from the aforementioned body of work, investigating lifestyle and biological data can aid in the early detection of Type II Diabetes Mellitus. With this method, doctors will be able to make more informed judgments about diabetes treatment in real time, which could decrease the need for hospital readmissions, clinical laboratory visits, and the overall cost of health checks. Moreover, such a system can benefit patients and individuals at risk by enabling early prediction and delaying the onset of the disease. Unawareness and under-resourced healthcare systems have resulted in a considerable number of individuals, approximately 232 million [15], being unaware of their diabetes status. Providing technological assistance to the general population can significantly address this issue.

2.2. Boosting Ensemble Learning

Ensemble learning is an efficient approach that uses various base learners to boost prediction and classification accuracy [16]. Each base learner, which produces a model from a collection of labeled inputs, contributes to the overall pre-

diction by considering different training sets and feature sets. The key concept behind ensemble learning is that errors made by individual base learners can be compensated for by the collective knowledge of the ensemble [17]. The overall objective of the learning process is to improve the effectiveness of the weak learners [18]. This is accomplished by compiling the results of the predictions made by each of the separate models, which can be done either by combining the results or through vote. In addition to this, every model in the ensemble is an improved and adapted version of the one that came before it, assigning more weight to misclassified samples in subsequent estimations [19]. Notably, there are a number of different boosting approaches that have developed over the course of the years in order to improve performance , including AdaBoost, Gradient Boosting Machine (GBM),Light Gradient Boosting Machine (LightGBM), Extreme Gradient Boosting (XGBoost), and Category Boosting (CatBoost), particularly suited for handling categorical data.

Freund and Schapire first presented AdaBoost in 1996, and since then, it has garnered substantial reputation in data mining and machine learning fields [20]. In AdaBoost, the base learner is trained using a training set, and the sample distribution is adjusted based on the performance of the base learner [21]. The algorithm assigns more attention to incorrectly predicted samples during subsequent training iterations. However, AdaBoost is prone to overfitting and underfitting, leading to poor performance on unseen data [22]. To address these limitations, researchers have proposed variations of AdaBoost, such as the AdaBoost-support vector regression model, which has shown improved performance in various prediction tasks [23].

GBM is an optimization technique that minimizes the loss function by iteratively adding weak learners or decision trees [24]. The objective is to create base learners that correlate most effectively with the negative gradient of the loss function when combined with the full ensemble [17]. Setting the number of trees in GBM is crucial, as choosing too many may lead to overfitting and too few may result in underfitting. To mitigate overfitting, stochastic gradient boosting techniques have been introduced, where trees are trained using small subsets of the original dataset [17]. The effectiveness of GBM has been demonstrated in various applications, such as protein solubility prediction [25].

LightGBM is a decision tree-based, fast gradient boosting approach, it offers improved computational efficiency and accuracy [26] [27]. Using exclusive functional grouping and histogram-based techniques, it eliminates occurrences with minor gradients and concentrates on those with big gradients to calculate information gain and decrease feature dimension [28]. The adoption of a tree leaf-wise strategy, with a maximum depth limit, further improves LightGBM's effectiveness [28] [29]. These strategies, along with others, contribute to LightGBM's superior computational efficiency and accuracy compared to other algorithms.

Figure 1 displays an overview of the structure of the leaf-wise strategy in LightGBM.

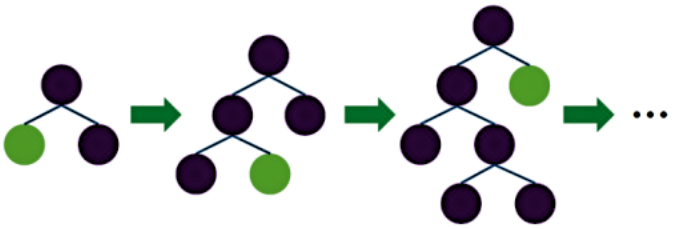

Figure 1. Leaf-wise growth.

Cheng W *et al.* [30] introduced the use of LightGBM in combination with a closed recurrent unit to predict weekday traffic congestion. The objective of their study was to built a model that could effectively capture and express features limited by traditional approaches. When compared to previous algorithms, the suggested model performed admirably and accurately predicted traffic congestion patterns.

In another application, using time series data, Hao X *et al.* [31] was able to accurately forecast the amount of free calcium oxide present in cement clinker by utilizing LightGBM in conjunction with Bayesian optimization. Bayesian optimization was used by the researchers to seek for optimal values for the hyperparameters and fine-tune the model, resulting in improved performance accuracy. Hyperparameter optimization methods, such as Bayesian optimization, are particularly valuable for algorithms that require extensive tuning to achieve optimal results. By incorporating these studies, we not only highlight the versatility of LightGBM in different domains but also emphasize its effectiveness in improving prediction accuracy and performance compared to alternative algorithms.

CatBoost, as highlighted by Shahriar SA *et al. [32]*, is a powerful gradient boosting package specifically designed to handle categorical data.It makes use of a refined version of the gradient boosting decision tree (GBDT) method, which is able to successfully deal with issues including noisy data, diverse feature sets, and complex dependencies. This algorithm has proven to be adept at handling categorical features [4]. Traditionally, categorical features are replaced by corresponding average label values when using the standard GBDT technique. However, CatBoost takes a different approach by utilizing oblivious trees as base predictors. In oblivious trees, the same splitting criterion is applied across an entire level of the tree [33] [34]. This results in balanced trees that are less prone to overfitting.

Gradient boosted oblivious trees have demonstrated their effectiveness in various learning tasks, as demonstrated by Gulin A *et al.* [35] [36]. Each leaf index in CatBoost can be represented as a binary vector of length 2. This vector's length is proportional to the tree's depth. Model predictions in CatBoost are computed using binary features, which are generated by first binarizing all float features, statistics, and one-hot encoded features [37]. **Figure 2** provides a visual representation of the symmetric or oblivious tree strategy employed by CatBoost.

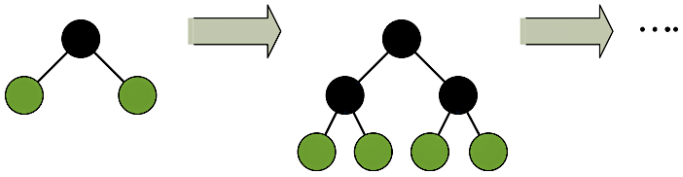

Figure 2. Level-wise tree growth in CatBoost.

Sibindi R *et al.* [38] proposed a boosting ensemble learning approach that combined the power of Light Gradient Boosting Machine (LightGBM) and Extreme Gradient Boosting (XGBoost) for predicting house prices. The hybrid model demonstrated superior performance with lower mean squared error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE) compared to individual baseline machine learning algorithms. However, it should be noted that the hybrid model's larger dataset, larger number of hyperparameters, and larger number of out-of-fold predictions led to longer computation times.

In a comparative study by Dorogush AV *et al.* [38], CatBoost, XGBoost, and LightGBM were evaluated. The results showed that CatBoost outperformed the other models in terms of computational efficiency, scoring around 25 times faster than XGBoost and approximately 60 times faster than LightGBM. Furthermore, among various models such as M5Tree, Random Forest (RF), XGBoost, CatBoost, and Support Vector Machines (SVM), CatBoost demonstrated satisfactory generalization capability and high computational efficiency [33].

Patel *et al.* [39] employed CatBoost, XGBoost, and LightGBM for predicting suicidal ideation in post-stroke patients. The objective of their study was to evaluate the efficiency of these boosting methods in predicting suicidal ideation based on clinical and psychological features. The results indicated that LightGBM had the least favorable performance, while XGBoost showed the best performance in terms of specificity, positive predictive value (PPV), and accuracy. On the other hand, CatBoost exhibited the best performance in terms of sensitivity, negative predictive value (NPV), and area under the curve (AUC).

3. Materials and Methods

3.1. Baseline Methods

The efficacy of the suggested hybrid LightGBM and CatBoost model in its application was validated by implementing various boosting techniques, including AdaBoost, GBM, XGBoost, Decision Tree, Random Forest, LightGBM, and CatBoost reinforcement models. This section provides an overview of the boosting techniques employed.

3.1.1. Adaptive Boosting

AdaBoost is a technique that takes multiple weak classifiers and combines them into one robust classifier.

Input: set of weak classifiers $\left\{\phi_{\mu}(x): \mu = 1, \cdots, M\right\}$. Labelled data

$\chi = \left\{\left(x^i, y^i\right) : i = 1, \cdots, N\right\}$ with $y^i \in \{\pm 1\}$.

Output: strong classifier:

$$S(x) = \sum_{\mu=1}^{M} \lambda_\mu \phi_\mu (x) \tag{1}$$

where $\{\lambda_\mu\}$ are parameters that need to be trained. The matching weak classifier $\phi_\mu(.)$ is not chosen since we prefer that most $\lambda_\mu = 0$. The AdaBoost method prioritizes the next iteration on the basis of the most inaccurate predictions, which are given larger weights.

3.1.2. Gradient Boosting Machine

The goal of the Gradient boosting machine algorithm is to integrate multiple base learners into a single robust learner. If we are given a dataset of the form $S = \{(x_i, y_i)\}_{i=1}^{n}$ with n observations, we want to obtain an estimation of the function $f^*(x)$ (from x inputs to target values y) using the formula $\hat{f}(x)$. To do so, we minimize the expectation of the loss function $L(y, f(x))$. To estimate $f^*(x)$, Gradient boosting machine creates a weighted sum of functions

$$f_t(x) = f_{t-1}(x) + \rho_t h_t(x), \tag{2}$$

where ρ_t denotes the weight of the t^{th} weak learner for $t = 1, \cdots, T$. Through the use of iterative construction, a constant estimation of $f^*(x)$ can be obtained by formulating it as:

$$f_0(x) = \arg\min_{\alpha} \sum_{i=1}^{n} L(y_i, \alpha), \tag{3}$$

where $L(y_i, \alpha)$ represents a loss function that can be differentiated. The weak learners seek to minimize

$$\left(\rho_t, h_t(x)\right) = \arg\min_{\rho, h} \sum_{i=1}^{n} L\left(y_i, f_{t-1}(x_i) + \rho h(x_i)\right). \tag{4}$$

In the process of optimizing gradient descent for f^*, each weak learner h_t can be thought of as a greedy step. As a result, a new dataset with the equation $S = \{(x_i, r_{ti})\}_{i=1}^{n}$ is trained using each model. The pseudo residuals r_{ti} are obtained using the following formula:

$$r_{ti} = \left[\frac{\partial L(y_i, f(x))}{\partial f(x)}\right]_{f(x) = f_{t-1}(x)} \tag{5}$$

In order to determine the value of the weight ρ_t, one must first solve the line search optimization issue. Gradient Boosting Machine is primarily focused with enhancing the accuracy of the model by decreasing the amount of error or residuals that it generates.

3.1.3. Extreme Gradient Boosting

The XGBoost algorithm is implemented as follows: In XGBoost, gradient boosting is used to fine-tune the trees.

Consider the output of a tree:

$$f(x) = w_q(x_i) \tag{6}$$

where x is the vector of input values and w_q is the score of the related leaf. A collection of K trees will provide the following result:

$$y_i = \sum_{k=1}^{K} f_k(x_i) \tag{7}$$

At each iteration t, the XGBoost algorithm seeks to optimize a certain objective function, denoted J.

$$J(t) = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{t-1} + f_t(x_i)\right) + \sum_{i=1}^{t} \Omega(f_i) \tag{8}$$

where the second term represents the regularization term that controls the complexity of the model and prevents overfitting. Train loss function L (such as mean squared error) between the real class y and the output $\hat{y}$ for the n samples is included in the first term.

In XGBoost, the complexity is defined as:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \tag{9}$$

where T is the total number of leaves, γ is the hyperparameter used to achieve pseudo-regularization (which varies between datasets), and λ is the $L2$ norm of the weights of the leaves.

Finding the optimal weights w using gradients to approximate the loss function at a second order, the optimal value of the objective function is:

$$J(t) = -\frac{1}{2}\sum_{j=1}^{T} \frac{\left(\sum_{i \in I} g_i\right)^2}{\sum_{i \in I} h_i + \lambda} + \gamma T \tag{10}$$

where $g_i = \partial_{\hat{y}^{t-1}} L\left(y, \hat{y}^{t-1}\right)$ and $h_i = \partial^2_{\hat{y}^{t-1}} L\left(y, \hat{y}^{t-1}\right)$ are the gradient statistics on the loss function, and I is the set of leaves.

3.1.4. Decision Trees

Given training vectors $x_i \in \mathbb{R}^n, i = 1, \cdots, I$ and a label vector $y \in \mathbb{R}^l$, a decision tree is a recursive partition of the feature space that combines training samples with the same labels or comparable target values together.

Let n_m samples of data from node m be represented by Q_m. Partition the data into $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$ subsets for each candidate split $\theta = (j, t_m)$ consisting of a feature j and threshold t_m.

$$\begin{aligned} Q_m^{left}(\theta) &= \left\{(x, y) \mid x_i \le t_m\right\} \\ Q_m^{right}(\theta) &= Q_m \setminus Q_m^{left}(\theta) \end{aligned} \tag{11}$$

Then, depending on the problem being solved (classification or regression), an impurity function or loss function $H()$ is chosen and used to calculate the quality of a potential split of node m.

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H\left(Q_m^{left}(\theta)\right) + \frac{n_m^{right}}{n_m} H\left(Q_m^{right}(\theta)\right) \tag{12}$$

Choose the parameters that minimises the impurity

$$\theta^* = \arg\min_{\theta} G\left(Q_m, \theta\right) \tag{13}$$

Recurse for subsets $Q_m^{left}\left(\theta^*\right)$ and $Q_m^{right}\left(\theta^*\right)$ until the maximum allowable depth is reached, $n_m < \min_{samples}$ or $n_m = 1$.

If a target is a classification outcome taking on values $0,1,\cdots,K-1$, for node m, assume that

$$Pmk = \frac{1}{n_m}\sum_{y\in Q_m} I\left(y = k\right) \tag{14}$$

is the fraction of node m's observations that belong to class k. It is assumed that m is a terminal node, and the $Predict_P roba$ for this area is changed to Pmk if this is the case. Common measures of impurity are the following.

Gini:

$$H\left(Q_m\right) = \sum_k Pmk\left(1 - Pmk\right) \tag{15}$$

Log Loss or Entropy:

$$H\left(Q_m\right) = \sum_k Pmk\log\left(Pmk\right) \tag{16}$$

3.1.5. Random Forest

This is how the Random Forest algorithm is put into practice: Assume that the training set of microarrays $D = \left\{\left(X_1, y_1\right),\cdots,\left(X_n, y_n\right)\right\}$ was chosen at random from a (potentially unknowable) probability distribution $\left(X_i, y_i\right) \sim \left(X, Y\right)$. The objective is to create a classifier that uses D as a data set to make predictions about y given X. Given a collection of classifiers, $h = \left\{h_1\left(X\right),\cdots,h_K\left(X\right)\right\}$. If each $h_k\left(X\right)$, some of which may be less accurate than others. The ensemble is a random forest if and only if each $h_k\left(X\right)$ is a decision tree. For the classifier $h_k\left(X\right)$, we define the tree's parameters as follows:

$$\Theta_k = \left(\theta_{k1}, \theta_{k2},\cdots,\theta_{kp}\right) \tag{17}$$

Tree structure, variable partitioning among nodes, etc. are all examples of such factors.

We occasionally write

$$h_k\left(X\right) = h\left(X \mid \Theta_k\right) \tag{18}$$

As a result, decision tree k leads to classifier $h_k\left(X\right) = h\left(X \mid \Theta_k\right)$.

How do we prioritize the characteristics to show in each branch of the K^{th} tree? Based on a random selection of Θ_k parameters from the model variable Θ.

A random forest is a type of classifier that is constructed using a family of classifiers $h\left(X \mid \Theta_1\right),\cdots,h\left(X \mid \Theta_K\right)$ on a classification tree, with parameters Θ_k selected at random from a model random vector Θ. Each tree contributes one vote to the final classification $f\left(X\right)$, which combines the classifiers $h_k\left(X\right)$, and the category that receives the greatest number of votes is deemed to be the most appropriate.

Specifically given data $D = \left\{\left(X_i, y_i\right)\right\}_{i=1}^n$, we train a family of classifiers $h_k\left(X\right)$.

Each classifier $h_k(X) \equiv h(X \mid \Theta_k)$ is in our case a predictor of n

$y = \pm 1$ = outcome associated with input X.

3.2. LightGBM

Introduction to LightGBM: LightGBM operates as a gradient boosting framework that uses a histogram-based algorithm, enhancing speed and efficiency. It stands out due to its leaf-wise tree growth strategy and specific mathematical optimizations that address overfitting, a common issue in predictive modeling.

Tree Growth Strategy and Mathematical Foundation: In the gradient boosting landscape, LightGBM introduces an innovative leaf-wise tree growth strategy, contrasting with traditional level-wise methods. This strategy optimizes the following objective, minimizing loss more efficiently:

$$\text{minimize } \mathbb{E}_{x,y}\left[L\left(y, F(x)\right)\right]$$

In each iteration t, the model computes the gradients:

$$g_i^{(t)} = \frac{\partial L\left(y_i, F_{t-1}(x_i)\right)}{\partial F_{t-1}(x_i)}$$

Using these gradients, LightGBM applies a leaf-wise strategy, selecting the leaf with the highest delta loss to grow. This method, governed by the following gain calculation, also integrates a regularization term λ to prevent overfitting:

$$\text{Gain} = \frac{1}{|I_L| + \lambda}\left(\sum_{i \in I_L} g_i^{(t)}\right)^2 + \frac{1}{|I_R| + \lambda}\left(\sum_{i \in I_R} g_i^{(t)}\right)^2 - \frac{1}{|I| + \lambda}\left(\sum_{i \in I} g_i^{(t)}\right)^2$$

Here, λ acts as a complexity penalization, ensuring the model doesn't overly adapt to training data nuances, a principle that is crucial for generalization and predictive accuracy in unseen data.

3.3. CatBoost

Introduction to CatBoost: CatBoost, known for its effectiveness with categorical features, takes gradient boosting further by addressing overfitting through algorithmic enhancements and sophisticated mathematical underpinnings.

Ordered Boosting and Mathematical Insights: CatBoost employs a unique permutation-driven scheme within its boosting approach, ensuring error correction in each sequential tree while avoiding repetitive learning from the same instances. The mathematical foundation for this involves computing gradients and Hessians for loss minimization:

Gradients:

$$g_i = \nabla L\left(y_i, F_{t-1}(x_i)\right)$$

Hessians:

$$h_i = \nabla^2 L\left(y_i, F_{t-1}(x_i)\right)$$

These values contribute to the construction of each tree, with the optimal leaf

value computed as follows:

$$\theta^* = -\frac{\sum g_i}{\sum h_i + \lambda}$$

In this formula, λ is a regularization parameter, adding a level of penalty against complexity, thereby safeguarding against overfitting. This regularization ensures that the model remains robust and maintains high accuracy by not mirroring the training data too closely.

3.4. The Proposed Hybrid LightGBM and CatBoost Model

The hybrid model was developed by constructing a super learner ensemble model by sequentially integrating the individual models with LightGBM and CatBoost as the foundational learning algorithms and data.

- **Data Preparation** Let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ be the training dataset where x_i are the feature vectors and y_i are the labels. Randomly split $\mathcal{D}_{\text{train}}$ into training ($\mathcal{D}_{\text{train-train}}$) and validation ($\mathcal{D}_{\text{train-valid}}$) sets.
- **Hyperparameter Optimization** Define the hyperparameter search space for LightGBM (Θ_{LGB}) and CatBoost (Θ_{CB}).

1) For each trial t:

- Sample hyperparameters θ_{LGB}^t from Θ_{LGB} and θ_{CB}^t from Θ_{CB}.
- Train models M_{LGB}^t and M_{CB}^t on $\mathcal{D}_{\text{train-train}}$ using θ_{LGB}^t and θ_{CB}^t respectively.
- Evaluate log-loss L_{LGB}^t and L_{CB}^t on $\mathcal{D}_{\text{train-valid}}$.

2) Select the best hyperparameters θ_{LGB}^* and θ_{CB}^* that minimize the log-loss L_{LGB} and L_{CB} respectively.

- **Model Training** Train models M_{LGB} and M_{CB} on the entire $\mathcal{D}_{\text{train}}$ using θ_{LGB}^* and θ_{CB}^* respectively.
- **Model Prediction** Let $P_{\text{LGB}} = M_{\text{LGB}}(\mathcal{D}_{\text{test}})$ and $P_{\text{CB}} = M_{\text{CB}}(\mathcal{D}_{\text{test}})$ be the probability predictions on the test set $\mathcal{D}_{\text{test}}$.
- **Weight Optimization** Define the loss function $\mathcal{L}$ as:

$$\mathcal{L}(\alpha) = -\sum_{i=1}^{N_{\text{test}}} y_i \log\left(\alpha P_{\text{LGB},i} + (1-\alpha) P_{\text{CB},i}\right) + (1-y_i)\log\left(1-\alpha P_{\text{LGB},i} - (1-\alpha) P_{\text{CB},i}\right)$$

Optimize α to minimize $\mathcal{L}$:

$$\alpha^* = \arg\min_{\alpha \in [0,1]} \mathcal{L}(\alpha)$$

- **Evaluation** Compute the hybrid model's predictions P_{Hybrid} as:

$$P_{\text{Hybrid}} = \alpha^* P_{\text{LGB}} + (1-\alpha^*) P_{\text{CB}}$$

Evaluate P_{Hybrid} using various metrics like log-loss, accuracy, precision, recall, F1-score, and AUC-ROC.

The diagram for the hybrid model is depicted in **Figure 3**.

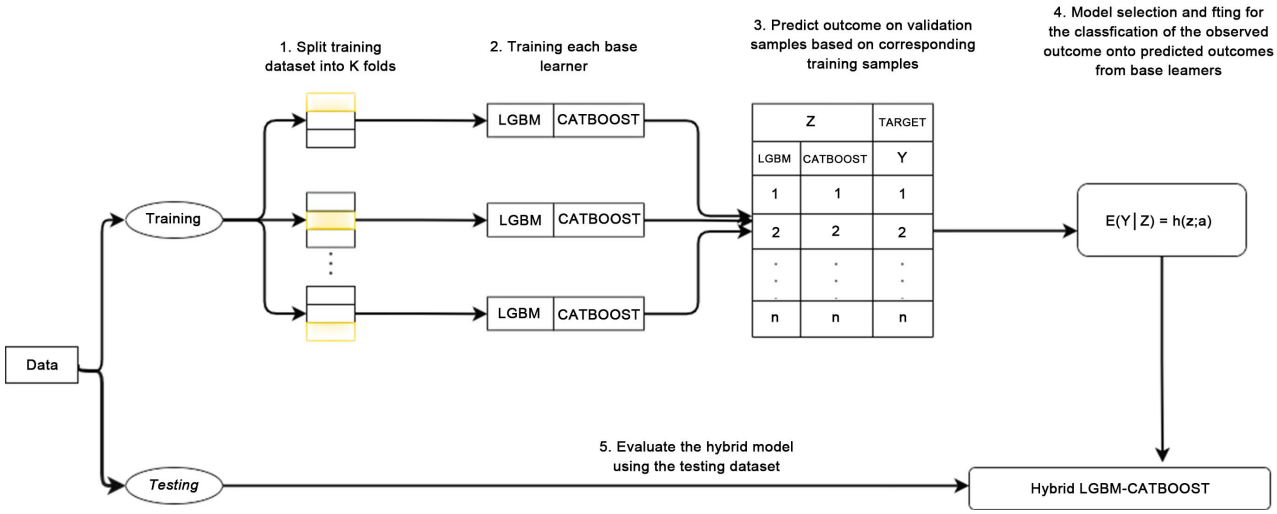

Figure 3. Diagram depicting the steps required to put into action the proposed LightGBM/CatBoost hybrid model.

Algorithm 1 A Pseudo-code of the Weight Averaging method in the Hybrid Model development

Input: Training data D_{train}, Test data D_{test}

▷ Step 1: Data Preparation

Randomly split D_{train} into training and validation sets

▷ Step 2: Hyperparameter Optimization using Optuna

Define the search space for LightGBM and CatBoost hyperparameters
Initialize Optuna study object for each model
for each trial t in Optuna study **do**
    Sample hyperparameters for LightGBM and CatBoost
    Train LightGBM and CatBoost on the training set using the sampled hyperparameters
    Evaluate LightGBM and CatBoost on the validation set
    Record the log-loss for each model
end for
Select the best hyperparameters for LightGBM and CatBoost based on minimum log-loss

▷ Step 3: Model Training

Train LightGBM and CatBoost on the entire D_{train} using the best hyperparameters

▷ Step 4: Model Prediction

Obtain probability predictions P_{LGB} and P_{CB} on D_{test} using LightGBM and CatBoost respectively

▷ Step 5: Optimize Hybrid Weight using Log-Loss

Define a loss function $\mathcal{L}$ to compute log-loss given α as weight

$$\mathcal{L}(\alpha) = \text{Log-Loss}(\alpha \times P_{\text{LGB}} + (1-\alpha) \times P_{\text{CB}})$$

Find $\alpha^* = \arg\min_\alpha \mathcal{L}(\alpha)$ using a numerical optimizer

▷ Step 6: Evaluation

Compute the hybrid model's probability predictions P_{Hybrid} on D_{test}

$$P_{\text{Hybrid}} = \alpha^* \times P_{\text{LGB}} + (1-\alpha^*) \times P_{\text{CB}}$$

Evaluate P_{Hybrid} using log-loss, accuracy, precision, recall, F1-score, and AUC-ROC
Return Evaluation metrics

Figure 4. A Pseudo-code of the weight averaging method in the hybrid model development.

3.5. Data Description and Tools

Dataset Overview: Diabetes Dataset from the Hospital Frankfurt, Germany

Source and Nature: This dataset has been assembled from varied health parameters pertinent to diabetes diagnosis, collected from a hospital in Frankfurt, Germany. It serves as a comprehensive representation of critical indicators used in the diagnosis and analysis of diabetes, demonstrating the scope of data-driven medical approaches in modern healthcare environments.

Composition and Attributes: The dataset contains 2000 individual instances, each providing insight into the health status of different subjects. It comprises 9

significant attributes, each contributing to a comprehensive understanding of diabetes indicators:

1) Pregnancies
2) Glucose
3) Blood Pressure
4) Skin Thickness
5) Insulin
6) Body Mass Index (BMI)
7) Diabetes Pedigree Function
8) Age
9) Outcome (e.g., positive or negative diabetes diagnosis)

Analytical Insights:

The dataset is subjected to a thorough analytical process, with a comprehensive representation of the descriptive statistics for each attribute. This meticulous analysis illuminates the fundamental statistical characteristics and dynamics of the data, providing essential insights that are pivotal for further research and exploration in the field of diabetes (Figure 5).

	count	mean	std	min	25%	50%	75%	max
Pregnancies	2000.0	3.70350	3.306063	0.000	1.000	3.000	6.000	17.00
Glucose	2000.0	121.18250	32.068636	0.000	99.000	117.000	141.000	199.00
BloodPressure	2000.0	69.14550	19.188315	0.000	63.500	72.000	80.000	122.00
Skin Thickness	2000.0	20.93500	16.103243	0.000	0.000	23.000	32.000	110.00
Insulin	2000.0	80.25400	111.180534	0.000	0.000	40.000	130.000	744.00
BMI	2000.0	32.19300	8.149901	0.000	27.375	32.300	36.800	80.60
DiabetesPedigreeFunction	2000.0	0.47093	0.323553	0.078	0.244	0.376	0.624	2.42
Age	2000.0	33.09050	11.786423	21.000	24.000	29.000	40.000	81.00
Outcome	2000.0	0.34200	0.474498	0.000	0.000	0.000	1.000	1.00

Figure 5. Descriptive statistics of the dataset.

Unique Characteristics: The geographical specificity and encompassing medical data bestow the dataset with a unique standpoint in diabetes studies.

3.5.1. Data Pre-Processing

In order to identify the hyperparameters that produce the best results for the objective function, the base learners used a Bayesian hyperparameter optimization strategy. This approach, introduced in 2019, effectively tunes the trial and error computing process to determine the most suitable hyperparameters [40]. To gain deeper insights into the hybrid model's behavior, Shapley Additive Explanation (Shap) values were utilized. Originally used in cooperative game theory within the economics sector, Shapley Additive Explanation values assess individual contributions in a predictive setting [41]. By quantifying the impact of each variable, an importance value is assigned to calculate the overall explanation. The proposed methodology was implemented using Python 3.9.13. The Python algorithms were executed in Jupyter notebook, an open-source web tool. For building the hybrid model, various ML modules including Scikit-learn, op-

tuna, Shap, lightgbm, and CatBoost were employed. The results were visually analyzed using Matplotlib and optuna visualization modules. The computational resources utilized for this implementation were an HP-Omen Gaming Laptop equipped with an NVIDIA GeForce GTX core i7, 16GB RAM, and a processing speed of 2.60 GHz.

3.5.2. Random Over-Sampling for Data Balancing

The Random Over-Sampling (ROS) technique is a widely employed method for addressing class imbalance in high-dimensional datasets, aiming to tackle various real-life scenarios [42]. Dealing with imbalanced datasets can be challenging as it often leads to poor model performance across multiple statistical metrics. In this study, prior to constructing the ML/EL models, the random over-sampling approach was employed to balance the classes and optimize the predictive capability of the framework. The random over-sampling technique involves over-sampling and augmenting the minority class present in the dataset by replicating existing minority samples, thereby increasing the size of the minority class. **Figure 6** illustrates the count of outcomes (class variable) before and after applying the ROS technique, as depicted in a previous study by [43].

3.5.3. Dataset Distribution

The distribution of the predicate variables Age, Insulin, Skin Thickness, Blood Pressure, Pregnancies, Glucose, BMI, and Diabetes Pedigree Function towards the target variable Outcome has been plotted using the FacetGrid method (Seaborn package). In this technique, the distribution of the dataset's observations was graphically represented using the Kernel Density Estimate (KDE) plot function. It uses a continuous probability curve to represent data samples in one or more dimensions. The range of samples is given along the horizontal or $x_a xis$, while the probability density function of a random variable is displayed vertically or $y_a xis$. The probability of value is the sum of the shaded region of the curve between x_1 and x_2, where K is the kernel function assigned to each data point x_i. We can estimate the kernel density as:

$$P(x) = \frac{1}{Nh}\sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right) \tag{19}$$

where,

- P = density at location x;
- K represents a non-negative kernel function;
- N represents the number of steps;
- The smoothing parameter is denoted by h;
- x denotes the maximum random value;
- The variable x_i determines the data sampling rate.

Figures 4-14 depict the frequency distribution of all lifestyle characteristics, with light green representing the non-diabetic class and dark green representing the diabetes class.

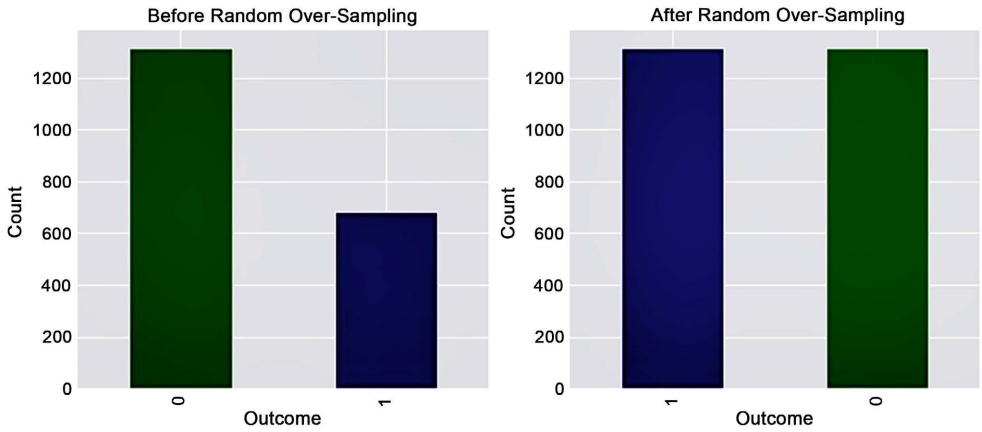

Figure 6. Class balancing of dataset using ROS technique.

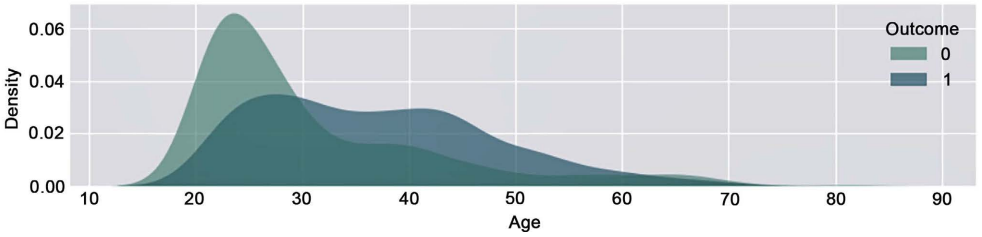

Figure 7. Age with respect to target.

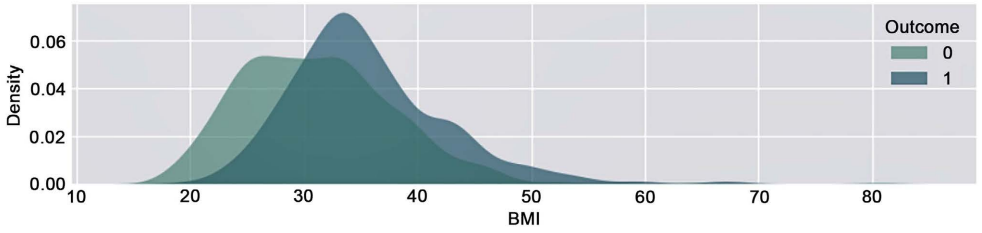

Figure 8. BMI with respect to target.

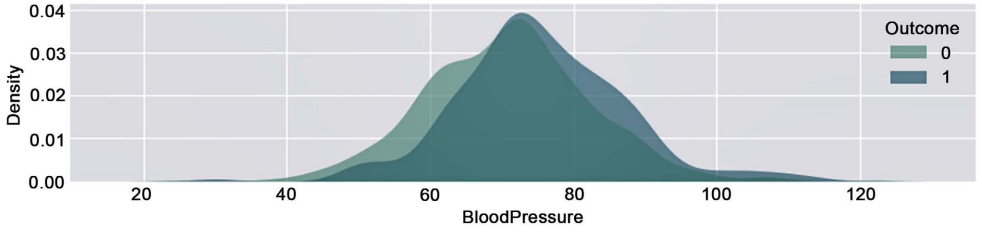

Figure 9. Blood Pressure with respect to target.

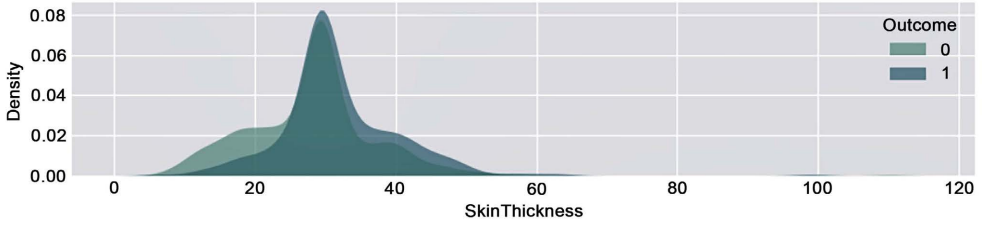

Figure 10. Skin Thickness with respect to target.

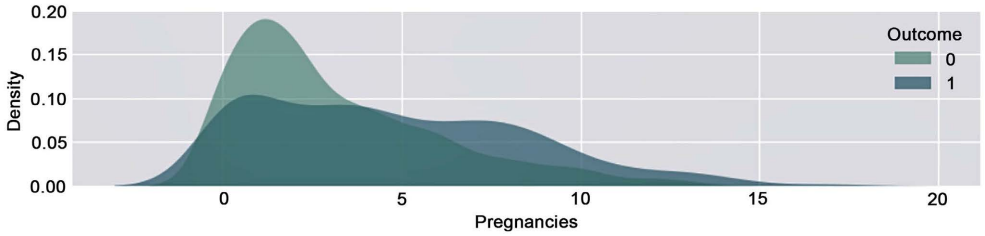

Figure 11. Pregnacies with respect to target.

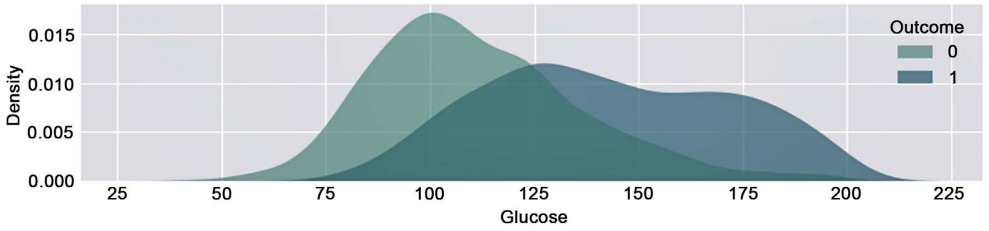

Figure 12. Glucose with respect to target.

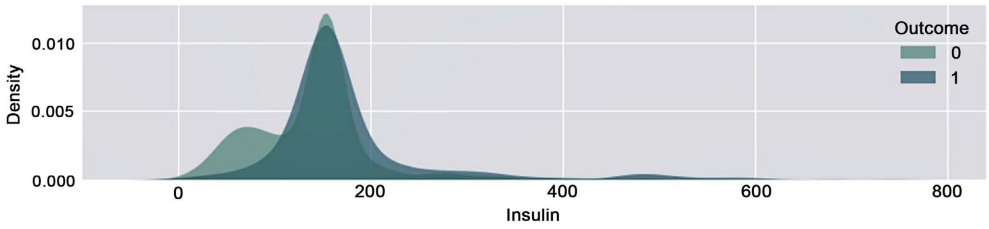

Figure 13. Insulin with respect to target.

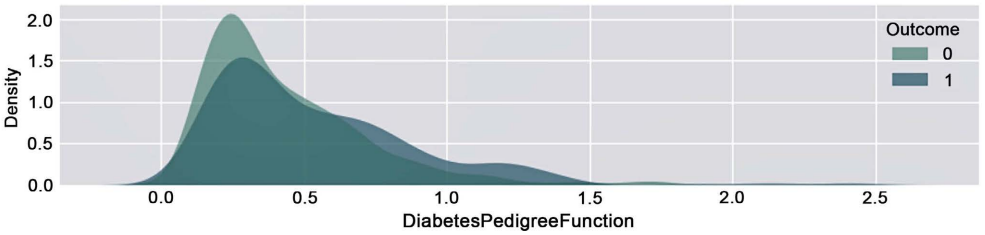

Figure 14. Diabetes Pedigree Function with respect to target.

3.5.4. K-Fold Cross-Validation and Splitting of Datasets

To mitigate dataset bias, researchers and professionals often employ the K-fold cross-validation technique [15]. As depicted in **Figure 15**, this study utilized 10-fold cross-validation, visually demonstrating the data splitting process. The dataset was divided into 10 equal-sized partitions randomly. During each iteration, one partition was designated as the validation set (testing set), while the remaining nine partitions were used for training the model. This method guaranteed that each partition only ever performed the validation procedure once. Summation was used to add up the results from each iteration. By utilizing this approach, the dataset effectively addressed the issues of overfitting and underfitting, thereby minimizing bias and producing realistic results in machine learning models. Notably, both training and testing datasets encompassed all data samples, ensuring comprehensive evaluation.

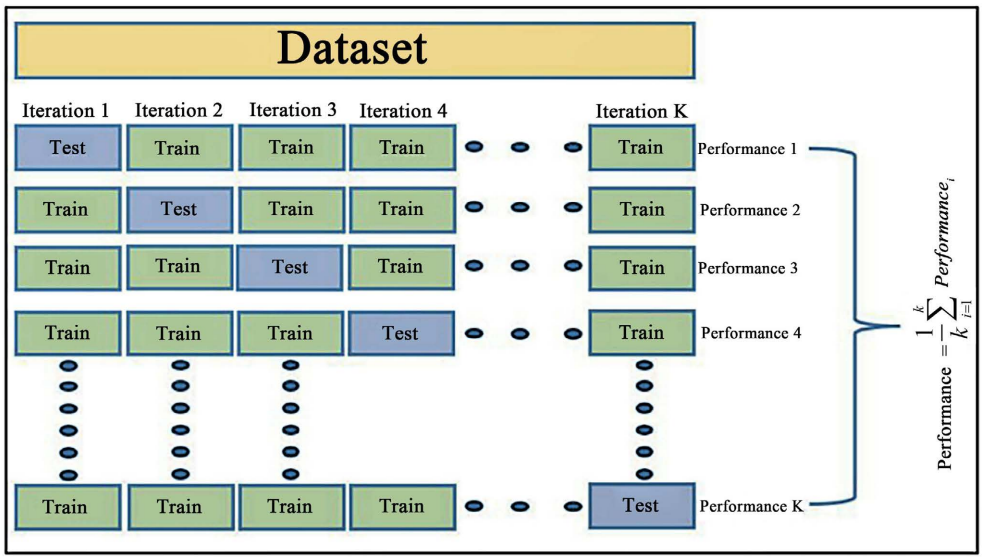

Figure 15. K-fold cross-validation technique.

3.5.5. Feature Engineering

Feature engineering is crucial to the process of constructing ML/EL models. Execution of a model can be negatively impacted by irrelevant or unsuitable features [44]. The training time is cut down and accuracy is increased with careful feature selection. Machine learning paradigms make use of a variety of feature selection methodologies, such as filter, wrapper, embedding, and hybrid approaches respectively [45]. Feature selection in this research was accomplished using the Information Gain and Correlation techniques. All of the characteristics that were used for the Category Boosting (CatBoost) classifier for TIIDM showcase prediction are shown in **Figure 16**. Glucose, Body Mass Index, Diabetes Pedigree Function, Age, Blood Pressure, Insulin, Pregnancies, and Skin Thickness are ranked/important in order from most to least in terms of outcome.

4. Description of Results

The primary aim of the weak learners was to maximize accuracy while minimizing squared error. Over the course of 50 iterations, the optimal set of hyperparameters was found. The hyperparameters were fine-tuned across a range of 100 - 300 iterations. Trial 50 had the optimal combination of hyperparameters for the weak learners and yielded the highest accuracy. The optimal trial is depicted in **Figure 17**.

Figure 18 and **Figure 19** serve as an illustration. The model provided the optimal set of hyperparameters to get the lowest possible error while boosting accuracy.

4.1. Hyperparameter Importance

The models' results were affected in different ways by the set of hyperparameters that brought about the minimum in the objective function. **Figure 20** shows that the LightGBM model's min child samples contributed 68% to the learning

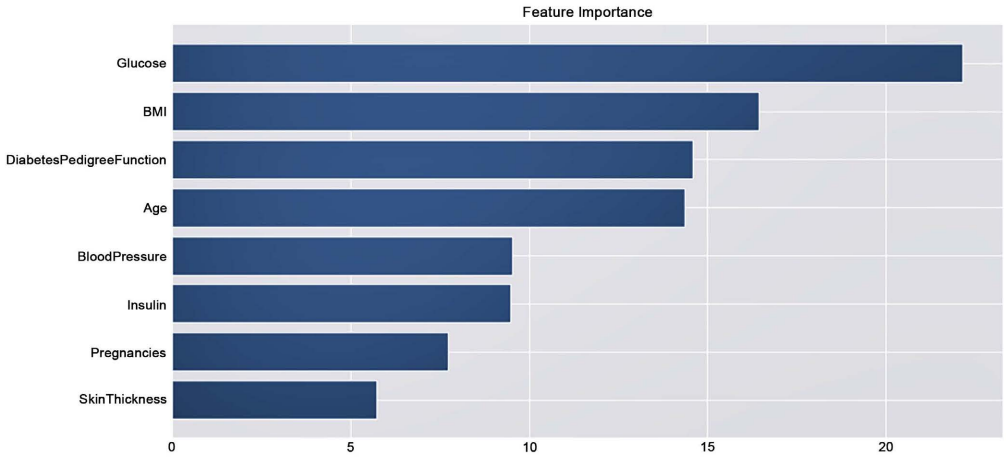

Figure 16. Feature Importance towards prediction of TIIDM.

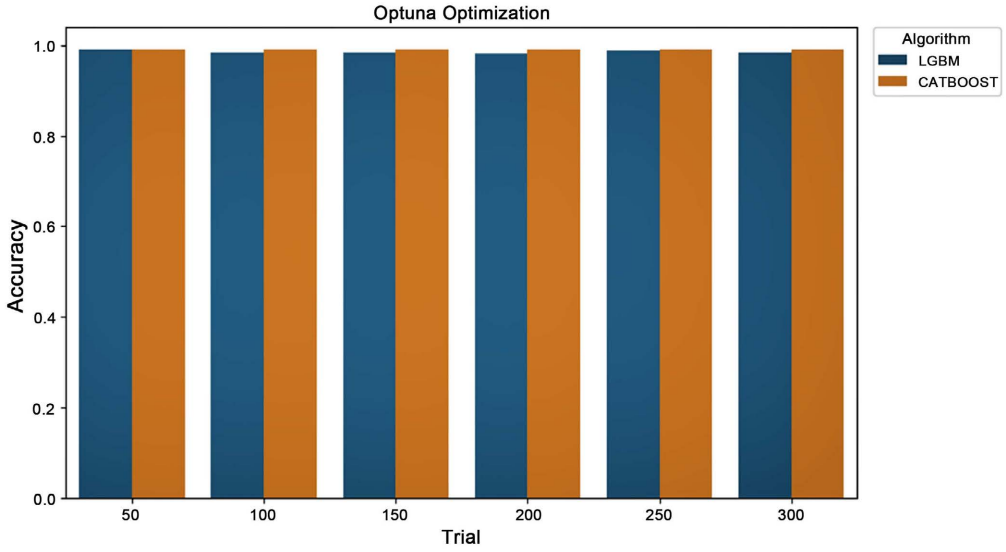

Figure 17. Optimization trials of hyperparameters to determine the optimal hyperparameter settings for the weak learners.

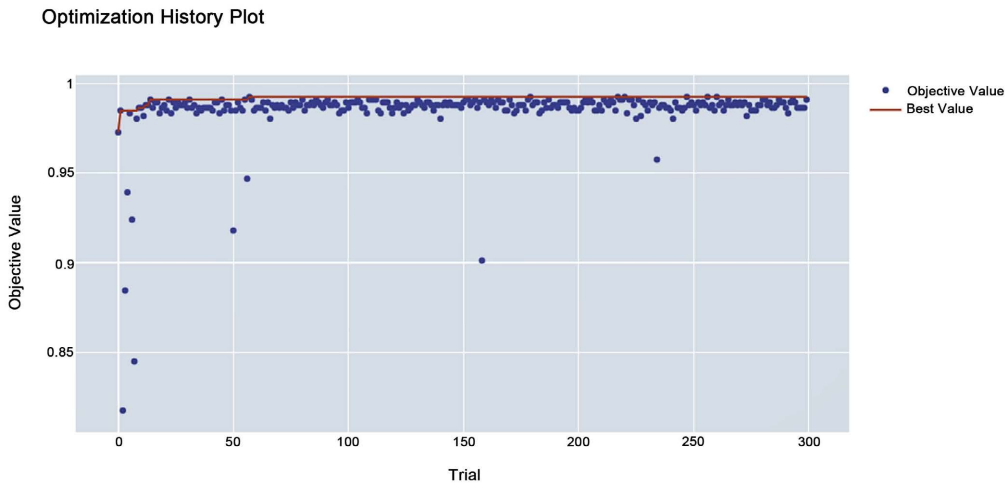

Figure 18. Catboost model's minimization of objective function optimization history over 300 tests.

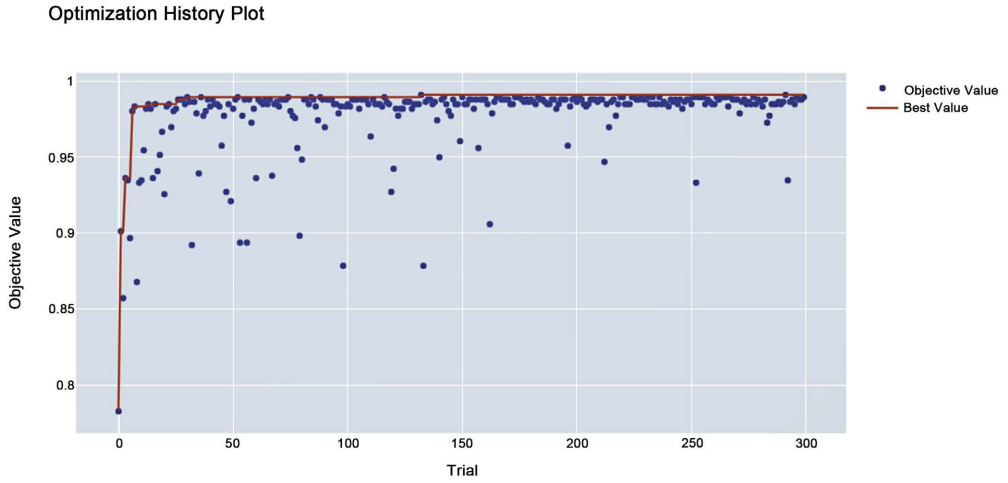

Figure 19. LightGBM model's minimization of objective function optimization history over 300 Tests.

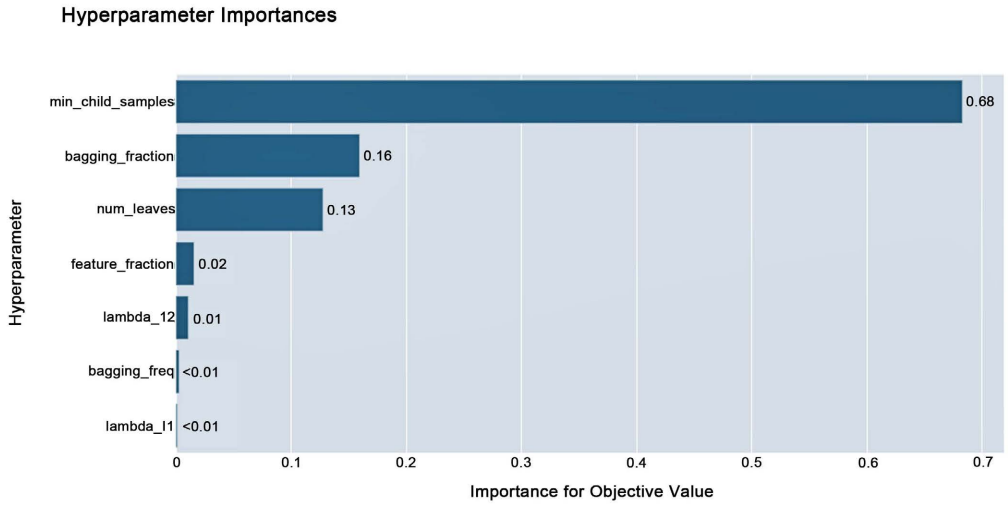

Figure 20. LightGBM model's minimization of the objective function: an optimization history of 300 attempts.

process by minimizing error, whereas bagging fraction contributed 16%. The other hyperparameters had the smallest influence, enhancing model performance by less than 14%. Bagging temperature, which defines the settings of the Bayesian bootstrap, had the most influence on the CatBoost model's objective function optimization, contributing 30% to the process, preceded by Iterations and Learning Rate contributing 26% and 22% respectively as shown in **Figure 21**. Less than 10% of the remaining hyperparameters affected the model's performance.

The base learners use various hyperparameters, although several of the hyperparameters had little effect on the model, this does not imply that they had no effect on enhancing performance. In order to maximize the objective function using the more important hyperparameters, even the small contribution was required.

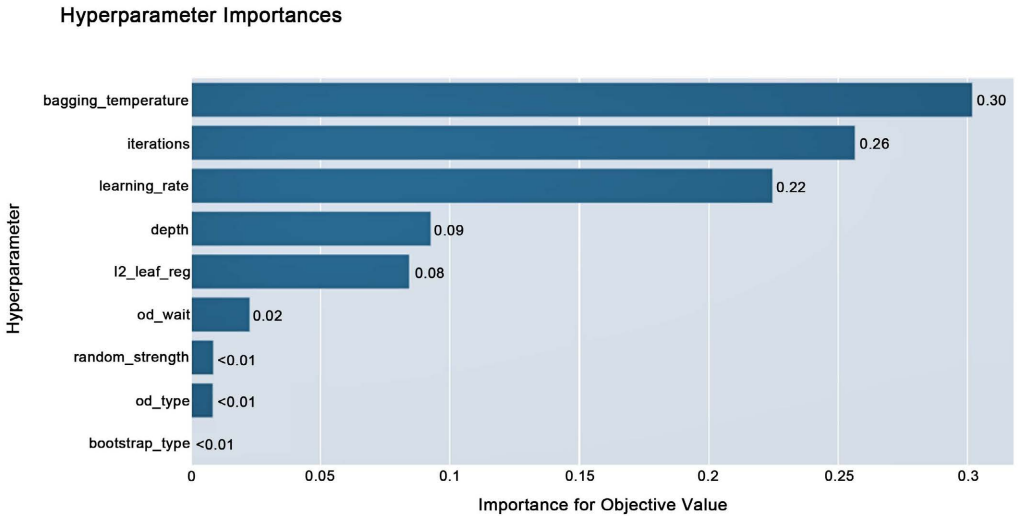

Figure 21. CatBoost model's minimization of the objective function: an optimization history of 300 attempts.

4.2. Hyperparameter Parallel Coordinate

Figure 22 depicts the correlation between hyperparameter tuning and LightGBM model objective function value optimization. High bagging fraction, low bagging frequency, medium feature fraction, low lambda, high alpha, low minimum child sample, and high minimum data in leaf were all associated with high objective values for various trials.

Figure 23 demonstrates that in the CatBoost model, high objective values were related to low bagging temperature, medium depth, high iterations, low l2 leaf reg, high learning rate, low od wait, and a high random strength.

Each possible value of one hyperparameter is tested against a range of other hyperparameters in order to find the optimal settings for base learners.

4.3. Overview of Base Learners Hyperparameters

The optimal values of the identical hyperparameters used to optimize the objective function for the weak learners and achieve the highest accuracy are shown in Table 1. The minimal number of child samples, bagging frequency, and total number of leaves for the LightGBM model were 3, 2, and 172, respectively. Iterations and Depth were set at 968 and 47 in the CATBOOST model, respectively.

4.4. Hybrid LightGBM and CATBOOST Model Interpretation

The Shap method was used to determine how much each of the weak learners contributed to the final results of the hybrid model. Figure 24 displays the distribution and impact of the features on TIIDM prediction, as well as their relative importance, in descending order. Low Glucose resulted in low chance to have TIIDM whilst high Glucose resulted in high chance to be TIIDM positive. For both skin thickness and blood pressure, the vast majority of samples had a shap value of zero, which had negligible impact on the model's predictions.

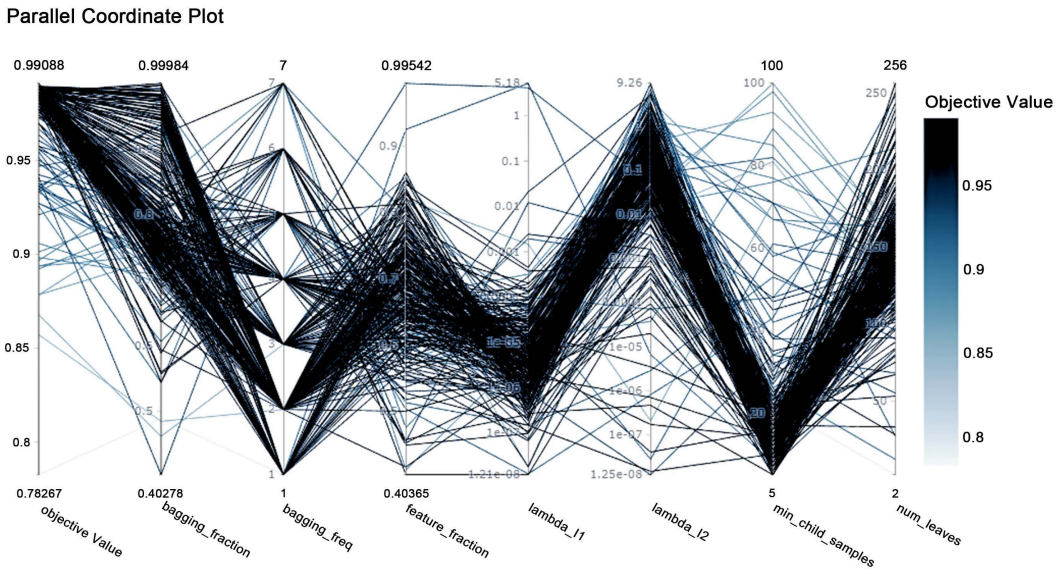

Figure 22. Parallel coordinates for LightGBM model hyperparameter and objective function values.

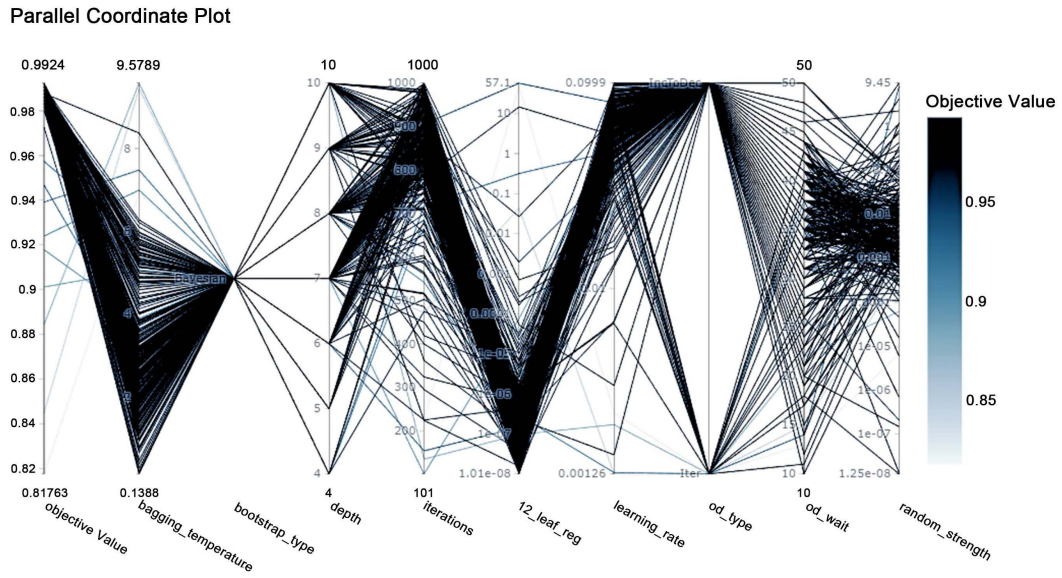

Figure 23. Parallel coordinates for CatBoost model hyperparameter and objective function values.

Table 1. Hyperparameter settings for the weak learners.

Hyperparameter	LGBM	Hyperparameter	CATBOOST	
α (L1 Regularization)	8.7767e−05	Iterations	968	a
λ (L2 Regularization)	0.32674	Learning Rate	0.067166	
Number of Leaves	172	Depth	7	
Feature Fraction	0.67151	L2 Leaf Reg	6.5419	
Bagging Fraction	0.72909	Random Strength	0.00971	
Bagging Frequency	2	Bagging Temperature	2.5224	
Min Child Samples	13	Od Wait	39	b

[a] Note for Iterations. [b] Note for Od Wait.

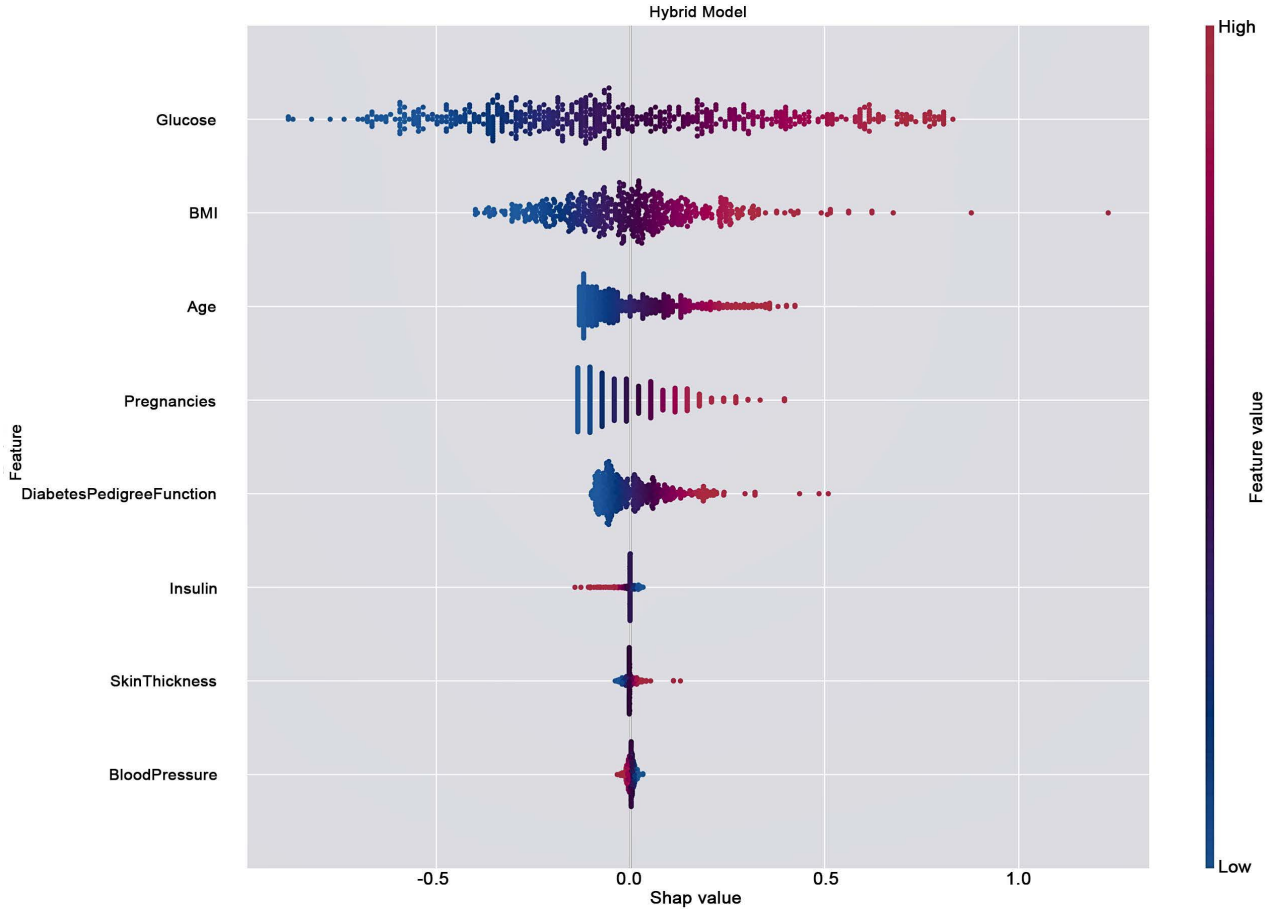

Figure 24. Feature Importance of Hybrid Model. Features' distribution and impact in TIIDM prediction.

Figure 25 shows how each weak learner contributed to the final output of the hybrid model based on the accuracy with which its predictions matched the target values. As can be seen from the length of the bars, CatBoost had a greater effect on the hybrid model's performance than LightGBM. LightGBM and CatBoost models contributed 40% and 60%, respectively, to the hybrid model. The difference between the two learners' contributions at the beginning is twenty percent. The higher performance accuracy can be attributed to the fact that both models contributed significantly to the hybrid model's output.

The weak learners are shown in Figure 26 and Figure 27 to have assisted in shifting the initial prediction of the hybrid model from the base value (the average output of the training set) to the target value. The initial estimate of −0.99 for the negative class was improved to 0.04, and the first forecast of 0.5 was improved to 0.93, both of which were very close to the target number. Both models made substantial contributions to this prediction, as shown by the length of the base learners.

5. Hybrid Model Summary

The hybrid model was superior at optimizing the objective function, which aims to reduce error while increasing performance. Table 2 displays the Log Loss for

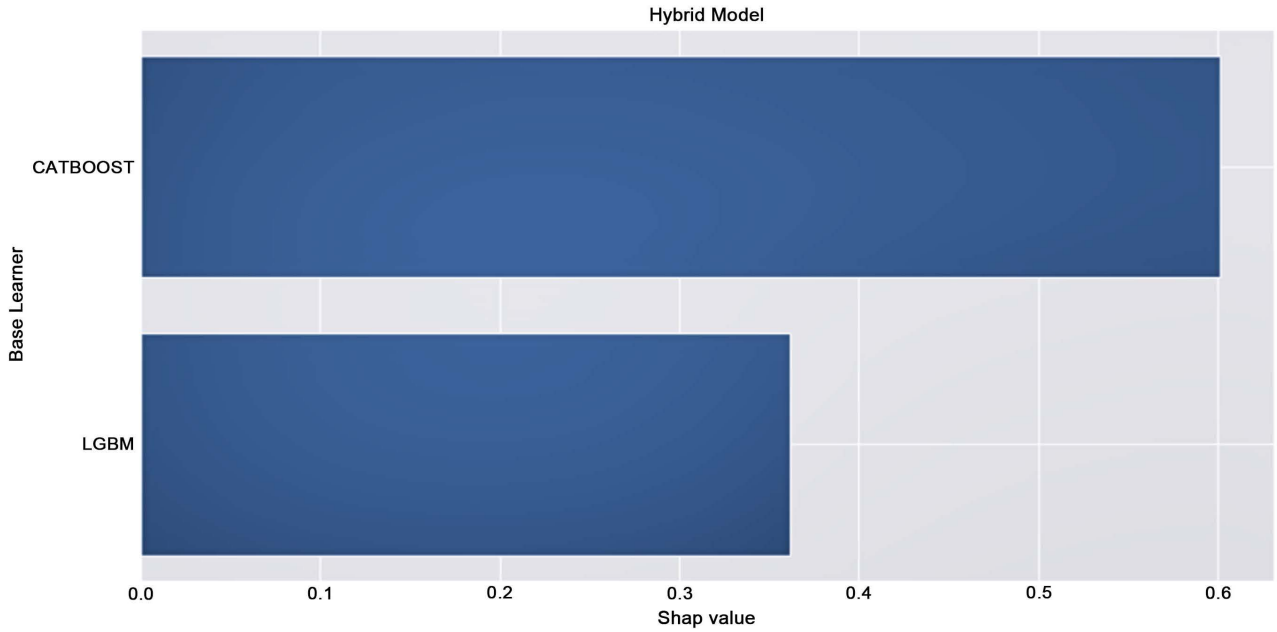

Figure 25. Predictive performance of the TIIDM hybrid model and the relative importance of its weak learners.

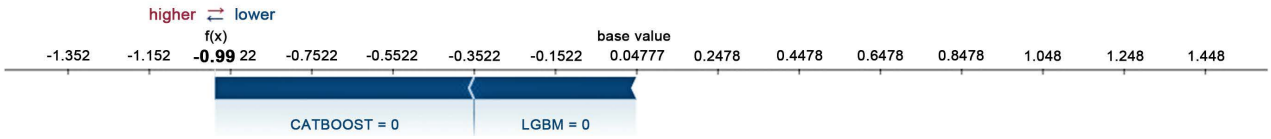

Figure 26. Individual hybrid model prediction explanation. The negative class prediction of each weak learner.

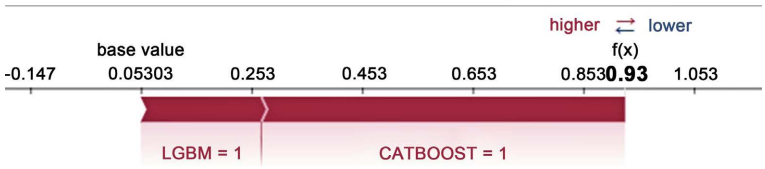

Figure 27. Individual hybrid model prediction explanation. The positive class prediction of each weak learner.

Table 2. Log Loss analysis of the combined efficiency of LightGBM and CATBOOST hybrids.

Model	Log Loss
LightGBM-CatBoost without Bayesian	0.699
LightGBM-CatBoost with Bayesian	0.255

the hybrid model before and after Bayesian hyperparameter tuning was performed on the base learners. The Log Loss of the hybrid model fell from 0.699 when the default hyperparameters were used to 0.262 when the optimal set of hyperparameters was used.

6. Hybrid Model Performance Evaluation

The performance of the hybrid model was assessed using the baseline techniques,

which include the weak learners, LightGBM and CatBoost models, and tree-based algorithms, AdaBoost, XGBoost, decision tree, random forest and GBM models. According to Table 3, the hybrid LightGBM and CatBoost model outperformed the other algorithms by having the lowest log loss that is 0.25 and the highest accuracy of 99.37%. The hybrid model outperformed its predecessors with minimal error, allowing for improved Type II diabetes mellitus prediction.

Table 3 provides a summary of the hybrid model's performance evaluation in predicting the output. The evaluation criteria used are detailed, including Accuracy, Log Loss, Precision, and Recall.

Other statistical/ML metrics for the test dataset, such as F1-score, ROC-AUC, and PR AUC, are shown in Table 4. The hybrid model, on the other hand, achieved an excellent performance rate of 99.37%, 99.91%, and 99.90% in terms of F1-score, ROC-AUC, and NPV, PR AUC.

The proposed hybrid LightGBM and CatBoost model performs better than baseline boosting methods to validate its effectiveness, demonstrating that it is the best model for predicting Type-II diabetes mellitus with decreased variance and increased accuracy.

6.1. Comparative Analysis with Existing Work

In Table 5, we compare our suggested framework to other research that have dealt with similar problems in terms of technique, dataset, and analysis to determine how effective it is. Most of these studies utilized similar lifestyle markers for comparison purposes. Our system demonstrated favorable results, particularly in terms of accuracy, for predicting Type-II Diabetes Mellitus (TIIDM). To ensure the validity of our results, we employed techniques such as hyperparameter tuning and K-fold cross-validation during the development of the proposed framework, aiming to achieve more robust and reliable outcomes compared to other related studies

6.2. Discussion on Hybrid Model Performance

The remarkable efficacy of the hybrid model, which integrates the LightGBM and CatBoost algorithms, is evident from the performance metrics presented in Table 3 and Table 4. This superiority is not coincidental but is attributable to several strategic and technical advantages, as discussed below:

1) **Precision in Learning from Data:** The synergy between LightGBM's efficiency in processing large datasets and CatBoost's adept handling of categorical features results in a model with enhanced learning precision. This precision significantly contributes to the model's high scores in accuracy and F1-score, ensuring a balanced harmony between precision and recall.

2) **Reduced Overfitting:** Both constituent models, LightGBM and CatBoost, have inherent features designed to combat overfitting. LightGBM's leaf-wise growth strategy, which is curtailed at a certain depth, and CatBoost's utilization of ordered boosting, collaboratively contribute to a model that generalizes well

Table 3. Performance evaluation summary.

Algorithms	Accuracy (%)	Log Loss	Precision (%)	Recall (%)
AdaBoost	82.78	5.94	83.33	82.50
GBM	86.96	4.50	86.49	80.00
LGBM	96.46	1.22	96.27	96.75
XGBoost	96.71	1.13	96.98	96.50
CatBoost	96.96	1.04	96.53	97.50
Random Forest	96.84	1.09	97.47	96.25
Decision Trees	94.30	1.96	95.63	93.00
LGBM-CatBoost	99.37	0.25	99.50	99.25

Table 4. Performance evaluation summary.

Algorithms	F1 Score (%)	ROC-AUC	PR AUC (%)
AdaBoost	82.91	82.27	77.61
GBM	87.24	86.94	82.18
LGBM	96.51	96.45	94.78
XGBoost	96.74	96.71	95.36
CatBoost	97.01	96.95	95.38
Random Forest	96.86	96.84	95.71
Decision Trees	94.30	94.32	92.48
LGBM-CatBoost	99.37	99.91	99.90

Table 5. Comparison with existing systems.

Authors	Technique Used	Dataset	Analysis
[46]	CART (Classification and Regression Trees)	Collected dataset through questionnaire	75% for CART
[47]	SVM, RF and LR	Demographic web-based questionnaire	80.17% for SVM
[48]	LR, GBC, LDA, ABC, ETC, NB, Bagging, RF, DT, SVC, Perceptron and KNN	Collected dataset from hospital	96% for LR
[49]	LR, KNN, SVM, NB, DT, RF	Offline and online questionnaire	94.10% for RF
[50]	LR, LDA, KNN, DT, NB, SVM, RFC and ANN	Noakhali Medical College Bangladesh	94.07% for ANN
[51]	LR, SVM, KNN, RF, NB, GB	Murtala Mohammed Specialist Hospital, Kano	88.76% for RF
[13]	BDT, RF, ET, AB, SGB, LR, SVM, and DT	Lifestyle dataset from geographical regions	99.14% for BDT
Our Proposed Study	AdaBoost, LGBM, CatBoost, XGBoost, GBM, DT, RF, Hybrid LGBM-CatBoost	Offline and online questionnaire	99.37% for Hybrid LGBM-CatBoost

to unseen data. This is empirically confirmed by the model's superior performance metrics, including the minimal log loss.

3) **Efficiency in Handling Various Data Types:** The hybrid model stands out in its ability to seamlessly process a diverse array of data types. This characteristic, coupled with the lack of a need for extensive data pre-processing, establishes the model's robustness, especially in real-world applications where data diversity is a given.

4) **Optimized Ensemble Learning:** The ensemble approach of the hybrid model leverages the individual strengths of both LightGBM and CatBoost. This method not only averages out individual biases and reduces variance but also enhances the model's resistance to overfitting, thereby optimizing performance. This strategic move is reflected in the model's higher accuracy and other metrics compared to those of standalone models.

5) **Superiority in Complex Predictive Tasks:** The task of predicting Type-II diabetes is intricate, given the disease's multifactorial nature. The hybrid model is well-equipped for such complexity, with its amalgamation of two potent algorithms that enable a more adaptive, accurate predictive analysis amidst the convoluted interaction of numerous risk factors.

In essence, the hybrid model's architectural innovation, advanced anti-overfitting approach, and capacity for handling diverse data types collectively contribute to its standout performance in predicting Type-II diabetes. The exemplary scores across all metrics underline the model's reliability and efficacy, promising substantial applicability in facilitating.

6.3. Potential Limitations of the Hybrid Model

The proposed hybrid model, despite its promising performance, is not without certain limitations in practical applications:

1) **Hyperparameter Sensitivity:** Significant reliance on the fine-tuning of hyperparameters, creating a dependency whereby slight alterations in data may necessitate a new round of exhaustive optimization.

2) **Complexity and Interpretability:** The integration of outputs from LightGBM and CatBoost contributes to a more complex model, potentially impeding straightforward interpretability-a crucial factor in healthcare settings.

3) **Data Quality Dependence:** The performance efficacy is tightly coupled with the input data quality, indicating that inadequate features or noisy, inconsistent data could undermine predictive capabilities.

6.4. Future Directions and Improvements

Considering the aforementioned limitations, future research and model refinement could explore the following avenues:

1) **Automated Feature Engineering:** Introduction of automated mechanisms for feature selection and engineering to fortify the model's adaptability to various datasets without necessitating manual intervention.

2) **Enhanced Interpretability:** Integration of model interpretability and explanation tools, offering clearer insights into prediction determinants and fostering trust among healthcare practitioners.

3) **Optimized Resource Allocation:** Refinement of computational resource usage through streamlined algorithms or parallel computing solutions, catering to the need for scalability and potentially enabling real-time application.

4) **Extensive Real-World Validation:** Prior to clinical deployment, conducting comprehensive testing in real-world environments using multi-center, diverse datasets to ascertain model reliability and effectiveness across different scenarios.

5) **Dynamic Learning Integration:** Adoption of a continuous learning framework allowing the model to evolve with new data, maintaining its relevancy and accuracy in the ever-changing clinical landscape.

7 . Conclusions and Suggestions

In this work, we develop a hybrid model for forecasting Type-II diabetes mellitus using lifestyle factors that combines the advantages of the Light Gradient Boosting Machine (LGBM) and the CatBoost algorithms. By minimizing overfitting and reducing variance, our hybrid model demonstrates improved accuracy compared to other classification techniques. Through the use of Bayesian hyperparameter optimization, we identified the optimal set of hyperparameters for the base learners, resulting in exceptional performance metrics such as accuracy, precision, recall, F1-score, and log loss. The proposed hybrid model achieved a high accuracy rate of 99.37%, making it a promising tool for early diabetes prediction in the healthcare industry. Furthermore, the framework shows potential for application to other datasets that share common characteristics with diabetes. Our findings highlight the effectiveness of combining LGBM and CatBoost algorithms and underscore the importance of addressing overfitting concerns in prediction models. Further research can explore the implementation of the hybrid model in real-world healthcare settings and investigate its applicability to other medical conditions. Overall, our study contributes to the advancement of predictive modeling for Type-II diabetes mellitus and offers valuable insights for future research in this field.

Data Availability

Data available at https://www.kaggle.com/johndasilva/diabetes.

Conflicts of Interest

The authors declare no potential conflict of interests.

References

[1] American Diabetes Association (2021) 5. Facilitating Behavior Change and Well-Being to Improve Health Outcomes: Standards of Medical Care in Diabetes—2021. *Diabetes*

Care, **44**, S53-S72. https://doi.org/10.2337/dc21-S005

[2] Friedman, J.H. (2001) Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, **29**, 1189-1232. https://doi.org/10.1214/aos/1013203451

[3] Ke, G.L., Meng, Q., Finley, T., Wang, T.F., Chen, W., Ma, W.D., Ye, Q.W. and Liu, T.Y. (2017) LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Proceedings of the* 31*st International Conference on Neural Information Processing Systems*, Long Beach, 4-9 December 2017, 3149-3157.

[4] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V. and Gulin, A. (2018) Catboost: Unbiased Boosting with Categorical Features. *Proceedings of the* 32*nd International Conference on Neural Information Processing Systems*, Montréal, 3-8 December 2018, 6639-6649.

[5] Snoek, J., Larochelle, H. and Adams, R.P. (2012) Practical Bayesian Optimization of Machine Learning Algorithms. *Proceedings of the* 25*th International Conference on Neural Information Processing Systems*, Nevada, 3-6 December 2012, 2951-2959.

[6] Zhang, G., Xu, J.M., Yu, M., Yuan, J. and Chen, F. (2020) A Machine Learning Approach for Mortality Prediction Only Using Non-Invasive Parameters. *Medical & Biological Engineering & Computing*, **58**, 2195-2238. https://doi.org/10.1007/s11517-020-02174-0

[7] Ganie, S.M., Malik, M.B. and Arif, T. (2022) Machine Learning Techniques for Diagnosis of Type 2 Diabetes Using Lifestyle Data. In: Khanna, A., Gupta, D., Bhattacharyya, S., Ella Hassanien, A., Anand, S. and Jaiswal, A., Eds., *International Conference on Innovative Computing and Communications*, Springer, Singapore, 487-497. https://doi.org/10.1007/978-981-16-3071-2_39

[8] Kopitar, L., Kocbek, P., Cilar, L., Sheikh, A. and Stiglic, G. (2020) Early Detection of Type 2 Diabetes Mellitus Using Machine Learning-Based Prediction Models. *Scientific Reports*, **10**, Article No. 11981. https://doi.org/10.1038/s41598-020-68771-z

[9] Ahmed, S., Shaikh, S., Ikram, F., Fayaz, M., Alwageed, H.S., Khan, F., Hassan Jaskani, F., *et al.* (2022) Prediction of Cardiovascular Disease on Self-Augmented Datasets of Heart Patients Using Multiple Machine Learning Models. *Journal of Sensors*, **2022**, Article ID: 3730303. https://doi.org/10.1155/2022/3730303

[10] Hasan, M.K., Alam, M.A., Das, D., Hossain, E. and Hasan, M. (2020) Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers. *IEEE Access*, **8**, 76516-76531. https://doi.org/10.1109/ACCESS.2020.2989857

[11] Rawat, V. (2019) A Classification System for Diabetic Patients with Machine Learning Techniques. *International Journal of Mathematical, Engineering and Management Sciences*, **4**, 729-744. https://doi.org/10.33889/IJMEMS.2019.4.3-057

[12] Zhang, L.Y., Wang, Y.K., Niu, M.M., Wang, C.J. and Wang, Z.F. (2020) Machine Learning for Characterizing Risk of Type 2 Diabetes Mellitus in a Rural Chinese Population: The Henan Rural Cohort Study. *Scientific Reports*, **10**, Article No. 4406. https://doi.org/10.1038/s41598-020-61123-x

[13] Ganie, S.M. and Malik, M.B. (2022) An Ensemble Machine Learning Approach for Predicting Type-II Diabetes Mellitus Based on Lifestyle Indicators. *Healthcare Analytics*, **2**, Article ID: 100092. https://doi.org/10.1016/j.health.2022.100092

[14] Hasan, M.K., Alam, M.A., Das, D., Hossain, E. and Hasan, M. (2020) Diabetes Prediction Using Ensembling of Different Machine Learning Classifiers. *IEEE Access*, **8**, 76516-76531. https://doi.org/10.1109/ACCESS.2020.2989857

[15] Kaur, P. and Sharma, M. (2018) Analysis of Data Mining and Soft Computing Techniques in Prospecting Diabetes Disorder in Human Beings: A Review. *International Journal of Pharmaceutical Science and Research*, **9**, 2700-2719, 2018.

[16] Sewell, M. (2008) Ensemble Learning. http://machine-learning.martinsewell.com/ensembles/ensemble-learning.pdf

[17] Sagi, O. and Rokach, L. (2018) Ensemble Learning: A Survey. *WIREs Data Mining and Knowledge Discovery*, **8**, e1249. https://doi.org/10.1002/widm.1249

[18] Basaran, K., Özçift, A. and Kılınç, D. (2019) A New Approach for Prediction of Solar Radiation with Using Ensemble Learning Algorithm. *Arabian Journal for Science and Engineering*, **44**, 7159-7171. https://doi.org/10.1007/s13369-019-03841-7

[19] Abou Omar, K.B. (2018) XGboost and LGBM for Porto Seguro's Kaggle Challenge: A Comparison. https://pub.tik.ee.ethz.ch/students/2017-HS/SA-2017-98.pdf

[20] Cui, S.Z., Yin, Y.Q., Wang, D.J., Li, Z.W. and Wang, Y.Z. (2021) A Stacking-Based Ensemble Learning Method for Earthquake Casualty Prediction. *Applied Soft Computing*, **101**, Article ID: 107038. https://doi.org/10.1016/j.asoc.2020.107038

[21] Ferreira, A.J. and Figueiredo, M.A.T. (2012) Boosting Algorithms: A Review of Methods, Theory, and Applications. In: Zhang, C. and Ma, Y., Eds., *Ensemble Machine Learning*, Springer, New York, 35-85. https://doi.org/10.1007/978-1-4419-9326-7_2

[22] Mayr, A., Binder, H., Gefeller, O. and Schmid, M. (2014) The Evolution of Boosting Algorithms. *Methods of Information in Medicine*, **53**, 419-427. https://doi.org/10.3414/ME13-01-0122

[23] Dargahi-Zarandi, A., Hemmati-Sarapardeh, A., Shateri, M., Menad, N.A. and Ahmadi, M. (2020) Modeling Minimum Miscibility Pressure of Pure/Impure CO_2-Crude Oil Systems Using Adaptive Boosting Support Vector Regression: Application to Gas Injection Processes. *Journal of Petroleum Science and Engineering*, **184**, Article ID: 106499. https://doi.org/10.1016/j.petrol.2019.106499

[24] Touzani, S., Granderson, J. and Fernandes, S. (2018) Gradient Boosting Machine for Modeling the Energy Consumption of Commercial Buildings. *Energy and Buildings*, **158**, 1533-1543. https://doi.org/10.1016/j.enbuild.2017.11.039

[25] Rawi, R., Mall, R., Kunji, K., Shen, C.H., Kwong, P.D. and Chuang, G.Y. (2018) Parsnip: Sequence-Based Protein Solubility Prediction Using Gradient Boosting Machine. *Bioinformatics*, **34**, 1092-1098. https://doi.org/10.1093/bioinformatics/btx662

[26] Nalluru, G., Pandey, R. and Purohit, H. (2019) Relevancy Classification of Multimodal Social Media Streams for Emergency Services. 2019 *IEEE International Conference on Smart Computing* (*SMARTCOMP*), Washington DC, 12-15 June 2019, 121-125. https://doi.org/10.1109/SMARTCOMP.2019.00040

[27] Chen, P., Deng, Y.M., Zhang, X.G., Ma, L., Yan, Y.L., Wu, Y.F. and Li, C.S. (2022) Degradation Trend Prediction of Pumped Storage Unit Based on MIC-LGBM and VMD-GRU Combined Model. *Energies*, **15**, Article 605. https://doi.org/10.3390/en15020605

[28] Liang, W.Z., Luo, S.Z., Zhao, G.Y. and Wu, H. (2020) Predicting Hard Rock Pillar Stability Using GBDT, XGBoost, and LightGBM Algorithms. *Mathematics*, **8**, Article 765. https://doi.org/10.3390/math8050765

[29] Machado, M.R., Karray, S. and de Sousa, I.T. (2019) Lightgbm: An Effective Decision Tree Gradient Boosting Method to Predict Customer Loyalty in the Finance Industry. 2019 14*th International Conference on Computer Science & Education* (*ICCSE*), Toronto, 19-21 August 2019, 1111-1116. https://doi.org/10.1109/ICCSE.2019.8845529

[30] Cheng, W., Li, J.L., Xiao, H.C. and Ji, L.N. (2022) Combination Predicting Model of Traffic Congestion Index in Weekdays Based on LightGBM-GRU. *Scientific Re-*

ports, **12**, Article No. 2912. https://doi.org/10.1038/s41598-022-06975-1

[31] Hao, X.C., Zhang, Z.P., Xu, Q.Q., Huang, G.L. and Wang, K. (2022) Prediction of f-CaO Content in Cement Clinker: A Novel Prediction Method Based on LightGBM and Bayesian Optimization. *Chemometrics and Intelligent Laboratory Systems*, **220**, Article ID: 104461. https://doi.org/10.1016/j.chemolab.2021.104461

[32] Shahriar, S.A., Kayes, I., Hasan, K., Hasan, M., Islam, R., Awang, N.R., Hamzah, Z., Rak, A.E. and Salam, M.A. (2021) Potential of ARIMA-ANN, ARIMA-SVM, DT and CatBoost for Atmospheric $PM_{2.5}$ Forecasting in Bangladesh. *Atmosphere*, **12**, Article 100. https://doi.org/10.3390/atmos12010100

[33] Kohavi, R. and Li, C.H. (1995) Oblivious Decision Trees, Graphs, and Top-Down Pruning. *Proceedings of the* 14*th International Joint Conference on Artificial Intelligence*, Montreal, 20-25 August 1995, 1071-1077.

[34] Langley, P. and Sage, S. (1994) Oblivious Decision Trees and Abstract Cases. Working Notes of the AAAI-94 Workshop on Case-Based Reasoning, Seattle, 113-117.

[35] Ferov, M. and Modr`y, M. (2016) Enhancing LambdaMART Using Oblivious Trees. arXiv: 1609.05610.

[36] Gulin, A., Kuralenok, I. and Pavlov, D. (2011) Winning the Transfer Learning Track of Yahoo!'s Learning to Rank Challenge with YetiRank. *Proceedings of the* 2010 *International Conference on Yahoo*! *Learning to Rank Challenge*, Haifa, 25 June 2010, 63-76.

[37] Dorogush, A.V., Ershov, V. and Gulin, A. (2018) Catboost: Gradient Boosting with Categorical Features Support. arXiv: 1810.11363.

[38] Sibindi, R., Mwangi, R.W. and Waititu, A.G. (2022) A Boosting Ensemble Learning Based Hybrid Light Gradient Boosting Machine and Extreme Gradient Boosting Model for Predicting House Prices. *Engineering Reports*, **5**, e12599. https://doi.org/10.1002/eng2.12599

[39] Patel, V., Choe, S. and Halabi, T. (2020) Predicting Future Malware Attacks on Cloud Systems Using Machine Learning. 2020 *IEEE* 6*th Intl Conference on Big Data Security on Cloud* (*BigDataSecurity*), *IEEE Intl Conference on High Performance and Smart Computing*, (*HPSC*) *and IEEE Intl Conference on Intelligent Data and Security* (*IDS*), Baltimore, 25-27 May 2020, 151-156. https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00036

[40] Pace, R.K. and Barry, R. (1997) Sparse Spatial Autoregressions. *Statistics & Probability Letters*, **33**, 291-297. https://doi.org/10.1016/S0167-7152(96)00140-X

[41] Matthews, S. and Hartman, B. (2021) mSHAP: SHAP Values for Two-Part Models. *Risks*, **10**, Article 3. https://doi.org/10.3390/risks10010003

[42] Zhang, J. and Chen, L. (2019) Clustering-Based Undersampling with Random over Sampling Examples and Support Vector Machine for Imbalanced Classification of Breast Cancer Diagnosis. *Computer Assisted Surgery*, **24**, 62-72. https://doi.org/10.1080/24699322.2019.1649074

[43] Shelke, M.S., Deshmukh, P.R. and Shandilya, V.K. (2017) A Review on Imbalanced Data Handling Using Undersampling and Oversampling Technique. *International Journal of Recent Trends in Engineering and Research*, **3**, 444-449. https://doi.org/10.23883/IJRTER.2017.3168.0UWXM

[44] Dutta, D., Paul, D. and Ghosh, P. (2018) Analysing Feature Importances for Diabetes Prediction Using Machine Learning. 2018 *IEEE* 9*th Annual Information Technology*, *Electronics and Mobile Communication Conference* (*IEMCON*), Vancouver, 1-3 November 2018, 924-928. https://doi.org/10.1109/IEMCON.2018.8614871

[45] Maniruzzaman, M., Rahman, M.J., Al-MehediHasan, M., Suri, H.S., Abedin, M.M., El-Baz, A. and Suri, J.S. (2018) Accurate Diabetes Risk Stratification Using Machine Learning: Role of Missing Value and Outliers. *Journal of Medical Systems*, **42**, Article No. 92. https://doi.org/10.1007/s10916-018-0940-7

[46] Anand, A. and Shakti, D. (2015) Prediction of Diabetes Based on Personal Lifestyle Indicators. 2015 1*st International Conference on Next Generation Computing Technologies* (*NGCT*), Dehradun, 4-5 September 2015, 673-676. https://doi.org/10.1109/NGCT.2015.7375206

[47] Patil, R. and Shah, K. (2023) Machine Learning in Healthcare: Applications, Current Status, and Future Prospects. In: Mangla, M., Shinde, S.K., Mehta, V., Sharma, N. and Mohanty, S.N., Eds., *Handbook of Research on Machine Learning*, Apple Academic Press, New York, 163-186. https://doi.org/10.1201/9781003277330-8

[48] Mujumdar, A. and Vaidehi, V. (2019) Diabetes Prediction Using Machine Learning Algorithms. *Procedia Computer Science*, **165**, 292-299. https://doi.org/10.1016/j.procs.2020.01.047

[49] Tigga, N.P. and Garg, S. (2020) Prediction of Type 2 Diabetes Using Machine Learning Classification Methods. *Procedia Computer Science*, **167**, 706-716. https://doi.org/10.1016/j.procs.2020.03.336

[50] Kowsher, M., Turaba, M.Y., Sajed, T. and Rahman, M.M.M. (2019) Prognosis and Treatment Prediction of Type-2 Diabetes Using Deep Neural Network and Machine Learning Classifiers. 2019 22*nd International Conference on Computer and Information Technology* (*ICCIT*), Dhaka, 18-20 December 2019, 1-6. https://doi.org/10.1109/ICCIT48885.2019.9038574

[51] Muhammad, L.J., Algehyne, E.A. and Usman, S.S. (2020) Predictive Supervised Machine Learning Models for Diabetes Mellitus. *SN Computer Science*, **1**, Article No. 240. https://doi.org/10.1007/s42979-020-00250-8