CAUCHY – Jurnal Matematika Murni dan Aplikasi
Volume 11 (1) (2026), Pages 45-60
p-ISSN: 2086-0382; e-ISSN: 2477-3344

Triple-Mutation Bat Algorithm–Optimized Extreme Learning Machine for Fetal Health Classification

Prabowo Wisnumurti, Syaiful Anam*, and Mohammad Muslikh

Brawijaya University, Malang, Indonesia

Abstract

Fetal health assessment is essential for preventing perinatal complications, yet manual interpretation of cardiotocography (CTG) signals is prone to variability and diagnostic delays. This study introduces TMBA–ELM, a hybrid intelligent model that optimizes Extreme Learning Machine (ELM) parameters using the Triple Mutation Bat Algorithm (TMBA). The novelty of this work lies in extending TMBA—originally designed for continuous optimization—into a mixed-variable optimization framework that simultaneously tunes the hidden-node size and the activation function. This is achieved through the integrated use of Cauchy, Gaussian, and time-based mutation strategies, representing the first adaptation of TMBA for ELM parameter optimization and its first application to CTG-based fetal health classification. The model was evaluated on an imbalanced CTG dataset comprising 2,126 samples and benchmarked against BA-ELM, EMD-FA-ELM, and PSO-EM-ELM. TMBA-ELM achieved 89.23% $\pm$ 0.44% accuracy, outperforming BA-ELM (ELM models with parameters tunned by ELM) with accuracy 87.37%$\pm$0.63%, PSO-EM-ELM (Error-minimizaed-ELM parameters tunned with particle swarm optimization) with accuracy 82.76% $\pm$ 1.83%, and EMD-FA-ELM (ELM parameters tunned with firefly algorithm and data decompositioned by empirical decomposition) with accuracy 87.76% $\pm$ 1.95%. However, TMBA-ELM required 164.23 $\pm$ 12.76 seconds of computation time, which is substantially higher than BA-ELM and PSO-EM-ELM with computing time 60.9 $\pm$ 10.24 seconds and 59.69 $\pm$ 5 seconds, respectively. Overall, TMBA-ELM provides improved accuracy compared with existing ELM-based models, while its increased computational cost represents a limitation for time-constrained applications.

Keywords: Fetal Health Classification; CTG; TMBA-ELM

1 Introduction

The fetus is an embryo over two months old during the pregnancy process. The fetus is also called a baby candidate. Sometimes, a fetus has abnormalities during the pregnancy process, such as miscarriage. Miscarriage not only has consequences for the emotional family breakdown but also for mothers' health and society's well-being, so miscarriage is often a challenge in global health discussions [1]. Fetal development monitoring is one of the most difficult challenges and complex processes in the medical world. 810 pregnant women die in one day even though early examinations have been carried out approximately according to data from the World Health Organization [1]. The Pregnant Mother Mortality Rate (MMR) in countries with data is very

[*] **Corresponding author**. E-mail: syaiful@ub.ac.id

Submitted: November 11, 2025 Reviewed: November 27, 2025 Accepted: December 21, 2025
DOI: https://doi.org/10.18860/cauchy.v11i1.37525

low, but high in countries that are not monitored. Common complications behind high MMR rates are preeclampsia, improper handling of the fetus and pregnant mother, and gestational diabetes. MMR can be treated and reduced with proper medical treatment. Standard treatment carried out in the third trimester is fetal monitoring which is carried out by examining the health of the prospective baby. Fetal growth affects the biological mother's health [2].

Fetal health can be measured by CTG. Parameters that can be used for measurement are: fetal heart rate, fetal movement, and uterine contraction. Necessary intervention assisted by that information. Worse results are fetal distress, birth asphyxia, and mother complications indirectly detected by CTG. Besides that, child risks and MMR can be reduced by CTG. Prenatal care is represented globally by the utilization of CTG data for automated fetal health tests. Simply put, CTG have several limitations. The limitation of manual CTG analysis is time-consuming evaluation. Obstericians must analyze multiple complex signal patterns—including fetal heart rate variability, accelerations, decelerations, and histogram characteristics. This complexity reduces efficiency, particularly when dealing with large volumes of data or ambiguous signal patterns. This limitations underscore the need for automated, machine-learning-based systems capable of providing faster evaluation time [1].

Machine learning is extracting knowledge from data machines [3]. Supervised learning, unsupervised learning, and reinforcement learning are three types of machine learning [4]. Supervised learning is machine learning that needs labeled data for the training process to obtain an optimal model [3]. Some machine learning applications in daily life are: thermal barrier coatings thickness prediction [5], rolling bearing classification [6], breast cancer type classification [7], and lithium-ion batteries health estimation [8]. Supervised learning has two types, that is classification and regression [3]. The classification used for data labeling [3]. One example of classification is fetal health classification.

Extreme Learning Machine (ELM) is a fast and efficient feedforward learning model widely applied in classification tasks [9]. However, its performance is highly dependent on the selection of hidden nodes and activation functions. Inappropriate parameter settings often lead to unstable convergence and reduced accuracy, making parameter optimization essential for robust ELM performance. Metaheuristic algorithms have been used to address this issue. The Bat Algorithm (BA), in particular, has been reported as an efficient optimizer due to its simplicity and minimal parameter requirements [10]. When integrated with ELM, BA-ELM has shown the ability to improve classification accuracy and stabilize convergence by effectively tuning hidden-node configurations. In several studies, BA-ELM outperforms standard ELM and other metaheuristic–ELM combinations due to BA's adaptive frequency and loudness mechanisms, which help maintain a balance between exploration and exploitation. However, BA-ELM still suffers from notable limitations. The standard BA search process tends to lose population diversity and easily falls into premature convergence, especially in high-dimensional or multimodal optimization problems [11]. To improve BA, the Triple Mutation Bat Algorithm (TMBA) was introduced by integrating Cauchy, Gaussian, and position-based mutation strategies [12]. These adaptations allow TMBA to optimize ELM parameters beyond continuous and discrete domains. In addition, the novelty introduced in this study remains consistent with previous research directions on TMBA and BA-ELM. TMBA was originally developed to enhance exploration and avoid premature convergence, while BA-ELM studies demonstrated that metaheuristic-based tuning can significantly improve ELM performance. The proposed TMBA-ELM builds upon these established strengths—leveraging TMBA's mutation-based exploration and the proven effectiveness of BA-type optimizers for ELM tuning—while extending them to mixed-variable optimization for the first time. This effect on ELM model while tunned with TMBA, TMBA can find the good parameters for ELM. In fact, existing TMBA has not been adapted for ELM optimization or applied to fetal health classification. This unaddressed limitation forms a clear research gap: there is no existing TMBA variant capable of optimizing discrete and categorical ELM parameters, and no study has investigated its application to fetal health classification based on CTG data. To fill this gap, the present study proposes a modified TMBA–ELM framework that enables TMBA to operate on mixed-variable search spaces. The modifications include: rounding the optimized hidden-node values to integers, encoding activation functions as integer indices during optimization, and decoding them back to categorical labels after optimization. The parameter optimization performed by TMBA primarily improves the overall classification accuracy and convergence stability of the ELM model. However, the confusion matrix in Table 6 indicates that classification of minority classes (suspect and pathological) remains challenging. A substantial number of suspect and pathological cases are still misclassified as normal, which reflects the inherent difficulty of learning minority-class decision boundaries in imbalanced CTG data. Although the metaheuristic tuning enhances global model performance, it does not fully mitigate minority-class misclassification. Thus, the proposed model provides a consistent and methodologically sound extension of TMBA to mixed-variable optimization and introduces its first application to fetal health classification.

This study aims to develop a TMBA-ELM model for classifying fetal health based on CTG data and to evaluate its performance in comparison to ELM and other combinations of ELM models. The contributions of this research are: parameter tuning with a metaheuristic algorithm improves classification accuracy and introducing Cauchy and Gaussian mutation to modify ELM based on bat algorithm optimization.

2 Methods

The procedures in this research have many steps. All procedure in this research using python programming application. The first step is to collect the dataset. The dataset used in this research is fetal health data. The second step is data learning, which consists: checking feature data type, checking data values that out-of-range values, and checking the data outliers with the winsorized mean (the average computed after limiting extreme values to reduce the influence of outliers). The third step is data preprocessing. The fourth step is applying TMBA-ELM, EMD-FA-ELM, BA-ELM, and PSO-EM-ELM to the dataset and running it 25 times to obtain a population of model performances and computation time. EMD-FA-ELM is chosen because it can reduce ELM randomness [13], BA-ELM is chosen because it fast convergences [11], PSO-EM-ELM is chosen because training convergence accelerated [14]. Parameters value of TMBA-ELM can be seen in Table 1. The maximum number of generations was set to 50 because this value is a sufficient iteration budget to achieve TMBA convergence on benchmark functions. The population size was fixed at 10 because TMBA maintains stable global–local search balance even with a small swarm while keeping the computational cost low. The decreasing amplitude coefficient α is 0.7 was selected because values in the range 0.5–0.9 have been shown to ensure a smooth decay of loudness, allowing the algorithm to gradually transition from exploration to exploitation. The increasing pulse-rate coefficient γ is 1 was adopted because this setting yields a fast but stable growth of the pulse rate, a behavior consistently recommended in earlier BA performance analyses. The minimum frequency $f_{min} = 0$ was chosen to enable broad exploration by allowing bats to take large position jumps during early iterations. Finally, the maximum frequency $f_{max} = 2$ was used because this upper bound provides controlled step sizes suitable for continuous optimization tasks [12]. TMBA-ELM are tunning of hidden nodes (between 5 - 200) [9] and activation function (sigmoid, ReLU, and tanh). Sigmoid, ReLU, and tanh functions can be seen in Eq. (1), Eq. (2), and Eq. (3), respectively.

$$sgn(x) = \frac{1}{1+e^{-x}} \tag{1}$$

$$ReLU(x) = max(0, x) \tag{2}$$

$$tanh(x) = \frac{2}{1+e^{-2x}} \tag{3}$$

Sigmoid function is chosen because its function is smooth and stable for simple feedforward neural network [9]. Tanh function is chosen because its function have strong gradient for small absolute x [9]. ReLU function is chosen because its sparsity, fast computing, avoid vanishing gradient, and well-used for big data [15]. The fifth step is to compare model accuracy and time computing by performing two populations and doing hypothesis test. The sixth step is to compare model computation time and doing hypothesis test. All the processes of this research can be seen in Figure 1.

Table 1: Parameters of TMBA-ELM

Parameter	**Value**	**Source**
Numer of generations	50	[12]
Number of bats	10	[12]
Decreasing amplitude coefficient (α)	0.7	[12]
Increasing pulse rate coefficient (γ)	1	[12]
Minimum frequency	0	[12]
Maximum frequency	2	[12]

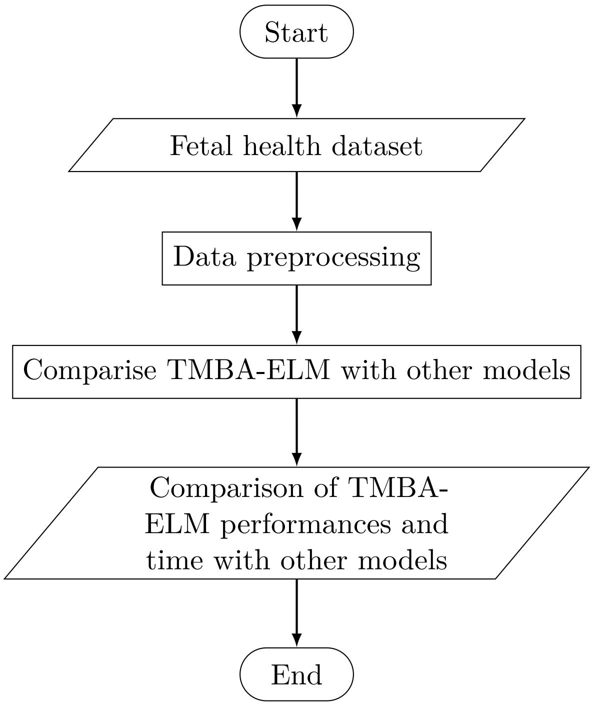
Start
Fetal health dataset
Data preprocessing
Comparise TMBA-ELM with other models
Comparison of TMBA-ELM performances and time with other models
End

Figure 1: Research stage

2.1 Dataset

The dataset is in Excel format and was obtained from an open-source platform, Kaggle.com [16]. It contains 20 numerical input features and one target feature. The target feature represents fetal health condition and is categorized into three classes: normal (0), suspect (1), and pathological (2). All 20 input features are of numerical type and are listed in Table 2. The dataset comprises a total of 2,126 samples, distributed in an imbalanced manner across the three fetal health classes: 1,655 samples (77.87%) are labeled as normal, 295 samples (13.87%) as suspect, and 176 samples (8.27%) as pathological. The dataset have no null, NaN, and outliers values.

Table 2: Data Features with Logic Range

No.	Feature	unit	Type	Range	Source	Skewness
1.	FHR baseline value	bpm	float	60 - 160	[17], [18]	0.02
2.	Accelerations	Event/sec	float	< 0.025	[17], [19]	1.2
3.	Fetal Movement	Event/sec	float	< 0.06	[20]	7.8
4.	Uterine Contraction	Event/sec	float	< 0.083	[17]	0.16
5.	Light Decelerations	Event/sec	float	< 0.015	[17]	1.72
6.	Severe Decelerations	Event/sec	float	< 0.002	[17]	17.34
7.	Prolonged Decelerations	Event/sec	float	< 0.005	[17]	4.32
8.	Percentage of Abnormal STV	%	float	< 70	[17]	-0.01
9.	STV Average Value	ms	float	< 8	[17]	1.66
10.	Percentage of Abnormal LTV	%	float	< 70	[17]	2.19
11.	LTV Average Value	ms	float	< 25	[17]	1.33
12.	FHR Histogram Width	bpm	float	$10 - 150$	[17], [19]	0.31
13.	FHR Histogram Minimum Value	bpm	float	$80 - 110$	[21]	0.12
14.	FHR Histogram Maximum Value	bpm	float	$150 - 190$	[22]	0.58
15.	FHR Histogram Peaks	Number	int	$1 - 10$	[23]	0.89
16.	Value of FHR Histogram with Zero Frequency	Number	int	< 20	[17], [23]	3.92
17.	FHR Histogram Mode	bpm	float	$80 - 200$	[17], [19], [23]	-0.99
18.	FHR Histogram Mean	bpm	float	$90 - 180$	[17], [18]	-0.65
19.	FHR Histogram Median	bpm	float	$90 - 180$	[17], [18]	-0.48
20.	FHR Histogram Variance	bpm^2	float	< 400	[17], [23]	3.21
21.	FHR Histogram Tendency		int			
22.	Fetal Health		int			

2.2 Data Preprocessing

Before applying TMBA-ELM model, dataset must be preprocessed. This plays a critical role in ensuring the quality and validity of the dataset before further analysis. Data preprocessing begins with out-of-logical-range values handling. Out-of-range value on feature no.3, 10-12, 15, and 17 on Table 2 handled by replace it value by feature median. Out-of-range value on feature no.8, 13, 14, 18, and 19 on Table 2 handled by replace it value by feature mean [24]. Then, categorical feature be converted (FHR histogram tendency feature was mapped as follows: -1 to “left tendency”, 0 to “center tendency”, and 1 to “right tendency”. Fetal health feature was mapped as follows: 0 to “normal”, 1 to “suspect”, and 2 to “pathological”), data splitting (70% for data training and 30% for data testing with random state value is 42, because it have same values for training and testing dataset [25]), imbalanced data training handling with SMOTE (Synthetic Minority Oversampling Techniques) [26], and encoding catrgorical value using one-hot-encoding. Finally, Min-Max scaler normalization applied in to dataset. The formula for Min-Max scaling of training data numerical feature is provided in Eq. (4) [27].

$$Z_{m,n} = \frac{dat_{m,n} - dat_{min,n}}{dat_{max,n} - dat_{min,n}}, \tag{4}$$

with $dat_{m,n}$ is m-th observation and n-th feature of dataset, $dat_{min,n}$ is n-th feature minimum value, and $dat_{max,n}$ is j-th feature maximum value. Normalization of numerical features in testing data using Eq. (4), $dat_{max,n}$ and $dat_{min,n}$ values are same as for data training normalization.

Due to the highly imbalanced class distribution, stratified 10-fold cross-validation is employed to ensure that the proportion of normal, suspect, and pathological samples is preserved in each fold [28]. It should be emphasized that stratified cross-validation does not resolve class imbalance, but rather provides a more reliable and unbiased estimation of model performance across all classes. To explicitly address the imbalance during training, the SMOTE oversampling technique is applied to the training dataset only, thereby increasing the representation of minority classes without discarding majority-class samples [26].

2.3 TMBA-ELM Model

TMBA-ELM is a modification of ELM by tuning number of nodes and activation function using TMBA. Activation function must be tunned because activation function selection effect to ELM ability in mapping complex pattern [9]. Number of hidden nodes must be tunned because number of hidden nodes determined model representation capacity (not enough hidden nodes causes underfitting, and to much hidden nodes causes overfitting) [9]. TMBA-ELM is a combination of BA by modificating microbats initial position, microbats position, and microbats best position.

Microbats initial position modificated by Eq. (5).

$$L = 1 - \exp\left(-\left(\frac{t_{max} - t}{t_{max} + 0.6t}\right)^{10}\right), \tag{5}$$

$$\overrightarrow{pos}_i^t = \overrightarrow{pos}_i^{t-1} + L\vec{v}_i^t, \tag{6}$$

where t_{max} denotes the maximum number of iterations. A time factor modification is applied to balance exploration and exploitation by adaptively adjusting the flight time throughout the iterative process [12]. Microbat position when a randomly generated number is less than or equal to the pulse rate modificated by Eq. (7).

$$\overrightarrow{pos}_i^t = \overrightarrow{pos}_i^t\left(1 + Cauchy(0,1)\right), \tag{7}$$

where $Cauchy(0,1)$ is a random number chosen from the distribution from Eq. (8).

$$C(x) = \frac{1}{2} + \frac{1}{\pi}\arctan x, \tag{8}$$

where $x \in \mathbb{R}$. A Cauchy mutation operator is used to enhance global search capability by enabling large perturbations that help escape local optima [12]. Microbats best position modificated by Eq. (9).

$$\overrightarrow{pos}_{best}^* = \overrightarrow{pos}_{best}\left(1 + Gauss(0,1)\right), \tag{9}$$

where $Gauss(0,1)$ is a random number chosen from the distribution from Eq. (10).

$$G(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right), \tag{10}$$

where $x \in \mathbb{R}$. A Gaussian mutation operator is used to strengthen local exploitation by introducing small, fine-tuned perturbations around the global best solution [12].

Besides three microbats postion modification, bats in TMBA-ELM are positioned in a two-dimensional search space ($\overrightarrow{pos} = (pos1, pos2)$), where the $pos1$-coordinate represents the number of ELM nodes, and the $pos2$-coordinate represents the encoded ELM activation function: sigmoid is encoded as 0.0, ReLU as 1.0, and tanh as 2.0. The activation function must be encoded into a numerical (float) form to allow TMBA's position-update equations—originally defined for continuous coordinates—to process this categorical choice within the search space. This numerical value, however, serves purely as a categorical label rather than a continuous variable; TMBA does not interpret or optimize it as a real-valued parameter. After each update, the $pos2$-coordinate is discretized and mapped back to the corresponding activation function class. Objective function $obj(pos1', pos2')$ in TMBA-ELM is the accuracy of ELM model with 10 fold cross validation, $pos1' = round(pos1)$ and $pos2' = trans(pos2)$ is the number of hidden node and activation function, respectively. Function $trans(x)$ formulated as Eq. (11).

$$trans(x) = \begin{cases} sigmoid, & \text{if } round(x) = 0 \\ ReLU, & \text{if } round(x) = 1 \\ tanh, & \text{if } round(x) = 2 \end{cases} \tag{11}$$

Algorithm 1 ELM with 10 fold cross validation

```
1: Initialize data train $\boldsymbol{dat}$ in to 10 folds ($\boldsymbol{dat} = \boldsymbol{dat}_1 \cup \boldsymbol{dat}_2 \cup \cdots \cup \boldsymbol{dat}_{10}$)
2: for each $k = 1$ to 10 do
3:     Train and test ELM with data test $\boldsymbol{dat}_k$ and data test is $\boldsymbol{dat}/\boldsymbol{dat}_k$, then accuracy of ELM obtained, that is $Acc_k$
4: end for
5: Return the accuracy mean ($Acc = 0.1(Acc_1 + Acc_2 + \cdots + Acc_{10})$)
```

The goal of TMBA-ELM is maximizing objective function. The steps of ELM with 10 fold cross validation can be seen in Algorithm 1. The ELM model can be seen in Eq. (12).

$$\boldsymbol{T'} = \boldsymbol{H'}\left(\left(\boldsymbol{H}^T\boldsymbol{H}\right)^{-1}\boldsymbol{H}^T\boldsymbol{T_0}\right), \tag{12}$$

where

$$\boldsymbol{H} = \begin{bmatrix} pos2'(\vec{w}_1 \cdot \overrightarrow{dat}_1 + b_1) & \cdots & pos2'(\vec{w}_{pos1'} \cdot \overrightarrow{dat}_1 + b_{pos1'}) \\ \vdots & \ddots & \vdots \\ pos2'(\vec{w}_1 \cdot \overrightarrow{dat}_N + b_1) & \cdots & pos2'(\vec{w}_{pos1'} \cdot \overrightarrow{dat}_N + b_{pos1'}) \end{bmatrix}_{N \times pos1'}, \tag{13}$$

$$\boldsymbol{T_0} = \begin{bmatrix} t_{1,1} & \cdots & t_{1,d} \\ \vdots & \ddots & \vdots \\ t_{N,1} & \cdots & t_{N,d} \end{bmatrix}_{N \times d} \tag{14}$$

$$\boldsymbol{H}' = \begin{bmatrix} pos2'(\vec{w}_1 \cdot \overrightarrow{dat}_{N+1} + b_1) & \cdots & pos2'(\vec{w}_{pos2} \cdot \overrightarrow{dat}_{N+1} + b_{pos1'}) \\ \vdots & \ddots & \vdots \\ pos2'(\vec{w}_1 \cdot \overrightarrow{dat}_{N+N'} + b_1) & \cdots & pos2'(\vec{w}_{pos1} \cdot \overrightarrow{dat}_{N+N'} + b_{pos1'}) \end{bmatrix}_{N' \times pos1'}, \tag{15}$$

$\left\{\left(\overrightarrow{dat}_m, \vec{t}_m\right)\right\}_{m=1}^{N}$ are the samples of learning data, $\left\{\left(\overrightarrow{dat}_m, \vec{t}_m\right)\right\}_{m=N+1}^{N+N'}$ are the samples of testing data, $\overrightarrow{dat}_m = \left[dat_{m,1}, \cdots, dat_{m,n'}\right]^T \in \mathbb{R}^{n'}\ \forall m = 1, \cdots, N + N'$, $\vec{t}_m = [t_{m,1}, \cdots, t_{m,d}]^T \in \mathbb{R}^d\ \forall m = 1, \cdots, N + N'$, and $\vec{w}_l = [w_{l,1}, \cdots, w_{l,n'}]^T \in \mathbb{R}^{n'}\ \forall l = 1, \cdots, pos1'$ is the weight vector [9]. Thus, all of TMBA-ELM algorithm can be seen in Algorithm 2.

TMBA-ELM using three modifications at once. If modification applied one by one, model have bias in exploration or exploitation. If modifications applied at once, population diversity can be maintained, premature convergence can be avoided, and convergence rate toward a stable global optimum solution can be enhanced [12]. Therefore, these combination leverages the global search ability of the BA, enhanced through specific tuning mechanisms designed to improve convergence speed and solution quality. By incorporating these rules, TMBA-ELM aims to overcome the limitations of standard ELM in terms of generalization performance and robustness across various datasets.

2.4 Hypothesis Test

Hypothesis testing is employed to determine whether the performance differences between TMBA-ELM and the comparative models are statistically significant. An independent two-sided t-test is applied at a significance level of $\alpha = 0.05$. The hypotheses are formulated as follows:

$$H_0 : \mu_1 = \mu_2 \tag{16}$$

$$H_1 : \mu_1 \neq \mu_2 \tag{17}$$

Algorithm 2 TMBA-ELM model

```
Initialize bat population $\vec{pos}_i(pos1_i, pos2_i)$ where $pos1_i$ = number of hidden nodes and $pos2_i$
    = activation function index
Initialize velocity $\vec{v}_i$, frequency $f_i$, loudness $A_i$, and pulse rate $r_i$
Evaluate initial fitness $obj(\vec{pos}_i)$ as the classification accuracy average of ELM with 10 fold
    cross validation of training data.
Determine the best solution $\overrightarrow{pos}_{best}$ and its fitness value $obj_{best}$
for each iteration $t = 1$ to $t_{max}$ do
    for each bat $i$ do
        $f_i \leftarrow f_{min} + \beta\,(f_{max} - f_{min})$, where $\beta \in [0, 1]$
        $\vec{v}_i \leftarrow \vec{v}_i + f_i(\vec{pos}_i - \overrightarrow{pos}_{best})$
        Update $pos_i$ with Eq. (6)
        if $rand() > r_i$ then
            Generate a local solution around $\overrightarrow{pos}_{best}$
        else
            Update $\vec{pos}_1$ with Cauchy mutation
        end if
        $pos1'_i \leftarrow round(pos1_i)$
        $pos2'_i \leftarrow trans(pos2_i)$, where $trans(x)$ described in Eq. (11)
        Compute fitness $obj(pos1'_i, pos2'_i)$ based on Algorithm 1 with $pos1'_i$ hidden nodes and
            $pos2'_i$ is activation function for all layers.
        if $obj(pos1'_i, pos2'_i)$ is maximum and $rand() < A_i$ then
            $\overrightarrow{pos}_{best} \leftarrow \overrightarrow{pos}_i$
            $r_i \leftarrow r_i\,(1 - \exp(-\gamma t))$, where $\gamma > 1$
            $A_i \leftarrow \alpha A_i$, where $\alpha \in (0, 1)$
        end if
        if $obj(pos1'_i, pos2'_i)$ is maximum then
            $\overrightarrow{pos}_{best} \leftarrow \overrightarrow{pos}_i$
        else
            Do Gauss mutation to $\overrightarrow{pos}_{best}$ (suppose the result is $\overrightarrow{postG}_{best}$ =
                $(postGbest1, postGbest2)$)
            $postGbest1'_i \leftarrow round(postGbest1_i)$
            $postGbest2'_i \leftarrow trans(postGbest2_i)$, where $trans(x)$ described in Eq. (11)
            Compute fitness $obj(postGbest1'_i, postGbest2'_i)$ based on Algorithm 1, with $postGbest1'_i$
                hidden nodes and $postGbest2'_i$ is activation function for all layers.
            if $obj(postGbest1'_i, postGbest2'_i)$ is the maximum solution then
                $\overrightarrow{pos}_{best} \leftarrow \overrightarrow{postG}_{best}$
            end if
        end if
    end for
end for
Return the optimal ELM parameters $\overrightarrow{pos}_{best} = (posbest1, posbest2)$.
Train and test ELM with $round(posbest1)$ hidden nodes and activation function
    $trans(posbest2)$, where $trans(x)$ defined in Eq. (11).
```

where μ_1 and μ_2 denote the mean performance (classification accuracy or computational time) of TMBA-ELM and the compared model, respectively. The t-test statistic is computed using Eqs. (16)–(17), with the degrees of freedom defined as:

$$df = n_1 + n_2 - 2. \tag{18}$$

For a two-sided test with $\alpha = 0.05$ and $df = 48$, the critical value is:

$$t_{\alpha/2,df} = t_{0.025,48} = 2.01. \tag{19}$$

For a two-sided independent t-test at a significance level of $\alpha = 0.05$, the null hypothesis $H_0 : \mu_1 = \mu_2$ is rejected in favor of the alternative hypothesis $H_1 : \mu_1 \neq \mu_2$ if the absolute value of the test statistic satisfies $|t_{test}| > t_{\alpha/2,df} = 2.01$. Otherwise, H_0 is not rejected. This decision rule is consistently applied to evaluate the statistical significance of differences in both classification accuracy and computational time between TMBA-ELM and the benchmark models.

Table 3: Decision rule for Two-sided Independent t-test ($\alpha = 0.05$, $df = 48$)

Condition Based on t-test Result	**Explanation**	**Statistical Decision**
The calculated t-test value lies within the non-rejection region	The difference between the two models is not statistically significant	Fail to reject the null hypothesis (H_0)
The calculated t-test value lies outside the non-rejection region	The difference between the two models is statistically significant	Reject the null hypothesis (H_0) and accept the alternative hypothesis (H_1)

3 Results and Discussion

Before applying any machine learning algorithm, the dataset must undergo a series of preprocessing steps to ensure data quality and compatibility with the selected models. Upon completion of preprocessing, the dataset is typically divided into two distinct subsets: one for training the model (1488 observations) and the other for evaluating its performance (638 observations). The categorical attributes contained within both the training and testing subsets are summarized and presented in Table 4, providing an overview of the distribution and variety of discrete features. Meanwhile, the corresponding numerical attributes—essential for capturing quantitative patterns—are detailed in Table 5, which illustrates their respective ranges, distributions, and statistical properties.

Table 4: Dataset Categorical Features

Feature	**Proportion in Train Data**	**Proportion in Test Data**
Left Tendency of FHR Histogram	7.73%	7.84%
Center Tendency of FHR Histogram	52.89%	51.41%
Right Tendency of FHR Histogram	39.38%	40.75%
Normal Fetal	80.78%	71%
Suspectible Fetal	12.16%	17.87%
Pathological Fetal	7.06%	11.13%

Table 4 presents the proportion of features in both the training and testing datasets. For FHR histogram tendencies feature, left tendency of FHR histogram feature have slightly minimum proportion (7.73% in training data and 7.84% in testing data) than center tendency histogram (52.89% in training data and 51.41% in testing data) and right tendency histogram (39.38% in training data and 40.75% in testing data). For fetal health feature, normal fetal have slightly

Table 5: Dataset Numerical Feature Values

Feature	Data Train Range	Data Test Range
FHR baseline value	106 – 160	110 – 159
Accelerations	0 – 0.018	0 – 0.019
Fetal Movement	0 – 0.481	0 – 0.477
Uterine Contraction	0 – 0.015	0 – 0.014
Light Decelerations	0 – 0.015	0 – 0.015
Severe Decelerations	0 – 0.001	0 – 0.001
Prolonged Decelerations	0 – 0.005	0 – 0.004
Percentage of Abnormal STV	12 – 87	12 – 86
STV Average Value	0.2 – 7	0.2 – 6
Percentage of Abnormal LTV	0 – 91	0 – 91
LTV Average Value	0 – 50.7	0 – 36.9
FHR Histogram Width	3 – 180	7 – 176
FHR Histogram Minimum Value	50 – 159	50 – 155
FHR Histogram Maximum Value	122 – 238	127 – 238
FHR Histogram Peaks	0 – 18	0 – 14
Value of FHR Histogram with Zero Frequency	0 – 5	0 – 10
FHR Histogram Mode	60 – 186	60 – 187
FHR Histogram Mean	73 – 182	76 – 171
FHR Histogram Median	77 – 186	79 – 176
FHR Histogram Variance	0 – 269	0 – 254

higher proportion (80.78% in training data and 71% in testing data) than suspectible fetal (12.16% in training data and 17.87% in testing data), where suspectible fetal have slightly higher proportion than pathological fetal (7.06% in training data and 11.13% in testing data). This proportion describes the data in fetal health feature as unbalanced data. Dominant proportion of normal fetal potentially causing misclassification on suspectible and pathological fetal. Thus, this feature in training data must be handled by SMOTE. After doing SMOTE, normal fetal, suspectible fetal, and pathological feta in training data have 1,202 observations.

Table 5 indicates that most numerical features show small differences in range between training and testing sets, indicating that the overall numerical features distribution is largely stable. This consistency potentially supports strong generalization at global level. But, a minority features not have subtle differences in range between training and testing sets. The dataset have 5 features that data train range wider than data test range and 1 features that data test range wider than data train range. When the training range is wider, the model becomes calibrated to a broader spectrum of values, including extremes that do not appear in the testing set. This can lead to an overestimation of the relevance of high-variation patterns and slightly weaken generalization. Conversely, when the testing range exceeds the training range, the model encounters unseen values during inference, increasing the likelihood of extrapolation errors and reducing prediction stability. Thus, further analysis of relationship between featres before applying TMBA-ELM need to be analyzed to remove weakly influential features. This analysis will be done in future research.

After the preprocessing stage, the dataset was divided into training and testing subsets using a 70/30 split, resulting in 1,488 samples for training and 638 samples for testing. Model optimization and performance estimation were conducted on the training data using stratified 10-fold cross-validation, while the final model evaluation was performed on the independent testing dataset. This strategy ensures that model optimization and performance estimation are conducted in a systematic and unbiased manner across multiple data partitions. Then, the confussion matrix applied to model-applying results. The confusion matrix presented in Table 6 is computed exclusively on the testing dataset (638 samples). The class distribution in the testing set consists of 453 normal, 114 suspect, and 71 pathological cases, which preserves the original class proportions of the full dataset.

Table 6: Confusion matrix of TMBA-ELM evaluated on the testing dataset (30% split, 638 samples)

	Normal Fetal Predicted	**Suspectible Fetal Predicted**	**Pathological Fetal Predicted**
Normal Fetal Actual	445	5	3
Suspectible Fetal Actual	31	78	5
Pathological Fetal Actual	21	4	46

The confusion matrix in Table 6 shows that TMBA-ELM correctly classified 569 out of 638 testing samples, corresponding to a test-set accuracy of 89.18%. This result indicates that the overall accuracy is largely influenced by the dominant normal class in the testing dataset. The model demonstrates strong discriminative capability for the normal class, correctly identifying 445 normal cases, thereby reducing the likelihood of unnecessary clinical interventions. However, the performance on the suspect and pathological classes remains more challenging, with 36 suspect cases and 25 pathological cases misclassified. Although some pathological cases are predicted as suspect—still triggering additional clinical monitoring—the presence of false negatives highlights limitations in minority-class recognition.

Although TMBA-ELM achieves higher overall classification accuracy and improved convergence stability, the improvement is primarily dominated by the majority (normal) class. As shown in the confusion matrix, a substantial number of suspect and pathological cases are still misclassified as normal. This indicates that metaheuristic-based parameter tuning enhances global model performance but does not guarantee robust discrimination of minority classes in highly imbalanced datasets.

After analyzing TMBA-ELM model, this model compared by other models. To ensure statistical robustness and account for variability in model performance, each model was executed 25 times under identical conditions. The average and the corresponding standard deviation of each model accuracy are visualized in Figure 2. The average computation time and the corresponding standard deviation of each model are visualized in Figure 3.

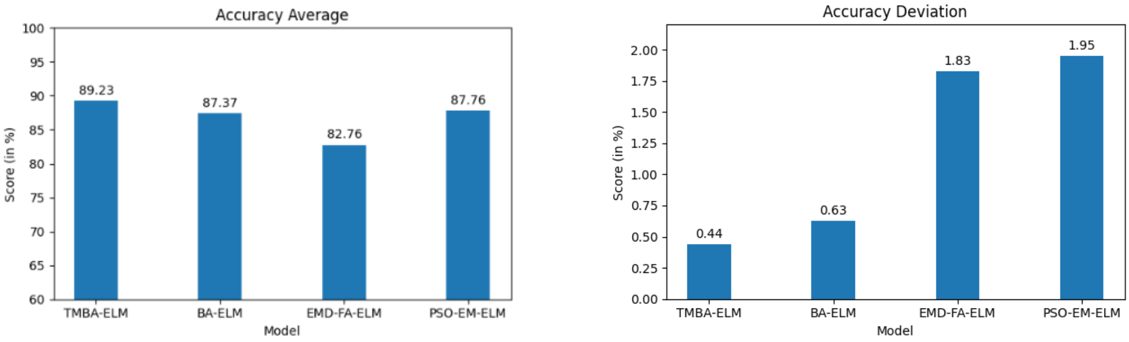

Figure 2: Model accuracy average (left side) and deviation (right side)

Figure 2 reveals the comparison of accuracy averages and deviations shows that TMBA-ELM achieves both the highest mean accuracy and the lowest variation across repeated runs, indicating superior convergence stability and robustness. Analytically, this advantage stems from TMBA's structural design, which integrates three adaptive mutation strategies—Cauchy mutation, Gaussian mutation, and time-factor–based position modification—resulting in a balanced exploration–exploitation dynamic throughout the search process. The Cauchy operator allows large jumps to escape local minima, the Gaussian operator provides fine-grained local refinement, and the time-adaptive position update stabilizes the search trajectory as iterations progress. This behavior is consistent with the findings of [12], who also reported that combining heterogeneous mutation operators with time-dependent control mechanisms significantly improves swarm-based optimization stability. In contrast, BA-ELM and PSO-EM-ELM rely on simpler update mechanisms that are more susceptible to premature convergence or sensitivity to initial conditions, leading to moderate performance and higher fluctuation. EMD-FA-ELM, which involves feature decomposition and swarm-based optimization, exhibits both lower accuracy and substantially greater deviation because the decomposition step amplifies variability between runs, and the Firefly Algorithm lacks TMBA's multi-layered control of search diversity. Consequently, the structural enhancements embedded in TMBA produce a more reliable and consistently optimal parameter-tuning process for ELM, reflected in both superior accuracy and minimal dispersion compared with the other models.

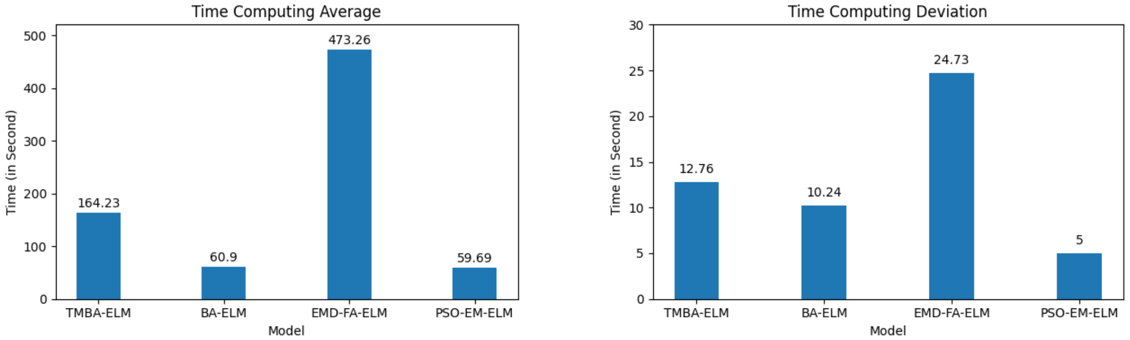

Figure 3: Model time computing average (left side) and deviation (right side)

Figure 3 presents the comparison of average computational time and its variability for all optimization-based ELM models. TMBA-ELM exhibits a higher average computation time than BA-ELM and PSO-EM-ELM, which is primarily attributed to the additional mutation operators integrated into the TMBA framework. The inclusion of Cauchy mutation, Gaussian mutation, and time-adaptive position updates increases the computational cost per iteration, as each mutation step requires extra evaluations of the ELM model. In contrast, BA-ELM and PSO-EM-ELM rely on simpler update mechanisms with fewer mutation operations, resulting in shorter execution times. EMD-FA-ELM demonstrates the highest computational cost due to its two-stage structure, where empirical mode decomposition is followed by swarm-based optimization, substantially increasing processing overhead. Overall, these results indicate that TMBA-ELM offers a trade-off between improved optimization stability and increased computational time, making it more suitable for accuracy-oriented applications rather than time-critical scenarios.

Significance different accuracy and time computing of TMBA-ELM must be tested with hypothesis test. Subsequently, the significant of TMBA-ELM model with other models must be measured by t hypothesis test with H_0 is the term that TMBA-ELM performance is not significantly different than other model performances and H_1 is the term that TMBA-ELM performance is significantly different than other model performances. Accuracy mean and deviation on Table 2 used to count t_{test} value of accuracy. Time computing mean and deviation on Table 3 used to count t_{test} value of time computing. The t_{test} of each comparison can be seen in Table 7.

Table 7: t_{test} values for each comparisons

	TMBA-ELM vs BA-ELM	**TMBA-ELM vs EMD-FA-ELM**	**TMBA-ELM vs PSO-EM-ELM**
Accuracy	8.99	21.47	4.75
Time computing	107.73	-252.36	124

From Table 7, the absolute values of the t-test statistics for all comparisons exceed the critical value of 2.01. Therefore, the null hypothesis H_0 is rejected and the alternative hypothesis H_1 is accepted for all comparisons, which means TMBA-ELM have significantly difference on accuracy and time computing than other models. Based on Figure 2, TMBA-ELM have biggest accuracy, so TMBA-ELM is the model that have best accuracy with other models, reminding TMBA-ELM have significantly different than other models. But, based on Figure 3, the model that have lower time computing average with significant different than TMBA-ELM is BA-ELM and PSO-EM-ELM. That is, TMBA-ELM is not most efficient model.

The additional mutation operations embedded in TMBA significantly increase the computational load per iteration, making the model less suitable for time-critical or resource-limited environments. But, this study focuses on overall classification accuracy and optimization stability. Class-specific performance metrics such as precision, recall, and F1-score for each class are not explicitly reported. Therefore, no definitive claim is made regarding improved performance for the suspect or pathological classes. Besides that, based on confussion matrix, TMBA-ELm cannot cassify minority class as well. It should be noted that the accuracy reported in the abstract and comparative analysis (89.23% ± 0.44%) corresponds to the mean classification accuracy obtained from stratified 10-fold cross-validation on the training dataset. In contrast, the confusion matrix represents the performance of the optimized TMBA-ELM model on a single independent testing split. The slight numerical difference between cross-validation accuracy and test-set accuracy is therefore expected due to the use of different evaluation protocols.

4 Conclusion

This study developed the TMBA-ELM model by integrating TMBA with ELM to optimize the number of hidden nodes and the activation function for fetal health classification using CTG data. Based on 25 independent experimental runs, TMBA-ELM achieved the highest classification accuracy and demonstrated the lowest performance variability among all benchmarked models, indicating strong convergence stability and reliable optimization behavior. The combination of Cauchy mutation, Gaussian mutation, and time-adaptive position adjustment effectively balanced exploration and exploitation, allowing the model to consistently identify high-quality ELM parameter configurations. Despite its accuracy advantage, TMBA-ELM presents several clear limitations. First, the model exhibits substantially higher computational time than BA-ELM and PSO-EM-ELM, making it less suitable for time-sensitive or resource-constrained applications. Second, while TMBA-ELM achieves superior overall accuracy compared with the benchmark models, the results demonstrate that the improvement is dominated by the majority (normal) class. The model does not yet provide robust discrimination for suspect and pathological classes, as reflected by the remaining false-negative cases. This limitation is consistent with the imbalanced nature of the dataset and indicates that metaheuristic-based parameter tuning alone is insufficient to guarantee strong minority-class performance. Future research should incorporate class-sensitive evaluation metrics such as precision, recall, and F1-score for each class, as well as cost-sensitive learning or imbalance-aware optimization strategies. These approaches are expected to improve minority-class recognition and reduce false-negative rates for suspect and pathological fetal conditions. Beside that, TMBA-ELM must tryed for other biomedical data like cancer dataset and skin desease dataset.

CRediT Authorship Contribution Statement

Prabowo Wisnumurti: Conceptualization, Methodology, Writing-Original Draft.
Syaiful Anam: Validation, Data Curation, and Supervision.
Mohammad Muslikh: Validation, Data Curation, and Supervision.

Declaration of Generative AI and AI-assisted technologies

In preparing this manuscript, Generative AI (ChatGPT by OpenAI) was used to aid inrefining grammar, clarity, and overall flow, in addition to providing paraphrasing and language polishing support.

Declaration of Competing Interest

The authors declare no competing interests.

Funding and Acknowledgments

This research received no external funding. The authors would like to extend their heartfelt appreciations to kaggle.com for access the dataset.

Data and Code Availability

The dataset used in this research is the fetal health conditions of 2126 correspondents. The dataset obtained from kaggle.com. The dataset is the open access dataset. Further details regarding the data are discussed in Section 2.1.

References

[1] R. R. Dixit, "Predicting fetal health using cardiotocograms: A machine learning approach," *Journal of Advanced Analytics in Healthcare Management*, vol. 4, no. 1, pp. 43–57, 2021. DOI: 10.1109/OCIT53463.2021.00056.

[2] A. Mehbodniya et al., "Fetal health classification from cardiotocographic data using machine learning," *Expert Systems*, vol. 39, no. 6, e12899, 2022. DOI: 10.1111/exsy.12899.

[3] A. C. Muller and S. Guido, *Introduction to Machine Learning with Python*, 1st ed. O'Reilly Media, 2017. Available online.

[4] I. E. Naqa, R. Li, and M. J. Murphy, *Machine Learning in Radiation Oncology*, 1st ed. Springer, 2015. DOI: 10.1007/978-3-319-18305-3.

[5] B. Yuan et al., "Nondestructive evaluation of thermal barrier coatings thickness using terahertz technique combined with pca-ga-elm algorithm," *Coatings*, vol. 12, no. 3, pp. 1–12, 2022. DOI: 10.3390/coatings12030390.

[6] W. Yuan, F. Liu, H. Gu, F. Miao, F. Zhang, and M. Jiang, "Accuracy-improved fault diagnosis method for rolling bearing based on enhanced esgmd-cc and ba-elm model," *Shock and Vibration*, pp. 1–16, 2024. DOI: 10.1155/2024/8026402.

[7] V. Lahoura et al., "Cloud computing-based framework for breast cancer diagnosis using extreme learning machine," *Diagnostics*, vol. 11, no. 2, pp. 1–19, 2021. DOI: 10.3390/diagnostics11020241.

[8] D. Ge, G. Jin, J. Wang, and Z. Zhang, "A novel ba-abc-elm model for estimating state of health of lithium-ion batteries," *Energy Reports*, vol. 13, pp. 465–476, 2025. DOI: 10.1016/j.egyr.2024.12.036.

[9] Q.-Y. Z. G.-B. Huang and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2006. DOI: 10.1016/j.neucom.2005.12.126.

[10] M. A. Albadr, S. Tiun, M. Ayob, and F. Al-Dhief, "Genetic algorithm based on natural selection theory for optimization pproblems," *Symmetry*, vol. 12, no. 11, pp. 1–31, 2020. DOI: 10.3390/sym12111758.

[11] W. Sun and C. Zhang, "A hybrid ba-elm model based on factor analysis and similar-day approach for short-term load forecasting," *Energies*, vol. 11, pp. 1–18, 2018. DOI: 10.3390/en11051282.

[12] Q. Liu, J. Li, L. Wu, F. Wang, and W. Xiao, "A novel bat algorithm with double mutation operators and its application to low-velocity impact localization problem," *Engineering Applications of Artificial Intelligence*, vol. 90, pp. 1–19, 2020. DOI: 10.1016/j.engappai.2020.103505.

[13] P. Satapathy, S. Chaine, S. Mishra, L. Tripathy, P.K Dash, and S. K. Dalai, "An evolutionary emd-fa-elm approach for short term wind power prediction using wind speed as input," *Odisha International Conferences*, vol. 2, pp. 1–19, 2022. DOI: 10.1109/ODICON54453.2022.10010060.

[14] Zhongda Tian, Yi Ren, and Gang Wang, "Short-term wind speed prediction based on improved pso algorithm optimized em-elm," *Energy Sources*, vol. 41, pp. 1–18, 2019. DOI: 10.1080/15567036.2018.1495782.

[15] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *Neural and Evolutionary Computing*, vol. 1, pp. 1–24, 2017.

[16] Larxel, *Fetal health classification*, (2020). Available online.

[17] D. Ayres-de-Campos, C. Y. Spong, and E. Chandraharan, "Figo consensus guidelines on intrapartum fetal monitoring: Cardiotocography," *International Journal of Gynecology and Obstetrics*, vol. 131, pp. 13–24, 2015. DOI: 10.1016/j.ijgo.2015.06.020.

[18] American College of Obstetricians and Gynecologists, "Acog practice bulletin no. 106: Intrapartum fetal heart rate monitoring: Nomenclature, interpretation, and general management principles," *Obstetrics & Gynecology*, vol. 114, no. 1, pp. 192–202, 2009. DOI: 10.1097/AOG.0b013e3181aef106.

[19] World Health Organization, *WHO Recommendations on Intrapartum Care for a Positive Childbirth Experience*. Geneva: World Health Organization, 2018. Available online.

[20] J. Li, Y. Jiang, L. Liu, and X. Liu, "Passive fetal movement signal detection system based on intelligent sensing technology," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–10, 2021. DOI: 10.1155/2021/1745292.

[21] I. Campos, H. Gonçalves, J. Bernardes, and L. Castro, "Fetal heart rate preprocessing techniques: A scoping review," *Bioengineering*, vol. 11, no. 4, pp. 1–20, 2024. DOI: 10.3390/bioengineering11040368.

[22] Z. Zhao, Y. Zhang, and Y. Deng, "A comprehensive feature analysis of the fetal heart rate signal for the intelligent assessment of fetal state," *Journal of Clinical Medicine*, vol. 7, no. 8, pp. 1–20, 2018. DOI: 10.3390/jcm7080223.

[23] J. Spilka, V. Chudacek, M. Huptych, G. Georgoulas, C. Stylios, and M. Koucky, "Using nonlinear features for fetal heart rate classification," *Biomedical Signal Processing and Control*, vol. 7, no. 4, pp. 350–357, 2012. DOI: 10.1016/j.bspc.2011.06.008.

[24] E. Acuña and C. Rodríguez, "The treatment of missing values and its effect on classifier accuracy," *Classification, Clustering and Data Mining Applications*, vol. 1, pp. 639–647, 2004. DOI: 10.1007/978-3-642-17103-1_60.

[25] K. G. Megalooikonomou and G. N. Beligiannis, "Random forests machine learning applied to peer structural performance experimental columns database," *Applied Sciences*, vol. 13, no. 23, pp. 1–24, 2023. DOI: 10.3390/app132312821.

[26] M. Mujahid et al., “Data oversampling and imbalanced datasets: An investigation of performance for machine learning and feature engineering,” *Journal of Big Data*, vol. 11, no. 87, pp. 1–32, 2024. DOI: 10.1186/s40537-024-00943-4.

[27] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd ed. CA: O’Reilly Media, 2019. Available online.

[28] D. Berrar, “Cross-validation,” *Reference Module in Life Sciences. Encyclopedia of Bioinformatics and Computational Biology*, vol. 1, pp. 542–545, 2019. DOI: 10.1016/B978-0-12-809633-8.20349-X.