MoM: Mortality During Maternity Prediction using Machine Learning

Tashin Chowdhury
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
tashin.chowdhury@northsouth.edu

Md Saeem Hossain Shanto
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
saeem.shanto@northsouth.edu

Adiba Sharif
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
adiba.sharif@northsouth.edu

Sifat Momen
Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
sifat.momen@northsouth.edu

***Abstract*—Predicting maternal mortality is important in terms of saving the lives of pregnant women as well as to improve the overall outcomes of maternal health. In spite of that, due to insufficient attention and resources, inadequate infrastructure, limited access to healthcare are some of the reasons why this issue gets overlooked sometimes.Early diagnosis of patients who may having health risks often helps to provide effective treatment.This research aims to develop mechanisms to detect the risk level and predict the reasons behind the outcome and to facilitate even more, a user-friendly web app has been built; where the model was deployed. In this work, the data set has been used as 1014 instances where three possible classes represent the risk level and the target variable is categorical. By using various feature engineering techniques to pre-process, the data set helped to improve the predictive power of the model.Furthermore, to improve the reliability of the model , the study employed a comprehensive approach from rigorous experimentation using multiple machine learning algorithms to it is deployment(best fitted model) of web app. 89% accuracy was obtained after implementing SMOTE and hyper-parameter tuning on the decision tree algorithm; which was the best among all the accuracy of the experiments. Finally, explainable AI tools like LIME and SHAP were applied to improve the model's interpretability and facilitate the communication between the patient and the health care provider.**

***Index Terms*—Maternal Health Risk,Machine Learning, Explainable AI, SMOTE, Feature Engineering, Django**

I. Introduction

Maternal mortality has a tragic impact on families and it acts as a crucial gauge of a health system's effectiveness [1]. Maternal mortality is a global health crisis, where around 285,000 women died due to pregnancy-related issues in 2020 [2]. According to WHO, the number of deaths due to pregnancy is around 100000 live births [3] and approximately 300000 women died from pregnancy-associated causes in 2017 [4] which sums up to 808 women died per day [5].Therefore, predicting the risks of maternal mortality will help to reduce the rates of maternal mortality and improve the health outcomes. And, Pregnancy can be categorized as low, moderate, or high risk depending on the presence of risk factors that have been demonstrated to increase the likelihood of pregnancy complications [6]. Pregnancies at high-risk can carry the potential for adverse outcomes for both the mother and the baby and especially the mother may experience complications like cardiovascular abnormalities in adulthood, preeclampsia, congenital anomalies, perinatal asphyxia, gestational diabetes, and the need for a cesarean section delivery [7].

Machine learning systems are being constantly developed to help doctors identify and treat the diagnostic data of mothers during their pregnancy. There are AI-powered programs available that help practitioners analyze diagnostic tests with more accuracy. This research aims to develop a sustainable and reliable machine learning system that predicts mortality during pregnancy, and to add credibility, explainable AI has been used to interpret the black box model, where it also looked into the possible reasons to extract new features and found various hyperparameter tuning techniques instead of manually tuning it.

This study followed several aspects that set our model apart from all other available models as a probable solution to this crisis, such as its comprehensive approach to improving the model's reliability, the selection and evaluation of all the used algorithms, the implementation of explainable AI for the model's interpretability, and lastly, the deployment of our model as a web app to improve its usage among the health service providers and the patients. The novelty of this work in several aspects is illustrated below:

- in order to get more relevant features (in this study, systolic and diastolic BP and age has been used) , addressing missing data and outliers (here, trimming and capping both techniques have been used), feature scaling and normalization (standardized feature scaling technique has been used) different techniques of feature engineering has been implemented.

- implementation of a web app improves a model's user accessibility, enables it to take real-time data as input, facilitates collaboration by integrating with frameworks(like Django) and ultimately helps to provide a user-friendly platform to predict maternal mortality health risk
- Implementation of explainable AI tools like LIME and SHAP provides transparency through offering insights while decision making, helps to validate the model's performance and facilitate the interpretability of the model's prediction.

The idea is to build an inexpensive system that can predict maternal mortality risk instantaneously, just as accurately as a gynecologist.

The structure of the paper is outlined below: We discuss detailed related works in Section II, following that in Section III the research work's methodology followed by the experiments results in Section IV. Finally, Section V concludes the paper with remarks on future work.

II. Related Work

Finding a solution to maternal mortality is crucial as it aligns with global development goals, promotes basic human rights and gender equality, alongside addresses health inequities [8]. Both healthcare providers and patients might experience critical deprivation which can lead to ineffective management and dire consequences for the vulnerable population [9]. Therefore, lots of considerable works are now available as a solution to get rid of this vital issue & here, some of the innovative works using different approaches will be discussed.

Reference	Applied technique	Advantage	Disadvantage
[5]	Ensemble learning-based feature engineering model for risk prediction	The "DT-BiLTCN" model incorporation of an IoT-based risk monitoring system enhances the model's capacity to monitor the continuous changes of health conditions during pregnancy.	Poor interpretability due to heavily dependency on deep learning techniques
[9]	Decision Tree Classification and Regression model for risk prediction	Focused on collecting row data and demonstrated promising results for predicting risk levels.	Dataset with low sample size.
[10]	IOT model for risk prediction	Allows to collect and transmit real-time data	It's effectiveness and accuracy depend on the availability and reliability of wearable sensing devices
[11]	Deep hybrid model for risk prediction	Able to classify accurately any instances of low-risk	Struggles to classify the instances of mid-risk
[12]	Traditional Machine Learning Methods for risk prediction	Implementation of multiple classifiers, provides a broader perspective to the performance of model	Low accuracy

TABLE I: A comparative analysis of related works.

III. Methodology

In this paper, we developed a webapp using Nextjs and Django,the latest and most popular frameworks, to predict the mortality level of a pregnant mother. Our machine learning model was developed by implementing feature engineering; after that, SMOTE was applied in order to synthetically oversample the minority class, which was then standardized to make all the features on a common scale without distorting the difference in the range of values. Following that, a range of machine learning algorithms were implemented to determine the best classifier. After that, to enhance the model's performance multiple hyperparameter tunings have been conducted, and lastly, explainable AI has been implemented in order to explain the generated black box model. In this section, we are going to discuss our approach in detail.

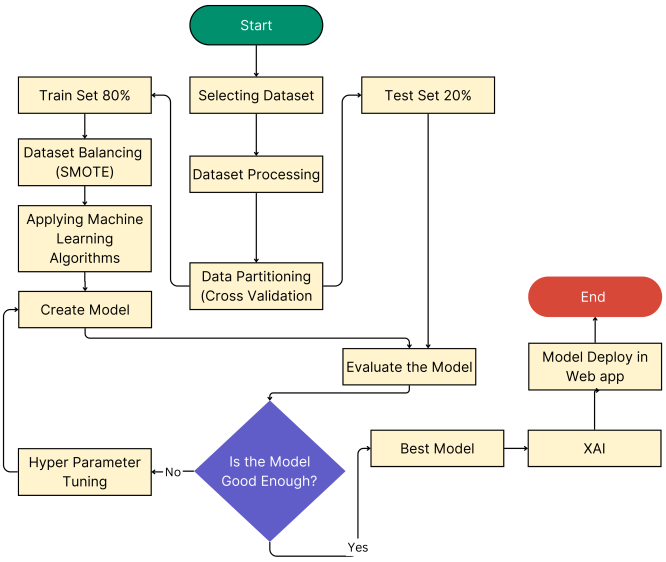

Fig. 1: Flowchart of our proposed method

A. Data description

The dataset we used for this study has been obtained from the University of California, Irvine (UCI) machine learning repository [10] which is collected by Ahmed from the rural areas of Bangladesh's hospital, clinics and mental health care centers [13]. The dataset contains six features and 1014 entries. The features are Age, SystolicBP, DiastolicBP, BS, BodyTemp, HeartRate, and Risk Level; where risk level is the targeted feature of this dataset. The targeted feature is categorized into three classes: "high risk", "mid risk", and "low risk".

B. Preprocessing

However, the data acquisition process stands out due to the unique approach we took to enhance the dataset through feature engineering based on the correlation we identified among all the attributes. And after feature engineering, BP named featured has been added, which is determined based on age, systolic BP, and diastolic BP. To establish credibility and viability for the new feature, age has been taken into account. For any age, low blood pressure is defined as systolic BP of 90 or below and diastolic BP of 60 or below. High blood pressure is classified as systolic BP of 120 or higher and diastolic BP of 80 or higher for mothers aged 39 or younger. For those aged 40 to 59, high blood pressure is indicated by systolic BP of 130 or higher and diastolic BP of 80 or higher. If the mother's age is 60 or older, then 140 or higher systolic BP and 90 or higher diastolic BP is considered as high blood pressure. Lastly, for the rest of the scenarios, the blood pressure would be considered normal [14]. Where 1 is classified as low pressure, 2 is classified as normal pressure, and 3 is classified as high pressure.

Later, the targeted feature was modified to turn the classes from categorical to numeric. The dataset had seven features after feature engineering; two of the features (Age, HeartRate) had abnormal values, so the outlier detection method has been applied to them. The dataset's outliers were examined using IQR. To handle the outliers, two methods have been followed: capping and trimming. These algorithms have been applied to both the original dataset and the feature-engineered dataset. After trimming, the "RiskLevel" variable has been added to the table, with its respective count, mean, standard deviation, minimum, 25th percentile, median (50th percentile), 75th percentile, and maximum values. After outlier detection, a standardized feature scaling technique has been applied to the updated dataset in order to interpret those features on the same scale.

Following that, the dataset had imbalanced classes, so in order to balance those imbalanced features, SMOTE (Synthetic Minority Over-sampling Technique) was applied, which over-sampled the imbalanced minority classes to 406 entries. We then split the data into training and testing sets with a ratio of 80:20, respectively.

C. Machine Learning Algorithms

We have applied many machine learning algorithms to experiment with our model, like Decision Tree, Random Forest, Logistic Regression, SVM, SGD, Gaussian Naive Bayes, KNN, ANN, and Voting Classifier. All these machine learning classifier algorithms use various techniques to train the model.

1) Logistic Regression: This is a classification-type algorithm that works on probability and can predict the dependent variable, which is categorical [15]. By using a method that predicts 0 or 1, yes or no, and success or failure, logistic regression makes predictions. The technique uses a link function that functions with a logistic function and has an S-shaped shape. If the value is below the threshold, it will trend to 0, and if it is above the point, it will tend to 1.

2) Random Forest: Random Forest is a machine learning method that relies on categorization and regression to reach conclusions. These two methods are applied in this approach to enhance the performance of the function and resolve a dataset's challenge. The decision tree is well-known to it, and each attribute contributes to the decision tree. Most of each tree's predictions are utilized by this algorithm [15].

3) Support Vector machine: It is an algorithmic type that solves categorization issues in machine learning. It is effective in resolving both linear and nonlinear problems in real-world situations. The quality is represented in the dataset as a coordinate, and this technique plots a hyperplane for each attribute. Because we will eventually set the most recent data to the best category, Hyperplane divides the class from one to another [15].

4) KNN: It is a type of algorithm that utilizes supervised learning strategies with both new and old data. Put the new data in the appropriate, related categories. Rather than regression, we use this for classification. To discover the closest neighbors and calculate the distance between data points, it uses Euclidean distance algorithms. If the training data are significant, it might be useful [15].

5) Decision Tree: In machine learning methods, it is a key classification technique, which can be used as enhancing search engines, developing medical apps, and locating missing data in a class. The advantages of the traditional decision tree algorithms ID3, C4.5, and C5.0 include quick classification times and powerful learning capacities [15].

6) Gaussian Naive Bayes: Built on the Bayes Theorem, it is a classification method which predicated on the idea of predictor independence. The naive Bayes classifier makes the assumption that the existence of a feature in a class has nothing to do with the presence of any other feature. It is mostly utilized for classification and clustering purposes based on the conditional likelihood of occurring [15].

7) ANN: To identify underlying connections in a set of data using a method, neural network is a collection of algorithms that aims to imitate how the human brain works. An artificial neural network acts similarly. It operates across three layers. The input layer receives data. The input is processed by the hidden layer. The determined output is then sent by the output layer [15].

8) SGD: It is a widely used optimization algorithm in machine learning and updates the model's parameters by computing gradients on randomly selected subsets of training data iteratively, making it efficient for large datasets. SGD's advantage lies in its scalability, as it can handle large amounts of data efficiently, and it often converges faster compared to traditional gradient descent methods [16].

9) Voting Classifier: A powerful ensemble method in machine learning which can combines predictions from multiple models to make a final decision. It works by aggregating the results of individual classifiers and selecting the majority vote as the final prediction [17].

D. SMOTE

To mitigate the effect of having few instances of the minority class in a data collection, SMOTE (Synthetic Minority Oversampling Technique) has been developed. Working in the "feature space" instead of working in the "data space" results in the creation of synthetic instances of the minority class [18].

E. Hyperparameter Tuning

We progressed with hyperparameter optimization to enhance the performance of the model. On algorithms such as Decision Tree, and Random Forest, we checked out a number of techniques, including Random Search CV, Auto Hyperparameter Tuning, Grid Search, Genetic Algorithm, and Optuna Hyperparameter Tuning.

1) Grid Search: One of the most popular techniques for perusing the hyper-parameter configuration space is grid search. It operates by analyzing the Cartesian product of a finite set of values that the user specifies. It is simple to parallelize and implement [19].

2) Random Search CV: Similar to grid search, random search randomly chooses a certain number of samples within the upper and lower boundaries as candidate hyper-parameter values, trains these candidates until the allocation of resources is used up, and then rejects all other values in the search space [19].

3) Genetic Algorithm: One of the most common meta-heuristic algorithms is the genetic algorithm, which is based on the evolutionary idea that individuals with the best survival potential and adaptability to the environment are more likely to survive. In Genetic Algorithm, random initialization, which assists to create the population initially with random values in the given search space, is frequently used to generate hyper-parameter configuration candidates for the initial population [19].

4) Auto Hyperparameter Tuning: One of the key functions of auto hyperparameter tuning is to automatically tune the hyperparameters, which reduces the load of manual work and enhances the performance, reproducibility, and fairness of various research [20].

5) Optuna Hyperparameter Tuning: Optuna is a software framework designed exclusively for machine learning that automatically optimizes hyperparameters. It defines hyper-parameter optimization as the process of minimization or maximization of an objective function that receives a set of hyperparameters as input and outputs the (validation) score. Optuna interacts with the trial object to gradually create the objective function. While executing the objective function, the search areas are dynamically created using techniques from the trial object [21].

F. Explainable AI

Explainable AI (XAI) is a collection of tools and frameworks that are natively linked with several Google products and services which help you to comprehend and interpret all the predictions provided by your models. With it, you can debug models to maximize their performance, and aid in the behavioral understanding of others [22]. To do so, we have used the tools LIME and SHAP to interpret our black box model.

1) Lime: Lime is a tool that can explain the prediction of any ML model. Locally learning an interpretable model around the forecast helps increase model interpretability and, more significantly, explain each prediction. Before examining the effect on the outcome, feature values in a single data sampling are changed [23].

2) SHAP: SHAP is a model-independent post-hoc approach that can be applied to any machine learning model. SHAP assigns a score to each feature in the model, which signifies its importance in the model output. It evaluates all collations between the features to cover all scenarios where all features and a subset of features is in the model to calculate the scores [24].

IV. Experiments

A. Experiment Setting-up

In this paper section, we discuss our machine learning model preparation. Various machine learning algorithms were employed in the experiments, with consistent steps across datasets. Initially, we mapped targeted features (e.g., 'high risk' as 3.0, 'mid risk' as 2.0, and 'low risk' as 1.0) on the original dataset. After splitting into train and test sets, we applied scale standardization. Training was performed using different classifiers, generating a classification report and confusion matrix for evaluation.

In another experiment, we maintained the same factors but applied SMOTE before training the classifiers on the original dataset. Similarly, we generated classification reports and confusion matrices for evaluation.

For our next experiment, we conducted feature engineering on the original data, focusing on age, systolic BP, and diastolic BP. We formed a new feature called BP and explored the correlation matrix. The correlation between BP and systolic BP, diastolic BP, and age was not significant, but fever showed high correlation with body temperature. The fever feature was dropped, resulting in a new dataset. All subsequent steps were consistent with previous experiments on the original dataset.

To improve performance, we applied hyperparameter tuning techniques to the top two classifiers. A voting classifier using the best performing classifier was used, and results were recorded.

Furthermore, we conducted an experiment using an Artificial Neural Network (ANN) instead of machine learning classifiers. The initial steps were the same, with standardization of the dataset. ANN experiments were conducted with a batch size of 32 and epochs set to 50. We generated accuracy and loss curves, along with classification

reports and confusion matrices. Results were recorded.

Lastly, explainable AI techniques were applied to the best performing machine learning model. Tools like LIME and SHAP were used for explanation. LIME provided insights into single output predictions, giving an idea of the factors and their probabilities. SHAP was implemented to determine feature importance and values for specific outputs.

B. Evaluation Metrics

The evaluating metrics were used to evaluate the trained classifier's capacity to generalize [25]. To evaluate the performance of our models, we evaluated metrics such as Accuracy, Precision, Recall and F1 Score. The accuracy, precision, recall, and f1-score were determined using a confusion matrix; which compares the actual target values with the predicted values by the machine learning model. Four key characteristics make up the confusion matrix, which determines the classifier's measuring metrics. These are the four numbers:

True Positive (TP): TP means model predicted positive, and it is true, meaning the predicted value corresponds to the actual value.

True Negative (TN): The model predicted negative, and it was correct, meaning the predicted value was the same as the actual value.

False Positive (FP): The model anticipated positive. However, it was incorrect, implying that the projected result was incorrect. Although the actual number was negative, the model projected that it would be positive.

False Negative (FN): When the model predicted a false negative value, implying that the anticipated value was incorrect. Although the actual number was positive, the model predicted that it would be negative.

Accuracy: To compare the efficiency of the classifier, the accuracy calculation is utilized. It takes into account the classifier's overall number of correct predictions. It is calculated using the following equation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Recall: By calculating the ratio of successfully detected positive inputs helps to determine the recall. It is the TP rate, which is calculated using the following equation:

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

Precision: Precision refers to the classifier's ability to forecast positive instances correctly. It is calculated using the following equation:

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

F1 score: The F1 score, commonly known as the F measure, measures the test's accuracy. It is a weighted average of precision and recall with a maximum value of 1 and a minimum value of 0.

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

C. Experiment Results

Table-II lists the results of the machine learning algorithms that have been used on each dataset and their performance results, both with and without using SMOTE. Where dataset-1 represents the original dataset and dataset-2 represents the feature engineered dataset, similarly, normally represents the same as without using the SMOTE technique, and the SMOTE column represents the same as with using SMOTE. The overall result of the experiment shows that the model using Decision Tree with SMOTE technique performs the best with an accuracy, precision, recall, and F1-score of 89%.

Classifier	**Dataset**	**Accuracy %**		**Precision%**		**Recall%**		**F1 Score%**	
		normal	SMOTE	normal	SMOTE	normal	SMOTE	normal	SMOTE
Logistic Regression	Dataset:1	66.0	64.0	68.0	65.0	68.0	66.0	66.0	64.0
	Dataset:2	63.0	67.0	64.0	69.0	65.0	67.0	62.0	69.0
Random Forest	Dataset:1	83.0	84.0	83.0	86.0	83.0	85.0	83.0	85.0
	Dataset:2	81.0	88.0	82.0	88.0	82.0	88.0	82.0	88.0
KNN	Dataset:1	67.0	67.0	69.0	69	68.0	67.0	68.0	68.0
	Dataset:2	70.0	70.0	71.0	71.0	68.0	70.0	69.0	70.0
Decision Tree	Dataset:1	80.0	87.0	82.0	88.0	81.0	87.0	81.0	87.0
	Dataset:2	81.3	89.0	82.0	89.0	82.0	89.0	82.0	89.0
Support Vector Machine	Dataset:1	60.0	73.0	68.0	75.0	59.0	73.0	58.0	73.0
	Dataset:2	61.0	59.0	68.0	61.0	61.0	59.0	59.0	60.0
Gaussian Naive Bayes	Dataset:1	58.0	61.0	62.0	64.0	58.0	61.0	55.0	59.0
	Dataset:2	61.0	61.0	64.0	65.0	63.0	60.0	59.0	58.0
ANN	Dataset:1	67.0	71.0	71.0	72.0	70.0	71.0	66.0	71.0
	Dataset:2	68.0	61.0	73.0	60.0	71.0	61.0	67.0	59.0
SGD	Dataset:1	64.0	59.0	66.0	59.0	64.0	59.0	63.0	58.0
	Dataset:2	65.0	62.0	68.0	68.0	65.0	63.0	64.0	63.0

TABLE II: Performance metrics (Machine Learning Algorithms Applied)

Table-III lists the results of the two best performing machine learning classifiers that have been used in this experiment after hyperparameter tuning. Where it is shown that among all the hyperparameter tuning techniques used, the Random Forest classifier using Optuna Hyperparameter tuning gives the highest accuracy of 88%, but for the decision tree classifier, Random Search CV gives better accuracy than Random Forest, which is 89%.

Classifier	Optuna Hyper parameter Tuning%	Genetic Algorithm%	Grid Search%	Auto Hyper parameter Tuning%	Random Search CV%
Random Forest	88.0	74.0	87.0	77.0	87.0
Decision tree	87.0	75.0	85.0	76.0	89.0

TABLE III: Comparing Accuracy Metric After Applying Different Hyperparameter Tuning Techniques on Dataset-2

V. Conclusion

This study presents comprehensive approaches which collectively improved not just the quality of the dataset we began to work with, but also led to more accurate predictions. After rigorous experiments and analysis using multiple other machine learning algorithms such as Decision tree, Random Forest, SVM, Gaussian Naive Bayes, SGD, Logistic Regression, KNN, Voting Classifier, and ANN; the best outcome our model provides after implementing smote and hyperparameter tuning on decision tree algorithm; which is 89%. Furthermore, the implementation of explainable AI and the deployment of a user-friendly web app allowed both the patients and the healthcare professionals to understand which factors are influencing the model's outcome. Thus, the study shows enough potential for this model to bring positive changes to the medical science domain already. Yet, in the future, by improving the model continuously, integrating with real-time data sources, conducting longitudinal analysis, and involving domain experts can help to evolve and contribute to even better maternal healthcare outcomes.

References

[1] A.-r. Y. Collier and R. L. Molina, "Maternal mortality in the united states: Updates on trends, causes, and solutions," *NeoReviews*, vol. 20, no. 10, 2019.

[2] [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/maternal-mortality

[3] [Online]. Available: https://data.worldbank.org/indicator/SH.STA.MMRT?end=2017&start=2000&view=chart

[4] M. Roser and H. Ritchie, "Maternal mortality," Nov 2013. [Online]. Available: https://ourworldindata.org/maternal-mortality

[5] A. Raza, H. U. R. Siddiqui, K. Munir, M. Almutairi, F. Rustam, and I. Ashraf, "Ensemble learning-based feature engineering to analyze maternal health during pregnancy and health risk prediction," *Plos one*, vol. 17, no. 11, p. e0276525, 2022.

[6] M. Y. Al-Hindi, T. A. Al Sayari, R. Al Solami, A. K. A. Baiti, J. A. Alnemri, I. M. Mirza, A. Alattas, and Y. A. Faden, "Association of antenatal risk score with maternal and neonatal mortality and morbidity," *Cureus*, vol. 12, no. 12, 2020.

[7] I. Chermesh, I. Solt, Y. Friedrich, L. Lowenstein *et al.*, "Achieving the recommended gestational weight gain in high-risk versus low-risk pregnancies." *The Israel Medical Association Journal: IMAJ*, vol. 20, no. 7, pp. 411–414, 2018.

[8] [Online]. Available: https://www.who.int/data/gho/data/themes/topics/sdg-target-3-1-maternal-mortality

[9] I. Umoren, F. Chigozirim, A. Silas, and B. Ekong, "Modeling and prediction of pregnancy risk for efficient birth outcomes using decision tree classification and regression model."

[10] M. Ahmed and M. A. Kashem, "Iot based risk level prediction model for maternal health care in the context of bangladesh," in *2020 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI)*. IEEE, 2020, pp. 1–6.

[11] T. O. Togunwa, A. O. Babatunde, and K.-U.-R. Abdullah, "Deep hybrid model for maternal health risk classification in pregnancy: Synergy of ann and random forest," *Frontiers in Artificial Intelligence*, vol. 6, p. 1213436.

[12] H. B. MUTLU, F. DURMAZ, N. YÜCEL, E. CENGİL, and M. YILDIRIM, "Prediction of maternal health risk with traditional machine learning methods," *NATURENGS*, vol. 4, no. 1, pp. 16–23, 2023.

[13] M. Ahmed, M. A. Kashem, M. Rahman, and S. Khatun, "Review and analysis of risk factor of maternal health in remote area using the internet of things (iot)," in *InECCE2019: Proceedings of the 5th International Conference on Electrical, Control & Computer Engineering, Kuantan, Pahang, Malaysia, 29th July 2019*. Springer, 2020, pp. 357–365.

[14] S. H. Salomon, "Average blood pressure by age: What you need to know," Apr 2023. [Online]. Available: https://www.verywellhealth.com/average-blood-pressure-by-age-5085328

[15] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, no. 1, pp. 381–386, 2020.

[16] L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM review*, vol. 60, no. 2, pp. 223–311, 2018.

[17] Y. Zhang, H. Zhang, J. Cai, and B. Yang, "A weighted voting classifier based on differential evolution," in *Abstract and applied analysis*, vol. 2014. Hindawi Limited, 2014, pp. 1–6.

[18] N. V. Chawla, A. Lazarevic, L. O. Hall, and K. W. Bowyer, "Smoteboost: Improving prediction of the minority class in boosting," in *Knowledge Discovery in Databases: PKDD 2003: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 7*. Springer, 2003, pp. 107–119.

[19] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

[20] L. Zahedi, F. G. Mohammadi, S. Rezapour, M. W. Ohland, and M. H. Amini, "Search algorithms for automated hyper-parameter tuning," *arXiv preprint arXiv:2104.14677*, 2021.

[21] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2623–2631.

[22] P. Gohel, P. Singh, and M. Mohanty, "Explainable ai: current status and future directions," *arXiv preprint arXiv:2107.07045*, 2021.

[23] A. Patil, A. Framewala, and F. Kazi, "Explainability of smote based oversampling for imbalanced dataset problems," in *2020 3rd International Conference on Information and Computer Technologies (ICICT)*. IEEE, 2020, pp. 41–45.

[24] A. Salih, Z. Raisi-Estabragh, I. B. Galazzo, P. Radeva, S. E. Petersen, G. Menegaz, and K. Lekadir, "Commentary on explainable artificial intelligence methods: Shap and lime," *arXiv preprint arXiv:2305.02012*, 2023.

[25] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *International journal of data mining & knowledge management process*, vol. 5, no. 2, p. 1, 2015.