DS Journal of Digital Science and Technology
Volume 3 Issue 1, 1-10, Jan – Mar 2024
ISSN: 2583-5416 / https://doi.org/10.59232/DST-V3I1P101

Original Article

Comparative Data Analysis on Fetal Health Using Machine Learning

Arosocohi Yosua Daeli[1], Dewi Agushinta R.[2], Emirul Bahar[3], Sri Hayuningsih[4]

[1]*Information Systems Department, Gunadarma University, Indonesia.*
[2]*Information Technology Department, Gunadarma University, Indonesia.*
[3]*Industrial Engineering Department, Gunadarma University, Indonesia.*
[4]*Midwifery Department, Gunadarma University, Indonesia.*

[1] archidaeli@gmail.com

Received: 07 January 2024; Revised: 10 February 2024; Accepted: 23 February 2024; Published: 08 April 2024;

Abstract - This research aims to find a machine learning classification algorithm to classify fetal health decisions using existing features of fetal health. The classification algorithms compared are support vector machine, logistic regression, and Random Forest Classifier. After hyperparameter tuning, the results obtained from this research are that the first support vector machine model obtained an accuracy of 0.88, a precision of 0.80, a recall of 0.76, and an f-1 score of 0.78. The second model is logistic regression, which gets an accuracy of 0.87, precision of 0.80, recall of 0.69, and f-1 score of 0.73. The third model is a Random Forest Classifier, which gets an accuracy of 0.94, precision of 0.91, recall of 0.88, and f-1 score of 0.89. Of these three algorithms, random forest is the best algorithm for detecting fetal health.

Keywords - Classification, Fetal health, Logistic Regression, Machine Learning, Random Forest Classifier, Support Vector Classifier.

1. Introduction

Every mother feels happiness and sorrow at times when they are pregnant. But the stages of pregnancy are a difficult period, one might even say critical, because the health of the fetus depends on the health of the mother. Mothers need to live a stable and stress-free life so that the fetus continues to be healthy in the process of recording development. Prenatal stress can indirectly cast a long shadow on a baby's well-being, increasing the risk of complications during birth that can have lasting health and developmental consequences [1].

According to the World Health Organization record (WHO), abortions have taken worldwide around 73 million each year. 6 out of 10 (61%) abortions are unintended pregnancies, and 3 out of 10 are induced abortions [2]. Induced abortion is a sensitive topic in reproductive health because it is defined that induced abortion is the act to terminate pregnancy before the fetus is viable. The act can be considered illegal or legal, depending on the law in the country the mother is living in, the reason for wanting termination, and the skill to get the procedure done [3]. Though technically abortion was prohibited by the criminal code, a 1970s judicial interpretation allowed medical professionals to discreetly perform abortions in Indonesia. These resulted in a significant increase in safe, medical abortions alongside a documented decline in morbidity and mortality associated with unsafe, illegal procedures [4].

CTG plays a crucial role in pregnancy and labor by monitoring fetal heart rate in relation to contractions, helping detect potential oxygen concerns and guiding decisions on further assessment or delivery methods [5]. In short, Cardiotocography (CTG) is a technique performed to monitor the fetal heartbeat and uterine contractions during pregnancy and labor [6, 7]. CTG signals are interpreted visually and are used to draw clinical inferences; however, their application has been rather inconsistent, subjective, and prone to the obstetrician's discretion, resulting in a significant false positive rate [8].

The model comparison between machine learning algorithms such as Support Vector Machine, Random Forest Classifier, Multilayer Perceptron, and K-Nearest Neighbor was implemented to classify fetal health and perform useful predictions UCI-Machine Learning Repository, which its data was recorded using SisPorto 2.0. Models' performances are then recorded and ranked based on metrics in Accuracy, Precision, Recall, F-1 Score, and Support. The comparison results to the Random Forest Classifier model having 94.5 accuracy [9].

A system that could be able to diagnose early Pre-Eclampsia, which is a disease afflicting expectant mothers, was built, characterized by the presence of elevated blood pressure and excessive protein levels in the urine occurring after the gestational period surpasses 20 weeks using the K-Nearest Neighbors model. The total record of data is 100 records which is taken from Hospital Ratu Zalecha, Banjar, South Kalimantan. K-fold cross-validation is applied by dividing the data into 10 folds and will have each turn become the training set to measure the system's accuracy. After the system was evaluated using a confusion matrix, the KNN model was able to provide an accuracy of 88% [10].

A study where machine learning implements a Decision Tree as well as Feature Selection to give an improvement in the performance of machines using the software RapidMiner was conducted. The dataset provides 2126 patients with 22 attributes, and one of the attributes will be the class, which will be categorized as Normal for 1, Suspect for 2, and Pathologic as 3. Decision trees, in general, are prediction models that use a tree structure or hierarchical structure of models and their possible consequences, which consist of root nodes, decision nodes, leaf nodes, and branches. On the other hand, Feature Selection is to select attributes that affect the output and remove the insignificance. Results show consistency where Feature Selection gives a 1.22% performance boost having 90.88% highest accuracy [11].

Research in identifying fetal heart rate using seven machine learning algorithms, namely Logistic Regression, Decision Tree, MLPClassifier, SVM, Random Forest, Naive Bayes, and KNN algorithms, was conducted. The dataset being used in the research is from the UCI Machine Learning Repository. Based on three trials of ratio data testing of 0.2, 0.25, and 0.3, the research results show that Logistic Regression has accuracy the highest is 88.40% [12].

A Random Forest Classifier to identify fetal health classes among Normal, Suspect, and Pathologic was implemented. The dataset used consists of 2126 cardiotocograms collected from the Maternity and Gynecological Clinic at the University Hospital of Porto, Portugal. The results show that the Random Forest Classifier can provide a high score of 93.6% [13].

There are several problems that Indonesian mothers face when they are pregnant, namely, the availability of adequate services and hospital services that are not fast and responsive. Predictions regarding the condition of the baby in the womb for pregnant women certainly provide a faster response so that not only can the health of the fetus be seen, but the actions that need to be taken can be more responsive. Based on these basic things, this research makes a comparison between three machine learning algorithms, namely Support Vector Classifier (SVC), Logistic Regression, and Random Forest Classifier, in determining three categories of fetal health, namely Normal, Suspicious, and Pathological.

2. Methods

This research aims to provide a comparative and comprehensive analysis to see the ability of three Machine Learning methods to process Fetal Health data. This research compares the Machine Learning Logistic Regression, Support Vector Classifier, and Random Forest Classifier models. The model process is shown in Figure 1.

The dataset is processed into Training Data and Testing Data, and the model used is given parameters to determine how the model is assembled. The model is assessed and the best parameters will be used for Data Testing as a comparison to determine the model's ability to process Fetal Health data.

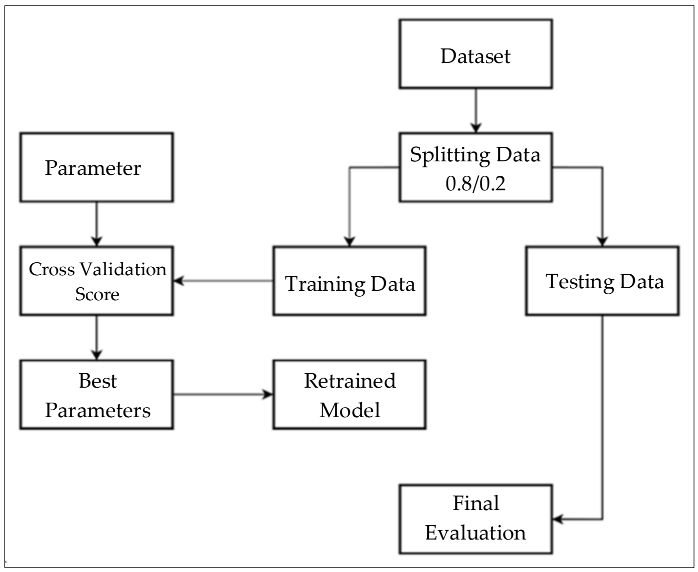

Fig. 1 Data process framework

3. Results and Discussion

3.1. Data Collection

In the data collection process, the dataset is reviewed to select the best data to use. The most common and complete datasets used for research and learning are in the UCI Machine Learning Repository [14]. However, in this study, the dataset was obtained from previous research on Kaggle with a dataset that had gone through a cleaning process and was used for previous research [15].

This research uses a dataset entitled Fetal Health which is available on kaggle.com with a version that has been cleaned for machine research use and has been downloaded. The dataset used is the fetal_health.csv file, as in Figure 2.

After the dataset is read, all the data contained in the dataset is used in research because only a small amount of data is contained in the file. In contrast, machine learning research requires a lot of data, so in this research, all existing datasets are used directly. The data in the dataset totals 2126 rows and 22 columns, shown in Figure 3.

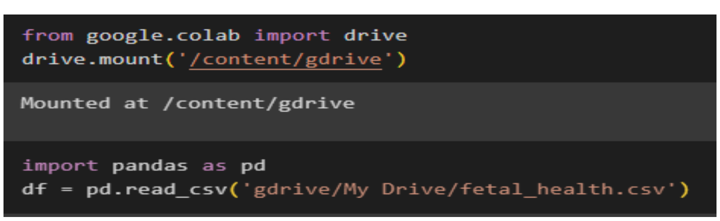

Fig. 2 Dataset reading process

Fig. 3 Rows and columns

3.2. Data Processing

Data Processing is a process where the data that has been taken is processed. Before processing, data in the form of tables and columns is reviewed to see data duplication or errors. Values that cannot be processed or that may interfere are not used so as not to interfere with the performance of the model to be assembled. After the data is clean, each data is reviewed in terms of dependence on each other.

Data pre-processing serves as the cornerstone of the data mining process, akin to meticulously preparing ingredients for a delectable dish. This crucial stage involves a multi-pronged approach to transform raw data into a well-structured and analyzable format [16, 17]. The diversity of features in fetal health creates diverse data, so the data must be explored, reviewing what features or factors most influence fetal health. These factors illustrate that only a few factors are crucial in influencing fetal health.

The data is then divided into Training Data and Testing Data. Training Data or training dataset is a data set that is shared for model learning. The model must learn the types and variations of data so that later, the model can provide patterns and show predictions according to the capabilities given using the parameters.

3.3. Data Exploration

The data exploration phase serves as the foundation for subsequent analysis, enabling the identification of patterns, data characteristics, and potential anomalies and the formulation of initial hypotheses. This critical stage equips researchers with the information necessary to align their research objectives with the insights gleaned from the dataset, ensuring a focused and informed investigation [18].

a) Statistical Analysis: collecting, analyzing, interpreting, and basic calculations carried out to see the distribution of data variables, which include calculations of median, mode, quartiles 1 to 4, average, and standard deviation.
b) Data Cleaning: is an action that includes cutting, repairing, standardizing, and adjusting data that was previously inaccurate to become more precise. Deleting data is also part of data cleaning because data with inaccurate information can have an impact on the performance of the dataset.
c) Graphic Visualization: is the process of taking raw data and turning it into graphs, charts, images, or even videos that explain the numbers and allow us to gain insight. These insights give us an overview, pattern, or trend. Generally, data visualization can be represented in the form of box plots, scatter plots, heat maps, pie charts, histograms, etc.
d) Correlation between Variables: correlation between variables is formed after the existing raw data has been provided with an overview through the exploration stage above. Data that has been processed in statistical form, providing images and visuals, can represent information to answer the objectives of the research and even provide more information than expected.

The first step is to see whether the data to be used has null values, as in Figure 4. Rows are numerical values used as processing material for machine learning algorithms in this research, while columns are features recorded. There are 23 features recorded in float type, shown in Figure 5, with the matrix correlation of them shown in Figure 6.

3.4. Model Building

Model Building is the manufacturing and design stage. The steps taken at this stage are selecting, designing, training, and classifying the performance of machine learning algorithms. The detailed stages are explained as follows:

1. The algorithm selection utilized in this study leverages findings and insights gleaned from prior research and existing literature.

2. A dedicated pipeline will be constructed to expedite and streamline the process of cross-validation accuracy score.
3. An extensive hyperparameter tuning process will be initiated to enhance the performance of each machine learning model for the subsequent stage.

Figure 7 showed the model-building stage begins by creating a new variable without including the fetal health column. The newly created dataset in X is the feature column, and the fetal health column becomes variable Y, which is the target column.

```
#check for missing values
df.isnull().sum()

baseline value                                            0
accelerations                                             0
fetal_movement                                            0
uterine_contractions                                      0
light_decelerations                                       0
severe_decelerations                                      0
prolongued_decelerations                                  0
abnormal_short_term_variability                           0
mean_value_of_short_term_variability                      0
percentage_of_time_with_abnormal_long_term_variability    0
mean_value_of_long_term_variability                       0
histogram_width                                           0
histogram_min                                             0
histogram_max                                             0
histogram_number_of_peaks                                 0
histogram_number_of_zeroes                                0
histogram_mode                                            0
histogram_mean                                            0
histogram_median                                          0
histogram_variance                                        0
histogram_tendency                                        0
fetal_health                                              0
dtype: int64
```

Fig. 4 Null value check

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2126 entries, 0 to 2125
Data columns (total 22 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   baseline value                                          2126 non-null   float64
 1   accelerations                                           2126 non-null   float64
 2   fetal_movement                                          2126 non-null   float64
 3   uterine_contractions                                    2126 non-null   float64
 4   light_decelerations                                     2126 non-null   float64
 5   severe_decelerations                                    2126 non-null   float64
 6   prolongued_decelerations                                2126 non-null   float64
 7   abnormal_short_term_variability                         2126 non-null   float64
 8   mean_value_of_short_term_variability                    2126 non-null   float64
 9   percentage_of_time_with_abnormal_long_term_variability  2126 non-null   float64
 10  mean_value_of_long_term_variability                     2126 non-null   float64
 11  histogram_width                                         2126 non-null   float64
 12  histogram_min                                           2126 non-null   float64
 13  histogram_max                                           2126 non-null   float64
 14  histogram_number_of_peaks                               2126 non-null   float64
 15  histogram_number_of_zeroes                              2126 non-null   float64
 16  histogram_mode                                          2126 non-null   float64
 17  histogram_mean                                          2126 non-null   float64
 18  histogram_median                                        2126 non-null   float64
 19  histogram_variance                                      2126 non-null   float64
 20  histogram_tendency                                      2126 non-null   float64
 21  fetal_health                                            2126 non-null   float64
dtypes: float64(22)
memory usage: 365.5 KB
```

Fig. 5 Fetal health features

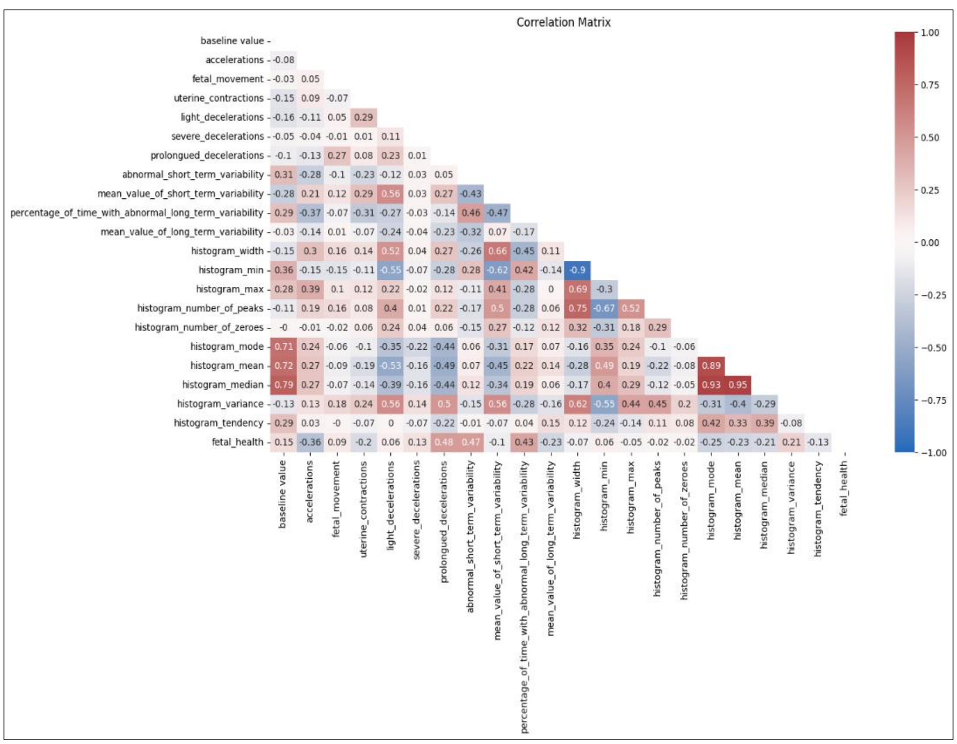

Fig. 6 Matrix correlation

```
#for x we drop fetal_health only so y be only having fetal_health column
x = df.drop(["fetal_health"],axis=1)
y = df["fetal_health"]
```

Fig. 7 Splitting feature and target column

```
#splitting dataframe itu data training and data testing
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

```
print(x_train.shape, x_test.shape, y_train.shape, y_test.shape)
```

```
(1700, 21) (426, 21) (1700,) (426,)
```

Fig. 8 Training set and testing set

After the feature column is separated from the target, the dataset is split into a training and a testing dataset. The presentation divides the training and testing set into a ratio of 8:2, which means the training is 80% of the dataset, and 20% of the dataset is the testing set. No evidence states the most optimal ratio in dividing datasets, but following the Pareto Principle, the 8:2 ratio is a rule of thumb used by researchers and practitioners [19]. The number of training and testing sets is shown in Figure 8.

The pipeline model in Figure 9 is used to perform many operations at once. In this research, the pipeline is used to carry out operations to find cross-validation scores for the three machine learning algorithms that have been selected. It puts a series of complex data processing steps, processes a labeled training dataset and produces a machine learning model [20].

The existence of a pipeline makes it easier for researchers to easily put together several ML models to cover the full analytic process starting from raw datasets. By encapsulating complex data processing tasks into a modular pipeline, researchers can focus on developing and optimizing their machine learning models. In contrast, the pipeline ensures consistent and efficient data preparation and training [21, 22]. Cross-validation is a common method for measuring the performance of machine learning algorithms.

The values obtained from the cross-validation process using a pipeline are a Support Vector Classifier of 0.835, a Logistic Regression of 0.857, and a Random Forest Classifier of 0.940.

```
#cross validation score
pipeline_lr=Pipeline([('lr_classifier',LogisticRegression())])

pipeline_rf=Pipeline([('rf_classifier',RandomForestClassifier(random_state=42))])

pipeline_svc=Pipeline([('svc_classifier',SVC(random_state=42))])
pipelines = [pipeline_lr, pipeline_rf, pipeline_svc]
pipe_dict = {0: 'Logistic Regression', 1: 'RandomForest', 2: "SVC"}

for pipe in pipelines:
    pipe.fit(x_train, y_train)
    cv_results_accuracy = []
for i, model in enumerate(pipelines):
    cv_score = cross_val_score(model, x_train,y_train, cv=10)
    cv_results_accuracy.append(cv_score)
    print("%s: %f " % (pipe_dict[i], cv_score.mean()))
```

Fig. 9 Build cross-validation score model

Table 1. Cross-validation score

Model	Values
Support Vector Classifier	0.853529
Logistic Regression	0.857059
Random Forest Classifier	0.940588

Every machine learning for classification has parameters that are used to improve the performance of accuracy, precision, and other measurements. In this stage, parameters are placed in each algorithm to see whether there is a difference between the performance before hyperparameter tuning is given, or not. This research provides parameters according to machine learning, but the parameters determined are based on GridSearchCV.

```
best_estimator_svc = CV_SVC.best_params_
print(f"Best estimator for RF model:\n{best_estimator_svc}")

Best estimator for RF model:
{'C': 1, 'cache_size': 42, 'kernel': 'linear', 'random_state': 8}
```

Fig. 10 Support Vector Classifier

```
best_estimator_svc = CV_SVC.best_params_
print(f"Best estimator for RF model:\n{best_estimator_svc}")

Best estimator for RF model:
{'C': 1, 'cache_size': 42, 'kernel': 'linear', 'random_state': 8}
```

Fig. 11 Logistic Regression hyperparameter tuning

```
best_estimator_RF = CV_rf.best_params_
print(f"Best estimator for RF model:\n{best_estimator_RF}")

Best estimator for RF model:
{'criterion': 'entropy', 'max_depth': 14, 'max_features': 'sqrt', 'n_estimators': 100, 'n_jobs': None}
```

Fig. 12 Random Forest Classifier hyperparameter tuning

3.5. Performance Enhancement and Model Evaluation

The attempt to improve the performance of each Machine Learning requires hyperparameter tuning. However, in this study, the hyperparameters were determined randomly, and their values were determined using the Grid Search Cross Validation or GridSearchCV process. GridSearchCV is a powerful tool in machine learning to find optimal hyperparameters. It identifies the set of hyperparameters that leads to the best model performance on a separate validation set [23, 24]. Hyperparameter tuning is given for each algorithm according to its type in Table 2, Table 3, and Table 4.

Table 2. Hyperparameter tuning for Support Vector Classifier

Parameter	Values
Kernel	Linear
random_state	8
Cache Size	42
C	1

Table 3. Hyperparameter tuning for Logistic Regression

Parameter	Values
max_iter	100
random_state	21
multi-class	ovr
n_jobs	1
C	1

Table 4. Hyperparameter tuning for Random Forest Classifier

Parameter	Values
n_estimators	100
max_features	sqrt
max_depth	14
criterion	entropy
n_jobs	None

Table 5. Confusion matrix

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

After providing hyperparameter tuning to each machine learning algorithm, the next stage is model evaluation. The method used to evaluate the model is to use the confusion matrix. confusion matrix calculations from metrics include Accuracy, Precision, f-1 Score, and Support values [25]. The confusion matrix table is described in Table 5.

- True Positive (TP): When the actual value is Positive, the predicted value is also Positive.
- False Negative (FN): When the actual value is Positive, but the predicted value is Negative.
- False Positive (FP): When the actual value is Negative but the predicted value is Positive.
- True Negative (TN): When the actual value is Negative, the predicted value is also Negative.

The final stage of this research is to view and evaluate the performance of each model. For the SVC model, it gets an accuracy of 0.88, a precision of 0.80, a recall of 0.76, and an f-1 score of 0.76.

	Precision	Recall	F1-Score	Support
1.0	0.93	0.95	0.94	334
2.0	0.64	0.62	0.63	60
3.0	0.82	0.72	0.77	32
Accuracy			0.88	426
Macro Avg.	0.80	0.76	0.78	426
Weighted Avg.	0.88	0.88	0.88	426

Fig. 13 Super Vector Classification confusion matrix

The next model is the Logistic Regression model, which gets an Accuracy value of 0.87, Precision of 0.80, Recall of 0.69, and f-1 Score of 0.73.

	Precision	Recall	F1-Score	Support
1.0	0.90	0.97	0.93	334
2.0	0.68	0.42	0.52	60
3.0	0.81	0.69	0.75	32
Accuracy			0.87	426
Macro Avg.	0.80	0.69	0.73	426
Weighted Avg.	0.86	0.87	0.86	426

Fig. 14 Logistic Regression confusion matrix

The final model is the Random Forest model, which gets an accuracy value of 0.94, precision of 0.91, recall of 0.88, and f-1 score of 0.89. The Random Forest gave the best performance, followed by SVC and Logistic Regression. The next step is to compare the performance of the model before being given hyperparameter tuning and after being given tuning. The comparison of model performance is only based on the level of accuracy and does not include Precision, Recall, and f-1 score indicators. The performance comparison is as in the following Table 6.

	Precision	Recall	F1-Score	Support
1.0	0.96	0.98	0.97	334
2.0	0.81	0.78	0.80	60
3.0	0.97	0.88	0.92	32
Accuracy			0.94	426
Macro Avg.	0.91	0.88	0.89	426
Weighted Avg.	0.94	0.94	0.94	426

Fig. 15 Random Forest confusion matrix

Table 6. Comparison of model performance accuracy

Model	Hyperparameter Tuning Accuracy	
	Before	After
Support Vector Classifier	0.853529	0.88
Logistic Regression	0.857059	0.87
Random Forest Classifier	0.940588	0.94

4. Conclusion

The research was to find out the most appropriate machine learning method to use in predicting fetal health. The three methods that are the focus of this research are to compare their performance based on the resulting accuracy, and accuracy after being given hyperparameter tuning. Hyperparameter tuning provides parameters for each model intending to improve model performance. The results of this research are that before hyperparameter tuning is given, the Support Vector Classifier (SVC) provides an accuracy of 0.853, Logistic Regression of 0.857, and Random Forest of 0.94. However, after being given hyperparameter tuning, SVC has an accuracy of 0.88, precision of 0.80, recall of 0.76, and f-1 score of 0.78. Logistics has an accuracy of 0.87, precision of 0.80, recall of 0.69, and f-1 score of 0.73. Random Forest has an accuracy of 0.94, Precision of 0.91, Recall of 0.88, and f-1 score of 0.89. Of the three algorithms that have been compared, Random Forest has the highest accuracy and is the most suitable for use in predicting fetal health.

Acknowledgments

This work is partially supported by the Directorate General of Higher Education, Ministry of Education and Culture, Republic of Indonesia - Ministry of Education, Culture, Research, and Technology - Matching Fund Grant - Batch 2, 2023.

References

[1] Ying Dong, and Jia-Lin Yu, "An Overview of Morbidity, Mortality and Long-Term Outcome of Late Preterm Birth," *World Journal of Pediatrics*, vol. 7, pp. 199-204, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[2] Jonathan Bearak et al., "Unintended Pregnancy and Abortion by Income, Region, and the Legal Status of Abortion: Estimates from A Comprehensive Model for 1990-2019," *The Lancet Global Health*, vol. 8, no. 9, pp. e1152-e1161, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Mary E. Coussons-Read, "Effects of Prenatal Stress on Pregnancy and Human Development: Mechanisms and Pathways," *Obstetric Medicine*, vol. 6, no. 2, pp. 52-57, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[4] Terence H. Hull, Sarsanto W. Sarwono, and Ninuk Widyantoro, "Induced Abortion in Indonesia," *Studies in Family Planning*, vol. 24, no. 4, pp. 241-251, 1993. [CrossRef] [Google Scholar] [Publisher Link]

[5] Zarko Alfirevic et al., "Continuous Cardiotocography (CTG) as a Form of Electronic Fetal Monitoring (EFM) for Fetal Assessment during Labour," *The Cochrane Database of Systematic Reviews*, vol. 2, pp. 1-113, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[6] Rosalie M. Grivell et al., "Antenatal Cardiotocography for Fetal Assessment," *The Cochrane Database of Systematic Reviews*, vol. 9, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[7] Dr. Emeil Kamel, Cardiotocography (CTG), 2017. [Online]. Available: https://www.dremeilkamel.com.au/patient-resources/obstetrics/cardiotocography/

[8] J.T. Parer et al., "Fetal Acidemia and Electronic Fetal Heart Rate Patterns: Is there Evidence of an Association?," *The Journal of Maternal-Fetal & Neonatal Medicine*, vol. 19, no. 5, pp. 289-294, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[9] Abolfazl Mehbodniya et al., "Fetal Health Classification from Cardiotocographic Data Using Machine Learning," *Expert Systems*, vol. 39, no. 6, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Dwi Normawati, Rizki Akbari, and Arfiani Nurhusna, "Early Diagnosis of Preeclamsia in Pregnant Women Using the K-Nearest Neighbor (KNN) Method," *Informatics Dynamics: Scientific Journal of Technology*, vol. 13, no. 2, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] I. Sulihati, A. Syukur, and A. Marjuni, "Fetal Health Detection Using Decision Tree and Forward Selection Feature," *Building of Informatics, Technology and Science*, vol. 4, no. 3, pp. 1658-1664, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] Irzan Fajari Nurahmadan et al., "Comparison of Machine Learning Algorithms for Fetal Heart Rate Classification," *National Seminar for Computer Science Students and Their Applications*, vol. 2, no. 1, pp. 733-740, 2021. [Google Scholar] [Publisher Link]

[13] Muhammad Arif, "Classification of Cardiotocograms Using Random Forest Classifier and Selection of Important Features from Cardiotocogram Signal," *Biomaterials and Biomechanics in Bioengineering*, vol. 2, no. 3, pp. 173-183, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[14] Cardiotocography, Donated on 9/6/2010, UCI Machine Learning Repository, 2010. [Online]. Available: https://archive.ics.uci.edu/dataset/193/cardiotocography

[15] Fetal Health Classification, Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/andrewmvd/fetal-health-classification

[16] Data Pre-Processing in Data Mining, GeeksforGeeks, 2023. [Online]. Available: https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/

[17] Kiran Maharana, Surajit Mondal, and Bhushankumar Nemade, "A Review: Data Pre-Processing and Data Augmentation Techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91-99, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Scott Robinson, Katie Terrell Hanna, and Jacqueline Biscobing, Data Exploration, Business Analytics, 2024. [Online] Available: https://www.techtarget.com/searchbusinessanalytics/definition/data-exploration

[19] V. Roshan Joseph, "Optimal Ratio for Data Splitting," *Statistical Analysis and Data Mining*, vol. 15, no. 4, pp. 531-538, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Giordano d'Aloisio, Antinisca Di Marco, and Giovanni Stilo, "Modeling Quality and Machine Learning Pipelines through Extended Feature Models," *arXiv*, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[21] Hannes Hapke, and Catherine Nelson, *Building Machine Learning Pipelines: Automating Model Life Cycles with TensorFlow*, O'Reilly Media, 2020. [Google Scholar] [Publisher Link]

[22] Stephen Oladele, How to Build an End-to-End ML Pipeline, neptune.ai, 2024. [Online] Available: https://neptune.ai/blog/building-end-to-end-ml-pipeline

[23] C.G. Siji George, and B. Sumathi, "Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 9, pp. 173-178, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[24] Muhammad Ariqleesta Hidayat, GRIDSEARCHCV, Algoritma Technical Blog, 2021. [Online] Available: https://algotech.netlify.app/blog/gridsearchcv/

[25] Confusion Matrix, Binus University School of Computer Science, 2020. [Online] Available: https://socs.binus.ac.id/2020/11/01/confusion-matrix/