Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 197 (2022) 162–171

Procedia Computer Science

www.elsevier.com/locate/procedia

Sixth Information Systems International Conference (ISICO 2021)

Comparison of machine learning algorithms to classify fetal health using cardiotocogram data

Nabillah Rahmayanti, Humaira Pradani, Muhammad Pahlawan, Retno Vinarti*

Information Systems Department, Faculty of Intelligent Electrical and Information Technology, Institut Teknologi Sepuluh Nopember (ITS), Indonesia

Abstract

Cardiotocogram (CTG) is one of the monitoring tools to estimate the fetus health in womb. CTG mainly yields two results fetal health rate (FHR) and uterine contractions (UC). In total, there are 21 attributes in the measurement of FHR and UC on CTG. These attributes can help obstreticians to clasify whether the fetus health is normal, suspected, or pathological. This paper compares 7 algorithms to predict the fetus health: Artificial Neural Network (ANN), Long-short Term Memory (LSTM), XG Boost (XGB), Support Vector Machine (SVM), K-Nearest Neighbour (KNN), Light GBM (LGBM), and Random Forest (RF). By employing three scenarios, this paper reports the performance measurements among those algorithms. The results show that 5 out of 7 algorithms perform very well (89-99% accurate). Those five algorithms are XGB, SVM, KNN, LGBM, RF. Furthermore, only one from five algorithm that always performs well through three scenarios: LGBM.

Keywords: Fetal health; cardiotocogram; machine learning; classification.

1. Introduction

Pregnancy is the happiest period in a woman's life. During pregnancy, a mother must really take care of her health with great care, since she is carrying a baby. To monitor fetal growth and development, several laboratory tests are suggested per trimester. One of the tests is cardiotocogram (CTG) which is commonly used in clinical evaluation to check the health state of the fetus in the uterus. Prenatal monitoring using two CTG signals, namely, fetal heart rate (FHR) and uterine contractions (UC).

* Corresponding author. Tel.: +6287766507967.
E-mail address: zahra_17@is.its.ac.id

10.1016/j.procs.2021.12.130

FHR represents the number of fetal heart beats per minute (BPM). In the clinical prenatal health diagnosis of the fetus, CTG is a tool used to monitor fetal activity and heart rate, as well as uterine contractions when the baby is in the womb. Through this examination, doctors can evaluate whether the fetus is healthy before and during delivery. By providing important physiological and pathological information to obstetricians, CTG results can prevent preterm birth and reduce risk of perinatal mortality. According to the International Federation of Gynecology and Obstetrics (FIGO), CTG test results can be classified into three classes: normal, suspect, or pathological. These classes are based on FHR, variability of FHR, accelerations and decelerations [1]. This is performed by obstetricians and can also be done by a software. According to Grivell et al, it was stated that there was a significant reduction in perinatal mortality with the use of computerized CTG with a relative risk result of 0.20 and a confidence interval of 95% when compared to traditional CTG. However, since this study has evidence of moderate quality, further studies to assess the impact of CTG on perinatal outcome are required [2]

In recent years, signal processing technology has used artificial intelligence to convert data from the human body into a diagnosis. Medical professionals are working to create an automated interpretation of CTG but the results have not been able to predict suspicious fetal conditions [3]. So that many researchers began to try to do research by applying various machine learning algorithms to predict the state of the fetus in the mother's stomach. As in this study, the aim is to develop a machine learning model that can identify pathologically suspicious or high-risk fetuses as accurately as possible as well as trained medical professionals.

2. Related works

Within more than 30 years after CTG began to be known and used to see the risk of the fetus being conceived. However, the predictive capacity of CTG is still controversial [4]. This refers to a review conducted by Devoe et al. Based on the results of a review of 45 studies that he conducted, he found that the reported sensitivity of the use of CTG ranged from 2% to 100% and the specificity ranged from 37% to 100% [5]. This is an opportunity for the use of machine learning to overcome the shortcomings of the sensitivity and specificity results based on a review of several studies conducted by Devoe et al, [5] and also eliminate the controversial effect regarding the predictive capacity of the use of CTG.

From several previous studies regarding machine learning for automatic CTG classification based on the same dataset, namely fetal health CTG records from the UCI Machine Learning Repository with a total of 2,126 data and 21 features and labeling categorized as Normal (N), Suspected (S) and Pathological (P) for the classification method using several machine learning methods such as Random Forest, Support Vector Machine (SVM), Decision Tree and k-Nearest Neighbor with Accuracy, F1-Score, and ROC results reaching above 90% [6][7]. However, there are several studies that also use the concept of ensemble learning [8] with the based model, namely Random Forest and the final candidate, namely LightGBM which is hybridized with Gaussian Naïve Bayes optimization which maps the log loss model by calculating it using K-Fold Cross Validation, namely k= 4, the resulting performance is 95% Accuracy, 89% Recall, 92% Precision and 91% F1-Score. In addition, what is quite a concern is the use of deep learning methods such as ANN to classify, and the results of the Precision, F1-Score and Recall performance obtained are quite good up to above 90% as Sundar et al, [3] implemented a model that utilizes an artificial neural network (ANN) to classify CTG data. In addition [3] proposed K-Means clustering for CTG classification.

Based on previous research using several machine learning methods, using either a single classifier, ensemble or hybrid method but with the aim of getting better performance, the contribution of this research is to compare the performance results of several machine learning methods such as several ensemble learning methods, such like Extreme Gradient Boosting (XGB), Light Gradient Boosting (LGBM), and Random Forest (RF) and other single classifiers namely Support Vector Machine (SVM), K-Nearest Neighbors (KNN), and the last are two methods that are still within the scope of machine learning but categorized into deep learning, namely the Artificial Neural Network (ANN) and Long-short Term Memory (LSTM) methods. We use these methods to compare the performance results in the hope of getting better Accuracy, Precision, Recall and F1-Score results.

3. Methods

3.1. Material and data

In this study, the dataset used was obtained from the University of California Irvine Machine Learning Repository, which is a public dataset. It consisted of 2,126 data on pregnant women who are in the third trimester of their pregnancy. This dataset contains of 21 attributes used in the measurement of FHR and UC on CTG in Table 1. According to the standards of the National Institute of Child Health and Human Development, the variables with the main risk used to determine the state of the fetus based on the description of the FHR are baseline heart rate, baseline variability, number of accelerations per second, number of early, late and variable decelerations per second, number of prolonged decelerations per second and sinusoidal patterns and uterine contractions which can be seen from the baseline uterine tone, contractions frequency, duration and strength. CTG results of pregnant women were classified by three experts in the field of obstetrics with interpretations of them being categorized as the gold standard. Fetal CTG is generated by SisPorto 2.0 (Speculum, Lisbon, Portugal) which is a program to automatically analyze CTG results [9].

Table 1. Cardiotocogram attributes used in the model.

Variable Symbol	Variable Description
BV	Baseline Value (FHR beats per minute)
AC	Accelerations (number of accelerations per second)
FM	Fetal Movement (number of fetal movement)
UC	Uterine Contractions (number of uterine contractions per second)
LD	Light Decelerations (number of light decelerations per second)
SD	Severe Decelerations (number of severe decelerations per second)
PD	Prolonged Decelerations (number of prolonged decelerations per second)
ASTV	Abnormal Short-Term Variability (percentage of time with abnormal short-term variability)
MSTV	Mean Value of Short-Term Variability
ALTV	Percentage of Time with Abnormal Long-Term Variability
MLTV	Mean Value of Long-Term Variability
HW	Histogram Width (Width of FHR histogram)
HMax	Histogram Max (Maximum of FHR histogram)
Hmin	Histogram Min (Minimum of FHR histogram)
NP	Number of Histogram Peaks
NZ	Number of Histogram Zeroes
HMo	Histogram Mode
HMe	Histogram Mean
HMed	Histogram Median
HV	Histogram Variance
HT	Histogram Tendency
NSP	Fetal Health (Fetal state class code, N=normal, S=Suspected, P=Pathological)

3.2. Pre-processing techniques

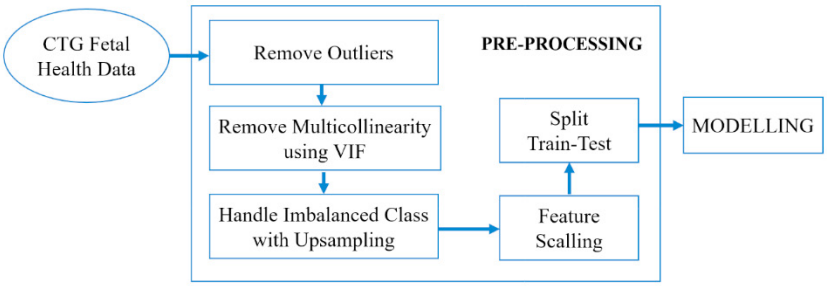

Fig. 1 Pre-processing flowchart

This stage aims prepare the data to improve the performance of the model used with certain actions taken. In this preprocessing stage there are steps defined in a flow chart Fig. 1. The preprocessing stage begins by entering the fetal health dataset derived from the CTG interpretation with a total of 2126 data and 21 columns, then the next process is to remove outliers which can reduce the accuracy of the model used. In this study, the outliers were found in 6 attributes (see Table 2).

Table 2. Number of outliers removed.

Attributes	Upper bound	Lower bound	Outliers Removed
HV	105.74	-68.12	44
HMed	180.71	96.21	17
HMe	178.71	92.08	17
HMo	180.29	97.08	17
ALTV	66.39	-45.78	57
MSTV	3.78	-1.17	30

After deleting the outliers, the number of rows of data becomes 1944 rows and the number of columns remains 21 columns. Elimination of outliers is done by using 3 standard deviations with a threshold of 3 or -3 so if the value exceeds 3 or -3 standard deviation, those values are deemed as outliers. Hence, they will be deleted [10].

After we know the outliers of data have been deleted, the next step is to check the correlation between variables whether there are variables that have a very high correlation level. By using a correlation heat map (Fig. 2), it can be seen that there are several variables that have a high correlation. The strong relationship between these predictor variables is called multicollinearity. Variable predictors that have a strong relationship are detected and removed to reduce fit using Variable Inflation Factors (VIF) [11].

In this study, a check was carried out to identify whether there is imbalanced data or not. The results shows that the data is imbalanced, it means that each class has extreme unequal number. Therefore, balancing technique was carried out using up sampling where the three classes, Normal (N), Suspected (S) and Pathological (P), have the same amount of data.

After balancing the data, we performed feature scaling which aims to consolidate or transfer the data into ranges and forms that are appropriate for modeling. Models trained on scaled data usually have significantly higher performance compared to the models trained on unscaled data. So data scaling is considered as an essential step in data preprocessing [12]. After the feature scaling process, the next process is to divide the data into 75% data for training and 25% data for testing before we use the data for modelling.

3.3. Algorithms

Machine learning is a powerful tool and widely used in healthcare, especially in fetal research study. For example, machine learning has been used to estimate fetal weight [13], predict probability of fetal hypoxia [14], predict fetal growth and gestational age [15]. Meanwhile, this study will focus on classifying fetal health using CTG data with machine learning and deep learning methods. Deep learning is a subset of machine learning that have special characteristics such as (1) extract high-level features from data, so good features can be learned automatically using a general-purpose learning procedure, (2) better performance on big dataset than basic machine learning methods, (3) work efficiently on high-end machines, (4) has many parameters so takes a long time to train (5) low interpretability [16]. Several algorithms were used to classify fetal health: Artificial Neural Network (ANN), Long-short Term Memory (LSTM), Extreme Gradient Boosting (XGB), Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Light Gradient Boosting (LGBM), and Random Forest (RF). XGB, RF and LGBM are ensemble algorithms that use decision tree principal [17] [18] [19]. KNN determines classifications based on the class belonging to the closest point, while SVM determines the class by the hyperplane and support vectors [20] [21]. ANN is a model that applies the principle of a neural network, and LSTM is a neural network development that includes deep learning [22]. Validation of the models were done with a test-set proportion of 25% from overall dataset. All processes from training models, validation models to parameter optimization used Python's Scikit-Learn and Keras module.

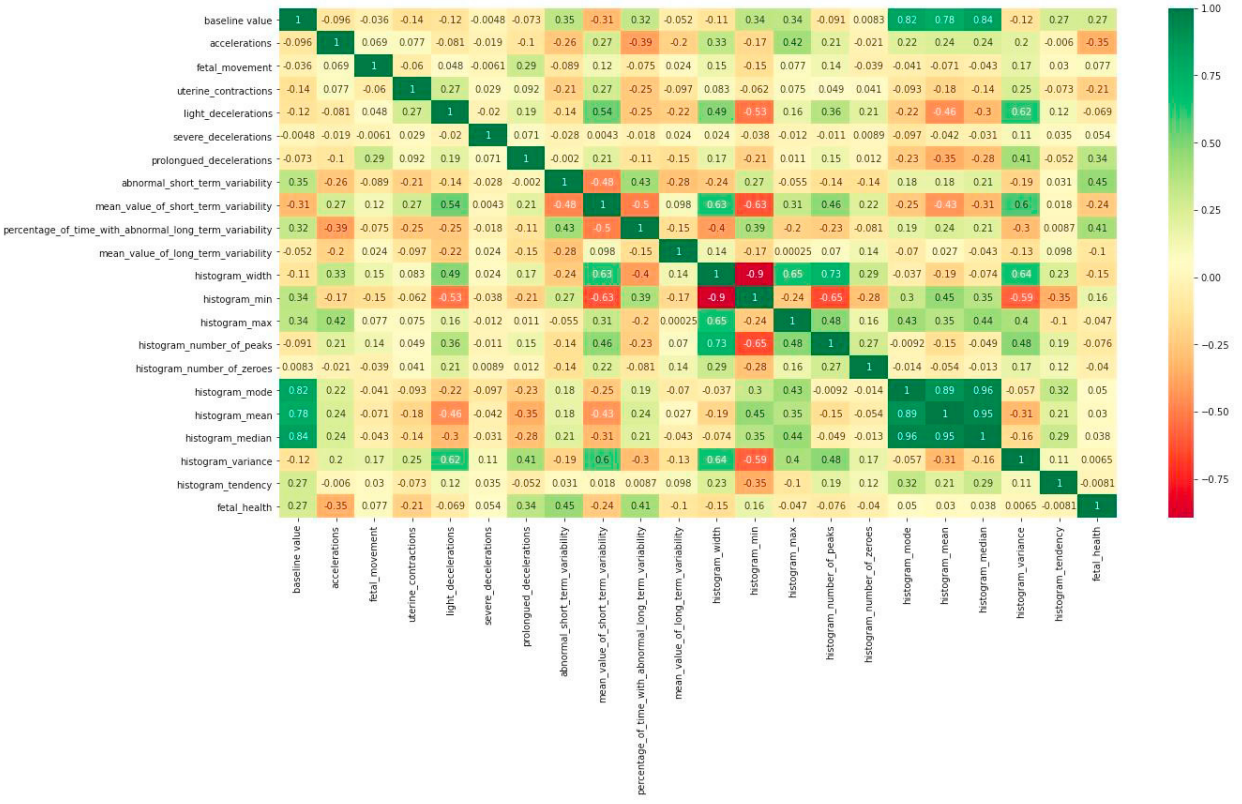

Fig. 2 Correlation heatmap

Table 3. Classifier parameters.

Algorithm	Parameters
ANN	Node in hidden layer, ptimizer, activation function, batch
LSTM	Node in hidden layer, ptimizer, activation function, batch
XGB	Minimum child weight, subsample
SVM	C, kernel function
KNN	Number of neighbors
LGBM	Number of leaves, type of boosting
RF	Minimum sample split, maximum features, number of estimators

For each model, parameter optimization was done by using grid search cross validation. The concept of this method is to create a combination of parameters in the form of a grid and test some points in the grid to produce the best model with best combination of parameters. Parameters were tuned in each algorithm listed in Table 3. For ANN and LSTM, we evaluated classification performance by the combination of node in hidden layer (n, 2n, 3n input), three optimizers (adam, rmsprop, sgd), three activation functions (rectified linear unit, softsign, tangent) and three batch counts (32, 64, 96) with using one hidden layer on each model. For XGB, we studied different numbers of minimum sum of instance weight needed in a child node, as known as min child weight (1,5,10) and subsample ratio of the training instances (0.6, 0.8, 1.0). For KNN, performances of numbers of neighbors with range 1-31, k ∈ (1,…,31), were evaluated. For SVM, performance of models was studied by different number of cost (0.1, 1, 10, 100) and kernel function (a polynomial function, a radial basis function (RBF), and a sigmoid function). For LGBM, the number of leaves from 6 to 31 and boosting type (Gradient Boosting Decision Tree and Dropouts meet Multiple Additive Regression Trees) were used for the evaluation. For Random Forest model, we tested different numbers of minimum sample split (2,4,6), maximum proportion of features used for model training (0.5, 1.0), and number of estimators (50, 150, 250) to avoid overfitting and underfitting.

3.4. Classifier evaluation

In this study, several metrics were used to measure the performance of classification models that had been produced. These metrics include accuracy, precision, recall, and f1 score, defined in Eq. (1) - (4). For multiclass classification, the confusion matrix is built for the analysis of the classification results on each model. Based on those confusion matrix, accuracy, precision, recall, and f1 score can be calculated. Accuracy captured percentage of correct predictions of overall test data, when precision and recall measured the ability of a model to identify relevant data points and to find all relevant cases within a dataset. High precision implied a low false positive rate, while high recall implied a low false negative rate. F1 score is a metric that combines precision and recall. High F1 score indicated a robust classification model. Moreover, the Area Under the Curve (AUC) score was also used to assess how much the model is capable of distinguishing between classes. A high number of AUC score implied the model can distinguish between Normal (N), Suspected (S) and Pathological (P) class [23].

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

$$Precision = \frac{TP}{TP+FP} \times 100\% \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \tag{3}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \times 100\% \tag{4}$$

4. Results and discussion

In this study, we use three experimental scenarios in the pre-processing stage, where the first scenario (S1) is pre-processing by removing outliers and using data that has been balancing with up sampling technique, but the difference is without removing multicollinearity. Multicollinearity is an independent variable that highly correlated with one or more of the other independent variables. Multicollinearity can cause problems when we fit the model and interpret the results. The variables of the dataset should be independent of each other to overdue the problem of multicollinearity. The second scenario (S2) is pre-processing without removing outliers and using normal data without balancing, and the last is the third scenario (S3) by using complete data pre-processing such as: removing outliers, removing multicollinearity, balancing data using up sampling techniques. All these processes are carried out before the data is split into 75% training and 25% testing. From the results of the three scenario processes that have been described, Fig. 3 produces a bar chart which explains that without using the multicollinearity removal process, the accuracy is higher than the other two scenarios. This is because eliminating multicollinearity variables reduces features and removes information to generate predictions. But whether doing multicollinearity removal does not really affect the prediction results. Machine learning algorithms such as decision trees are robust in handling multicollinearity. For example, if in this data there are 2 features which are 99% correlated, when deciding to split, the tree selects only one of them.

Three scenarios that we can see in Fig. 3 produces accuracy in each method used. From the results of the three scenarios, the S1 accuracy results are superior to the others. That is because S1 does not eliminate multicollinearity which has an impact on removing several features that affect the lack of information on the data needed to predict. Although the accuracy results by eliminating multicollinearity on S3 are not too different from the accuracy results on S1 which produce higher accuracy.

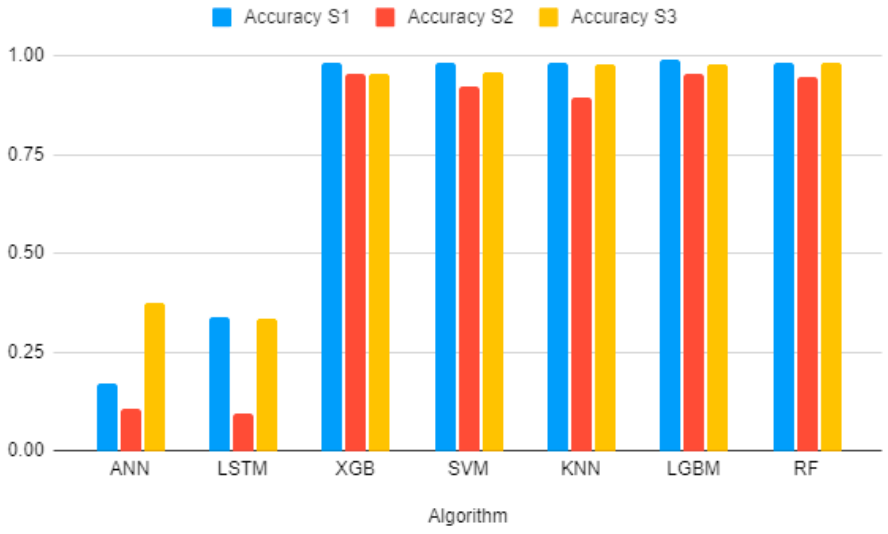

Fig 3. Accuracy result on 3 scenario

Experiments were carried out based on data pre-processing result and the tuning parameters as mentioned in Table 3 where the accuracy were used to help choose the best model. Table 4 show the results with all classifier models on testing set. LightGBM and Random Forest can produce the best AUC and F1-Score values with their respective values 0.99 and 0.98. LightGBM has a higher model performance in terms of accuracy of 0.99 while Random Forest is only 0.98. Therefore, LightGBM is the most suitable method to use in this experiment based on the measurements that have been made. Fig. 4 is the confusion matrix of the LGBM model. That matrix is a summary of the prediction result by LGBM. Based on that matrix, we can see that almost all the target classes are can correctly be predicted by the model. The LGBM model can produce an accuracy value, f1-score, and AUC has an almost perfect value of 0.99. It

is not surprising to see the confusion matrix generated by the model as is shown in Fig. 4. The LGBM model only incorrectly predicts the actual data which should be normal, to be suspected of only 12 data. For other classes, suspected and pathological, the model can predict well.

In this experiment, as we can see Table 4, all machine learning models produce impressive model performance. From the measurements made, almost all of them reached values above 0.94 and some even reached 1 or perfect. Machine learning models can produce better performance than deep learning models like LSTM and ANN. Based on the experiments, best model of ANN for scenario 1 is 14 nodes in hidden layer, 16 batch size, 'relu' activations, 'adam' optimizer, 14 nodes in hidden layer, 32 batch size, 'relu' activations, 'sgd' optimizer for scenario 2, and 63 nodes in hidden layer, 64 batch size, 'relu' activations, 'adam' optimizer for scenario 3. For the LSTM model, best model of for scenario 1 is 28 nodes in hidden layer, 32 batch size, 'relu' activations, 'adam' optimizer, 14 nodes in hidden layer, 32 batch size, 'softsign' activations, 'sgd' optimizer for scenario 2, and 42 nodes in hidden layer, 32 batch size, 'softsign' activations, 'sgd' optimizer for scenario 3. The deep learning model performance used is LSTM with AUC score of 0.50 and F1-Score is 0.17 while the ANN model's performance with AUC score of 0.38 and F1-Score is 0.13. Based on this experiments, deep learning not suitable for this dataset because cannot produce the accuracy impressively. The deep learning model has increased accuracy in the 3rd scenario where data preprocessing uses multicollinearity. This can be an indication that deep learning is not doing well with high-dimensional data.

Table 4. Result of all classifiers.

Algorithm	Experimental Scenario								
	Scenario 1			Scenario 2			Scenario 3		
	F1-Score	AUC	Accuracy	F1-Score	AUC	Accuracy	F1-Score	AUC	Accuracy
XGB	0.98	0.98	0.98	0.92	0.9341892738	0.9530075188	0.96	0.966899534	0.9550374688
SVM	0.98	0.98	0.98	0.85	0.8811842787	0.9210526316	0.96	0.9688067707	0.9575353872
KNN	0.98	0.99	0.98	0.8	0.8550916953	0.8947368421	0.98	0.9835124573	0.9775187344
LGBM	0.99	0.99	0.99	0.92	0.9301560314	0.954887218	0.98	0.9822555746	0.9758534555
RF	0.99	0.99	0.98	0.91	0.9155201859	0.9454887218	0.98	0.9859550562	0.9808492923
ANN	0.13	0.38	0.17	0.11	0.5043607093	0.1071428571	0.29	0.5376216985	0.3738551207
LSTM	0.17	0.50	0.34	0.06	0.5	0.09398496241	0.33	0.5008283865	0.3355537052

Table 5. Result of experimental scenario 1.

Algorithm	Precision			Recall			F1-Score	AUC	Accuracy
	N	S	P	N	S	P			
XGB	0.99	0.94	1.00	0.94	0.99	1.00	0.98	0.98	0.98
SVM	0.98	0.95	0.99	0.95	0.98	1.00	0.98	0.98	0.98
KNN	1.00	0.95	1.00	0.94	1.00	1.00	0.98	0.99	0.98
LGBM	1.00	0.97	1.00	0.97	1.00	1.00	0.99	0.99	0.99
RF	0.99	0.95	1.00	0.95	0.99	1.00	0.99	0.99	0.98
ANN	0.13	0.19	0.00	0.14	0.38	0.00	0.13	0.38	0.17
LSTM	0.34	0.00	0.00	1.00	0.00	0.00	0.17	0.50	0.34

In terms of precision and recall measurements, the precision value of the "Suspected" class is generally lower than the other two classes. Meanwhile, the recall score in the "Normal" class tends to be lower than the other classes. After further analysis through the results of the confusion matrix that is formed, it was known that this is due to some prediction errors that mostly occur on these classes. Approximately 3-6% data points that have the actual label as "Normal" class were misclassified as "Suspected" class and this affects the recall value of "Normal" class. Meanwhile, some data that were predicted to be classified as "Suspected" were in "Normal" class and this affected the precision

value of "Suspected" class.

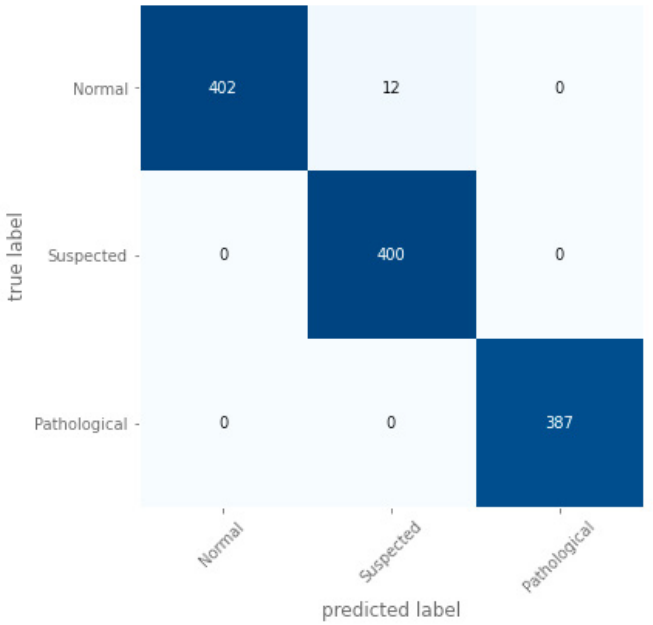

Fig. 4. Confusion matrix (LGBM)

5. Conclusion

This study conducts any kind of pre-processing stage before entering the modeling stage such as outlier removal with up sampling dataset and eliminates multicollinearity. Data without pre-processing stage are also compared to know how the effect of the pre-processing stage on model performance. Based on the results of the experiments, data that does not contain multicollinearity features cannot improve the model performance. Hence proved that multicollinearity is not a big deal for feature engineering in this study case. Based on the performance results of the 7 algorithms used in this study to predict fetal health, we can see the comparison. The use of algorithms other than deep learning algorithms, namely XGB, SVM, KNN, LGBM and RF, produces a fairly high level of accuracy, ranging from 89-99% accurate in each scenario used. While the use of deep learning algorithms, namely ANN and LSTM, only produces accuracy with a range of 9-37% accurate in each scenario used. It can be concluded from the performance results obtained that the use of tree-based classifiers such as XGB, LGBM and RF is superior in each scenario used, followed by the level of accuracy generated by SVM and KNN. However, the difference is that SVM and KNN are not superior to the tree-based classifier algorithm in the 2nd scenario and the lowest accuracy results are in the ANN and LSTM deep learning algorithms in all scenarios we used in this study. but deep learning model has increased accuracy in the 3rd scenario where data preprocessing uses multicollinearity. This can be an indication that deep learning is not doing well with high-dimensional data.

This study is not without limitations. First, the data used in this study only comes from one main source. Meanwhile, other data sources that are generated from tools and other geographic locations may provide different data characteristics. Second, in this study the researchers compared the algorithm's performance in general. There is not much discussion of each algorithm. Even so, this research can certainly be the first step for further research that can review the performance of the algorithm in more detail.

The next research that can be done is to compare the algorithm performance with more representative data conditions such as the geographical location, age, and type of CTG can certainly be considered as variable input for this study case. The dataset from CTG contains many variables which have potential future research on how to deals with that many variables. This study already implements some of the feature engineering techniques such as up

sampling data, outlier removal, and removing multicollinearity on some features to remove which features are considered not relevant. In the next future research, can consider for use other feature engineering techniques such as Principle Component Analysis (PCA) to handling big dimensional data.

References

[1] Ayres-De-Campos, D., C. Y. Spong, and E. Chandraharan. (2015) "FIGO GUIDELINES FIGO consensus guidelines on intrapartum fetal monitoring: Cardiotocography" *FIGO Intrapartum Fetal Monitoring Expert Consensus Panel Int J Gynaecol Obs* **131**: 13-24.

[2] Gyte, G. ML., and D. D. Grivell R. M. (2019) "Cochrane Database of Systematic Reviews Antenatal cardiotocography for fetal assessment (Review)." *Antenatal cardiotocography for fetal assessment Cochrane Libr* **9**: CD007863.

[3] S. C, M. C. M. Chitradevi, and G. Geetharamani. (2012) "Prediction of neonatal state by computer analysis of fetal heart rate tracings: The antepartum arm of the SisPorto® multicentre validation study." *Int. J. Comput. Appl* **47 (14)**: 19-25.

[4] Ayres-De-Campos, D., C. Costa-Santos, and J. Bernardes. (2005) "Prediction of neonatal state by computer analysis of fetal heart rate tracings: The antepartum arm of the SisPorto® multicentre validation study." *Eur. J. Obstet. Gynecol. Reprod. Biol* **118 (1)**: 52-60.

[5] Devoe, L.D, R. A. Castillo, and D. M. Sherline. (1985) "The nonstress test as a diagnostic test: A critical reappraisal." *Am. J. Obstet. Gynecol* **152 (8)**: 1047–1053.

[6] Friedman, J. H. (2001) "Greedy Function Approximation : A gradient Boosting Machine" *Annals of statistics*: 1189-1232.

[7] Cömert Z., and A. F. Kocamaz. (2017) "Comparison of machine learning techniques for fetal heart rate classification." *Acta Phys. Pol. A* **132 (3)**: 451-454.

[8] Maranhão, A., V. Dadario. (2021) "Classification of Fetal State through the application of Machine Learning techniques on Cardiotocography records : Towards Real World Application.", *medRxiv* [Online]. Available: https://www.medrxiv.org/content/10.1101/2021.06.03.21255808v1.

[9] Garrido, A., D. Ayres-de-campos, and J. Marques-de-sa. (2000) "Cardiotocograms." *Journal of Maternal-Fetal Medicine* **9 (5)**: 311-318.

[10] Seo, S., P. D. Gary M. Marsh. (2006) "A review and comparison of methods for detecting outliersin univariate data sets." *Dep. Biostat. Grad. Sch. Public Heal* 1-53.

[11] Shrestha, N. (2020) "Detecting Multicollinearity in Regression Analysis." *Am. J. Appl. Math. Stat* **8 (2)**: 39-42.

[12] Cao, X. H., I. Stojkovic, and Z. Obradovic. (2016) "A robust data scaling algorithm to improve classification accuracies in biomedical data." *BMC Bioinformatics* **17 (1)**: 1-10.

[13] Solt, Ido, Or Caspi, Ron Beloosesky, Zeev Weiner, and Eyal Avdor. (2019) "Machine learning approach to fetal weight estimation." *American Journal of Obstetrics and Gynecology* **220 (1)**: S666-S667.

[14] Alsaggaf, Wafaa, Zafer Cömert, Majid Nour, Kemal Polat, Hani Brdesee, and Mesut Tog˘açar. (2020) "Predicting fetal hypoxia using common spatial pattern and machine learning from cardiotocography signals." **167**: 107429.

[15] Ananthet, Cande V, and Justin S Brandt. (2020) "Fetal growth and gestational age prediction by machine learning." *The Lancet Digital Health* **2 (7)**: e336-e337.

[16] Tiwari, T., T. Tiwari, & S. Tiwari. (2018) "How Artificial Intelligence, Machine Learning and Deep Learning are Radically Different?" *International Journal of Advanced Research in Computer Science and Software Engineering* **8**: 1-9.

[17] Chen, T., Carlos Guestrin. (2016) "XGBoost: A Scalable Tree Boosting System.", In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*: 785-794.

[18] Breiman. (2001) "Random Forest." *Machine Learning* **45 (1)**: 5-32.

[19] Ke, Guolin, Qi Meng, Thomas Finley, and Tie Yen Liu. (2017) "LightGBM: A Highly Efficient Gradient Boosting Decision Tree." *Advances in neural information processing systems* **30**: 3146-3154.

[20] Cunningham, Padraig, and Sarah Delany. (2007) "k-Nearest neighbour classifiers." *arXiv preprint arXiv:2004.04523* [Online]. Available: https://arxiv.org/abs/2004.04523.

[21] Srivastava, Durgesh, and Lekha Bhambhu. (2017) "Data classification using support vector machine." *Journal of Theoretical and Applied Information Technology* **12**: 1-7.

[22] Staudemeyer, Ralf, and Eric Morris. (2019) "Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks." arXiv preprint arXiv:1909.09586 [Online]. Available: https://arxiv.org/abs/1909.09586.

[23] Hossin, Mohammad, and Sulaiman M.N. (2015) "A Review on Evaluation Metrics for Data Classification Evaluations." *International Journal of Data Mining & Knowledge Management Process* **5**: 1-11.