The Acquiring Optimal Models of Random Forest and Support Vector Machine Through Tuning Hyperparameters in Classifying the Imbalanced Data

Dwija Wisnu Brata[1], Arif Djunaidy[2]*, Daniel Oranova Siahaan[3], Samingun Handoyo[4]
Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia [1, 3]
Department of Information Systems, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia [2]
Department of Information Systems, Brawijaya University, Malang, Indonesia [1]
Data Science Study Program, Brawijaya University, Malang, Indonesia [4]

***Abstract*—Machine learning models most often misclassify the positive class in the dataset with class imbalance. Besides, a sophisticated model involves the hyperparameters that need to be tuned to the optimal values. The study aims to tune hyperparameters of random forest (RF) and support vector machine (SVM) models using 5-fold cross-validation data, to build the best RF and SVM for two data scenarios: the original and oversampling training data, and to compare the models' performances in either the training or testing data. The RF hyperparameters: the instance number in the leaf node and tree depth of the RF, were acquired (500, 10), respectively. Whereas, the SVM hyperparameters: the values of gamma and constant, were acquired (0.001, 500), respectively. The benchmark models achieved around 98% across the accuracy, precision, recall, and F1 score metrics. However, it performed worse on the Mathew's Correlation Coefficient (MCC) and Area Under the Curve (AUC): 0.0000 and 0.5000, respectively. The models trained on the class-imbalance dataset failed to predict the positive class. Although the best RF and SVM models trained on the oversampled dataset perform worse than both benchmark models across four standard metrics, the RF best model shows improvements of approximately 7% (from 0.000 to 0.067) and 11% (from 0.500 to 0.612) while the SVM best model show slightly different improvements of approximately 6% (from 0.000 to 0.056) and 11% (from 0.500 to 0.611) in MCC and AUC, respectively. Both the RF and SVM models improve in predicting the positive class, and the best RF model performs slightly better.**

***Keywords*—Area under the curve; cross-validation folds; Matthew's correlation coefficient; optimal hyperparameters; oversampling technique**

I. Introduction

Choosing the right decision from among the decision candidates will lead to the appropriate policy decision-maker. Decision-makers often face binary choices across fields, from economics to the medical sciences. Financial analysts evaluated the company profile, that is, features related to the practice of good accounting, to recommend the company's status, which is fraudulent or not fraudulent [1]. Nutritionists assessed the baby's status as stunted or normal [2]. Midwives decide whether a pregnant woman should give birth through surgery or give birth normally [3]. Unfortunately, binary-class datasets, particularly medical datasets, often exhibit class imbalance. The severity of imbalance with proportions of 80% vs. 20% or 90% vs. 10% is still classified as a standard imbalance class, as Liu et al. stated. [4], but the proportion of 95% vs. 5% or larger gap is categorized as an extreme imbalance class, as stated by Mienye and Sun[5]. Because class imbalance can lead machine learning models to perform poorly at predicting the positive class, the imbalance should be addressed immediately to produce a balanced class dataset.

Machine learning linear models, including logistic regression, linear discriminant analysis, Bayesian, and naïve Bayes, employ a linear hyperplane as the separating boundary to classify instances from different classes [6]. Unfortunately, the linear separating boundary has many possible solution lines, and the optimal solution line is not unique. A Decision tree categorized as a model called a random forest (RF), is a choice for better performance than a decision tree classification model [8]. Some studies have reported satisfactory performance of the RF model, including Speiser et al. [9], Alam et al. [10], and El-Sappagh et al. [11]. Although RF models are relatively resistant to overfitting, some hyperparameters, such as the tree depth and the minimum number of instances required to split a leaf, strongly influence the optimal RF model. Tuning RF hyperparameters should receive significant attention to obtain the best RF model. Another classification model type, the support vector machine (SVM), is a popular model with superior performance on nonlinear hyperplane boundaries [12]. Nevertheless, determining an optimal SVM model is difficult [13]. The SVM with a radial basis function (RBF) kernel has two hyperparameters: a constant term, which is the inverse of the regularization term in the SVM objective function, and a gamma parameter, which controls the width of the RBF kernel [14]. As with RF modeling, tuning the SVM hyperparameters should receive significant attention. The works conducted by Kumar et al. [15], Bektaş [16], and Ganaie et al. [17] are some examples of applications of the SVM classification model. Again, both hyperparameters of the SVM with the RBF kernel must be tuned systematically to obtain the optimal SVM model. The performance evaluation metric is another critical factor in determining the best model [18]. A confusion matrix was used to compute performance metrics [19]. Using different metrics for model performance evaluation can lead to different best models, particularly in imbalanced-class classification.

*Corresponding author.

There are three central issues encountered in developing the best RF and SVM models: oversampling to address class imbalance, tuning the model's hyperparameters using 5-fold cross-validation, and evaluating model performance using six metrics. The imbalanced-class problem is addressed using oversampling, as employed by Kovács [20] and Xu et al. [21], which can significantly improve model performance in classifying instances from the positive class. The hyperparameters of both models were tuned systematically using k-fold cross-validation, as in Passos and Mishra [22] and Nematzadeh et al. [23]. They randomly partitioned the training set into k folds, yielding k pairs of training and validation sets. The performance of both the optimal RF and SVM models will be evaluated by not only four famous metrics, that is, accuracy, precision, recall, and F1 score [24], but also MCC and AUC metrics that should measure the shifting prediction of models to the positive class[25].

The research goals are to develop the best RF and SVM models by oversampling for class imbalance and tuning the models' hyperparameters using cross-validation data. The RF's optimal hyperparameters (minimum number of instances per tree leaf and tree depth) and the SVM's optimal hyperparameters (constant and gamma values) were determined by averaging the k-fold validation accuracies obtained via grid search. The models with optimal hyperparameters of both RF and SVM were trained using either the training set or the oversampled training set. The best model's performance is evaluated on the training and test data using six metrics. Furthermore, the important features of the best RF model are explored. The rest of the sections are organized as follows: reviewing the related works given in Section two, the material data and proposed methods are explored and explained in Section three, the results consisting of the hyperparameter tuning, the model's performance, and the discussion presented in Section four, and finally, Section five presents the conclusion.

II. Related Works

The dataset characteristics, such as high dimensionality and imbalanced classes, tend to influence the machine learning model's performance. In fact, two issues can arise across all aspects of human life, including business (e.g., predicting fraudulent firms [26]), industry (e.g., manufacturing quality prediction [27]), and public health (e.g., cancer classification [28]). Fraud business, unusual products, and cancer patients are minority groups that control the class imbalance in the dataset. Some efforts have been made to handle class imbalance to improve classifier performance. The techniques range from simple methods with basic principles to creating a balanced class through random sampling without replacement, employing oversampling or under sampling methods [29] to sophisticated techniques that were innovated either by hybridization between simulated annealing algorithm for under-sampling and machine learning model [30] or ensemble techniques that concern on combining individual techniques such as under sampling technique, Real Ada-boost, cost-sensitive weight modification, and adaptive boundary decision strategy to acquire classifier models with better performance[31].

The deployment of artificial intelligence across most aspects of human life has made machine learning methods a popular approach and a challenging area of research. In general, machine learning techniques are grouped into unsupervised and supervised methods based on whether a target feature is available [32]. A popular method in unsupervised learning is clustering, which groups similar instances into the same group [33]. The other unsupervised method, namely the ranking method (e.g., the Analytic Hierarchy Process), is often used in decision support systems [34]. When the target feature has a numerical scale, the appropriate model is called a regression model, which can be used to predict [35] or forecast future value [36]. The supervised predictive modeling technique that uses the target feature category is known as a classification model [37]. The categorical target features can be binary [38] or multiclass [39].

Classification models for binary classification can have either a linear decision boundary, such as logistic regression [40] and linear discriminant analysis [41], or a nonlinear one, such as a decision tree [42] and a support vector machine [43]. The decision tree implemented in various fields outperformed the other machine learning models. The ensemble of decision tree models creates a random forest (RF) model that not only mitigates the disadvantages of the decision tree model, such as reducing the risk of overfitting, but also achieves higher performance than the DT model [44]. The RF classifier performed very satisfactorily for the brain-stroke classification by Subudhi et al. [45] and for Arabic sentiment analysis of social media related to COVID-19 conspiracy theories by Al-Hashedi et al. [46]. Nevertheless, similar to the DT model, the RF model also has hyperparameters that should be tuned systematically.

Another classification model with a different approach is the support vector machine (SVM), which uses a hyperplane as the decision boundary [47]. When the decision boundary is complex and nonlinear, SVMs outperform logistic regression [48] and k-means clustering [49]. It also performs comparably to neural network ensembles and boosted shallow trees [50]. The superior performance of both RF and SVM has prompted numerous researchers to compare them, including fraud detection in credit card transactions by Hussain et al. [51], classification of invasive and expansive species by Sabat-Tomala et al. [52], and optimization modeling of callus growth and development in Cannabis sativa done by Hesami and Jones [53]. Based on the above-cited research, RF and SVM show comparable performance, with the difference between the two models not significant. The model's hyperparameters should be adequately tuned to improve the model's performance. Efforts to find the RF-optimal hyperparameters were carried out using various approaches, including the grid search technique [54], Bayesian optimization [55], and an RF model hybridized with a genetic algorithm [56]. On the other hand, the SVM/s hyperparameters, i.e., gamma value and kernel parameter, became the focus of SVM's hyperparameter tuning. Al-Mejibli et al. [57] investigated SVM performance across hyperparameters (e.g., gamma values) and kernel parameters. Wainer and Fonseca [58] employed grid search, random search, Bayesian optimization, simulated annealing, particle swarm optimization, Nelder Mead, and others as methods for SVM tuning hyperparameters.

High-dimensional features and class imbalance are not only common challenges in building machine learning models but can also lead to poor classifier performance. The

implementation of artificial intelligence across most aspects of modern human life has brought machine learning to the forefront as a popular method and a central challenge in the field. It assumes that both RF and SVM will perform satisfactorily, high-performing classifiers for predicting instances with an unknown label when trained with optimal model hyperparameters, so their hyperparameters must be tuned systematically. Comparing the optimal RF and SVM performance remains a challenging research topic, especially for class-imbalanced datasets.

III. The Material and Proposed Method

The dataset containing 52159 patient records for joint replacement surgery is a preprocessed subset of a data mining project. The original data were published as public data and were collected from a Taiwanese hospital in 2021. The target feature, the outcome, had binary classes (infected or uninfected) and should be influenced by 29 predictors consisting of categorical and numerical features. Table I presents the categorical features and their associated label distributions. Table II presents the numerical features and their corresponding basic statistics, i.e., the minimum, maximum, and variance for each feature.

Table I presents all features followed by their class distribution. The target feature (outcome) has two labels: not infected (class_0) or infected (class_1), with a label distribution of [51280, 879], indicating that the target feature is highly class-imbalanced. The not-infected class dominated the infected class, accounting for 98% of the dataset. This means that simply guessing an unknown instance label, without using a classifier, will achieve 98% accuracy. There are 17 categorical predictors in the dataset. where their distribution is 11 features with two classes, two features with three classes, and the remaining features with 4, 6, 14, and 16 classes. Predictors with two classes dominate the others, and their class frequencies are heterogeneous. Nevertheless, the predictor features have a more balanced class distribution than the target feature.

TABLE I. The Categorical Features and Their Label Distribution

Feature Name	Distribution Label	Labels number	Feature Name	Distribution Label	Labels number
Outcome	[51280, 879]	2	Commercial_ALBC	[47355, 4804]	2
Psychoses	[51999, 160]	2	Drain	[38550, 13609]	2
Alcohol Abuse	[51541, 618]	2	Paralysis	[52025, 134]	2
Coagulopathy	[51802, 357]	2	SEX	[34786, 17279]	2
Solid Tumor without Metastasis	[50089, 2070]	2	Non_commercial_ALBC	[30418, 21741]	2
Peptic Ulcer Disease, excluding bleeding	[49244, 2915]	2	Anemia	[51510, 629, 20]	3
Rheumatoid Arthritis/collagen	[49890, 2269]	2	Psychiatric disorder	[51114, 986, 59]	3
Cancer history	[50001, 1964, 193, 1]				4
Diagnosis	[40786, 7023, 2038, 1310, 906, 96]				6
elx_index	[29700, 9366, 6165, 3474, 1768, 898, 464, 188, 74, 32, 18, 9, 2, 1]				14
cci_index	[35002, 8003, 4343, 2301, 1172, 656, 289, 131, 117, 84, 29, 13, 12, 4, 2, 1]				16

TABLE II. The Numerical Features and their Simple Statistics

Feature name	Min.	Max.	Var.
CBC_Platelet	16.1	992	1509.13
AGE	12	99	148.93
LOS	1	72	8.35
OP_time_minute	2	1539	1030.98
OP_time_hour	0.03	25.65	0.29
BUN	1.8	140.67	25.76
GOT	4	15643	4824.68
GPT	2	506	171.28
ALB	2	7.01	0.17
Na	22.1	169.32	3.07
K	2.3	8.13	0.17
UA	1.1	18.74	0.8

The predictors with numerical scales and their simple summaries are displayed in Table II. There seems to be a clear issue with a commensurate measure. Among the features, there are not only significant gaps in the range statistic but also very heterogeneous variances. In addition, the incomparable unit scales are resolved by converting them into standardized normal scores. Finally, all numerical predictors have an N (0,1) distribution and the range of [-3, 3].

The study builds the best RF and SVM classifiers, with the model development stages covering 4 main processes sequentially: addressing the class imbalance with oversampling, using cross-validation data in the hyperparameters tuned, training the model candidates employing the obtained hyperparameters, and assessing the model's performance on six performance metrics for both the training and testing data. Before splitting the dataset into training and testing data. The preprocessing step converts the original scale to Z-scores to ensure a commensurate measure is obtained. Fig. 1 displays the steps for building the SVM classifier model; the steps for building the RF classifier model are similar, except for performing PCA and using both standardized and categorical data as inputs.

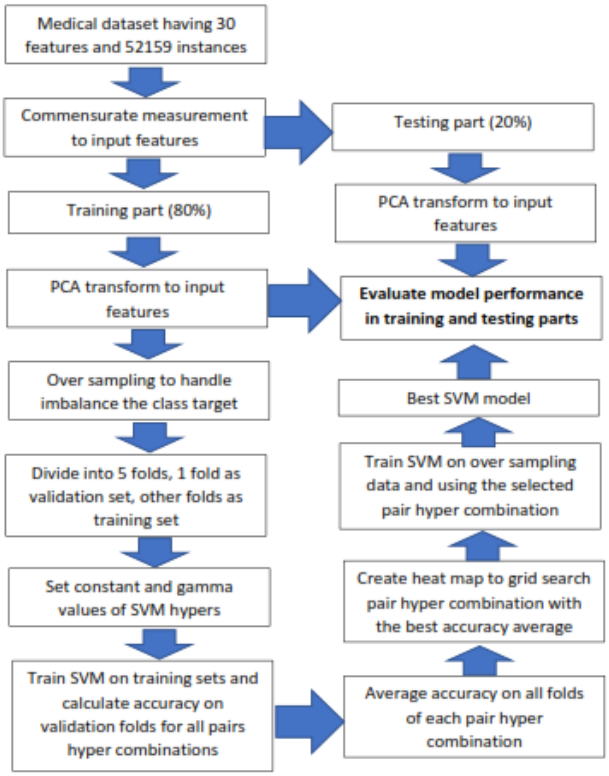

Fig. 1. The steps of the building SVM classifier.

A. The K-Folds Cross-validation to Create Pairs of Training and Validation Sets

Splitting a dataset into training and test data is a fundamental step in machine learning modeling, as the test data is needed to assess how well the model performs. Usually, the dataset is randomly split into 80% for training and 20% for testing. Most machine-learning models involve hyperparameters that should be carefully tuned [59]. The k-fold cross-validation method is a popular approach that requires substantial computational effort to obtain the optimal hyperparameters of machine learning models [60]. Fig. 2 describes splitting the dataset into training and test sets, dividing the training set into 5 folds, and finally creating pairs of training and validation sets. In detail, the stages in producing the training and validation sets are as follows.

- The validation set was the first fold, and all succeeding folds were the training set.
- The validation set was the second fold, and the other folds were the training set.
- The validation set was the third fold, and the other folds were the training set.
- Finally, the validation set was the k-fold, and all preceding folds were the training set.

After the domain values of the hyperparameters that cover the optional ones are determined, the model's candidate with a hyperparameter pair is trained on the training set, and performance is evaluated on the validation set using the accuracy metric. The trained model with the highest average accuracy was selected via grid search [61].

Training part					Testing part
fold_1	fold_2	fold_3	. . .	fold_k	
val_1	Training set_1				
fold_1	**val_2**	fold_3	. . .	fold_k	
fold_1	fold_2	**val_3**	. . .	fold_k	
. . .	. . .	. . .	. . .	. . .	
Training set_k				**val_k**	

Fig. 2. Formatting the dataset to produce pairs of the training and validation sets.

B. Random Forest Classifier Model

The Random Forests (RF) method was proposed by Leo Breiman in 2001. It is an ensemble method designed explicitly for decision tree classifiers. There are two sources of randomness: bagging and a random input vector. Bagging means that each tree was grown using a bootstrapped sample of the data. A random input vector was used at each decision node, where the best split was chosen from a random sample of m features or a subset of attributes instead of all attributes [62].

Fig. 3 depicts the steps for building the RF model. They include selecting the M-dimensional predictors from the dataset, setting m < M, randomly selecting the m predictors, and calculating the information gains (IG). Furthermore, determine the splitting feature with the highest IG value. The RF algorithm was presented as follows[63]:

for $b = 1$ to B,

- Take the bootstrapped dataset (Db) with size N from the D dataset.
- Grow the decision tree (Tb) based on Db, and repeat the steps for each decision node until the leaf node.
  - Select m of M features randomly with (m < M)
  - Compute the IG of the m features
  - Select the best splitting feature and the IG
  - Divide the node into two child nodes
- The RF output was yielded by the majority vote.

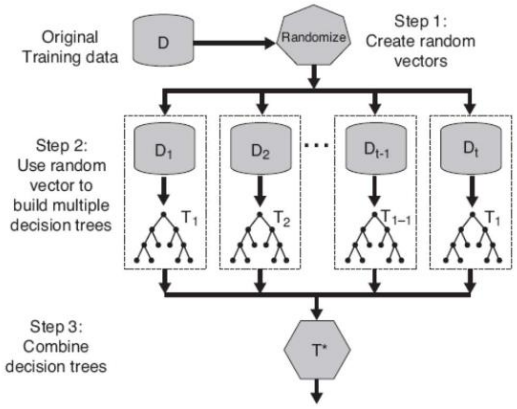

Fig. 3. The concept of developing the RF model.

The RF's hyperparameters, including the tree depth and the number of instances per tree leaf node, should be determined fairly. The grid search method selects the hyperparameter pair with the highest average accuracy obtained from evaluating the trained RF on 5-fold validation data.

C. Modeling Support Vector Classifier Model

The conventional classifier models, namely logistic regression (LR) and Fisher linear discriminant (FDL), do not have a unique decision boundary due to their lack of well-definedness. An example of a well-defined classifier model is a support vector machine (SVM). The decision boundary of the SVM is produced by maximizing the distance between the boundary and the nearest instances. The SVM model is known as the maximum-margin classifier.

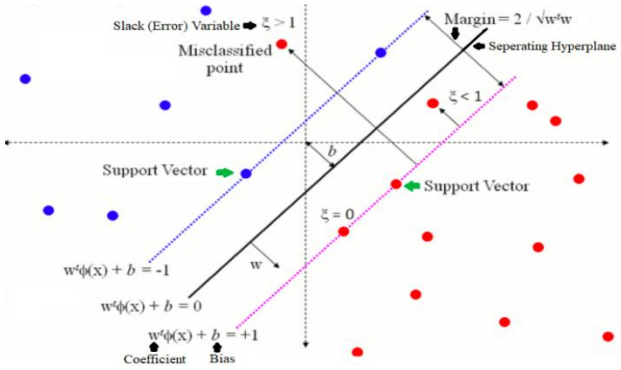

Fig. 4. The SVM model including the decision boundary and support vectors.

The margin of an SVM classifier is the smallest distance between the decision boundary and any training instance, called a support vector[64]. Fig. 4 describes the relationship among the components of an SVM model, including the decision boundary (separating hyperplane), decision margin, and support vectors. A support vector is a training point that lies on the decision-margin line. The SVM concept involves finding support vectors that minimize the distance between the two decision-margin lines. In other words, the SVM task is the same as maximizing the vectors of x nearest to the decision boundary[65].

Fig. 4 depicts three kinds of points in the SVM modeling: points well-classified are $(y_i(w^T \cdot x + w_0) \geq 1)$, points lying within the margin are $(0 < y_i(w^T \cdot x + w_0) < 1)$, and misclassified points are $(y_i(w^T \cdot x + w_0) < 0)$. Eq. (1) describes the three types of points in the SVM model.

$$y_i(w^T \cdot x + w_0) \geq 1 - \xi_i \quad (1)$$

where $\xi_i \geq 0$, is the slack variable. The problem formulation in SVM modeling maximizes the margin and minimizes the slack variable. Mathematically, the formulation is presented by Eq. (2) as follows:

$$\text{minimize } J(\boldsymbol{w}, w_0, \xi) = \frac{1}{2}||w||^2 + C\sum \xi_i \quad (2)$$

Subject to

$$y_i(w^T \cdot x_i + w_0) \geq 1 - \xi_i \; \alpha\nu\delta \; \xi_i \geq 0$$

Consider Eq. (2) and rewrite it by adding the Lagrange multiplier to produce Eq. (3) as follows:

$$\mathcal{L}(w, b, \alpha, \xi) = \frac{1}{2}w^T w + C\sum_i \xi_i - \sum_i \alpha_i(y_i(w^T x_i + b) - 1 + \xi_i) - \sum_i \beta_i \xi_i \quad (3)$$

Where are $\alpha_i \geq 0$, and $\beta_i \geq 0, \forall i = 1,2,3,\ldots,N$. Minimize $\mathcal{L}(w, b, \alpha, \xi)$ through setting $\nabla \mathcal{L}(w, b, \alpha, \xi) = 0$. Furthermore, by rewriting the dual problem formulation given in Eq. (4) as follows [66]:

$$\mathcal{L}(\boldsymbol{w}, b, \alpha) = \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j \quad (4)$$

Subject to:

$$\sum_i \alpha_i y_i = 0 \; and \; 0 \leq \alpha_i \leq C \quad \forall i = 1,2,3,\ldots,N$$

The inverse of the regularization (C constant) is a crucial hyperparameter for the SVM. The C value describes the behavior as follows: as C increases, the model overfits. In contrast, as C decreases, the model underfits.

Consider the last term of Eq. (4) that can be replaced with a kernel function that serves as a nonlinear mapping from a low-dimensional to a high-dimensional feature space, thereby resulting in a simple form of the optimization problem. The method is known as the kernel trick. Furthermore, the formula for a kernel function is given as follows:

$k(x_1, x_2) = \langle \varphi(x_1), \varphi(x_2) \rangle$ where k and φ respectively are the kernel and the feature map. The decision function of the SVM model in the space of feature-map is stated as follows:

$$y(x) = w^T \emptyset(x) + b \;\; \omega\eta\varepsilon\rho\varepsilon \;\; w = \sum_{n=1}^{N} a_n t_n \emptyset(x_n)$$

An instance x with unknown class is classified by involving a kernel with the formula as follows:

$y(x) = \sum_{n=1}^{N} a_n t_n k(x, x_n) + b$ where t_n is the target class, and the bias b is calculated by the formula as follows:

$$b = \frac{1}{N_{\mathcal{M}}} \sum_{n \in \mathcal{M}} (t_n - \sum_{m \in S} a_m t_m k(x_n, x_m))$$

Both formulas of w and b are obtained by satisfying the constraints of the Karush Kuhn Tucker (KKT).

Whereas the Radial Basis Function (RBF) kernel is presented as follows:

$$k(x, x^l) = exp\left(-\frac{||x - x^l||^2}{2\sigma^2}\right) \quad (5)$$

The denominator in Eq. (5) is referred to as the gamma value only for the RBF kernel. Gamma is a model hyperparameter for the RBF kernel. The gamma value has a behavior: when gamma increases, the SVM overfits, and when gamma decreases, the SVM underfits.

D. Classifier Model Performance Metrics

The metrics for evaluating the classifier's performance are calculated using a confusion matrix. The confusion matrix elements are presented in Eq. (6) as follows:

$$confusion\ matrix = \begin{bmatrix} TN & FN \\ FP & TP \end{bmatrix} \quad (6)$$

Where are TN, FN, FP, and TP representing true negatives, false negatives, false positives, and true positives, respectively

[67]. The study employs the six metrics to evaluate the classifier model's performance, presented in Eq. (7) as follows:

$$\begin{aligned} Accuracy &= \frac{(TP+TN)}{(TP+TN+FP+FN)} \\ Precision &= \frac{TP}{(TP+FP)} \\ Recall &= \frac{TP}{(TP+FN)} \\ F1\ score &= \frac{2\times Recall\times Precision}{(Recall+Precision)} \\ MCC &= \frac{TP\times TN-FP\times FN}{\sqrt{(TP+FP)\times(TP+FN)\times(TN+FP)\times(TN+FN)}} \\ AUC &= \int_0^1 ROC(x)dx \end{aligned} \tag{7}$$

Performance metrics, including accuracy, precision, recall, and F1 score, are widely used to evaluate classifier models in binary classification with balanced classes. Two other metrics, namely Matthews Correlation Coefficient (MCC) and Area Under the ROC Curve (AUC), were employed to evaluate the classifier's performance more comprehensively [68].

The MCC metric is widely used to evaluate the performance of classifier models in biomedical research. Both MCC and AUC are elective metrics for reaching a consensus on the best practices for the development and validation of predictive models for personalized medicine[69]. The AUC value ranges from 0.0 to 1.0, describing the capability of the binary classifier to separate instances of the positive class (class_1) from instances of the negative class (class_0).

IV. Performance Metrics of Classification Model

A commensurate measure of the dataset's numerical features is the first issue encountered during preprocessing. The numerical predictors with the heterogeneous unit scale are converted to standardized Z-scores with a mean of zero and a variance of one. The advantage of standardized data is that it is straightforward to identify the extreme values. The value of a feature of an instance is considered an outlier when the absolute Z-score exceeds 3 in the instance where an outlier value of any feature is dropped from the dataset. Although the action entails a risk of dropping an instance from the minority class, further observation to identify the factors underlying the occurrence of extreme features was not straightforward. The former version of the dataset used an incomparable unit scale, and there were no outliers in either the training or testing data.

A. Do Over Sampling and Format the Training and Validation Subsets

The dataset with all predictor features was randomly split into training (80%) and testing (20%) sets. The random forest (RF) and support vector machine (SVM) models were developed using a training set and five pairs of training and validation subsets for hyperparameter tuning. The test set serves as out-of-sample data to evaluate the performance of the best model, i.e., the one with the optimal hyperparameters. Because the dataset is imbalanced, the issue must be addressed by oversampling. For simplicity, the research oversampled the training data only once before dividing the training set randomly into five cross-validation folds. The target feature has a class-label distribution of [class_0, class_1] = [36155, 611], indicating an imbalance. Oversampling was performed on the lower-frequency class (class_1) until its frequency matched that of class_0. Fig. 5 shows that both classes had the same frequency.

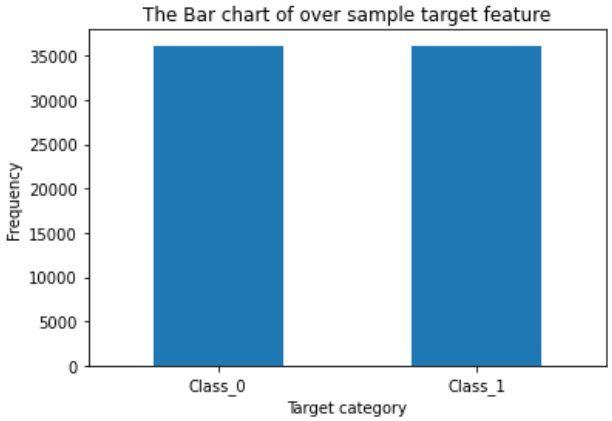

Fig. 5. The class distribution after oversampling.

By applying oversampling, the class_1 frequency increased from 611 to 36155, bringing the total number of instances in the oversampled training set to 72310. Oversampling introduces a trade-off in model training, and hyperparameter tuning requires longer processing time. Furthermore, the oversampled training set was randomly split into five folds, producing five pairs of the training and validation subsets. The number of instances in each validation subset was 14462. The training subset was used to train model candidates with a given associated hyperparameter pair, whereas the validation subset was used to calculate the model's accuracy. Each pair of hyperparameters was used to train a model candidate on five training subsets, producing five accuracy values on five validation subsets. The average accuracy indicates the model's performance on the pair of hyperparameters.

B. Tuning Hyperparameters of Random Forest Classifier

The research builds an RF classifier model that considers not only tree depth but also the number of instances per leaf node as hyperparameters. Both are expected to affect the RF's performance significantly. Their ranges are first determined by trial and error to cover the optimal value. Furthermore, the first hyperparameter was set to [300, 500, 700, 1000, 1200], and the second one to [7, 8, 9, 10, 12, 14]. The RF model candidate was trained by evaluating each hyperparameter pair on four folds of training data, then calculating the accuracy metric on the remaining fold as validation data. There are not only five pairs of training and validation data but also five pairs of hyperparameters; as a result, each pair of training and validation data is used 5 times to train an RF model. The accuracy metric of each trained RF model is calculated on its corresponding validation data. The model's accuracy for a pair of hyperparameters was calculated and recorded. The average accuracy represents the RF model's performance across the pair of hyperparameters. Fig. 6 is a heatmap of the average accuracy across 30 cells.

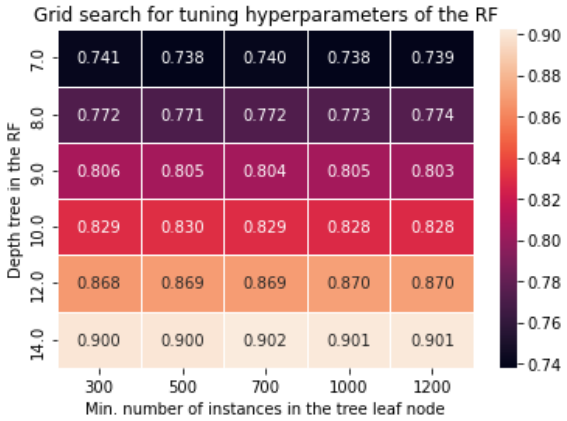

Fig. 6. The average accuracy presented in a heatmap.

Fig. 6 shows the RF's performance: average accuracy increases with tree depth. For example, at a tree depth of 10, the average accuracy increases by around 6% when the tree depth increases by around two points. On the other hand, a different pattern emerges when the tree depth exceeds 10, with the RF's performance increasing less than at the preceding depth level. There is only a marginal increase in the RF's performance of approximately 4% as the tree depth increases from 10 to 12 points. The phenomenon indicates that the model overfits when the RF tree depth exceeds 10. The number of instances per leaf node does not significantly affect the shift in mean accuracy. RF performance at a tree depth of 10 is in the range [0.828, 0.830]. The highest mean accuracy occurs with the cell hyperparameter pair (500, 10). The obtained hyperparameter pair is used to train the RF model on the oversampled training data. The confusion matrix of the optimal RF model is presented for the training and testing data, as presented in Table III:

TABLE III. THE OPTIMAL RF MODEL CONFUSION MATRIX

Actual value	Prediction training		Prediction testing	
	Class_0	Class_1	Class_0	Class_1
Class_0	26329	9826	6512	2511
Class_1	36	575	84	85

Table III displays the confusion matrices for the optimal RF model on both the training and test data with the hyperparameter pair (500, 10). The best RF incorrectly classified 9826 of 36155 instances in the negative class (class_0) on the training data and incorrectly classified 2511 of 9023 instances in class_0 on the testing data. On the other side, the best RF incorrectly classified 36 of 611 instances in the positive class (class_1) in the training data and incorrectly classified 84 of 169 instances in class_1 in the testing data.

C. Tuning Hyperparameter of the Support Vector Classifier Model

In developing the support vector machine (SVM) classifier, the constant and gamma hyperparameters are tuned using cross-validation. The optimal hyperparameter range was determined carefully. First, the SVM model candidate was trained with a given constant value and various gamma values to identify the gamma value with the highest accuracy, and select it as the selected gamma value. Furthermore, the SVM model was retrained with the selected gamma and various constant values. Finally, the sets of gamma and constant values are obtained as [0.0001, 0.0005, 0.001, 0.005, 0.01] and [100, 500, 1000, 3000, 5000, 10000]. For each hyperparameter combination (30 pairs), an SVM model candidate was trained on the training set (the union of the four folds), and the accuracy on the associated validation set (the remaining fold) was calculated. The average accuracy across all folds and gamma-constant pairs is presented as a heat map in Fig. 7.

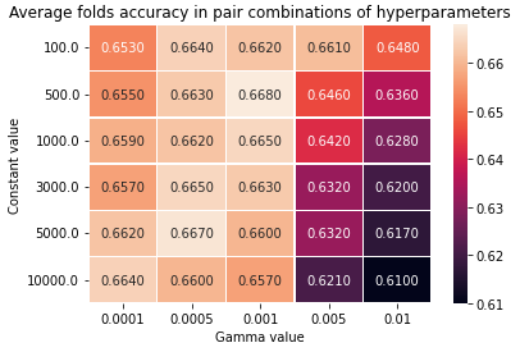

Fig. 7. The heat map of the SVM average accuracy.

Fig. 7 shows the average accuracy of the SVM model candidates in gamma coordinates and constant hyperparameter coordinates. They appear to be randomly distributed and compete tightly with one another. The highest accuracy was achieved with the pair [0.001, 500] for the gamma and constant values, respectively. The hyperparameter pair for the SVM model achieved an average accuracy of 0.668. Furthermore, the hyperparameter pair (0.001, 500) was used to train an SVM candidate model on the oversampled training data, and the confusion matrices for both the training and test sets are presented in Table IV.

TABLE IV. THE SVM MODEL CONFUSION

Actual value	Prediction training		Prediction testing	
	Class_0	Class_1	Class_0	Class_1
Class_0	22813	13342	5461	3562
Class_1	192	419	65	104

Table IV displays the confusion matrix of the best SVM, in which the classifier model incorrectly classified 13342 of 36155 instances of the negative class (class_0) in the training data and incorrectly classified 3562 of 9023 instances of class_0 in the testing data. In addition, the best SVM model incorrectly classified 192 of 611 instances of the positive class (class_1) in the training data and incorrectly classified 65 of 169 instances of class_1 in the testing data. The best SVM model performs better in classifying the class_1 instance in the testing data than the best RF model does. Nevertheless, the best RF outperformed the best SVM on the training data. The best SVM model's average accuracy, displayed in the heat map, showed a random distribution, suggesting it has no evidence of overfitting. Nevertheless, it should be noted that training the SVM model on the 5-fold validation data took longer than training the RF model.

D. The Models' Performance Comparison and the Important Features

The comparison of the best RF and SVM models was conducted in the six performance metrics, i.e., accuracy, precision, recall, F1 score, MCC, and AUC, on both the training and testing data. In addition, the acquired optimal hyperparameters were employed to train both models, using the original training data (rather than the oversampled training data) as the benchmark. Table V displays the benchmark and the best model's performance on both the training and testing data. Table V consists of two primary columns: the first presents RF performance, and the second presents SVM performance. Each primary column contains four columns: the first and second columns present the benchmark model performance, and the two remaining columns display the best model's performance.

TABLE V. THE BEST RF AND SVM PERFORMANCE COMPARISON METRICS

Performance measures	Random Forest (RF)				Support Vector Machine (SVM)			
	Benchmark		*Best model*		Benchmark		Best model	
	Training	*Testing*	*Training*	*Testing*	Training	Testing	Training	Testing
Accuracy	0.984	0.982	0.732	0.718	0.983	0.982	0.632	0.605
Precision	0.980	0.960	0.980	0.970	0.970	0.970	0.970	0.980
Recall	0.980	0.980	0.730	0.720	0.980	0.980	0.630	0.610
F1-score	0.980	0.970	0.830	0.820	0.970	0.980	0.760	0.740
MCC	0.150	0.000	0.190	0.067	0.000	0.000	0.085	0.056
AUC	0.512	0.500	0.835	0.612	0.500	0.500	0.658	0.611

On the training data, the RF benchmark model achieved 98% across four metrics: accuracy, precision, recall, and F1 score. Furthermore, it achieved the MCC and AUC metrics of 15% and 51.2%, respectively. The best RF model performs worse than the RF benchmark across four metrics. However, it achieves the same precision as the RF benchmark and improves by 4% and 32.3% in MCC and AUC metrics, respectively.

The RF model's performance is assessed on the test data; the RF benchmark outperforms the best RF in three metrics, namely accuracy, recall, and F1 score, by 26.4%, 26%, and 15%, respectively. However, the best RF outperforms the RF benchmark in precision, MCC, and AUC, with improvements of 1%, 6.7%, and 11.2%, respectively. The RF benchmark achieves an MCC of 0% and an AUC of 50%, which is equivalent to a coin toss; thus, the RF benchmark model has no performance. Mathematically, an MCC of 0% is obtained when the numerator in the MCC formula equals 0, i.e., when both the true positives (TP) and false positives (FP) are 0. Although the best RF model performed below the RF benchmark on three standard metrics, it significantly outperformed MCC and AUC, which were 6.7% and 61.2%, respectively. Oversampling increased the MCC and AUC by 6.7% and 11.2%, respectively.

When evaluated on the training set, the benchmark SVM model achieved the same precision (97%) as the best SVM model, and the two models showed a performance difference of 35.1%, 35%, and 21% for accuracy, recall, and F1 score, respectively. Furthermore, the best SVM outperformed the benchmark by 8.5% and 15.8% in MCC and AUC, respectively. When evaluating the SVM models on the test set, the benchmark SVM model outperformed the best SVM by 37.7%, 37%, and 24% in accuracy, recall, and F1 score, respectively. In contrast, the best SVM outperformed the benchmark model in precision, MCC, and AUC, with improvements of 1%, 5.6%, and 11.1%, respectively.

The SVM benchmark's performance on the test set is nearly identical to that of the RF benchmark. Both benchmark models perform poorly, as demonstrated by an MCC of 0 and an AUC of 0.5. This explains why they cannot correctly predict labels for unknown positive-class instances on the test set. Extensive attempts have been made to add hyperparameters to produce a well-performing classifier, but to no avail. The best-performing RF model is similar to the best SVM model on both the training and test sets. On the training set, the best RF performs slightly better than the best SVM, with differences of 10%, 1%, 10%, 7%, 10.5%, and 17.7% for accuracy, precision, recall, F1 score, MCC, and AUC, respectively. On the test dataset, the best RF achieves slightly higher accuracy, recall, F1 score, MCC, and AUC than the best SVM, with differences of 11.3%, 11%, 8%, 1.1%, and 0.1%, respectively. The best SVM outperforms the best RF on all five metrics, except precision, by 1%.

The results confirmed that oversampling to address class imbalance during machine learning model training does not always improve model performance. This is similar to the work done by [20], [29], and [30]. The MCC or AUC metrics clearly explain performance improvements. The MCC values increased by 6.7% and 5.6% for the best RF and SVM models, respectively, whereas the AUC values for both models were approximately 11%. Although the best RF model performs slightly better than the best SVM model, hyperparameter optimization for the RF model requires careful attention. Otherwise, there is a potential for overfitting problems[55]. Additionally, the benchmark model identified other findings warranting significant attention. The benchmark model achieves approximately 98% performance across both the training and test sets in four metrics: accuracy, precision, recall, and F1 score. However, both benchmark models performed poorly in the test set, as indicated by an MCC value of 0 and an AUC value of 0.5, where their performance was only equal to the probability of a

balanced coin toss[25]. The event implicitly suggests that using four widely used metrics to evaluate model performance on imbalanced binary classes is misguided.

In machine learning modeling, assessing model performance on test data is crucial because it confirms the model's ability to classify out-of-sample data. This study presents model performance assessments on both training and test data to explicitly demonstrate that assessing model performance based on training data can lead to biased conclusions. The model performed worse on the test data, but the pattern of performance metrics was similar when it did not overfit [52]. Optimally fitting an RF model is constrained by the difficulty in determining the hyperparameter domain that covers optimal values. However, training the RF model on cross-validation data requires less time. On the other hand, the hyperparameter space of an SVM model is straightforward to determine, but training it on cross-validation data is more time-consuming.

Another advantage of building an RF model is that it is a key feature that warrants exploration. The degree of importance of each feature can be used to determine the dominant features or for dimension reduction[10]. Developing a model that involves a few predictor features is preferable because the system mechanism underlying the model can be explained and interpreted well[42]. Fig. 8 presents the degree of essential features produced by the best RF model.

The bar chart in Fig. 8 presents the ranking of important features in increasing order. The numerical features appear to dominate the importance measure, serving as the splitting features in developing the RF model. The first five important features are CBC_Platelet, AGE, K, GOT, and LOS, with importance levels of approximately 10%, 9%, 8%, 8%, and 8%, respectively. This indicates that the five features account for approximately 43% of the RF model's development. The next group comprises four features, each with a degree of importance of approximately 6%, followed by Na and GPT features with approximately 5%. There were five features with a degree of importance of roughly 3%. The importance of the first 16 critical features is approximately 92%. From a dimension-reduction perspective, 16 features account for approximately 92% of the dataset's total variance. The variance explained by feature importance represents the contribution of the original features to the RF model. On the other hand, the variance explained by the PCA transformation is the sum of the variances of the principal components, which are mutually orthogonal and account for the dataset's variance. Note that all 12 numerical features are considered important.

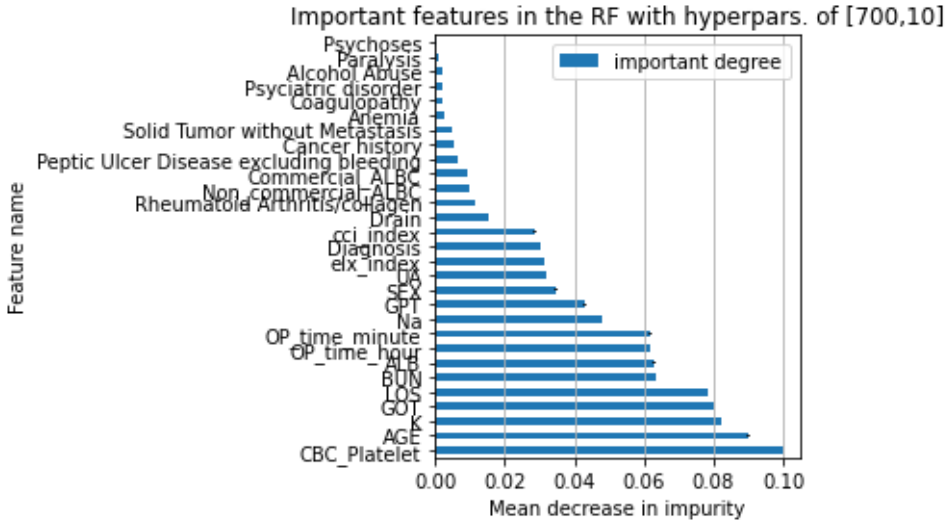

Fig. 8. The best RF model's important features.

V. Conclusion

Oversampling significantly increases the size of the training data, doubling it. This consumes substantial time, particularly when training the SVM model on cross-validation data for hyperparameter tuning. The optimal hyperparameters acquired for the RF model are (500, 10) for the number of instances per tree leaf and the tree depth, respectively. The optimal hyperparameters acquired for the SVM model are (0.001, 500) for the gamma and the constant, respectively. Both RF and SVM benchmarks achieve satisfactory performance on 4 metrics: accuracy, precision, recall, and F1 score, all at 98%. However, they perform worse on 2 metrics: MCC and AUC, with 0% and 50%, respectively. In contrast, the best RF and SVM performed similarly on 6 metrics. Oversampling reduces both the best models across four metrics: accuracy, precision, recall, and F1 score, but it increases on two metrics, namely MCC and AUC. The best RF and SVM outperformed the benchmark on the MCC and AUC metrics. The performance gaps between the best RF and the RF benchmark are (6.7% and 11.2%) for the MCC and AUC metrics, respectively. Whereas the performance gaps between the best SVM and the SVM benchmark are (5.6% and 11.1%) for the MCC and AUC metrics, respectively. In addition, the best RF produces important features that can serve as predictors for other models. In the future, the challenging research topics should focus on building an RF model using the original feature scale, employing MCC or AUC as selection criteria to obtain optimal hyperparameters, and conducting oversampling on the cross-

validation data. They are expected to significantly improve model performance.

REFERENCES

[1] A. W. Widodo, S. Handoyo, G. Irianto, N. W. Hidajati, D. S. Susanti, and I. N. Purwanto, "Finding the Best Performance of Bayesian and Naïve Bayes Models in Fraudulent Firms Classification through Varying Threshold," International Journal of Emerging Technology and Advanced Engineering, vol. 12, no. 12, pp. 94–106, Dec. 2022, doi: 10.46338/ijetae1222_11.

[2] S. Syahrial, R. Ilham, Z. F. Asikin, and St. S. I. Nurdin, "Stunting Classification in Children's Measurement Data Using Machine Learning Models," Journal La Multiapp, vol. 3, no. 2, 2022, doi: 10.37899/journallamultiapp.v3i2.614.

[3] W. Cnota et al., "The Killer Placenta" — a threat to the lives of young women giving Birth by cesarean section," Ginekol Pol, vol. 93, no. 4, 2022, doi: 10.5603/GP.a2021.0235.

[4] T. Liu, W. Fan, and C. Wu, "A hybrid machine learning approach to cerebral stroke prediction based on the imbalanced medical dataset," Artif Intell Med, vol. 101, 2019, doi 10.1016/j.artmed.2019.101723.

[5] I. D. Mienye and Y. Sun, "Performance analysis of cost-sensitive learning methods with application to imbalanced medical data," Inform Med Unlocked, vol. 25, 2021, doi: 10.1016/j.imu.2021.100690.

[6] A. V. Barmak, Y. V. Krak, E. A. Manziuk, and V. S. Kasianiuk, "Information technologyof separating hyperplanes synthesisfor linear classifiers," Journal of Automation and Information Sciences, vol. 51, no. 5, 2019, doi: 10.1615/JAutomatInfScien.v51.i5.50.

[7] Priyanka and D. Kumar, "Decision tree classifier: A detailed survey," International Journal of Information and Decision Sciences, vol. 12, no. 3, 2020, doi: 10.1504/ijids.2020.108141.

[8] T. Lan, H. Hu, C. Jiang, G. Yang, and Z. Zhao, "A comparative study of decision tree, random forest, and convolutional neural network for spread-F identification," Advances in Space Research, vol. 65, no. 8, 2020, doi: 10.1016/j.asr.2020.01.036.

[9] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," Expert Systems with Applications, vol. 134. 2019. doi: 10.1016/j.eswa.2019.05.028.

[10] M. Z. Alam, M. S. Rahman, and M. S. Rahman, "A Random Forest based predictor for medical data classification using feature ranking," Inform Med Unlocked, vol. 15, 2019, doi: 10.1016/j.imu.2019.100180.

[11] S. El-Sappagh, F. Ali, T. Abuhmed, J. Singh, and J. M. Alonso, "Automatic detection of Alzheimer's disease progression: An efficient information fusion approach with heterogeneous ensemble classifiers," Neurocomputing, vol. 512, 2022, doi: 10.1016/j.neucom.2022.09.009.

[12] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," Neurocomputing, vol. 408, 2020, doi: 10.1016/j.neucom.2019.10.118.

[13] W. Pannakkong, K. Thiwa-Anont, K. Singthong, P. Parthanadee, and J. Buddhakulsomsiri, "Hyperparameter Tuning of Machine Learning Algorithms Using Response Surface Methodology: A Case Study of ANN, SVM, and DBN," Math Probl Eng, vol. 2022, 2022, doi: 10.1155/2022/8513719.

[14] X. Ding, J. Liu, F. Yang, and J. Cao, "Random radial basis function kernel-based support vector machine," J Franklin Inst, vol. 358, no. 18, 2021, doi: 10.1016/j.jfranklin.2021.10.005.

[15] A. Kumar, N. Sinha, and A. Bhardwaj, "A novel fitness function in genetic programming for medical data classification," J Biomed Inform, vol. 112, 2020, doi: 10.1016/j.jbi.2020.103623.

[16] J. Bektaş, "EKSL: An effective novel dynamic ensemble model for unbalanced datasets based on LR and SVM hyperplane-distances," Inf Sci (N Y), vol. 597, 2022, doi: 10.1016/j.ins.2022.03.042.

[17] M. A. Ganaie and M. Tanveer, "KNN weighted reduced universum twin SVM for class imbalance learning," Knowl Based Syst, vol. 245, 2022, doi: 10.1016/j.knosys.2022.108578.

[18] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," BMC Genomics, vol. 21, no. 1, 2020, doi: 10.1186/s12864-019-6413-7.

[19] G. Canbek, T. Taskaya Temizel, and S. Sagiroglu, "BenchMetrics: a systematic benchmarking method for binary classification performance metrics," Neural Comput Appl, vol. 33, no. 21, 2021, doi: 10.1007/s00521-021-06103-6.

[20] G. Kovács, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," Applied Soft Computing Journal, vol. 83, 2019, doi: 10.1016/j.asoc.2019.105662.

[21] Z. Xu, D. Shen, T. Nie, Y. Kou, N. Yin, and X. Han, "A cluster-based oversampling algorithm combining SMOTE and k-means for imbalanced medical data," Inf Sci (N Y), vol. 572, 2021, doi: 10.1016/j.ins.2021.02.056.

[22] D. Passos and P. Mishra, "A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks," Chemometrics and Intelligent Laboratory Systems, vol. 223. 2022. doi: 10.1016/j.chemolab.2022.104520.

[23] S. Nematzadeh, F. Kiani, M. Torkamanian-Afshar, and N. Aydin, "Tuning hyperparameters of machine learning algorithms and deep neural networks using metaheuristics: A bioinformatics study on biomedical and biological cases," Comput Biol Chem, vol. 97, 2022, doi: 10.1016/j.compbiolchem.2021.107619.

[24] R. Yacouby and D. Axman, "Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models," 2020. doi: 10.18653/v1/2020.eval4nlp-1.9.

[25] D. Chicco and G. Jurman, "The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification," BioData Min, vol. 16, no. 1, 2023, doi: 10.1186/s13040-023-00322-4.

[26] S. Handoyo, Y. P. Chen, G. Irianto, and A. Widodo, "The varying threshold values of logistic regression and linear discriminant for classifying fraudulent firm," Mathematics and Statistics, vol. 9, no. 2, 2021, doi: 10.13189/ms.2021.090207.

[27] H. Zhou, K. M. Yu, Y. C. Chen, and H. P. Hsu, "A Hybrid Feature Selection Method RFSTL for Manufacturing Quality Prediction Based on a High Dimensional Imbalanced Dataset," IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3059298.

[28] A. S. Sumant and D. V. Patil, "Parallel Ensemble Feature Subset Selection based Deep Learning Approach for Imbalanced High Dimensional Cancer Datasets," Indian Journal of Computer Science and Engineering, vol. 13, no. 6, 2022, doi: 10.21817/indjcse/2022/v13i6/221306034.

[29] T. Wongvorachan, S. He, and O. Bulut, "A Comparison of Undersampling, Oversampling, and SMOTE Methods for Dealing with Imbalanced Classification in Educational Data Mining," Information (Switzerland), vol. 14, no. 1, 2023, doi: 10.3390/info14010054.

[30] A. S. Desuky and S. Hussain, "An Improved Hybrid Approach for Handling Class Imbalance Problem," Arab J Sci Eng, vol. 46, no. 4, 2021, doi: 10.1007/s13369-021-05347-7.

[31] W. Lu, Z. Li, and J. Chu, "Adaptive Ensemble Undersampling-Boost: A novel learning framework for imbalanced data," Journal of Systems and Software, vol. 132, 2017, doi: 10.1016/j.jss.2017.07.006.

[32] P. Roy and C. Chowdhury, "A Survey of Machine Learning Techniques for Indoor Localization and Navigation Systems," Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 101, no. 3, 2021, doi: 10.1007/s10846-021-01327-z.

[33] Marji, S. Handoyo, I. N. Purwanto, and M. Y. Anizar, "The effect of attribute diversity in the covariance matrix on the magnitude of the radius parameter in fuzzy subtractive clustering," J Theor Appl Inf Technol, vol. 96, no. 12, 2018.

[34] I. N. Purwanto, A. Widodo, and S. Handoyo, "System for selection starting lineup of a football players by using analytical hierarchy process (AHP)," J Theor Appl Inf Technol, vol. 96, no. 1, 2018.

[35] S. Handoyo, Marji, I. N. Purwanto, and F. Jie, "The fuzzy inference system with rule bases generated by using the fuzzy C-means to predict regional minimum wage in Indonesia," International Journal of Operations and Quantitative Management, vol. 24, no. 4, 2018.

[36] L. Liu, N. Jia, L. Lin, and Z. He, "A Cohesion-Based Heuristic Feature Selection for Short-Term Traffic Forecasting," IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2018.2889814.

[37] W. H. Nugroho, S. Handoyo, and Y. J. Akri, "An influence of measurement scale of predictor variable on logistic regression modeling and learning vector quntization modeling for object classification," International Journal of Electrical and Computer Engineering, vol. 8, no. 1, 2018, doi: 10.11591/ijece.v8i1.pp333-343.

[38] Marji and S. Handoyo, "PERFORMANCE OF RIDGE LOGISTIC REGRESSION AND DECISION TREE IN THE BINARY CLASSIFICATION," J Theor Appl Inf Technol, vol. 100, no. 15, 2022.

[39] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," Comput Electron Agric, vol. 174, 2020, doi: 10.1016/j.compag.2020.105507.

[40] S. Handoyo, N. Pradianti, W. H. Nugroho, and Y. J. Akri, "A Heuristic Feature Selection in Logistic Regression Modeling with Newton Raphson and Gradient Descent Algorithm," International Journal of Advanced Computer Science and Applications, vol. 13, no. 3, 2022, doi: 10.14569/IJACSA.2022.0130317.

[41] N. Han et al., "Transferable Linear Discriminant Analysis," IEEE Trans Neural Netw Learn Syst, vol. 31, no. 12, 2020, doi: 10.1109/TNNLS.2020.2966746.

[42] Y. T. Mursityo, I. Rupiwardani, W. H. N. Putra, D. S. Susanti, T. Handayani, and S. Handoyo, "Relevant Features Independence of Heuristic Selection and Important Features of Decision Tree in the Medical Data Classification," Journal of Advances in Information Technology, vol. 15, no. 5, pp. 591–601, 2024, doi: 10.12720/jait.15.5.591-601.

[43] P. H. Huynh and V. H. Nguyen, "A Novel Ensemble of Support Vector Machines for Improving Medical Data Classification," Engineering Innovations, vol. 4, 2023, doi: 10.4028/p-h0cef4.

[44] M. M. Chen and M. C. Chen, "Modeling road accident severity with comparisons of logistic regression, decision tree, and random forest," Information (Switzerland), vol. 11, no. 5, 2020, doi: 10.3390/INFO11050270.

[45] A. Subudhi, M. Dash, and S. Sabut, "Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier," Biocybern Biomed Eng, vol. 40, no. 1, 2020, doi: 10.1016/j.bbe.2019.04.004.

[46] A. Al-Hashedi et al., "Ensemble Classifiers for Arabic Sentiment Analysis of Social Network (Twitter Data) towards COVID-19-Related Conspiracy Theories," Applied Computational Intelligence and Soft Computing, vol. 2022, 2022, doi: 10.1155/2022/6614730.

[47] M. Aslani and S. Seipel, "Efficient and decision boundary aware instance selection for support vector machines," Inf Sci (N Y), vol. 577, 2021, doi: 10.1016/j.ins.2021.07.015.

[48] A. Widodo and S. Handoyo, "The classification performance using logistic regression and support vector machine (SVM)," J Theor Appl Inf Technol, vol. 95, no. 19, 2017.

[49] T. Y. Wen and S. A. Mohd Aris, "Hybrid Approach of EEG Stress Level Classification Using K-Means Clustering and Support Vector Machine," IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3148380.

[50] G. Bologna and Y. Hayashi, "A Comparison Study on Rule Extraction from Neural Network Ensembles, Boosted Shallow Trees, and SVMs," Applied Computational Intelligence and Soft Computing, vol. 2018, 2018, doi: 10.1155/2018/4084850.

[51] S. K. Saddam Hussain, E. Sai Charan Reddy, K. Gangadhar Akshay, and T. Akanksha, "Fraud Detection in Credit Card Transactions Using SVM and Random Forest Algorithms," in Proceedings of the 5th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2021, 2021. doi: 10.1109/I-SMAC52330.2021.9640631.

[52] A. Sabat-Tomala, E. Raczko, and B. Zagajewski, "Comparison of support vector machine and random forest algorithms for invasive and expansive species classification using airborne hyperspectral data," Remote Sens (Basel), vol. 12, no. 3, 2020, doi: 10.3390/rs12030516.

[53] M. Hesami and A. M. P. Jones, "Modeling and optimizing callus growth and development in Cannabis sativa using random forest and support vector machine in combination with a genetic algorithm," Appl Microbiol Biotechnol, vol. 105, no. 12, 2021, doi: 10.1007/s00253-021-11375-y.

[54] 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, 2020, doi: 10.14569/IJACSA.2020.0110920.

[55] D. Sun, J. Xu, H. Wen, and D. Wang, "Assessment of landslide susceptibility mapping based on Bayesian hyperparameter optimization: A comparison between logistic regression and random forest," Eng Geol, vol. 281, 2021, doi: 10.1016/j.enggeo.2020.105972.

[56] M. Daviran, A. Maghsoudi, R. Ghezelbash, and B. Pradhan, "A new strategy for spatial predictive mapping of mineral prospectivity: Automated hyperparameter tuning of random forest approach," Comput Geosci, vol. 148, 2021, doi: 10.1016/j.cageo.2021.104688.

[57] I. S. Al-Mejibli, J. K. Alwan, and D. H. Abd, "The effect of gamma value on support vector machine performance with different kernels," International Journal of Electrical and Computer Engineering, vol. 10, no. 5, 2020, doi: 10.11591/IJECE.V10I5.PP5497-5506.

[58] J. Wainer and P. Fonseca, "How to tune the RBF SVM hyperparameters? An empirical evaluation of 18 search algorithms," Artif Intell Rev, vol. 54, no. 6, 2021, doi: 10.1007/s10462-021-10011-5.

[59] W. H. Nugroho, S. Handoyo, H. C. Hsieh, Y. J. Akri, Zuraidah, and D. Dwinita Adelia, "Modeling Multioutput Response Uses Ridge Regression and MLP Neural Network with Tuning Hyperparameter through Cross Validation," International Journal of Advanced Computer Science and Applications, vol. 13, no. 9, 2022, doi: 10.14569/IJACSA.2022.0130992.

[60] T. Yan, S. L. Shen, A. Zhou, and X. Chen, "Prediction of geological characteristics from shield operational parameters by integrating grid search and K-fold cross validation into stacking classification algorithm," Journal of Rock Mechanics and Geotechnical Engineering, vol. 14, no. 4, 2022, doi: 10.1016/j.jrmge.2022.03.002.

[61] G. N. Ahmad, H. Fatima, Shafiullah, A. Salah Saidi, and Imdadullah, "Efficient Medical Diagnosis of Human Heart Diseases Using Machine Learning Techniques with and Without GridSearchCV," IEEE Access, vol. 10, 2022, doi: 10.1109/ACCESS.2022.3165792.

[62] Z. Chai and C. Zhao, "Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification," IEEE Trans Industr Inform, vol. 16, no. 1, 2020, doi: 10.1109/TII.2019.2915559.

[63] L. Gigović, H. R. Pourghasemi, S. Drobnjak, and S. Bai, "Testing a new ensemble model based on SVM and random forest in forest fire susceptibility assessment and its mapping in Serbia's Tara National Park," Forests, vol. 10, no. 5, 2019, doi: 10.3390/f10050408.

[64] A. Murugan, S. A. H. Nair, and K. P. S. Kumar, "Detection of Skin Cancer Using SVM, Random Forest and kNN Classifiers," J Med Syst, vol. 43, no. 8, 2019, doi: 10.1007/s10916-019-1400-8.

[65] Y. Zhang, D. Xiao, and Y. Liu, "Automatic Identification Algorithm of the Rice Tiller Period Based on PCA and SVM," IEEE Access, vol. 9, 2021, doi: 10.1109/ACCESS.2021.3089670.

[66] D. D. Shankar and A. S. Azhakath, "Minor blind feature based Steganalysis for calibrated JPEG images with cross validation and classification using SVM and SVM-PSO," Multimed Tools Appl, 2020, doi: 10.1007/s11042-020-09820-7.

[67] A. W. Widodo, S. Handoyo, I. Rupiwardani, Y. T. Mursityo, I. N. Purwanto, and H. Kusdarwati, "The Performance Comparison between C4.5 Tree and One-Dimensional Convolutional Neural Networks (CNN1D) with Tuning Hyperparameters for the Classification of Imbalanced Medical Data," International Journal of Intelligent Engineering and Systems, vol. 16, no. 5, pp. 748–759, 2023, doi: 10.22266/ijies2023.1031.63.

[68] D. Chicco and G. Jurman, "A statistical comparison between Matthews correlation coefficient (MCC), prevalence threshold, and Fowlkes–Mallows index," J Biomed Inform, vol. 144, 2023, doi: 10.1016/j.jbi.2023.104426.

[69] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric," PLoS One, vol. 12, no. 6, 2017, doi: 10.1371/journal.pone.0177678.