Journal of Electronics, Electromedical Engineering, and Medical Informatics
Multidisciplinary: Rapid Review: Open Access Journal **Vol. 6, No. 2, April 2024, pp: 276-288; eISSN: 2656-8632**

RESEARCH ARTICLE **OPEN ACCESS**

Manuscript received May 2, 2024; revised May 23, 2024; accepted May 27, 2024; date of publication July 8, 2024
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v6i3.453

How to cite: Helma Ghinaya, Rudy Herteno, Mohammad Reza Faisal, Andi Farmadi, and Fatma Indriani, "Analysis of Important Features in Software Defect Prediction using Synthetic Minority Oversampling Techniques (SMOTE), Recursive Feature Elimination (RFE) and Random Forest", Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 6, no. 3, pp. 276-288, July 2024.

Analysis of Important Features in Software Defect Prediction using Synthetic Minority Oversampling Techniques (SMOTE), Recursive Feature Elimination (RFE) and Random Forest

Helma Ghinaya, Rudy Herteno, Mohammad Reza Faisal, Andi Farmadi, and Fatma Indriani

Computer Science Department, Lambung Mangkurat University, Banjarbaru, South Kalimantan, Indonesia

Corresponding author: rudy.herteno@ulm.ac.id

ABSTRACT Software Defect Prediction (SDP) is essential for improving software quality during testing. As software systems grow more complex, accurately predicting defects becomes increasingly challenging. One of the challenges faced is dealing with imbalanced class distributions, where the number of defective instances is significantly lower than non-defective ones. To tackle the imbalanced class issue, use the SMOTE technique. Random Forest as a classification algorithm is due to its ability to handle non-linear data, its resistance to overfitting, and its ability to provide information about the importance of features in classification. This research aims to evaluate important features and measure accuracy in SDP using the SMOTE+RFE+Random Forest technique. The dataset used in this study is NASA MDP D", which included 12 data sets. The method used combines SMOTE, RFE, and random forest techniques. This study is conducted in two stages of approach. The first stage uses the RFE+Random Forest technique; the second stage involves adding the SMOTE technique before RFE and Random Forest to measure the accurate data from NASA MDP. The result of this study is that the use of the SMOTE technique enhances accuracy across most datasets, with the best performance achieved on the MC1 dataset with an accuracy of 0.9998. Feature importance analysis identifies "maintenance severity" and "cyclomatic density" as the most crucial features in data modeling for SDP. Therefore, the SMOTE+RFE+RF technique effectively improves prediction accuracy across various datasets and successfully addresses class imbalance issues.

INDEX TERMS Software Defect Prediction, Important Features, SMOTE, RFE, Random Forest

I. INTRODUCTION

In today's technology-driven world, software quality and reliability are becoming increasingly important. Defective software can result in significant economic losses and endanger user safety. Therefore, minimizing defects in software is a top priority for software companies. The goal of any software company is to produce software that has no defects at all [1]. Software Defect Prediction (SDP) utilizes historical data mined from software repositories to determine the quality and reliability of software modules for software quality assurance. Software Defect Prediction (SDP) is the most crucial task throughout the testing stage of the software development process because it might be challenging to identify modules prone to defects. Software Defect Prediction (SDP) is most helpful during the testing phase [2]. A class label and a set of metrics define each software module or component. The status of a module is indicated by its class label, which is either non-faulty or defective [3]. SDP models are constructed using the obtained metric values. Identifying and repairing defects takes a lot of time and resources, so it is almost impossible to eliminate all the defects. However, it is possible to reduce the number of defects that exist. The quality and reliability of the software are enhanced by standard practices and techniques such as unit testing and code inspection [1], [4]. The publicly available NASA dataset consists of two sources: the NASA MDP (Metrics Data Program) repository and the Predictor Models in Software Engineering (PROMISE) repository. NASA MDP is a

software metric for data frequently used in software defect prediction research [4]. This study uses NASA's clean software defect datasets for experiments, including CM1, JM1, KC1, KC3, MC1, MC2, MW1, PC1, PC2, PC3, PC4, and PC5.

Machine learning algorithms encounter challenges when exposed to imbalanced data sets. Imbalanced data means that data samples from one class far outnumber data samples from another class, thus hindering most software defect prediction techniques. The forecasts have deceiving accuracy and are skewed. This is due to a lack of information about the minority class. Machine learning algorithms often classify every test case sample into the majority class to increase the accuracy metric since they assume that data sets are balanced with equal class weights. The application of sampling techniques is one potential fix for this issue. Studies have demonstrated that for many base classifiers, an imbalanced data set results in a worse overall classification than a balanced data set [5], [6].

Resampling procedures are frequently employed in imbalanced datasets to correct class distribution imbalances. One possible solution to the issue of class imbalance is to employ sampling techniques, namely oversampling and undersampling. Oversampling causes overfitting because it duplicates the same minority class, whereas undersampling eliminates the majority class until the distribution is balanced. Synthetic Minority Over-sampling Technique (SMOTE) is a popular method that balances datasets by synthesizing minority class examples [6]. Through the removal of redundant, noisy, irrelevant, or non-beneficial predictor variables from the training dataset or feature space, feature selection techniques are frequently used in remote sensing analyses to increase the prediction accuracy and interpretability of machine learning classification algorithms. Typically, this is done by identifying and selecting the best feature combination that maximizes the accuracy of a particular classification model of interest as measured using an accuracy metric [7].

Pre-processing tasks, such as feature selection, have been recognized as crucial components of the prediction process because they improve the data quality, increasing the prediction models' efficiency. Subsets of the original software's features that can most accurately depict the original features without devaluing them are the focus of feature selection. Feature selection approaches use labeled datasets to identify a collection of germane features and assess the qualities of the accessible features. Consequently, the significant dimensionality issue in SDP datasets can be lessened by implementing feature selection approaches in SDP processes [8]. Recursive Feature Elimination (RFE) is a wrapper-style feature selection technique that generates many classification models and iteratively eliminates features that do not increase classification accuracy to find the best feature combinations. Recursive Feature Elimination (RFE) uses backward selection, which means that after starting with the entire feature set, it iteratively removes characteristics that either improve or decrease the classification's accuracy until the best possible feature combination is discovered [7].

Machine learning used to predict is usually C4.5, Support Vector Machine (SVM), Naïve Bayes, K-nearest Neighbors (KNN), Decision Tree, Random Forest, and others. An ensemble learning method known as Random Forest (RF) has a track record of performing exceptionally well. High accuracy, processing of thousands of input variables, and integrated metrics of varying importance are only a few of its many positive attributes. Furthermore, Random Forest (RF) is resistant to noise and outliers. The computing procedure is quick, and the RF parameter adjustment is straightforward. RF has excelled [9], [10].

This research [9] investigated the potential of the Random Forest (RF) method for extracting and mapping five forest types found in Yanqing, north China, using multi-source data. One hundred twenty-five features were obtained using the DEM, GF-1 WFV pictures, Google Earth photos, and forest inventory data. The recursive feature elimination (RFE) method chose thirty-two characteristics for mapping five forest types. The findings yielded an overall accuracy rate of 87.06% and a Kappa coefficient of 0.833.

In other research, Metric Data Program (MDP) datasets from NASA were used for the experiments. Software defect history logs based on multiple complexity indicators are available in the NASA MDP. However, the NASA MDP has an uneven distribution of modules that are and are not malfunctioning. The percentage of faulty modules in the distribution is lower than zero. It may lower the effectiveness of software defect detection. It is necessary to duplicate the distribution of the faulty module. The distribution between defective and non-defective modules is balanced in this study by applying the Synthetic Minority Oversampling Technique (SMOTE). On the NASA MDP dataset, software defect identification using fuzzy association rule mining (FARM) in conjunction with dataset balance using SMOTE and CFS complexity metric selection yields accuracy and sensitivity of 91.63% and 85.51%, respectively [11].

The research by [12] presents a Decision Tree (DT) classification technique to examine the risk factors associated with cervical cancer. SMOTETomek, a combination of under and oversampling techniques, was used with recursive feature elimination (RFE) and most minor absolute shrinkage and selection operator (LASSO) feature selection strategies. A comparative analysis of the suggested model has been carried out to demonstrate the efficacy of feature selection and class imbalance based on the classifier's accuracy, sensitivity, and specificity. The DT using the chosen features from SMOTETomek and RFE performs better, achieving 100% sensitivity and 98.72% accuracy. When features are decreased, and a significant class imbalance is handled, the DT classifier performs better when addressing classification problems.

In this research, the SMOTE-MCT variant will be used to balance the data, recursive feature elimination (RFE) will be

used for feature selection, and random forest will be used for classification models. Based on the previous explanation, this research was carried out to find out about the most important features and measure the level of accuracy and AUC-ROC in SDP by integrating feature selection techniques such as the SMOTE-MCT variant, RFE, and the RF algorithm. By combining these techniques, this research can provide deeper insight into the importance of specific features in SDP predictions and improve the quality of the classification models built.

In previous research, the main focus was on using random forests for mapping forest types in Yanqing [9]. In contrast, other studies utilized SMOTE and FARM to enhance software defect detection on the NASA MDP dataset [11]. However, in this study, we integrate the RFE feature selection technique with a variant of SMOTE-MCT to improve prediction accuracy and AUC-ROC and understand the important features of SDP. We employ Random Forest as our classification model and identify the pivotal features contributing to SDP predictions. The primary goal of this research is to enhance the accuracy of SDP predictions by combining the RFE feature selection technique, SMOTE-MCT variants, and the Random Forest algorithm. Additionally, we strive to pinpoint the most influential features in SDP to deepen our understanding of software quality and reliability. Research Contributions:

- Integrates SMOTE-MCT, RFE, and RF variants to improve SDP prediction accuracy and AUC-ROC.
- Handling class imbalance problems more effectively compared to previous methods.
- Identifying important features influencing software defect prediction provides deeper insight into software quality and reliability.

With this approach, this research contributes to improving the accuracy and AUC-ROC of SDP predictions and better understanding the important features in software defect detection.

II. MATERIAL AND METHODS

In general, the research process include data collection, splitting data into training and test sets, model testing, and evaluation. The proposed model can be seen in FIGURE 1. In essence, this research was carried out in two stages: the first used the RFE + RF technique, and the second used the SMOTE + RFE + RF technique. The research process began with the collection of 12 NASA MDP datasets. After collecting the data, preprocessing was done, including coding and handling missing values. The data was then split into a 70:30 ratio, with 70% used for training and 30% for testing. Additionally, the SMOTE + RFE + RF process used stratified 10-fold cross-validation to ensure balanced class distribution in each fold.

In the first stage, important features were identified using the Recursive Feature Elimination (RFE) technique combined with the Random Forest (RF) algorithm. The second stage applied the Synthetic Minority Over-sampling Technique (SMOTE) before the RFE + RF process to handle a class imbalance in the data. The results from both stages were then evaluated by comparing the levels of accuracy and the Area Under the Curve (AUC-ROC). Furthermore, the important features identified in both stages were compared to understand the effectiveness of each method in the context of this research.

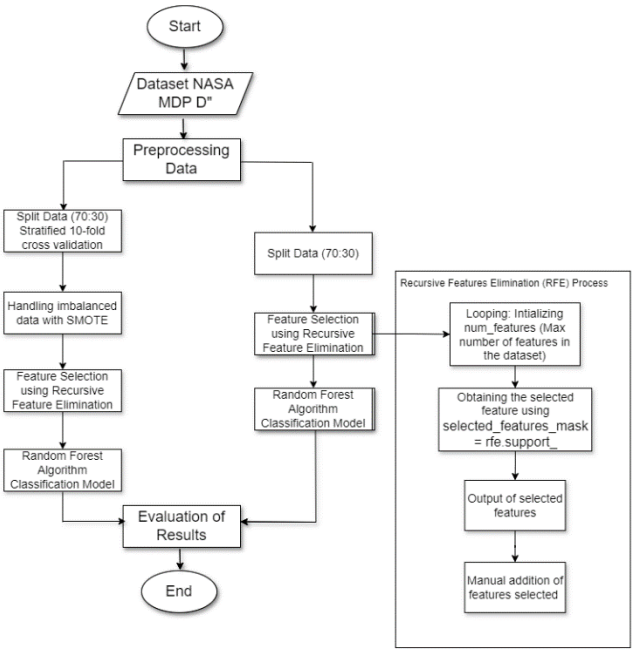

FIGURE 1. Research methods

A. DATASET

The data used in this research come from the Facility Metrics Data Program (MDP) repository of the National Aeronautics and Space Administration (NASA), accessible at NASA MDP|Fighshare, known by the acronym NASA MDP. MDP is widely used in the field of software defect prediction [13]. The NASA MDP datasets are utilized to assess our approach. Numerous software defect history logs based on Branch Count, McCabe, and Halstead metrics are available in NASA MDP datasets. This publicly available dataset has been the subject of numerous prior studies. In this study, 12 datasets were used, namely CM1, JM1, KC1, KC3, MC1, MC2, MW1, PC1, PC2, PC3, PC4, and PC5.

TABLE 1
Dataset NASA MDP D"

Dataset	number of features	amount of data
CM1	38	327
JM1	22	7720
KC1	22	1162
KC3	40	194
MC1	39	1952
MC2	40	124
MW1	38	250
PC1	38	679
PC2	37	722
PC3	38	1053
PC4	38	1270
PC5	39	1694

B. PREPROCESSING DATA

Data preprocessing is a critical task in data mining. The data must be optimal, making it appropriate for various ML models [14]. Data preprocessing is a series of steps or stages on raw data before the data is used for analysis or model development. The main goal of data preprocessing is to improve data quality, guarantee accurate analysis results, and overcome problems or deficiencies that may arise with raw data. The purpose of preprocessing datasets is to get them ready for algorithmic processing. The data is balanced now, and null values are verified and fixed. Anything that now affects the machine learning model's performance can be managed more deftly [15]. Data preprocessing is the label encoding process, which converts each table containing string or text data into numerical form.

Moreover, label encoding values are transformed into numerical form using label encoding so that machine learning algorithms can handle them more efficiently [15], [16]. This prepares the data for machine learning by converting the labels into a proper numeric format. In this study, the label "Y" was changed to "1" and the label "N" to "0." This label change can be seen in one of the datasets, namely CM1 TABLE 2 and TABLE 3.

TABLE 2
Before preprocessing

id	LOC_ BLANK	BRANCH_ OUT	CALL_ PAIRS	…	NUMBER_OF_ LINES	PERCENT_ COMMENTS	LOC_ TOTAL	Defective
1	2	3	0	…	9	47.06	9	N
2	3	3	0	…	19	26.67	13	N
3	38	35	4	…	218	41.90	109	N
4	1	7	5	…	68	22.64	41	Y
5	9	15	4	…	73	57.14	41	N
…	…	…	…	…	…	…	…	…
323	67	29	10	…	228	42.50	119	N
324	9	3	5	…	40	40.00	18	N
325	3	3	1	…	18	21.43	12	N
326	6	9	3	…	61	59.26	32	N
327	1	3	4	…	12	0.00	10	N

TABLE 3
After preprocessing

id	LOC_ BLANK	BRANCH_ OUT	CALL_ PAIRS	…	NUMBER_OF_ LINES	PERCENT_ COMMENTS	LOC_ TOTAL	Defective
1	2	3	0	…	9	47.06	9	0
2	3	3	0	…	19	26.67	13	0
3	38	35	4	…	218	41.90	109	0
4	1	7	5	…	68	22.64	41	1
5	9	15	4	…	73	57.14	41	0
…	…	…	…	…	…	…	…	…
323	67	29	10	…	228	42.50	119	0
324	9	3	5	…	40	40.00	18	0
325	3	3	1	…	18	21.43	12	0
326	6	9	3	…	61	59.26	32	0
327	1	3	4	…	12	0.00	10	0

C.SPLIT DATA

After the dataset undergoes preprocessing stages involving normalization and handling of missing values, the data is split using Stratified K-Fold Cross Validation with k=10. This process ensures that the data split remains at a 70:30 ratio between training and testing data while ensuring the target classes are balanced within each cross-validation fold.

In each iteration of the K-Fold Cross Validation process, the training data comprises 70% of the total data, and the testing data comprises 30%. This division is done while maintaining the proportions of the target classes in each fold to ensure a balanced representation of these classes in both the training and validation sets. In FIGURE 2, you can see the illustration of stratified K-fold cross validation.

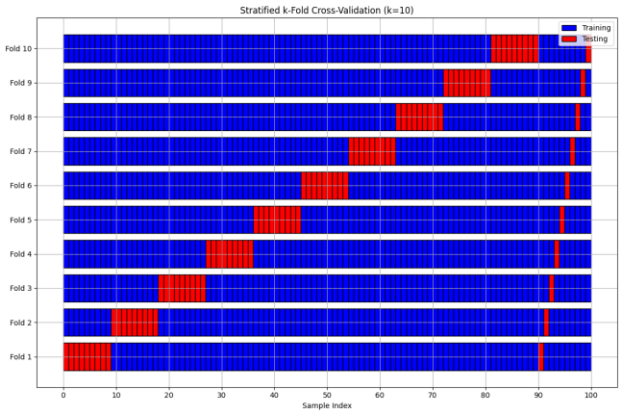

FIGURE 2. **Illustration of Stratified K-flod Cross Validation**

D. SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE (SMOTE)

A condition where the class distribution is imbalanced in a dataset occurs when the majority class (majority class) has a larger amount of data than the minority class (minority class) [17] [18]. Sampling methods are very popular for balancing the class distribution. Methodologies for oversampling and undersampling have drawn a lot of interest in mitigating the impact of imbalanced data sets. Therefore, the goal of applying oversampling techniques is to match the number of samples from the majority class with the number of samples from the minority class. To create a balanced dataset using the over-sampling technique, instances from the minority class are randomly duplicated until the required ratio is reached. One of the benefits of over-sampling is that no information is lost. Nevertheless, oversampling can lead to overfitting and may not be useful in enhancing minority class detection, even though it is commonly utilized [19], [20], [21], [22]. The difference between a sample's feature vector and that of its closest neighbor must first be computed to create synthetic samples using the SMOTE approach. Subsequently, the acquired disparity is multiplied by a value chosen at random from the 0 to 1 range [23] and appended to the feature vector of the sample under consideration. The new point that is generated at random along the line segment that links two features is the outcome of this operation. Stated differently, the utilization of this methodology broadens the minority class's decision-making domain. The SMOTE algorithm operates in the following manner: FIGURE 3 [24], [25]. The SMOTE technique formula is as follows [18], [26], [27]:

$$X_{new} = Xi + (Xknn - Xi) \times \delta \quad (1)$$

In the data synthesis process, X_new represents the synthesized data, while Xi is the original data to be replicated. X_knn denotes the data point closest to Xi in the feature space. During synthesis, the algorithm analyzes Xi and its nearest neighbor, X_knn, to generate X_new, maintaining the original dataset's patterns. The degree of variation in the synthesized data is controlled by the random parameter δ, sampled from a uniform distribution between 0 and 1. This approach ensures the creation of additional data points that closely resemble the original dataset, contributing to tasks such as machine learning model training or data quality assessment.

TABLE 4
Dataset after SMOTE-MCT

Dataset	Before SMOTE		After SMOTE	
	0	1	0	1
CM1	199	29	199	199
JM1	4276	1128	4276	4276
KC1	607	206	607	607
KC3	110	25	110	110
MC1	1341	25	1341	1341
MC2	55	31	55	55
MW1	157	18	157	157
PC1	437	38	437	437
PC2	494	11	494	494
PC3	646	91	646	646
PC4	766	123	766	766
PC5	865	320	865	865

In TABLE 4, you can see the change in the number of minority data points, symbolized by the label "1," after using the synthetic minority oversampling techniques (SMOTE) method for the MCT variant. The amount of data with the label "1" is balanced by adjusting the amount of data in the majority, namely the label "0." As in the example in TABLE 3, in the CM1 dataset, the amount of data before SMOTE shows that the "0" label is 199 and the "1" label is 29, so after SMOTE, the amount of data on the "1" label changes to 199, adjusting to the number of "0" labels. As with all datasets on NASA MDP D", after SMOTE is carried out, the amount of data on label "1" is adjusted to the amount of data on label "0" so that all data is balanced.

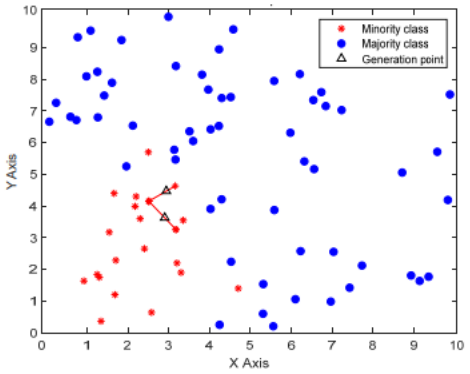

FIGURE 3. **Illustration of SMOTE algorithm**

D. RECURSIVE FEATURE ELIMINATION (RFE)

Many previous studies employed feature selection to improve the model's prediction accuracy [28]. The best feature subset can be found using the wrapper-based feature-ranking technique known as recursive feature elimination (RFE). In essence, it involves repeatedly building a model until the ideal feature subset is chosen [29]. Depending on how many features we desire, RFE will choose the best features. RFE also functions by first fitting the model, after which it determines how important each feature is. Subsequently, the least significant feature will be removed from the feature set, and a new model fitting and feature importance calculation will be initiated based on the remaining features, with the least significant feature being removed [30]. The calculation stops when the feature set reaches a predefined number of features [31].

Recursive Feature Elimination (RFE) with a Random Forest estimator will be used as the main method for feature selection. This decision was based on considerations of computational time efficiency, where research results showed that RFE-RF required a shorter time than RFE with Support Vector Machines (SVM). Although the use of RFE-SVM is more common in the literature. In this study, RFE-RF was chosen because of its better computational time efficiency, thus allowing the analysis to be carried out more efficiently and effectively.

Algorithm RFE

1. Initialization: Initialize the dataset $D = \{X, y\}$, where X is the feature matrix, and y is the target vector.
2. Model Building: Perform model building on the dataset D to learn patterns or relationships between features and the target.
3. Feature Assessment: Evaluate the relative importance of each feature in the dataset D, and determine the most important features for classification or prediction purposes.
4. Feature Selection: Based on the previous assessment, select a subset of the most important features using algorithms such as Recursive Feature Elimination (RFE).
5. Stop Condition Checking: Check if the stopping criteria have been met, such as reaching the desired number of features or achieving adequate model performance.

E. RANDOM FOREST (RF)

Random forest (RF), a powerful machine-learning method [32] based on the theory of decision trees [33], [34] which was proposed in 2001, is composed of multiple decision trees. Based in FIGURE 4, Random Forest combines many Decision Trees based on the bagging technique [35]. According to the theory behind ensemble learning, each learner should be "good but different," meaning that they should have a comparatively high recognition rate that sets them apart from the rest. Nevertheless, the number of decision trees is predetermined when choosing a single decision tree [33], [36]. RF establishes numerous decision trees, which improves generalization performance as an ensemble learning technique. N sample sets must be created to train each decision tree in an RF with N trees [33]. Independent sample vector values, distributed identically across each tree in the model, are used to build each decision tree. Random forests can improve model performance by combining predictions from these trees [37]. The Random Forest classifier was chosen primarily for its ability to manage high-dimensional data's complexity and provide robust predictions. The Random Forest technique formula is as follows [15]:

$$Gini\ Index\ (D) = 1 - \sum_{i=1}^{m} P_i^2 \qquad (2)$$

Where Pi is the proportion of the number of attributes in each class, and m is the number of each attribute. The feature that has the lowest total Gini Index value will be the root node in the tree. The total Gini Index at an internal node (e.g., K) is calculated in the following equation (3).

$$Tot.Gini\ index\ (K) = \frac{T_1}{T}\ Gini\ Index\ (D_1) + \frac{T_2}{T}\ Gini\ Index\ (D_2) \qquad (3)$$

Where $T1$ is the total records belonging to the first class, $T2$ is the total records belonging to the second class, and T is the total records of all classes. This process continues with the formation of child nodes until all nodes in the tree cannot be split. After the entire tree is formed, the classification stage continues using the voting method [15]. The following are some of the steps involved in the random forest [38]:

• Random forest selects random records from a data set of k records.
• A distinct decision tree is constructed for each sample.
• Each decision tree yields a result.
• In classification, the final result is determined by majority voting.

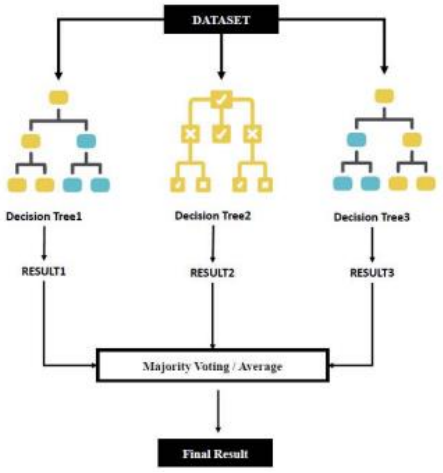

FIGURE 4. Illustration of How Random Forest Works

F. PERFORMANCE EVALUATION

In this research, the evaluation of the results was carried out with accuracy and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Evaluation metrics should be appropriately selected to assess machine learning models' performance accurately. Accurate is the most commonly used metric for evaluating classification models [39]. Accuracy is one of many metrics that may be used to assess classification methods. The percentage of accurate forecasts among all feasible guesses is known as accuracy. According to the suggested model, accuracy is a gauge of how well the model can predict a consumer's opinion of a product—whether they would like it or not [40]. In classification, the evaluation metrics are calculated from true positive (TP), false positive (FP), true negative (TN), and false negative (FN) [39]. TP indicates the number of clean software instances correctly classified as clean, while TN indicates the number of defective software instances. FP indicates the number of clean software instances correctly classified as clean, and FN indicates the number of software instances incorrectly classified as defective. Several problematic software instances were mistakenly considered clean [41]. The accuracy can be calculated by using Equations [42], [43]

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN} \qquad (4)$$

Besides accuracy, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) was also used as a performance metric. The AUC-ROC metric provides a comprehensive measure of a model's ability to distinguish between classes across all classification thresholds. It is particularly useful for evaluating models on imbalanced

datasets as it considers both the true positive rate (sensitivity) and the false positive rate, offering a balanced view of the model's performance. The categories for the range of AUC values can be seen in TABLE 5 [23].

$$AUC = \frac{(\frac{TP}{TP+FN}) x (\frac{TN}{TN+FP})}{2} \quad (5)$$

TABLE 5
Category on AUC Values

Category	AUC Values
Excellent	0.90 -1.00
Good	0.80 -0.90
Fair	0.70 -0.80
Poor	0.60 -0.70
Failure	0.50 -0.60

III. RESULT

This research is conducted in two steps: RFE+RF and SMOTE+RFE+RF. In the first stage, the research focused on calculating the level of accuracy of NASA MDP data using the recursive feature elimination (RFE) technique and random forest (RF) classification. Then, in the second stage, additional testing was carried out to calculate the level of accuracy of NASA MDP data using synthetic minority over-sampling technique (SMOTE), followed by applying the Recursive Feature Elimination (RFE) technique and Random Forest classification. The test results in this research focus on identifying important features using RFE method and then comparing the level of accuracy and AUC-ROC that has been carried out in the two previous stages. In this way, important features in the calculations can be identified.

In analyzing the dataset after implementing SMOTE-MCT, this study used several important statistical tests to determine the significance level and evaluate the results. First, this research applies RFE, a feature selection technique that aims to improve model performance by selecting the most relevant features. RFE removes the least important features based on the importance determined by a base model such as Random Forest (RF) and then builds a model with the remaining features, repeating this process until the desired number of features is reached. After selecting the most relevant features using RFE, this research builds a classification model with Random Forest (RF). Random Forest is an ensemble learning algorithm used for classification and regression. In the context of this analysis, RF is used for classification by building many decision trees during training. The output is the mode of the class (classification) of each tree. This research uses SMOTE (Synthetic Minority Over-sampling Technique) to overcome the problem of class imbalance in the dataset. SMOTE aims to balance the dataset by creating synthetic samples from minority classes through interpolation between existing minority samples so that the model has a better chance of learning patterns from minority classes. This study compares two approaches: RFE+RF and SMOTE+RFE+RF. In the RFE+RF approach, RFE is used first to select the most relevant features from the dataset; then Random Forest is used to build a classification model.

TABLE 6
Comparison of Different Techniques in Terms of Accuracy

Dataset	RFE+RF		SMOTE+RFE+RF	
	n_features	Acc	n_features	Acc
CM1	2, 7, 8, 10, 16	0.8182	19, 21, 33	0.9833
JM1	16, 18	0.8044	22	0.9228
KC1	18	0.7908	8, 15, 16	0.8795
KC3	22	0.7966	36	0.9545
MC1	2, 34	0.9795	36	**0.9988**
MC2	30, 38, 40	0.7895	12, 13, 20, 22	0.9091
MW1	2-5, 7, 8, 11, 13-38	0.8667	5, 7, 12, 13, 18, 19, 20, 24, 25	0.9895
PC1	17, 18, 19, 27, 38	0.9559	17, 21, 27	0.9924
PC2	23, 26 - 37	0.9862	3, 7, 15-37	0.9966
PC3	21	0.8956	21	0.9871
PC4	26	0.9160	30	0.9848
PC5	31	0.8114	15, 26	0.8690

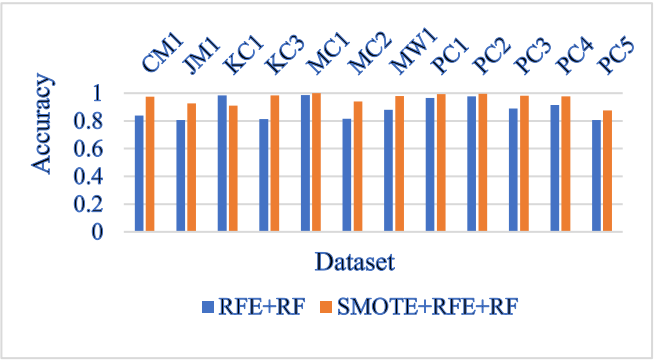

FIGURE 5. Comparison visualization of different techniques in term of accuracy.

Based on the results in FIGURE 5 and TABLE 6, after comparing the first and second stages of the research, it was found that the SMOTE effect could increase the accuracy values in all datasets. SMOTE increases the accuracy value in CM1 by 0.1735, JM1 by 0.1184, KC1 by 0.0887, KC3 by 0.1579, MC1 by 0.0193, MC2 by 0.1196, MW1 by 0.1228, PC1 by 0.0365, PC2 by 0.0104, PC3S by 0.0889, PC4 by 0.0688, and PC 5 is 0.0576. The best classification performance of RFE+Random Forest is on the PC2 dataset, with an accuracy of 0.9862. The best classification performance of SMOTE+RFE+Random Forest is on the MC1 dataset, with an accuracy of 0.9988. Therefore, in the research above, SMOTE has a good effect in increasing accuracy.

TABLE 7
Comparison of Accuracy and AUC-ROC

Dataset	RFE+RF		SMOTE+RFE+RF	
	Acc	AUC-ROC	Acc	AUC-ROC
CM1	0.8182	0.6384	0.9917	1
JM1	0.8044	0.7067	0.9228	0.9679
KC1	0.7908	0.7081	0.8795	0.9592
KC3	0.7966	0.6246	0.9545	0.9843
MC1	0.9795	0.8431	**0.9988**	1
MC2	0.7895	0.6811	0.9091	0.9148
MW1	0.8667	0.8069	0.9895	1

PC1	0.9559	0.9338	0.9924	1
PC2	0.9862	0.8691	0.9966	1
PC3S	0.8956	0.7991	0.9845	1
PC4	0.9160	0.9831	0.9848	0.9997
PC5	0.8114	0.8241	0.8690	0.9527

The level of significance in this analysis is seen from two primary metrics: Accuracy and AUC-ROC). Accuracy indicates the percentage of correct predictions out of the total predictions made by the model, with higher values indicating the model is better at making correct predictions. Meanwhile, AUC-ROC measures the model's ability to differentiate between positive and negative classes, with a higher value indicating a model better at distinguishing between these classes, where a value of 1 indicates a perfect model.

TABLE 8
Confidence Interval for Model Performance Metrics

Category	SE	ME	CI	
			Lower	Upper
Acc RFE+RF	0.022	0.0432	0.8244	0.9108
AUC RFE+RF	0.033	0.0648	0.7201	0.8496
Acc SMOTE+RFE+RF	0.0139	0.0273	0.9288	0.9834
AUC SMOTE+RFE+RF	0.008	0.0156	0.9660	0.9971

TABLE 8 displays the confidence intervals (CI) for the model performance metrics with four model categories evaluated, namely Acc RFE+RF, AUC RFE+RF, Acc SMOTE+RFE+RF, and AUC SMOTE+RFE+RF. The confidence interval for accuracy (Acc) and Area Under Curve (AUC) each indicate the range of values within which the model performance lies with 95% confidence. SE stands for Standard Error, which measures the accuracy with which a sample represents a population. ME stands for Margin of Error, the range within which the actual value is expected to fall. A smaller SE indicates a more precise estimate of the population parameter, while the ME provides the extent of the possible error in the estimate. The tight confidence intervals, reflected by small SE and ME values, suggest that the model performance metrics are estimated with a high level of precision and reliability. Based on the TABLE 8, the confidence intervals (CI) for the model performance metrics appear tight, indicating that the model performance estimates have a high degree of certainty. For example, for Acc RFE+RF, the accuracy range is between 0.8244 and 0.9108, while for AUC RFE+RF, the AUC range is between 0.7201 and 0.8496. Similarly, Acc SMOTE+RFE+RF has an accuracy range between 0.9288 and 0.9834, while AUC SMOTE+RFE+RF has an AUC range between 0.9660 and 0.9971. This CI indicates that these models have stable and reliable performance in predictions. The analysis results of important features in the research, as shown in TABLE 9, indicate that "MAINTENANCE_SEVERITY" and "CYCLOMATIC_DENSITY" are the most frequently appearing important features. Further examination reveals that features such as "CYCLOMATIC_DENSITY," "NORMALIZED_CYCLOMATIC_COMPLEXITY," and "MAINTENANCE_SEVERITY" consistently appear in both methods. These findings suggest that these features possess characteristics necessary for the modeling process, warranting further investigation.

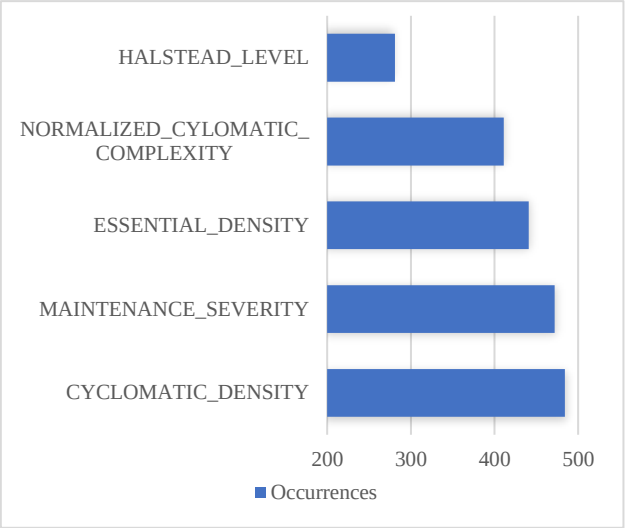

FIGURE 6. Visualization of important features

TABLE 10
List of Important Features

No	Important Features	Occurrences
1	CYCLOMATIC DENSITY	484
2	MAINTENANCE SEVERITY	472
3	ESSENTIAL DENSITY	441
4	NORMALIZED CYCLOMATIC COMPLEXITY	411
5	HALSTEAD LEVEL	281
6	LOC CODE AND COMMENT	271
7	DECISION DENSITY	217
8	DESIGN DENSITY	215
9	GLOBAL DATA DENSITY	187
10	CYCLOMATIC COMPLEXITY	166

Based on the results identified in the data above, FIGURE 6 and TABLE 10 summarizes the ten important features with the highest occurrences. Notably, "CYCLOMATIC_DENSITY," with 484 occurrences, and "MAINTENANCE_SEVERITY," with 472 occurrences, emerge as the most critical features in data modeling. This underscores the significant role of cyclomatic complexity and maintenance severity in determining the predictive model's accuracy. These features offer valuable insights into pivotal factors requiring predictive model development consideration. Understanding and incorporating the roles of "CYCLOMATIC_DENSITY," "MAINTENANCE_SEVERITY," and other important features facilitate improved and more accurate decision-making.

TABLE 9
Comparison of the appearance of features

Data set	RFE+RF Important features	SMOTE+RFE+RF Important features	number of occurrences	Data set	RFE+RF Important features	SMOTE+RFE+RF Important features	number of occurrences
CM1	MAINTENANCE_SEVERITY	CYCLOMATIC_DENSITY	38	MW1	DECISION_DENSITY	ESSENTIAL_DENSITY	38
	CYCLOMATIC_DENSITY	HALSTEAD_ERROR_EST	37		LOC_CODE_AND_COMMENT	PARAMETER_COUNT	37
	NORMALIZED_CYLOMATIC_COMPLEXITY	MODIFIED_CONDITION_COUNT	36		MODIFIED_CONDITION_COUNT	LOC_CODE_AND_COMMENT	36
JM1	HALSTEAD_LEVEL	LOC_CODE_AND_COMMENT	22	PC1	NORMALIZED_CYLOMATIC_COMPLEXITY	NORMALIZED_CYLOMATIC_COMPLEXITY	38
	LOC_CODE_AND_COMMENT	NUM_UNIQUE_OPERATORS	21		CYCLOMATIC_DENSITY	CYCLOMATIC_DENSITY	37
	LOC_BLANK	CYCLOMATIC_COMPLEXITY	20		MAINTENANCE_SEVERITY	MAINTENANCE_SEVERITY	36
KC1	HALSTEAD_LEVEL	HALSTEAD_LEVEL	22	PC2	CYCLOMATIC_DENSITY	CYCLOMATIC_DENSITY	37
	LOC_CODE_AND_COMMENT	LOC_CODE_AND_COMMENT	21		DESIGN_DENSITY	DESIGN_DENSITY	36
	CYCLOMATIC_COMPLEXITY	CYCLOMATIC_COMPLEXITY	20		NORMALIZED_CYLOMATIC_COMPLEXITY	MAINTENANCE_SEVERITY	35
KC3	LOC_CODE_AND_COMMENT	NODE_COUNT	40	PC3	NORMALIZED_CYLOMATIC_COMPLEXITY	NORMALIZED_CYLOMATIC_COMPLEXITY	38
	DECISION_DENSITY	EDGE_COUNT	39		MAINTENANCE_SEVERITY	MAINTENANCE_SEVERITY	37
	LOC_COMMENTS	CYCLOMATIC_COMPLEXITY	38		ESSENTIAL_DENSITY	ESSENTIAL_DENSITY	36
MC1	CYCLOMATIC_DENSITY	ESSENTIAL_DENSITY	39	PC4	CYCLOMATIC_DENSITY	CYCLOMATIC_DENSITY	38
	GLOBAL_DATA_DENSITY	NORMALIZED_CYLOMATIC_COMPLEXITY	38		NORMALIZED_CYLOMATIC_COMPLEXITY	NORMALIZED_CYLOMATIC_COMPLEXITY	37
	NORMALIZED_CYLOMATIC_COMPLEXITY	GLOBAL_DATA_DENSITY	37		MAINTENANCE_SEVERITY	HALSTEAD_LEVEL	36
MC2	ESSENTIAL_DENSITY	ESSENTIAL_DENSITY	40	PC5	HALSTEAD_LEVEL	HALSTEAD_LEVEL	39
	GLOBAL_DATA_DENSITY	NORMALIZED_CYLOMATIC_COMPLEXITY	39		CYCLOMATIC_DENSITY	CYCLOMATIC_DENSITY	38
	MAINTENANCE_SEVERITY	GLOBAL_DATA_DENSITY	38		MAINTENANCE_SEVERITY	MAINTENANCE_SEVERITY	37

In this research, the use of the SMOTE+RFE+RF technique has had a significant impact on increasing accuracy on several datasets, such as CM1, JM1, KC1, KC3, MC1, MC2, MW1, PC1, PC2, PC3, PC4, and PC5. With a reasonably significant increase in accuracy on most datasets, especially on MC1, with an accuracy value reaching 0.9998, the SMOTE+RFE+RF technique shows strong potential for improving classification model performance. These results

show that using the SMOTE technique can help balance unbalanced datasets, thereby improving the quality of the resulting classification model. These findings indicate that both features “CYCLOMATIC_DENSITY” and “MAINTENANCE_SEVERITY” may strongly correlate with the predicted target variables in the software context. Therefore, a deep understanding of the role of these important features can help develop better and more accurate predictive models.

IV. DISCUSSION

The results of this study, found in TABLE 6 and FIGURE 5, demonstrate significantly enhanced prediction accuracy across most of the evaluated datasets when employing the SMOTE+RFE+RF technique. Specifically, our method achieves a notable average increase in accuracy compared to baseline models that do not utilize SMOTE or RFE. TABLE 6 provides a detailed accuracy comparison between RFE+RF and SMOTE+RFE+RF across various datasets. For example, in dataset CM1, the accuracy increases from 0.8182 to 0.9833 after applying SMOTE. Similarly, dataset KC1 shows an accuracy improvement from 0.7908 to 0.8795 using SMOTE+RFE+RF. However, most datasets, such as PC1 and PC4, exhibit substantial enhancements, with accuracy rising from 0.9559 to 0.9924 and from 0.9160 to 0.9848, respectively.

Compared to other state-of-the-art methods, such as Support Vector Machine (SVM) and Decision Trees (DT), the combination of SMOTE, RFE, and Random Forest consistently shows superior performance in accuracy and handling class imbalances. Recent studies, such as those by Zhang et al. [13] and Kim et al. [14], also corroborate the benefits of using ensemble methods like Random Forest in conjunction with SMOTE for software defect prediction. These results further emphasize the consistent increase in accuracy across diverse datasets, reinforcing the potential of SMOTE as a valuable tool in improving prediction models within the software defect prediction domain. However, unlike previous studies focusing on individual techniques, integrating RFE for feature selection further boosts the model's performance by eliminating redundant and irrelevant features.

Important feature analysis highlights the critical role of features "MAINTENANCE_SEVERITY" and "CYCLOMATIC_DENSITY" in software defect prediction, as shown in TABLE 10 and FIGURE 6. These findings are consistent with previous studies [7] that identified cyclomatic complexity and maintenance severity as crucial factors influencing prediction model accuracy. Differences in findings compared to other studies may be due to the diverse nature of datasets and the specific features analyzed. For instance, while some studies have emphasized code churn and developer activity metrics, this study underscores the importance of maintenance-related metrics.

The insights gained from this research offer valuable guidance for researchers and practitioners in building more effective software defect prediction models. In real-world software development and testing scenarios, understanding critical features like "CYCLOMATIC_DENSITY" and "MAINTENANCE_SEVERITY" can facilitate more targeted feature selection and lead to the development of more accurate prediction models. Furthermore, employing SMOTE to handle class imbalances should be considered a best practice in predictive model development because it can enhance the representation of the minority class without introducing significant noise.

While this study contributes valuable insights, it is essential to acknowledge its limitations. The research is confined to the NASA MDP dataset, which may limit the generalizability of the findings. This dataset's specific characteristics might not represent other domains or types of software projects. Additionally, using SMOTE may introduce synthetic examples that do not perfectly reflect real-world minority class samples, potentially affecting the robustness of the predictions. Furthermore, addressing the issue of overfitting should be a key focus for future research endeavors. Overfitting was identified as one of the limitations of this study, potentially impacting the robustness and generalizability of the predictive models. Future studies can ensure the development of more robust and reliable software defect prediction models by prioritizing the prevention or mitigation of overfitting.

The rationale for choosing the SMOTE+RFE+RF technique over others is its combined ability to handle class imbalances, reduce dimensionality, and enhance prediction accuracy. SMOTE addresses the skewness in class distribution, RFE ensures that only the most relevant features are used, and Random Forest provides robustness and high accuracy in classification tasks. Reflecting on the broader impact of this research, the findings have significant implications for software defect prediction and machine learning. By demonstrating the effectiveness of the SMOTE+RFE+RF technique, this study provides a robust methodology that can be adapted and extended to other predictive modeling tasks, including those in different disciplines such as finance, healthcare, and cybersecurity.

In conclusion, this research enhances the accuracy of software defect predictions and contributes to a deeper understanding of the critical features influencing these predictions. Integrating SMOTE, RFE, and Random Forest offers a comprehensive approach to tackling class imbalances and feature selection, setting a new standard for future studies in this domain.

V. CONCLUSION

This research aims to evaluate important features and measure accuracy in software defect prediction (SDP) using the SMOTE+RFE+RF technique. The study was conducted in two stages to measure accuracy on the NASA MDP dataset: initially employing the RFE+RF technique and subsequently integrating the SMOTE technique before RFE+RF in the second stage. Results demonstrate that incorporating SMOTE

enhances accuracy across most datasets, with the MC1 dataset achieving the highest accuracy of 0.9998. The analysis of important features identified "MAINTENANCE_SEVERITY" and "CYCLOMATIC_DENSITY" as the most frequently occurring features, underscoring their significant role in SDP modeling. This finding aligns with previous studies [7] and highlights the consistency in recognizing these features and their importance across various research efforts.

The novelty of this research lies in integrating the SMOTE+RFE+RF techniques, demonstrating their potential to enhance SDP classification model performance while identifying pivotal features. These insights offer valuable contributions to the field by providing a more practical approach to SDP and emphasizing the importance of feature selection in predictive modeling. Despite the promising results, it is essential to recognize the study's limitations. The research is confined to the NASA MDP dataset, which may limit the generalizability of the findings. Future research could explore the applicability of the SMOTE+RFE+RF technique across different datasets or domains to further validate its effectiveness.

In conclusion, this study showcases the potential of the SMOTE+RFE+RF technique in improving SDP classification model performance and highlights the importance of feature selection in predictive modeling. The findings contribute valuable insights to the field and pave the way for future research to build upon this foundation.

ACKNOWLEDGMENT

We extend our heartfelt gratitude to everyone involved in the Computer Science program, Faculty of Mathematics and Natural Sciences, Lambung Mangkurat University. The support, resources, and collaboration provided were instrumental in the successful completion of this research. We also appreciate the dedication of our project team members; their contributions greatly enriched the quality and outcomes of this study. The valuable insights and suggestions from our colleagues in the Computer Science Department significantly enhanced our research, reflecting a collective commitment to excellence in this project.

REFERENCES

[1] A. O. Balogun, S. Basri, S. J. Abdulkadir, and A. S. Hashim, "Performance analysis of feature selection methods in software defect prediction: A search method approach," *Applied Sciences (Switzerland)*, vol. 9, no. 13, Jul. 2019, doi: 10.3390/app9132764.

[2] M. A. Mabayoje, A. O. Balogun, H. A. Jibril, J. O. Atoyebi, H. A. Mojeed, and V. E. Adeyemo, "Parameter tuning in KNN for software defect prediction: an empirical analysis," *Jurnal Teknologi dan Sistem Komputer*, vol. 7, no. 4, pp. 121–126, Oct. 2019, doi: 10.14710/jtsiskom.7.4.2019.121-126.

[3] K. K. Bejjanki, J. Gyani, and N. Gugulothu, "Class imbalance reduction (CIR): A novel approach to software defect prediction in the presence of class imbalance," *Symmetry (Basel)*, vol. 12, no. 3, Mar. 2020, doi: 10.3390/sym12030407.

[4] A. Suryadi, "Integration of Feature Selection with Data Level Approach for Software Defect Prediction," *Journal Publications & Informatics Engineering Research*, vol. 4, no. 1, 2019, doi: 10.33395/sinkron.v3i1.10137.

[5] S. Mishra, "Handling Imbalanced Data: SMOTE vs. Random Undersampling," *International Research Journal of Engineering and Technology*, 2017, [Online]. Available: www.irjet.net

[6] I. de Zarzà, J. de Curtò, and C. T. Calafate, "Optimizing Neural Networks for Imbalanced Data," *Electronics (Switzerland)*, vol. 12, no. 12, Jun. 2023, doi: 10.3390/electronics12122674.

[7] C. A. Ramezan, "Transferability of Recursive Feature Elimination (RFE)-Derived Feature Sets for Support Vector Machine Land Cover Classification," *Remote Sens (Basel)*, vol. 14, no. 24, Dec. 2022, doi: 10.3390/rs14246218.

[8] A. O. Balogun *et al.*, "Impact of feature selection methods on the predictive performance of software defect prediction models: An extensive empirical study," *Symmetry (Basel)*, vol. 12, no. 7, Jul. 2020, doi: 10.3390/sym12071147.

[9] F. Wu, Y. Ren, and X. Wang, "Application of Multi-Source Data for Mapping Plantation Based on Random Forest Algorithm in North China," *Remote Sens (Basel)*, vol. 14, no. 19, Oct. 2022, doi: 10.3390/rs14194946.

[10] Y. Zhang, T. Li, Z. Li, Y. M. Wu, and H. Miao, "Software Defects Prediction Based on Hybrid Beetle Antennae Search Algorithm and Artificial Bee Colony Algorithm with Comparison," *Axioms*, vol. 11, no. 7, Jul. 2022, doi: 10.3390/axioms11070305.

[11] Čhulālongkǭnmahāwitthayālai. Khana Witthayāsāt, Mahāwitthayālai Būraphā. Faculty of Informatics, Institute of Electrical and Electronics Engineers, IEEE Thailand Section, and C. Electrical Engineering/Electronics, *2019 JCSSE : the 16th International Joint Conference on Computer Science and Software Engineering : "Knowledge Evolution Towards Singularity of Man-Machine Intelligence" : July 10-12, 2019, Amari Pattaya, Chonburi, Thailand*.

[12] J. J. Tanimu, M. Hamada, M. Hassan, H. A. Kakudi, and J. O. Abiodun, "A Machine Learning Method for Classification of Cervical Cancer," *Electronics (Switzerland)*, vol. 11, no. 3, Feb. 2022, doi: 10.3390/electronics11030463.

[13] Y. Yuan, C. Li, and J. Yang, "An Improved Confounding Effect Model for Software Defect Prediction," *Applied Sciences (Switzerland)*, vol. 13, no. 6, Mar. 2023, doi: 10.3390/app13063459.

[14] A. Ghavidel, P. Pazos, R. Del Aguila Suarez, and A. Atashi, "Predicting the Need for Cardiovascular Surgery: A Comparative Study of Machine Learning Models," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 92–106, Feb. 2024, doi: 10.35882/jeeemi.v6i2.359.

[15] Y. F. Zamzam, T. H. Saragih, R. Herteno, Muliadi, D. T. Nugrahadi, and P. H. Huynh, "Comparison of CatBoost and Random Forest Methods for Lung Cancer Classification using Hyperparameter Tuning Bayesian Optimization-based," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 125–136, Apr. 2024, doi: 10.35882/jeeemi.v6i2.382.

[16] K. Marzuki, L. Ganda Rady Putra, H. Hairani, L. Zazuli Azhar Mardedi, and J. Ximenes Guterres, "Performance Improvement of The Random Forest Method Based on Smote-Tomek Link on Lombok Tourism Analysis Sentiment," *Jurnal Bumigora Information Technology (BITe)*, vol. 5, no. 2, pp. 151–158, 2023, doi: 10.30812/bite/v5i1.3166.

[17] H. Shi, J. Ai, J. Liu, and J. Xu, "Improving Software Defect Prediction in Noisy Imbalanced Datasets," *Applied Sciences (Switzerland)*, vol. 13, no. 18, Sep. 2023, doi: 10.3390/app131810466.

[18] A. J. Mohammed, "Improving Classification Performance for a Novel Imbalanced Medical Dataset using SMOTE Method," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 3, pp. 3161–3172, Jun. 2020, doi: 10.30534/ijatcse/2020/104932020.

[19] M. Alrumaidhi, M. M. G. Farag, and H. A. Rakha, "Comparative Analysis of Parametric and Non-Parametric Data-Driven Models to Predict Road Crash Severity among Elderly Drivers Using Synthetic Resampling Techniques," *Sustainability (Switzerland)*, vol. 15, no. 13, Jul. 2023, doi: 10.3390/su15139878.

[20] Z. Liang, L. Zhang, and X. Wang, "A Novel Intelligent Method for Fault Diagnosis of Steam Turbines Based on T-SNE and XGBoost," *Algorithms*, vol. 16, no. 2, Feb. 2023, doi: 10.3390/a16020098.

[21] N. Anđelić, I. Lorencin, S. Baressi Šegota, and Z. Car, "The Development of Symbolic Expressions for the Detection of Hepatitis C Patients and the Disease Progression from Blood Parameters Using Genetic Programming-Symbolic Classification Algorithm," *Applied*

Sciences (Switzerland), vol. 13, no. 1, Jan. 2023, doi: 10.3390/app13010574.
[22] N. Anđelić, S. Baressi Šegota, I. Lorencin, and M. Glučina, "Detection of Malicious Websites Using Symbolic Classifier," *Future Internet*, vol. 14, no. 12, Dec. 2022, doi: 10.3390/fi14120358.
[23] Angga Maulana Akbar, R. Herteno, S. W. Saputro, M. R. Faisal, and R. A. Nugroho, "Optimizing Software Defect Prediction Models: Integrating Hybrid Grey Wolf and Particle Swarm Optimization for Enhanced Feature Selection with Popular Gradient Boosting Algorithm," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 169–181, Apr. 2024, doi: 10.35882/jeeemi.v6i2.388.
[24] L. Zhang, Y. Liu, J. Zhou, M. Luo, S. Pu, and X. Yang, "An Imbalanced Fault Diagnosis Method Based on TFFO and CNN for Rotating Machinery," *Sensors*, vol. 22, no. 22, Nov. 2022, doi: 10.3390/s22228749.
[25] A. A. Hussin Adam Khatir and M. Bee, "Machine Learning Models and Data-Balancing Techniques for Credit Scoring: What Is the Best Combination?," *Risks*, vol. 10, no. 9, Sep. 2022, doi: 10.3390/risks10090169.
[26] A. M. de Carvalho and R. C. Prati, "DTO-SMOTE: Delaunay tessellation oversampling for imbalanced data sets," *Information (Switzerland)*, vol. 11, no. 12, pp. 1–22, Dec. 2020, doi: 10.3390/info11120557.
[27] S. Rout, P. K. Mallick, A. V. N. Reddy, and S. Kumar, "A Tailored Particle Swarm and Egyptian Vulture Optimization-Based Synthetic Minority-Oversampling Technique for Class Imbalance Problem," *Information (Switzerland)*, vol. 13, no. 8, Aug. 2022, doi: 10.3390/info13080386.
[28] G. Alfian *et al.*, "Deep neural network for predicting diabetic retinopathy from risk factors," *Mathematics*, vol. 8, no. 9, Sep. 2020, doi: 10.3390/math8091620.
[29] N. Zhang *et al.*, "Forest Height Mapping Using Feature Selection and Machine Learning by Integrating Multi-Source Satellite Data in Baoding City, North China," *Remote Sens (Basel)*, vol. 14, no. 18, Sep. 2022, doi: 10.3390/rs14184434.
[30] R. C. Chen, W. E. Manongga, and C. Dewi, "Recursive Feature Elimination for Improving Learning Points on Hand-Sign Recognition," *Future Internet*, vol. 14, no. 12, Dec. 2022, doi: 10.3390/fi14120352.
[31] X. Fan *et al.*, "Sentinel-2 Images Based Modeling of Grassland Above-Ground Biomass Using Random Forest Algorithm: A Case Study on the Tibetan Plateau," *Remote Sens (Basel)*, vol. 14, no. 21, Nov. 2022, doi: 10.3390/rs14215321.
[32] M. A. Kabir, S. Begum, M. U. Ahmed, and A. U. Rehman, "CODE: A Moving-Window-Based Framework for Detecting Concept Drift in Software Defect Prediction," *Symmetry (Basel)*, vol. 14, no. 12, Dec. 2022, doi: 10.3390/sym14122508.
[33] Z. Li, X. Guan, K. Zou, and C. Xu, "Estimation of knee movement from surface emg using random forest with principal component analysis," *Electronics (Switzerland)*, vol. 9, no. 1, Jan. 2020, doi: 10.3390/electronics9010043.
[34] R. De Fazio, R. Di Giovannantonio, E. Bellini, and S. Marrone, "Explainabilty Comparison between Random Forests and Neural Networks—Case Study of Amino Acid Volume Prediction," *Information (Switzerland)*, vol. 14, no. 1, Jan. 2023, doi: 10.3390/info14010021.
[35] N. H. Arif, M. R. Faisal, A. Farmadi, D. T. Nugrahadi, F. Abadi, and U. A. Ahmad, "An Approach to ECG-based Gender Recognition Using Random Forest Algorithm," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 107–115, Apr. 2024, doi: 10.35882/jeeemi.v6i2.363.
[36] Y. Wang, Y. Li, Y. Song, and X. Rong, "Facial expression recognition based on random forest and convolutional neural network," *Information (Switzerland)*, vol. 10, no. 12, Dec. 2019, doi: 10.3390/info10120375.
[37] M. K. Suryadi, R. Herteno, S. W. Saputro, M. R. Faisal, and R. A. Nugroho, "A Comparative Study of Various Hyperparameter Tuning on Random Forest Classification with SMOTE and Feature Selection Using Genetic Algorithm in Software Defect Prediction," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 137–147, Apr. 2024, doi: 10.35882/jeeemi.v6i2.375.
[38] I. Ul Hassan, R. H. Ali, Z. Ul Abideen, T. A. Khan, and R. Kouatly, "Significance of Machine Learning for Detection of Malicious Websites on an Unbalanced Dataset," *Digital*, vol. 2, no. 4, pp. 501–519, Dec. 2022, doi: 10.3390/digital2040027.
[39] A. Alqarni and H. Aljamaan, "Leveraging Ensemble Learning with Generative Adversarial Networks for Imbalanced Software Defects Prediction," *Applied Sciences*, vol. 13, no. 24, p. 13319, Dec. 2023, doi: 10.3390/app132413319.
[40] S. M. A. Shah *et al.*, "An Ensemble Model for Consumer Emotion Prediction Using EEG Signals for Neuromarketing Applications," *Sensors*, vol. 22, no. 24, Dec. 2022, doi: 10.3390/s22249744.
[41] A. Alsaeedi and M. Z. Khan, "Software Defect Prediction Using Supervised Machine Learning and Ensemble Techniques: A Comparative Study," *Journal of Software Engineering and Applications*, vol. 12, no. 05, pp. 85–100, 2019, doi: 10.4236/jsea.2019.125007.
[42] N. Z. Al Habesyah, R. Herteno, F. Indriani, I. Budiman, and D. Kartini, "Sentiment Analysis of TikTok Shop Closure in Indonesia on Twitter Using Supervised Machine Learning," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 2, pp. 148–156, Apr. 2024, doi: 10.35882/jeeemi.v6i2.381.
[43] N. Anđelić, S. Baressi Šegota, I. Lorencin, and Z. Car, "The Development of Symbolic Expressions for Fire Detection with Symbolic Classifier Using Sensor Fusion Data," *Sensors*, vol. 23, no. 1, Jan. 2023, doi: 10.3390/s23010169.

BIBLIOGRAPHY

Helma Ghinaya originated from Amuntai, a city located in South Kalimantan, Indonesia. Since 2020, she was been involved in the academic world as a student in the Department of Computer Science at Lambung Mangkurat University, the Faculty of Mathematics and Natural Sciences of Lambung Mangkurat University. Her current area of research lies within the realm of software engineering. She has selected this particular interest due to her affinity towards software engineering. In addition, her thesis requires her to undertake research to improve the accuracy and AUC-ROC of SDP predictions and better understand the important features in software defect detection.
Email : 2011016320011@mhs.ulm.ac.id.

Rudy Herteno was born in Banjarmasin, South Kalimantan. After graduating from high school, he pursued his undergraduate studies in the Computer Science Department at Lambung Mangkurat University and graduated in 2011. After completing his undergraduate program, he worked as a software developer to gather experience for several years. He developed a lot of software, especially for local governments. In 2017, He completed his master's degree in Informatics from STMIK Amikom University. Currently, he is a lecturer in the Faculty of Mathematics and Natural Science at Lambung Mangkurat University. His research interests include software engineering, software defect prediction, and deep learning.
Email: rudy.herteno@ulm.ac.id.

Mohammad Reza Faisal was born in Banjarmasin. Following his graduation from high school, he pursued his undergraduate studies in the Informatics department at Pasundan University in 1995, and later majored in Physics at Bandung Institute of Technology in 1997. After completing his bachelor's program, he gained experience as a training trainer in the field of information technology and software development. Since 2008, he has been a lecturer in computer science at Universitas Lambung Mangkurat, while also pursuing his master's program in Informatics at Bandung Institute of Technology in 2010. In 2015, he furthered his education by pursuing a doctoral degree in Bioinformatics at Kanazawa University, Japan. To this day, he continues his work as a lecturer in Computer Science at Universitas Lambung Mangakurat.
Email: reza.faisal@ulm.ac.id.

Andi Farmadi serves as a faculty member within the Department of Computer Science at Lambung Mangkurat University. His research interest is centered around the field of Data Science. Prior to assuming his role as a lecturer, he successfully obtained his bachelor's degree in Physics from Hasanuddin University in 1999, followed by the completion of his master's degree at Bandung Institute of Technology in 2007. It was in 2008 that he commenced his tenure as a lecturer within the Department of Computer Science at Lambung Mangkurat University. The primary focus of his research endeavors lies within the realm of Data Science.

Fatma Indriani is a lecturer in the Department of Computer Science, Lambung Mangkurat University. She completed her undergraduate program in the Informatics Department at Bandung Institute of Technology. In 2008, she started working as a lecturer in the Computer Science department at Lambung Mangkurat University. She then obtained her master's degree at Monash University, Australia (2012) and a doctorate degree in Bioinformatics at Kanazawa University, Japan (2022). Her research interest is applied Data Science.