Genetic algorithm-based oversampling approach to prune the class imbalance issue in software defect prediction

C. Arun1 • C. Lakshmi1

Accepted: 2 August 2021 / Published online: 29 August 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract
Class imbalance is the potential problem that has been existent in machine learning, which hinders the performance of the classification algorithm when applied in real-world applications such as electricity pilferage, fraudulent transactions, anomaly detection, and prediction of rare diseases. Class imbalance refers to the problem where the distribution of the sample is skewed or biased toward one particular class. Due to its intrinsic nature the software fault prediction dataset falls into the same category where the software modules contain fewer defective modules compared to the non-defective modules. The majority of the oversampling techniques that has been proposed is to address the issue by generating synthetic samples of minority class to balance the dataset. But the synthetic samples generated are near duplicates that also results in over-generalization issue. We thus propose a novel oversampling approach to introduce synthetic samples using genetic algorithm (GA). GA is a form of evolutionary algorithm that employs biologically inspired techniques such as inheritance, mutation, selection, and crossover. The proposed algorithm generates synthetic sample of minority class based on the distribution measure and ensures that the samples are diverse within the class and are efficient. The proposed oversampling algorithm has been compared with SMOTE, BSMOTE, ADASYN, random oversampling, MAHAKIL, and no sampling approach with 20 defect prediction datasets from the promise repository and five prediction models. The results indicate that the genetic algorithm oversampling approach improves the fault prediction performance and reduced false alarm rate.

Keywords Class imbalance • Software fault prediction • Synthetic samples • Generating samples of minority class • Oversampling techniques • Genetic algorithm • False alarm rate • Evolutionary algorithm

1 Introduction
Software defect prediction (SDP) is one of the predominant happenings of the testing phase in Software Development Life Cycle (Rathore and Kumar 2019; Song et al. 2011). The main task of SDP is to identify the defect-prone modules that call for extensive testing (Wang and Yao 2013; Menzies et al. 2007). A defect is an error, bug, fault, flaw, or malfunction of the software component which results in an incorrect behavior or an unexpected outcome. Figure 1 illustrates the basic machine learning framework for SDP. SDP provides the list of artifacts that assist the development team to focus on, prioritize, and allocate the resources for testing the modules toward those artifacts. SDP models aim to classify fault-prone modules of the software component which helps in focusing and diverting the efforts of testing activity toward the faulty components which in turn increase the quality of the software components by more focused and prioritized testing activity. SDP helps are obtaining better quality product with optimized cost and efforts, since faulty modules are identified prior to the testing activity. The accuracy of the SDP models has always been hindered by the inherent nature of the dataset (Gray et al. 2011; Bennin et al. 2016). The performance of the prediction depends on the quality of the data given for training, and thus, the dataset needs to be balanced. The
intrinsic nature of the software defect dataset is imbalanced (Japkowicz 2000; Arun and Lakshmi 2020; Pai and Dugan 2007; Vandecruys et al. 2008; He and Garcia 2009; Sivaganesan 2021; Smys et al. 2020; Manoharan 2021) in which the number of non-defective modules is quite smaller (Wang and Yao 2013; Sun et al. 2012; Fenton and Ohlsson 2000) than the defective modules. Prediction model trained on the imbalanced dataset is biased or skewed toward the majority class samples (Provost 2000; Weiss and Provost 2001; Yoon and Kwek 2007), and hence, the defective modules are prone to be labeled as a non-defective one.

The prevailing techniques to solve the imbalance problem is to use datasampling approaches (Rathore and Kumar 2019; Ebo Bennin et al. 2017), which includes both over- and under-sampling techniques (Peters et al. 2013). The objective of the sampling approach is to create a balanced dataset, so the classifiers perform better. Oversampling approaches increase the sample space of minority class by introducing synthetic samples (Chawla et al. 2002; He et al. 2008; Han et al. 2005; Barua et al. 2014), whereas under-sampling approach reduces the sample space of the majority class by randomly selecting a part of the original samples. Though the sampling methods improve the training set to create more balanced distribution, they undergo limitations as well. Random oversampling (ROS) approach increases the count of minority samples by randomly replicating it, which results in overfitting due to replication of original samples (Seiffert et al. 2008). Random under-sampling (RUS), the majority class instances are discarded at random until the desired balanced attained. While in ROS vast amount of information is lost due to discarding of samples, the RUS results in overfitting of the classifier, making the generalization performance of the classifier extremely meager (Rathore and Kumar 2019).

Chawla et al. proposed an oversampling approach called synthetic minority oversampling technique (SMOTE) (Chawla et al. 2002; Pak et al. 2018), which introduce synthetic samples toward balancing the dataset which results in improved performance on the prediction models. Han et al. Borderline-SMOTE (Han et al. 2005) model enhances SMOTE which concentrates primarily on the samples along the boundary for synthetic sample generation (Chawla et al. 2003). He et al. coined a novel approach ADASYN (He et al. 2008) which generated synthetic samples based on Euclidean distance and its density distribution. On the other hand, oversampling techniques tend to balance the dataset by generating synthetic samples, which may fall out of the minority boundary due to the generation of noisy minority class samples, which effects in enlargement of decision boundary (Barua et al. 2014). An inaccurately inflated region is likely to overlap with the majority class region that poses a difficulty toward classification. Despite the improvement in the accuracy of classification the above-discussed techniques tend to produce high false alarm rate.

We thus propose a novel sampling technique which provides improvisation both in terms of accuracy and reduced false alarm rate. The recommended oversampling technique generates synthetic samples based on the distribution-based measure that ensures samples are highly diverse and reside within the region of minority sample which results in eliminating over-generalization and reduced false alarm rate. Proposed oversampling techniques are inspired based on the phenomenon of chromosomal theory of inheritance which ensure the generated synthetic sample is diverse because of the diversity measure, i.e., Mahalanobis distance (MD). Ebo Bennin et al. (2017) concluded that synthetic samples generated using similar samples reduce the possibility of overfitting. The proposed methodology eliminates over-generalization by generating diverse sample, overfitting by generating synthetic samples from dissimilar parent, improved classification accuracy and lower false alarm rate.

2 Motivation

Chawla et al. SMOTE (Chawla et al. 2002) generates synthetic instances of minority class by selecting a minority class instance at random, finding its nearest neighbors and choosing one of its nearest neighbors by connecting a line segment of those two randomly chosen points as a convex combination. Though SMOTE improves the prediction performance, it also introduces certain bias toward the minority class.

Han et al. developed a method for creating synthetic samples called Borderline-SMOTE (BSMOTE) (Han et al. 2005) which creates synthetic samples using the samples along the decision boundary. BSMOTE generates synthetic sample by finding nearest neighbors for all minority samples. Based on the neighbors if all are from the majority
class, then the sample is considered as noisy and eliminated, if near equal to that of minority class, then the sample is considered to be safe and introduced to the synthetic sample generation by computing the difference between itself and one of its random neighbors and multiply the difference by a random number.

The above-discussed oversampling methods are based on the k-nearest neighbor (Chawla et al. 2002; He et al. 2008; Han et al. 2005) technique to select the nearest samples for oversampling. For each instance of minority class, the samples which are of close proximity to the dataset are in consideration based on the distance measures. However, selecting the nearest or closest sample imposes challenge of generating diverse samples, since only the nearest neighbors are used for oversampling which introduce mere duplicate samples and does not provide any new information which hinders the classification task. The techniques are similar to the ROS approach, which duplicates the minority samples in random fashion until the set is balanced to the desired level.

Synthetic samples introduced by the oversampling techniques widened the boundary of the minority samples which leads to the impact of over-generalization (Wong et al. 2013). Figure 2a illustrates the ratio of minority to majority samples in a class imbalance dataset. Figure 2b illustrates a smaller portion of minority samples clustered around the center before performing oversampling on the minority samples. The KNN-based approach primarily focuses on the data points that are closest to the sample in consideration which results in near duplication and results in increasing the number of sub-clusters in the minority classes due to sparsely generated synthetic samples which is evident from Fig. 2c–f. The minority samples are oversampled using SMOTE, ROS, BSMOTE, and ADASYN which results in more dispersed samples and some of them may fall outside the existing minority boundary.

Chyi (2003) such as nearest, farthest, average nearest, average farthest distance from majority sample as training and include in the prediction process. This approach primarily focuses on majority sample which proved to ineffective in case of real-time applications. Yen and Lee (2009) proposed an under-sampling approaches for skewed datasets which groups the dataset samples in different cluster and compute the ratio of imbalance existing in the cluster, based on which compute the number of majority samples selected from each cluster and combine with minority class sample to form the training data. Even though has better classification accuracy the system results higher false alarm rate. Zhang and Wang (2011) proposed a distribution-based re-sampling approach called D-SMOTE which generated synthetic sample based on the cluster density and form the training dataset by combining the synthetic samples with the actual dataset and performs the prediction. The performance of the model is explored on with the credit card dataset.

The proposed methodology selects samples of minority class from different clusters for synthetic sample generation which ensure the new samples lie within the existing cluster boundary and avoid over-generalization problem.

3 Theory of inheritance and measure of diversity

Sutton (1903) proposed the theory of inheritance which states the new offspring inherit its traits from both parents in equal quantity, i.e., 50 percentage of traits from each parent which ensure the offspring has similarities toward both parents at the same time maintaining unique property. To uphold diversity within classes, the sex (gender) of the species is of great prominence because it helps in selecting two opposite members that can reproduce. The large number of population-based heuristics algorithm has been proposed by researchers among which genetic algorithm has proved to be a successful one (David 1997; Jong and Spears 1989), which is inspired from the biological evolution process. Genetic algorithm is an evolutionary algorithm inspired from the biological evolution operators such as inheritance, recombination, selection, and mutation (David 1997; Jong and Spears 1989).

GA is a population-based search algorithm based on theory of survival of fittest (David 1997). The genetic algorithm is a computational simulation in which a population of abstract representations of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem (called chromosomes, genotypes, or genomes) evolves toward better solutions (David 1997; Jong and Spears 1989). The progression starts from a population of random selected individuals called as population. Two individuals are selected as parents from the
population which undergo biological operations such as mutation, crossover to produce the new offspring based on the fitness evaluations, the offspring is added to population for future generations. The evolution process is iterative, where the fittest offspring propagates to the next generations until reaching the maximum generation reached or stopping criteria reached. Algorithm1 provides an overview of basic genetic algorithm to generate the offspring.
Algorithm 1: Pseudo code of Genetic Algorithm to generate the offspring:

1. Initialize the population by random selection
2. Evaluate the Populations
3. Genetic Algo Operations
   while max_generation not reached:
   1. GA_S = Selection (Population)
   2. GA_C = Crossover (GA_S)
   3. GA_M = Mutation (GA_C)
   4. Goffs = Evaluate (GA_M)
   5. Gpop = Survival (Gpop, Goffs)
4. end

The above pseudocode illustrates the generic evolution process of creating new offspring. The idea of evolution starts from the possible set of solutions, i.e., creating an initial population for the sampling process incorporating any random selection method. From the selected population, fitted individuals from the population that yield better offspring are to be selected according to their fitness out of the population. On selecting the parent genetic operations are applied to generate offspring’s, which include various steps like selection, crossover, and mutation. During the process of crossover, the selected portion of the gene from two different parents will be combined to form a new offspring. If the offspring that does not satisfy the fitness property is further mutated to yield a new offspring for the next generation. It iterates that through all generations and in every generation, the best child is selected for the propagation of next generation by evaluating the fitness function. The algorithm generally stops either on reaching the maximum generation or by satisfying the fitness level of some specified stopping criteria.

The principle of evolutionary algorithm can be applied to the imbalanced defect dataset to generate new instance of faulty samples. The oversampling algorithm generates offspring based on the nearest neighbor which results in less diverse samples. Prof. P. C. Mahalanobis introduced Mahalanobis distance (MD) which measures the distance between a point and the distribution (Mahalanobis 1936). Euclidean distance usually measures the distance between two points in a straight line which results in a new sample of same as the original one which is less in diverse. Based on theory of inheritance and MD measure new synthetic samples are introduced to the population which is more diverse and reside within the region on minority boundary. In view of retaining all the information from the minority samples even after offspring is generated, the parents involved in the generations are retained in the population.

4 Proposed methodology: genetic algorithm-based oversampling approach for imbalance dataset

4.1 Overview

Inspired from the theory of inheritance proposed by Walter, the feature vectors of the defective samples i.e different metrics calculated for the software components are considered as Chromosomes (Sutton 1903). Also, the existing oversampling approach majorly depends on the distance metric which uses techniques such as Euclidian, Manhattan, Minkowski distance which merely compute how close the two different points, when those two samples involved the sampling process results in at most duplicate sample which adversely increase over-generalization. The proposed approach uses diversity-based measure called MD which compute the measure of diversity of sample from population and then rank the samples according to the measure and grouped into clusters with diversity. Using genetic algorithm parents are chosen from the cluster involved in the sampling process which results in more diverse sample. We strive to generate synthetic samples that takes its trait from two different instances that has been selected from the population. To alleviate the over-generalization, issue the samples that are apart from each other is considered for generating synthetic samples and is evaluated for the fitness. Only the fittest are involved in the process after ensuring that generated samples reside inside the boundary of minority samples. The oversampling approach involves three phases.
In the first phase the minority samples are separated from majority ones, then discrimination measure for the individual minority samples is calculated against the minority distribution based on MD. The second phase involves partitioning the datasets of minority class into two different clusters based on the MD. The third phase involves the generation of synthetic samples based on the genetic algorithm-based approach using operators such as crossover, selection, and fitness value.

4.1.1 Phase 1

Most of the oversampling approach uses Euclidean metric to compute the synthetic samples which usually compute the shortest distance between two given points that’s why those models lack in diversity of sample. Also, defect dataset prone to contain noise or duplicate data which will not recognize by the Euclidean measure. To counter all these issues MD uses the correlation measure to compute the diversity of samples from the distribution. MD’s effectiveness has also been demonstrated in a variety of applications, including determining whether a sample is an outlier, whether a method is in control, and whether a sample is a member of a community or not. To ensure the diversity of the sample, we have computed the MD for all the minority samples and ranked them according to the computed results. The minority samples from the dataset...
are segregated from the majority and MD is computed for all the vectors of minority samples using Eq. 1.

$$\Delta 2 = (x - \mu)T \ast \sum^{-1}(x - \mu)$$  \hspace{1cm} (1)

where $x$ is a row vector consisting of the feature vector, $\mu$ is the mean of the class, and $\sum^{-1}$ is the covariance matrix of the sample.

The computed MD is appended to the corresponding row vector and the entire minority samples are sorted based on the MD which represented in the algorithm 2 from steps 2 to 6.

### 4.1.2 Phase 2

The midpoint of the MD is identified and the data points are divided into two clusters $P_1$ and $P_2$ where $P_1$ will contain the sample which had values till midpoint and cluster $P_2$ will contain the data points whose MD values are greater than midpoint. The row vector in each cluster is labeled sequentially in both the clusters which primarily form the population for generation process. The first row vector from $P_1$ is paired with first row vector from $P_2$ which forms as first set of parents, similarly all the vectors with same label are paired as parent which is expressed in algorithm 2 in step 7 to step 9. Since the parents are chosen from different clusters, distance apart will eliminate the problem of duplication which occurs in Euclidean distance. Also choosing parents from different clusters ensures the generated child resides within the minority boundary.

### 4.1.3 Phase 3

Selected samples are provided as input for the genetic algorithm which returns the synthetic samples. Parent $p_{1i}$ is selected from the cluster $P_1$ and similarly labeled parent $p_{2i}$ from cluster $P_2$ is selected from the initial population. The selected parent undergoes crossover operation using multipoint crossover. Crossover operator yields a new offspring from existing parents in the mating pool by applying the selection operator where it exchanges the gene information from the parents yield in two new offspring. Here we exchange multiple such sections of the parent1 with parent2 yielding two different children. After Child1 and Child2 have been added to the distribution to compute the MD for the Child1 ($d1$) and Child2 ($d2$), the distance of the child is verified to be resided within the minority boundary and if the distance value is less than the midpoint it will be added to Cluster $P_1$ otherwise added to Cluster $P_2$. Eqs. 2 and 3, if the values are large than the highest distance of $P_2$ then the Child will be discarded. Before child added to the cluster the fitness of the child is assessed using fitness function $f$ based on the performance measure (pf and pd) and also the convex property using Eq. 4, only child which are fittest are added to the respective cluster for future generations. Based on the percentage of balancing the number of synthetic samples to be generated is computed and process iteratively repeated for all the pair of the parents and keep iterating till the sufficient balance is attained.

### 5 Experiment methodology

The objective of this oversampling approach is to improve the performance of fault prediction performance by reducing the false alarm rate on imbalanced datasets. The comparative analysis of the proposed method performance with the existing oversampling approach is carried out on 20 different benchmark datasets (NASA Data Repository, http://mdp.ivv.nasa.gov; PROMISE Data Repository, http://openscience.us/repo/; Software Defect Datasets, https://ieee-dataport.org/) with different oversampling approaches such as SMOTE (Chawla et al. 2002), ROS (Seiffert et al. 2008), BSMOTE (Han et al. 2005), ADA-SYN (He et al. 2008), and no sampling approach. Oversampled data performance has been assessed with various standard predictors in machine learning and the performance is evaluated using statistical measures.

#### 5.1 Benchmark datasets

The experiment has been conducted on 20 different benchmark software fault datasets (NASA Data Repository, http://mdp.ivv.nasa.gov; PROMISE Data Repository, http://openscience.us/repo/; Software Defect Datasets, https://ieee-dataport.org/) that are extracted from the promise repository. Promise Software Engineering Repository consists of freely available datasets and tools which enable the researchers to carry out research predictive software modules (Shirabad and Menzies 2005; Vandecruys et al. 2008; Sivaganesan 2021). The characteristics of the software components is a measure using various levels of metrics such as code metrics, process metrics, CK metrics, representative process metrics, and so on. Metrics are the quality indicators which define the quality of a software component and the process used (Chidamber et al. 1998; Chidamber and Kememer 1994; Li and Henry 1993, 1996; Basili et al. 1996). Dataset contains the value for these
software metrics which is calculated across class, modules code metrics (Table 1) of the projects that are written in Java. Table 1 provides the description of the various static code metrics that represented in the datasets. The bug represents the status of defects, if it contains the defect the numbers of defect found in the module are mentioned else zero is mentioned. A summary of the datasets (Table 2) includes the name and its release version, number of modules, number of non-defect, number of defects and percentage of defective samples.

5.2 Inception

The proposed oversampling approach using genetic algorithm-based approach is compared extensively with five other oversampling approaches and no sampling approach (None).

5.2.1 SMOTE

Chawla et al. proposed SMOTE (He et al. 2008), which oversamples the minority class by creating new synthetic samples in order to balance the dataset based on the given pfp value. SMOTE introduces the synthetic samples by selecting the samples that are close enough in the sample space using the nearest neighbor approach, drawing a line between the samples in the sample space and drawing a new point along the same line. The sample space of minority which is close to the majority sample cluster will result in introduction of noisy samples will increase overfitting.

5.2.2 Borderline-SMOTE (BSMOTE)

Han et al. proposed a new oversampling approach as an extension to the SMOTE called as BSMOTE (Han et al. 2005) which also works on the basis on the nearest neighbor but consider only the sample which lies in the decision boundary. Primary focus BSMOTE lies on the samples that are harder to classify which are identified as danger samples. BSMOTE classifies the minority sample and identifies the neighbor for each of it. The sample of minority class has the majority of neighbor from majority class, then those samples are not included for synthetic process. Sample which has both majority and minority samples will consider for the synthetic sample generation using the nearest neighbor which aids in strengthening the borderline.

5.2.3 ADASYN

He et al. proposed ADASYN (He et al. 2008) oversampling approach based on the distribution of samples measure. ADASYN considers the density distribution, which computes the number of synthetic samples needs to generate for each minority sample by finding the nearest neighbor by

| Table 1 Details of static code metrics | Abbreviation | Description |
|----------------------------------------|--------------|-------------|
| WMC                                    | Weighted method per class |
| DIT                                    | Depth of inheritance tree |
| NOC                                    | Number of children |
| CBO                                    | Coupling between objects classes |
| RFC                                    | Response for a class |
| LCOM                                   | Lack of cohesion in methods |
| CA                                     | Afferent coupling |
| CE                                     | Efferent coupling |
| NPM                                    | Number of public methods |
| LCOM3                                  | Lack of cohesion in methods, differ from LCOM |
| LOC                                    | Lines of code |
| DAM                                    | Data access metrics |
| MOA                                    | Measure of aggregation |
| MFA                                    | Measure of functional abstraction |
| CAM                                    | Cohesion among methods of class |
| IC                                     | Inheritance coupling |
| CBM                                    | Coupling between methods |
| AMC                                    | Average methods complexity |
| MAX_CC                                 | Maximum value of CC methods of investigate class |
| AVG_CC                                 | Arithmetic mean of the CC value in the investigate class |
| DEFECTS                                | Counts of bugs detected in the class |
computing the Euclidean distance. If the sample has more number of majority, then more synthetic sample will be generated for it. New synthetic sample is computed by adding weight to the sample which is multiple of vector difference in sample space and random value.

5.2.4 ROS

Random oversampling (ROS) (Seiffert et al. 2008) technique balances the dataset by introducing random sample of minority class by duplicating, i.e., replicating the samples of minority class. ROS will suffer from overfitting due to the duplication of existing sample which results in no new information.

**Table 2** Summary of the 20 imbalanced datasets from promise repository

| S. No | Dataset | # Modules | # defective | # n-defective | Defects (%) |
|-------|---------|-----------|-------------|---------------|-------------|
| 1     | ant-1.3 | 125       | 20          | 105           | 0.160       |
| 2     | ant-1.4 | 178       | 40          | 138           | 0.225       |
| 3     | ant-1.5 | 293       | 32          | 261           | 0.109       |
| 4     | ant-1.6 | 351       | 92          | 259           | 0.262       |
| 5     | arc     | 234       | 27          | 207           | 0.115       |
| 6     | camel-1.4 | 875       | 145         | 730           | 0.166       |
| 7     | camel-1.6 | 965       | 188         | 777           | 0.195       |
| 8     | ivy-1.4 | 241       | 16          | 225           | 0.066       |
| 9     | ivy-2.0 | 352       | 40          | 312           | 0.114       |
| 10    | jedit-4.0 | 306       | 75          | 231           | 0.245       |
| 11    | jedit-4.1 | 312       | 79          | 233           | 0.253       |
| 12    | jedit-4.2 | 367       | 48          | 319           | 0.131       |
| 13    | log4j-1.0 | 135       | 34          | 101           | 0.252       |
| 14    | pbeans2 | 51        | 10          | 41            | 0.196       |
| 15    | readktor | 176       | 27          | 149           | 0.153       |
| 16    | Synapse-1.0 | 157     | 16          | 141           | 0.102       |
| 17    | systemdata | 65       | 9           | 56            | 0.138       |
| 18    | tomcat | 858       | 77          | 781           | 0.090       |
| 19    | xerces-1.2 | 440       | 71          | 369           | 0.161       |
| 20    | xerces-1.3 | 453       | 69          | 384           | 0.152       |

**Table 3** Confusion matrix

| Predicted positive | Predicted negative |
|--------------------|--------------------|
| Positive           | TP                 | FN       |
| Negative           | FP                 | TN       |

**Fig. 3** Experimental framework
5.2.5 MAHAKIL

Ebo Bennin et al. proposed MAHAKIL (Ebo Bennin et al. 2017) approach which is based on the theory of inheritance which introduce synthetic samples using diversity-based measure called MD. P. C. Mahalanobis proposed the distance metric which computes the diversity of the sample toward its distribution. The minority samples are separated from the majority and the MD is calculated for each minority sample and the dataset is divided into two bins ($c_1, c_2$) based on the distance value. $c_1$ contains the samples with value less than the midpoint and $c_2$ contains the sample with value greater than midpoint and the sample in each bin is labeled. The synthetic samples are introduced by computing the average of the samples with same label in each bin. The process repeated till the required balance attained.

5.3 Experimental setup

The performance on the proposed methodology has been evaluated with five classification algorithms which has been already tested with imbalanced datasets (Ma et al. 2006). Classification methods such as SVM (Xing et al. 2005), C4.5 decision tree (Quinlan 2014), Naive Bayesian (Pai and Dugan 2007; He and Garcia 2009), and random forest (Lessmann et al. 2008; Puntumapon et al. 2016; Rodriguez et al. 2007) model utilized to evaluate the performance of the proposed oversampling technique. Random forest (ensemble) has outperformed other classification method which is already proven by Lessmann et al. (2008), Breiman (2001), Liaw and Wiener (2002). Other classification models such as KNN and NB also used by Yan-Ping et al. used on imbalanced datasets.

The dataset obtained from the repository might contain noisy or missing values which will be eliminated by applying preprocessing technique. Preprocessed data and the pfp (balancing percentage) are fed into the oversampling approach which produces the balanced dataset. The oversampling approaches using different prediction models based on false alarm rate. e Performance comparison of oversampling approaches using different prediction models based on each pfp levels.

![Fig. 4](image1)
![Fig. 5](image2)
Genetic algorithm-based oversampling approach to prune the class imbalance issue in software...

(a) Comparison on the performance Random Forest model on different over sampling tecl

(b) Comparison on the performance Support Vector Machine model on different over sampling techniques
(c) Comparison on the performance C4.5 model on different over sampling techniques

(d) Comparison on the performance KNN model on different over sampling techniques

Fig. 5 continued
sampled dataset is divided into 2/3 of the sample size as training and 1/3 of the sample size as testing data. The division ensures the same percentage of balance in both testing and training data. Figure 3 shows the steps followed for the evaluation of GA-based oversampling approach using various machine learning models. To reduce the effect of bias the whole experiment is repeated for tenfold and the values are average across multiple runs. The experiments were conducted on each of the 20 datasets after re-sampling with each oversampling technique at different pfp levels. Since the highest level of balancing among the experiment dataset is being 26 percentage, we conduct experiments with three pfp values as 30, 40, and 50 on all sampling approaches, since all the datasets have balance percentage less than 30 percentile and the maximum balance is 50 percentage. The experiments were conducted using Python and the library package imblearn.over_sampling for the sampling approaches and sklearn for the machine learning models.

5.4 Performance evaluation

Defect dataset is a two-class problem where the class labels of minority, i.e., faulty, are labeled as positive and the majority are negative. Table 3 presents the confusion matrix where the first row denotes the numbers of positive (TP) and negative (TN) samples that are correctly classified and the second row denotes the number of misclassified positive (FP) and the negative (TN) instances.

Numeric performance evaluation measures such as accuracy, specificity, f-measure, G-means, false alarm rate, recall, and precision have been used to evaluate performance. Accuracy provides the probability of faulty modules is classified correctly. Recall (pd) is the measure tells the module contained fault is classified correctly. False alarm rate (pf) is the measure where non-faulty modules are wrongly classified as faulty one which ultimately over-compensate the process of fault prediction (Buckland and Gey 1994; Joshi et al. 2001). Lessmann et al. found that relying on accuracy indicators is ineffective for fault prediction (Basili et al. 1996). Based on the study, we observed that the recall and false alarm rate are the ideal metrics for the SDP. The objective of the fault prediction is to reduce the false alarm rate, and hence pd, pf is the ideal measure to evaluate the performance which are defined in equations (Eq. 5) and (Eq. 6).

Recall (pd) = TP/(TP + FN)  

Fig. 5 continued
False Alarm (pf) = FP/(TN + FP) 

5.5 Performance comparison

To compare the performance of the proposed method against other oversampling approaches using different machine learning models, we adopt the win–tie–loss statistics proposed by Kocaguneli et al. (2013, 2012) and Brunner et al. (2002). Predictor output (model * sampling techniques) for all possible pairs of predictors in a pairwise manner is compared, a tie is declared if the results are not substantially different. If one of the predictors have higher value, it is counted as win for the one with higher values and loss for the lower one. Finally, all the performance of the respective pfP level is calculated to estimate the best predictor.
6 Empirical results

The performance of the proposed methodology and other sampling approaches on the prediction models are evaluated against each percentage of balancing (pfp). The performance comparison of the proposed oversampling approach using various prediction models is evaluated using the following metrics accuracy, recall, and false alarm rate. It is evident from Fig. 4a–c that the random forest model overshadowed all other prediction models in all aspects. Also, the proposed oversampling performance has been compared against other oversampling approach with various rates of balancing, where the proposed methodology has performed better compared to other models in terms of false alarm rate which is presented in Fig. 4d.

From Fig. 5a–e, we observed the following

1) The proposed oversampling technique and the MAHAKIL provide better performance in terms of false alarm rate, while ROS and ADASYN performance are least
2) The proposed method with random forest and c4.5 outperforms other oversampling approaches in comparison with other prediction models
3) The performance increases with the amount of balance attained, i.e., the best performance has obtained when the dataset is balanced with pfp rate of 50%
4) It is observed that proposed oversampling approach provides an average false alarm rate over 5 to 10 percentage in all the prediction model which better performs in SMOTE, BSMOTE and ADASYN techniques.

6.1 Statistical comparison

The performance of the proposed oversampling approach has been examined in order to prove its effectiveness, where the proposed methodology performs better in case of reducing the false alarm rate and results in improved performance accuracy measure. The comparison analysis is made use of Brunner’s statistical method of win–tie–loss ratio results of the proposed method against those of SMOTE, Borderline-SMOTE, ADASYN, MAHAKIL, and ROS per each value of balancing for performance measures of recall and false alarm rate across all the 20 benchmark datasets. From Fig. 5, we observed that proposed method performs better compared to all other sampling techniques, with the pfp increasing from 30 to 50 in case of false alarm rate measure. The detailed comparisons using the statistical technique of the proposed method against other oversampling techniques using various prediction models are presented in Table 4.

7 Discussion

7.1 External validity

In this work, we have used benchmark datasets obtained from the promise repository. We assume that the data belong to both classes are segregated by a clear boundary. But in the real word the segregation is a threat, since there could be many small clusters in some of the datasets, which ideally increase the possibility of overfitting. This could be avoided, if data are divided into multiple small clusters, work in local blotches will improve the performance. Also, the work carried out across only limited set of software metrics. The examination of the performance using other dataset and metrics is left to future works. The effect of the proposed methodology on cross-project prediction will be left for future study (Turhan et al. 2009).

7.2 Internal validity

We have used Mahalanobis distance to compute the diversity of the sample against the distribution, where the computation of covariance matrix is high cost when the dataset is large and featured set is also large. Also, if the number of instances in the minority class is very low and highly correlated then finding the covariance matrix makes the task difficult [base] which can be overcome by the generalized inverse method and the impact of the generalized method is left to future studies.

8 Future work

The objective of the work is to improve the prediction performance on an imbalance dataset by introducing synthetic samples of minority class in view of balancing the dataset and thereby reducing the false alarm rate. Our approach outperforms other oversampling techniques with respect to the recall, accuracy, and false alarm rate. The oversampling approach implemented in the work considers all the datapoints of minority class as a single cluster. In reality the minority samples are outspread and are a challenge to group it into a single cluster. Additionally, some majority sample might overlap the cluster region which increases the false alarm rate. Further different sampling approaches can be adopted which yields better performances. Most of the software prediction models were evaluated with only historical data, in case of new project where the dataset is not readily available can be tested...
using cross project or company prediction which is left to future study. To further reduce the pf rate and provide better performance the dataset can be partitioned into smaller cluster and synthesis of minority samples can be done in local patches which can be extended in future works.

9 Conclusion

Most real-world SDP datasets are highly imbalance in nature, sampling approaches which include both over- and under-sampling approaches have been proposed to create a balanced dataset. Sampling approaches create synthetic samples in view of balancing the dataset, which results in increased accuracy, on the other hand increase the false alarm rate due to generation of mere duplicate data erroneously. Class imbalance is a threat to the learning model which lower the performance and results in overfitting and high false alarm rate. Most of the SDP models built using sampling techniques which includes oversampling, undersampling, cost-sensitive have been proposed to tackle the class imbalance issue within the project. The majority of the existing oversampling approaches are based on the Euclidean metrics which result in duplication of samples which results in over-generalization problem. By taking these challenges into account, the proposed methodology generates more diverse samples of minority class by using diversity-based measure and evolutionary approach which ensures the synthesized sample is diverse in nature also reside in the cluster boundary and eliminate the possible outliers using distribution measure. Performance of the proposed methodology has been evaluated by comparing with other five oversampling approaches using five prediction models and observed that the proposed method outperforms other models in terms of reduced false alarm rate and better recall value. Further research is required to improve upon the GA to reduce the computation time by finding variations of crossover and mutation. Also, the synthesis of minority samples can be catered using multi-cluster approach and generate sample in local patches.

Compliance with ethical standards

Conflicts of interest All authors declare that they have no conflicts of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Arun C, Lakshmi C (2020) Class imbalance in software fault prediction data set. In: Artificial intelligence and evolutionary computations in engineering systems. Springer, pp 745–757
Barua S, Islam MM, Yao X, Murase K (2014) MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning. IEEE Trans Knowl Data Eng 26(2):405–425
Basili VR, Briand LC, Melo WL (1996) A validation of object-oriented design metrics as quality indicators. IEEE Trans Softw Eng 22(10):751–761
Bennin KE, Keung J, Monden A, Kamei Y, Ubayashi N (2016) Investigating the effects of balanced training and testing datasets on effort-aware fault prediction models. In: Proc IEEE 40th annu comput softw appl conf, pp 154–163
Breiman L (2001) Random forests. Mach Learn 45(1):5–32
Brunner E, Munzel U, Puri ML (2002) The multivariate nonparametric Behrens–Fisher problem. J Stat Plan Inference 108(1):37–53
Buckland MK, Gey FC (1994) The relationship between recall and precision. J Am Soc Inform Sci 45(1):12–19
Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) Smote: synthetic minority over-sampling technique. J Artif Intell Res 16:321–357
Chawla NV, Lazarevic A, Hall AO, Bowyer KW (2003) SMOTE-Boost: improving prediction of the minority class in boosting. In: European conference on principles of data mining and knowledge discovery
Chidamber S, Kemerer C (1994) Ametrics suite for object-oriented design. IEEE Trans Softw Eng 20(6):476–493
Chidamber S, Darcy D, Kemerer C (1998) Managerial use of metrics for object oriented software: an exploratory analysis. IEEE Trans Softw Eng 24(8):629–639
Chyi Y-M (2003) Classification analysis techniques for skewed class distribution problems. Master thesis, Department of Information Management, National Sun Yat-Sen University
Cover T, Hart P (1967) Nearest neighbor pattern classification. IEEE Trans Inform Theory 13(1):21–27
David E (1997) Goldberg, genetic algorithms in search, optimization and machine learning. Addison Wesley, Boston
Ebo Bennin K, Keung J, Phannachitta P, Monden A, Mensah S (2017) MAHAKIL: diversity based oversampling approach to alleviate the class imbalance issue in software defect prediction. IEEE Trans Softw Eng 44:534–550
Fenton NE, Ohlsson N (2000) Quantitative analysis of faults and failures in a complex software system. IEEE Trans Softw Eng 26(8):797–814
Gray D, Bowes D, Davey N, Sun Y, Christianson B (2011) The misuse of the NASA metrics data program data sets for automated software defect prediction. In: Proc 15th annu conf eval assessment softw eng, pp 96–103
Han H, Wang W-Y, Mao B-H (2005) Borderline-smote: a new oversampling method in imbalanced data sets learning. In: Proc int conf intell comput, pp 878–887
He H, Garcia EA (2009) Learning from imbalanced data. IEEE Trans Knowl Data Eng 21(9):1263–1284
He H et al (2008) Adasyn: adaptive synthetic sampling approach for imbalanced learning. In: Proc IEEE int joint conf neural netw IEEE world cong comput intell, pp 1322–1328
Japkowicz N (2000) The class imbalance problem: significance and strategies. In: Proceedings of the 2000 international conference on artificial intelligence (IC-AI’2000). Special Track on Inductive Learning Las Vegas, Nevada
De Jong KA, Spears WM (1989) Using genetic algorithm to solve NP-complete problems. In: Proceedings of 3rd international conference on genetic algorithms, pp 124–132
Joshi MV, Kumar V, Agarwal RC (2001) Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In: Proceedings IEEE international conference on data mining, pp 257–264
Kocaguneli E, Menzies T, Bener A, Keung JW (2012) Exploiting the essential assumptions of analogy-based effort estimation. IEEE Trans Softw Eng 38(2):425–438
Kocaguneli E, Menzies T, Keung J, Cok D, Madachy R (2013) Active learning and effort estimation: finding the essential content of software effort estimation data. IEEE Trans Softw Eng 39(8):1040–1053
Laradji IH, Alshayeb M, Ghotui L (2015) Software defect prediction using ensemble learning on selected features. Inform Softw Technol 58:388–402
Lessmann S, Baesens B, Mues C, Pietsch S (2008a) Benchmarking classification models for software defect prediction: a proposed framework and novel findings. IEEE Trans Softw Eng 34(4):485–496
Li W, Henry S (1993) Object-oriented metrics that predict maintainability. J Syst Softw 23(2):111–122
Li W, Henry W (1996) A validation of object-oriented design metrics as quality indicators. IEEE Trans Softw Eng 22(10):751–761
Liaw A, Wiener M (2002) Classification and regression by random forest. R News 3(2):18–22
Liu A, Ghosh J, Martin CE (2007) Generative oversampling for mining imbalanced datasets. In: Proceedings international conference on data mining, pp 66–72
Ma Y, Guo L, Cukic B (2006) A statistical framework for the prediction of fault-proneness. In: Advances in machine learning applied software engineering. Idea Group Inc, pp 237–265
Mahalanobis PC (1936) On the generalized distance in statistics. Proc Nat Inst Sci Calcutta 2:49–55
Manoharan JS (2021) A novel user layer cloud security model based on chaotic Arnold transformation using fingerprint biometric traits. J Innov Image Process JIIP 3(01):36–51
Menzies T, Greenwald J, Frank A (2007) Data mining static code traits. J Innov Image Process JIIP 3(01):36–51
Lessmann S, Baesens B, Mues C, Pietsch S (2008a) Benchmarking classification models for software defect prediction: a proposed framework and novel findings. IEEE Trans Softw Eng 34(4):485–496
Li W, Henry S (1993) Object-oriented metrics that predict maintainability. J Syst Softw 23(2):111–122
Li W, Henry W (1996) A validation of object-oriented design metrics as quality indicators. IEEE Trans Softw Eng 22(10):751–761
Liaw A, Wiener M (2002) Classification and regression by random forest. R News 3(2):18–22
Liu A, Ghosh J, Martin CE (2007) Generative oversampling for mining imbalanced datasets. In: Proceedings international conference on data mining, pp 66–72
Ma Y, Guo L, Cukic B (2006) A statistical framework for the prediction of fault-proneness. In: Advances in machine learning applied software engineering. Idea Group Inc, pp 237–265
Mahalanobis PC (1936) On the generalized distance in statistics. Proc Nat Inst Sci Calcutta 2:49–55
Manoharan JS (2021) A novel user layer cloud security model based on chaotic Arnold transformation using fingerprint biometric traits. J Innov Image Process JIIP 3(01):36–51
Menzies T, Greenwald J, Frank A (2007) Data mining static code traits. J Innov Image Process JIIP 3(01):36–51
Lessmann S, Baesens B, Mues C, Pietsch S (2008a) Benchmarking classification models for software defect prediction: a proposed framework and novel findings. IEEE Trans Softw Eng 34(4):485–496
Li W, Henry S (1993) Object-oriented metrics that predict maintainability. J Syst Softw 23(2):111–122
Li W, Henry W (1996) A validation of object-oriented design metrics as quality indicators. IEEE Trans Softw Eng 22(10):751–761
Liaw A, Wiener M (2002) Classification and regression by random forest. R News 3(2):18–22
Liu A, Ghosh J, Martin CE (2007) Generative oversampling for mining imbalanced datasets. In: Proceedings international conference on data mining, pp 66–72
Ma Y, Guo L, Cukic B (2006) A statistical framework for the prediction of fault-proneness. In: Advances in machine learning applied software engineering. Idea Group Inc, pp 237–265
Mahalanobis PC (1936) On the generalized distance in statistics. Proc Nat Inst Sci Calcutta 2:49–55
Manoharan JS (2021) A novel user layer cloud security model based on chaotic Arnold transformation using fingerprint biometric traits. J Innov Image Process JIIP 3(01):36–51
Menzies T, Greenwald J, Frank A (2007) Data mining static code traits. J Innov Image Process JIIP 3(01):36–51
Lessmann S, Baesens B, Mues C, Pietsch S (2008a) Benchmarking classification models for software defect prediction: a proposed framework and novel findings. IEEE Trans Softw Eng 34(4):485–496
Li W, Henry S (1993) Object-oriented metrics that predict maintainability. J Syst Softw 23(2):111–122
Li W, Henry W (1996) A validation of object-oriented design metrics as quality indicators. IEEE Trans Softw Eng 22(10):751–761
Liaw A, Wiener M (2002) Classification and regression by random forest. R News 3(2):18–22
Liu A, Ghosh J, Martin CE (2007) Generative oversampling for mining imbalanced datasets. In: Proceedings international conference on data mining, pp 66–72
Ma Y, Guo L, Cukic B (2006) A statistical framework for the prediction of fault-proneness. In: Advances in machine learning applied software engineering. Idea Group Inc, pp 237–265
Mahalanobis PC (1936) On the generalized distance in statistics. Proc Nat Inst Sci Calcutta 2:49–55
Manoharan JS (2021) A novel user layer cloud security model based on chaotic Arnold transformation using fingerprint biometric traits. J Innov Image Process JIIP 3(01):36–51
Menzies T, Greenwald J, Frank A (2007) Data mining static code traits. J Innov Image Process JIIP 3(01):36–51
