

An Empirical Comparison of Model-Agnostic Techniques for Defect Prediction Models

Gichan Lee and Scott Uk-Jin Lee
 Department of Computer Science and
 Engineering
 Hanyang University
 Ansan, Republic of Korea
 {fantasyopy, scottlelee}@hanyang.ac.kr

Abstract—Recently, software defect prediction studies have attempted to make black-box defect prediction models explainable and actionable. State-of-the-art defect prediction studies have utilized various model-agnostic techniques derived from the explainable AI domain as key tools to make the predictions easier for practitioners to understand. However, it has not been sufficiently investigated whether there is inconsistent information within local explanations generated by different model-agnostic techniques when interpreting a defect prediction. If local explanations generated by heterogeneous model-agnostic techniques consist of different information, the derivable insights to understand and act upon defect predictions becomes less viable and it may cause ineffective or even incorrect defect corrections. In this research, we empirically analyzed 323,844 local explanations generated by three different model-agnostic techniques: (1) LIME, (2) SHAP, and (3) BreakDown. These local explanations were analyzed in terms of how the contributions of features were distributed, how the contributions were ranked, and whether the contributions were contradictory. We concluded that (i) different model-agnostic techniques provide practitioners with local explanations where average contributions of the top-ranked features are different; (ii) different model-agnostic techniques provide practitioners with local explanations consisting of different contribution rankings and inconsistent contribution directions of top-ranked features. Therefore, we recommend that practitioners should avoid using model-agnostic techniques interchangeably and must perform a multi-faceted manual validation when planning actions based on the local explanations.

Keywords— *Explainable Software Analytics, Software Quality Assurance, Defect Prediction Models, Model-Agnostic Technique*

I. INTRODUCTION

Recently, software defect prediction studies have attempted to make black-box defect prediction models explainable and actionable [1], [2]. State-of-the-art defect prediction research has utilized various model-agnostic techniques derived from the explainable AI (XAI) domain as key tools for enabling practitioners to understand defect predictions [3]. Model-agnostic techniques have received significant attention in that they provide local explanations containing information about how features contributed to each prediction of any Machine Learning and Artificial Intelligence (ML/AI) models [32]. Since local explanations contain information about the degree and direction of contribution for each feature in the sample, they provide hints as to what action practitioners should take on individual defect predictions. For instance, the local explanation “feature X contributed 15% to predicting that a method is defective” leads practitioners to infer that they can reduce the probability of the risk to be defective by reducing feature X [3].

Recent defect prediction studies analyzed the characteristics of local explanations from model-agnostic techniques such as Local Interpretability Model-Agnostic Explanations (LIME) [8], Shapley Additive Explanations (SHAP) [9], and BreakDown [10] to investigate the applicability to defect prediction models [4], [5]. In addition, with the use of various model-agnostic techniques, other studies have used various model-agnostic techniques in proposing several means to support decision-making and maintenance activities for software defects by selecting most-contributing features to make the defect models more explainable [13], detecting defective lines of source codes [6], [12], and providing feedback plans to reduce future defect-introducing commits [7], [11]. These studies recommend that model-agnostic techniques should be actively used as they can help practitioners understand ML/AI-based defect prediction models and their predictions.

However, it has not yet been sufficiently investigated whether there is a significant difference in information provided by local explanations when interpreting defect predictions using heterogeneous model-agnostic techniques. If there is a discrepancy in the information, practitioners may obtain different or even opposite insights depending on which model-agnostic technique is used. This can lead practitioners to select the inappropriate way to deal with software defects, resulting in having a risk of unnecessarily consuming resources for defect correction. Moreover, the various findings presented in the aforementioned studies heavily depend on local explanations of a specific model-agnostic technique used. If local explanations vary according to model-agnostic techniques, the research results based on them may also change, potentially jeopardizing the reliability of the research results presented. Despite the possible risks, there is no research conducted to identify the discrepancy between local explanations or to validate their accordance by examining the distributions of contribution sizes, rankings, and the directions of features within local explanations of heterogeneous model-agnostic techniques.

In that vein, we empirically analyze 323,844 local explanations generated by LIME, SHAP and BreakDown based on defect datasets for 32 releases of 9 open source software systems. We compare local explanations of each model-agnostic technique by focusing on the top features in the contribution rankings that practitioners are most likely to examine to obtain actionable guidance. Through the comparisons, we analyze the differences in contributions of top-ranked features, the similarities of contribution rankings, and the inconsistency in directions of contributions for top-ranked features to answer the following research questions:

(RQ1) How different are the contribution sizes of top-ranked features in different local explanations?

(RQ2) How different are the contribution rankings and the contribution directions of features in different local explanations?

In order to answer the RQ2 more precisely, an appropriate statistical method that can reflect the contributions of features to the similarity measure is adopted to fairly investigate the similarity between contribution rankings. Also, we first investigate whether the insights of local explanation could shift by comparing the contribution directions of the identical features appearing in local explanations of each model-agnostic techniques for the same defect prediction.

To the best of our knowledge, the contributions of this paper are the followings:

- The first large-scale empirical investigation focusing on contribution sizes, rankings and directions of features, to identify the differences in local explanations of heterogeneous model-agnostic techniques.
- Novel statistical analysis which reflects contribution sizes of features to identify the similarities of local explanations generated by heterogeneous model-agnostic techniques.
- The dataset publicly available online for explainable and actionable software analytics research, containing 323,844 local explanations of defect predictions [36].

Structure of the paper: Section 2 presents model-agnostic techniques in the field of defect prediction and further describes motivations for our research. Section 3 describes our case study design for analyzing local explanations of heterogeneous model-agnostic techniques. Section 4 explains the experimental results in detail and section 5 discusses the implications of findings. Section 6 presents the threats to validity. Finally, in section 7, we conclude our study and suggests the direction of future research.

II. RELATED WORKS AND MOTIVATIONS

Most of the research that uses model-agnostic techniques to provide practitioners with local explanations for individual defect predictions have newly appeared after 2020. They have investigated the value of the application for model-agnostic techniques, the practical use of local explanations, and the agreements between model-agnostic techniques.

A. Model-Agnostic Techniques For Defect Prediction

Jiarpakdee et al. [4] first applied LIME, LIME with Hyper Parameter Optimisation (LIME-HPO), and BreakDown to ML/AI defect prediction models to investigate whether local explanations of the individual predictions could provide explainable analytics to practitioners. Through a qualitative survey, they found that the majority of practitioners rated local explanations to be necessary and useful for understanding defect predictions and agreed that local explanations could build appropriate trusts for defect predictions. Based on the experimental results, they suggested that model-agnostic techniques could be used in the field of defect prediction, claiming that model-agnostic techniques are needed to explain the individual predictions of defect models. Jiarpakdee et al. [5] also investigated the possible future directions for next-

generation defect prediction models by conducting a survey on the goals of defect prediction models and practitioners' perceptions of model-agnostic techniques. They revealed that even though more than 82% of the respondents demanded explanations for predictions to be provided as a goal for utilizing defect prediction models, only 4% of the surveyed defect production research actually provides such explanations. Furthermore, they reported that model-agnostic techniques from the XAI domain could certainly help practitioners understand defect predictions while confirming that practitioners' agreement on LIME explanations was the highest among the visualized explanations of the eight different model-agnostic techniques.

Meanwhile, various research results to provide more practical benefits to practitioners based on local explanations of defect predictions have also been published. Wattanakriengkrai et al. [6] proposed a LINE-DP framework which interprets the predictions of ML/AI models using LIME to indicate where the actual defective lines are in the class predicted to have a defect. The LINE-DP has demonstrated the ability to find the actual location of defects predicted by a ML/AI model with 63% accuracy. Pornprasit et al. [7] developed PyExplainer where LIME's internal algorithms are improved to create explainable ML/AI models that immediately detects defect-introducing commits. PyExplainer provides practitioners with what-if explanations based on the values of features (e.g., if Churn > 100 and # of reviewers < 2, it is a defect-introducing commit) for humans to understand ML/AI models that predicts defect-introducing commits Just-In-Time. They claimed that PyExplainer's what-if explanations would help practitioners to focus on the features most relevant to defect-introducing commits. Additionally, Khanan et al. [11] proposed the JITBot to induce feedback from developers by providing the riskiness of defect-introducing commits predicted by the JIT models. Through the JITBot, they quantitatively expressed how risky the commits are and provided plans to reduce the risk of probability to be defective based on LIME's local explanations. Lastly, Esteves et al. [13] interpreted the defect predictions of ML/AI-based models with SHAP and filtered the features that contribute significantly to the predictions of the models. They emphasized that reducing the number of features in the dataset based on SHAP's local explanations can reduce the complexity of the model and provide more explainable insights to practitioners.

However, although aforementioned studies have interchangeably used heterogeneous model-agnostic techniques, consistency of information in the local explanations generated by these techniques have not yet been properly and thoroughly investigated. Inconsistencies in contribution distributions, rankings, and contribution directions of features in local explanations can cause practitioners to perform unnecessary maintenance activities and consequently affect the actual actionability provided. For instance, if different or even contradictory local explanations for the same defect prediction are generated by heterogeneous model-agnostic techniques, it will be very difficult for practitioners to know which local explanations to use to plan maintenance activities. In addition, even if practitioners want to choose a model-agnostic technique that provides more accurate local explanations, the proper means to quantitatively evaluate them have not yet been established [32]. It naturally makes practitioners reluctant to perform actions based on local explanations since they have to bear the burden of manual

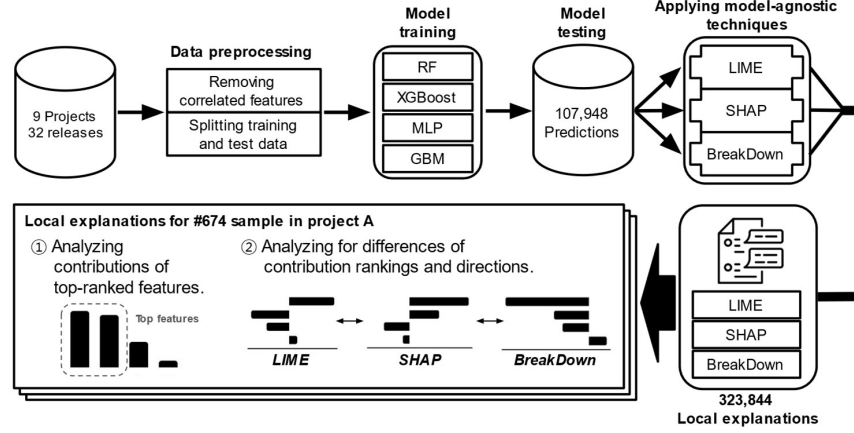


Fig. 1. Overview of experimental design.

validation to know which local explanations are reliable. Therefore, **it is essential to analyze the differences of information in the local explanations generated by heterogeneous model-agnostic techniques** to determine whether local explanations for defect predictions can really serve as actionable guidance.

B. Investigating Agreements between Local Explanations

There are also few studies which investigate the agreements between local explanations for defect prediction models. Rajbahadur et al. [14] raised an issue with the interchangeable use of model-specific methods (i.e., gini importance) and model-agnostic techniques when related studies have interpreted the predictions of defect models. They examined the agreements between model-agnostic techniques using the Kendall's tau [16] and revealed that there is not always a strong agreement between model-specific methods and model-agnostic techniques. Additionally, they reported that the top 1 and the top 3 ranks were similar when local explanations between different model-agnostic techniques were compared. Similarly, Yang et al. [15] investigated the agreements between local explanations of LIME, SHAP, and BreakDown for just-in-time defect predictions using the Kendall's tau. They found that the contribution rankings of features within the local explanations generated by different model-agnostic techniques were in strong agreement. Both Rajbahadur et al. and Yang et al. used Kendall's tau and reported that different model-agnostic techniques generates similar local explanations.

However, the Kendall's tau, which has been used by various related studies to measure the similarities between the feature rankings in local explanations, has a **limitation in that the contributions of features cannot be reflected in the similarity measurement**. For example, when measuring the similarity between two contribution rankings with 10 features, the case where the top 5 features are similar and the case where the bottom 5 features are similar are semantically different because the top 5 features have a greater contribution. However, the Kendall's tau will evaluate the both cases to have the same similarity because it does not consider the contributions of the top 5 features. Moreover, even if only the top 5 features are trimmed and considered when measuring the similarity between two contribution rankings, the Kendall's tau assume both ranked lists to be consist of the same elements. Hence, if the top 5 features in two contribution rankings differ,

the Kendall's tau cannot measure the similarity between the top 5 rankings. These limitations of the Kendall's tau makes it very difficult to analyze the agreement between model-agnostic techniques from a detailed point of view and may also cause a distortion in measuring true similarity due to sub-features with insignificant contributions to defect predictions.

Furthermore, the studies conducted by Rajbahadur et al. and Yang et al. did not consider **direction in which features within local explanations contribute**. The direction of contributions is determined by which a feature contributed positively or negatively to the prediction results of the model. For example, considering a case where a model predicts a commit to be defective, if a feature influences to predict this commit to be defective it is considered having a positive direction; otherwise, it is considered having a negative direction. Therefore, if the contribution directions of a feature that appears in each local explanations for the same defect prediction are opposite, the corresponding insights provided by each local explanation can be also opposite. To overcome the limitations of Kendall's tau, a novel statistical similarity measurement method is used and the contribution directions of features are also examined in this research (as detailed in following section).

III. RESEARCH METHODOLOGY

In this section, we introduce the research questions of this research and the experiment design to answer them.

A. Research Questions and Approach

(RQ1) How different are the contribution sizes of top-ranked features in different local explanations?

(RQ2) How different are the contribution rankings and the contribution directions of features in different local explanations?

The main objective of this study is to identify whether there exists a difference in information about defect prediction within local explanations generated by heterogeneous model-agnostic techniques. To answer the RQ1, we compare the contribution sizes of the top features in the contribution ranking for each model-agnostic technique to determine the difference in the amount of information that practitioners can obtain. To answer RQ2, we examine the similarities between the contribution rankings and the contribution directions of

TABLE I. OVERVIEW OF THE DATASET.

<i>Project Name</i>	<i>Description</i>	<i>Studied Releases</i>	<i># of features</i>	<i># of training samples</i>	<i># of testing samples</i>
ActiveMQ	Messaging and Integration Patterns server	5.0.0, 5.1.0, 5.2.0, 5.3.0, 5.8.0	26-28	1884-3420	691-1258
Camel	Enterprise Integration Framework	1.4.0, 2.9.0, 2.10.0, 2.11.0	25-26	1515-8846	554-3293
Derby	Relational Database	10.2.1.6, 10.3.1.4, 10.5.1.1	28	1963-2705	723-995
Groovy	Java-syntax-compatible OOP for JAVA	1.5.7, 1.6.0.Beta_1, 1.6.0.Beta_2	26-28	757-884	274-319
HBase	Distributed Scalable Data Store	0.94.0, 0.95.0, 0.95.2	25-27	1059-1834	380-674
Hive	Data Warehouse System for Hadoop	0.9.0, 0.10.0, 0.12.0	24-27	1416-2662	514-981
JRuby	Ruby Programming Lang for JVM	1.1, 1.4, 1.5, 1.7	29-31	731-1614	263-595
Lucene	Text Search Engine Library	2.3.0, 2.9.0, 3.0.0, 3.1.0	27-31	805-2806	292-1036
Wicket	Web Application Framework	1.3.0.beta1, 1.3.0.beta2, 1.5.3	27	1672-2578	610-953

features for determining the differences in local explanations generated by different model-agnostic techniques. An experiment is designed and conducted to answer the research questions as shown Fig. 1. We describe the processes to conduct the experiment in Fig. 1 in the following subsections.

B. Context Selection and Data Preprocessing

In our experiment, we use the dataset with ground-truths labeled by Yatish et al. [18] using the realistic approach that leverages the earliest affected release that is realistically estimated by a software development team for a given defect. The dataset consists of 32 releases of 9 Apache open-source software systems where they are actively used in many related studies to explain the predictions of defect models using model-agnostic techniques. By using the same dataset as the closely related studies which evaluated the value of applying model-agnostic techniques to defect prediction models [4]–[6], we aim to extend these studies and increase the generalizability of our results.

Accordingly, we apply the AutoSpearman function [19] to remove correlated features and use out-of-sample bootstrap sampling [20], [21] to obtain training and test samples, as in the study of Jiarpakdee et al. [4]. In this study, we set the dependent variable (i.e., target variable) as a binary variable “RealBug” feature, and the independent variables as the result of applying the AutoSpearman function to the features of each project release. To split the training and testing sets, we generate a bootstrap sample of size N with replacement for each release. Because all generated bootstrap samples are generated with replacement, an average of 36.8% of the original dataset did not appear in the bootstrap samples. By setting these samples as the testing set, we allow our models to predict samples that have never been seen during training process. Table 1 shows the dataset of this study obtained by performing the aforementioned data preprocessing. We train defect prediction models on a training set configured for each release as described in the following subsection.

C. Model Training and Testing

In this research, 3 ensemble methods which are Random Forest (RF), Gradient Boosting Machine (GBM), and XGBoost Tree (XGBT) as well as the deep learning method which is Multi-Layer Perceptron (MLP) are selected to construct defect prediction models. These algorithms are frequently selected for their high performance in the defect prediction field and their robustness to the correlation between features in high-dimensional datasets [6], [7], [11]–[15], [22]–

[27]. We intend to increase the generalizability of the research results by selecting these black-box ML/AI algorithms in which practitioners are likely to interpret individual defect predictions using model-agnostic techniques.

This study uses the implementations provided in the *scikit-learn* package [28] to acquire ML/AI models. The *RandomForestClassifier*, the *GradientBoostingClassifier*, and the *MLPClassifier* which are *scikit-learn-1.1.2* implementations of RF, GBM, and MLP as well as the *XGBoostClassifier* [29], in which the XGBT algorithm is implemented as an API version of *scikit-learn*, are used in the experiment. We set the classification threshold of the algorithms to 0.5 for the trained models to perform binary classification on the presence or absence of a defect (i.e., if the predicted probability ≥ 0.5 , the model predicted the defect, otherwise clean). We divide the entire dataset into a training and testing set where the training set is used for the training and performance evaluation of the models, and the testing set is used only once for prediction to have the same effect as the models predicting a new data in the wild.

The performance of the models is evaluated through 5-fold cross validation of the training set [33]. The k-fold cross validation is a method of measuring performance by dividing the dataset into k-folds where k-1 folds are used for training ML/AI algorithms and 1-fold is used for testing by repeating the process k times. When dividing the training set by 5-fold, we apply stratified sampling to keep the defect and clean ratio of the training set equal to avoid the bias caused by imbalance. In addition, according to the recommendation of Hall et al. [30] indicating that performance can be overestimated due to the randomness of dividing the folds, we repeat the 5-fold cross validation more than 100 times and then average all the evaluation results to check the performance of the models.

D. Collecting Local Explanations

We use three widely-used model-agnostic techniques (i.e., LIME [8], SHAP [9], and BreakDown [10]) from XAI domains, as well as those used in related studies. These model-agnostic techniques calculate the contribution of each feature using their own algorithms which can be summarized as follow:

LIME generates new samples with features that randomly approximate the features of the target sample and let the target model predict these new samples. Then, it creates a new training set with the features of new samples and the predicted

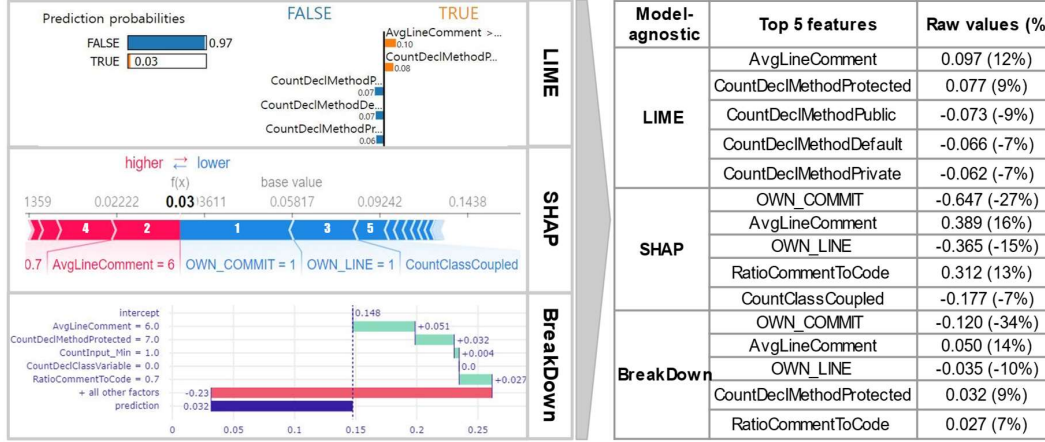


Fig. 2. A summarized example of local explanations from heterogeneous model-agnostic techniques.

values (e.g. a probability of the risk to be defective) of the target model. With a new training set, several local regression models are trained and the model with the smallest error among the trained models are selected. Finally, LIME have the model predict the target sample and calculate the contribution of the features using the coefficients of the selected local regression model.

SHAP creates a power set based on all features of the training set, and trains the target models to all combinations included in the power set. Subsequently, it compares the predicted values of the model trained on the subset with the target feature to the predicted values of the model trained on the subset without the target feature but all remaining features are the same. SHAP then sums all the differences between the predicted values generated by the target feature. SHAP repeats this process for all features and calculates the contribution of each feature.

BreakDown make the target model to predict all samples in the training set and then average the predicted values. Afterwards, it picks one of the features of the target sample and feeds all samples which have the same value of the feature to target model. The contribution of the feature is the difference between the average predicted value for all samples in the training set and the average predicted value for the samples in which the feature value was picked. BreakDown continues picking one of the other features in the target sample until the contribution of all features are calculated.

As described above, LIME, SHAP, and BreakDown calculate the contribution and direction of each feature in different manners. Therefore, it is difficult to compare the contributions of features from different model-agnostic technique as the units of contributions are different. In order to resolve this issues, we replace the contribution value of each feature with a percentage divided by the sum of the contribution values of all features in the same local explanation to fairly compare the contributions of features from each model-agnostic technique.

Fig. 2 illustrates the summarized example of local explanations for the same defect prediction in this study using *lime-0.2.0.1*, *shap-0.41.0*, and *dalex-1.4.1* (including BreakDown package). It includes only the summarized version of each local explanation from different model-agnostic techniques due to the page limit. LIME shows the

directional contributions of the top 5 features in descending order where the orange bars represent the features contributed to make the model predict that the sample has a defect, and the blue bars represent the opposite. SHAP also shows the contributions with red and blue bars (consistent with the orange and blue bars in LIME) with various lengths. The length of each bar represents the contribution size of each feature. Since the number of features to be represented are restricted due to the page limit, BreakDown explicitly shows the features that contributed positively and summarizes all the other features that contributed negatively with the label named “+ all other factors”. The table on the right lists the top 5 features and their contribution sizes calculated for each model-agnostic technique. Raw values are original contribution values from each model-agnostic technique and their signs mean the contribution directions. Since the units for raw contribution values are different for each model-agnostic technique, it is difficult to compare which feature has a relatively greater contribution in their local explanation. Hence, the replaced contribution percentages presented within the parentheses are used to enable comparison between the contributions of features for each model-agnostic technique.

E. The Methods to Analyse Local Explanations

In this study, we compare the contribution sizes as well as the contribution rankings and the directions of features in local explanations generated by heterogenous model-agnostic techniques to answer the derived research questions. For each comparative analysis, we focus on the top features in contribution rankings from which practitioners can gain actionable guidance.

Firstly, we examine the contribution sizes of the top features for each sample and calculated the average contribution sizes of the features for each release. Then the average contribution sizes of top-ranked features of LIME, SHAP, and BreakDown are compared to investigate whether there are differences in the amount of information that practitioners can acquire according to model-agnostic technique.

Secondly, we use the Ranked Biased Overlap (RBO) [17] to measure the similarity between LIME, SHAP and BreakDown in terms of contribution rankings. The RBO enables the similarity measure between indefinite ranked lists by using the average overlap ratio which can be calculated by

means of all the overlap ratios between the two ranked lists at each depth. Since the “ratio” of overlap is measured, if RBO is 0, it means “completely different” rankings, and if RBO is 1, it means the “same” rankings. We set the value of 0.5 as a threshold to classify the $RBO < 0.5$ as “different”, otherwise “similar”.

One of main reasons for selecting the RBO is because a weight can be assigned to the overlaps up to a specific depth when RBO measure the similarity between two ranked lists. Furthermore, the RBO can measure the similarity even if the two rankings do not have the same elements as the nature of the RBO is in calculating overlap. The RBO weights the depth of the ranking through the value p representing the persistence of the user. Since the RBO parameterizes the user's patience with the degree to which the user check the depth of ranking with p , we can apply Weight of RBO (WRBO) to the overlap up to the top d rank by adjusting this p value, as in (1).

$$W_{RBO}(1 : d) = 1 - p^{d-1} + \frac{1-p}{p} \cdot d \cdot \left(\ln \frac{1}{1-p} - \sum_{i=1}^{d-1} \frac{p^i}{i} \right) \quad (1)$$

The WRBO is simply a weight indicating how much influence the top d rank exerts on the final RBO value. For example, if $p=0.9$ and $d=10$, the WRBO will have a value of 0.85 approximately, meaning that the overlap ratio up to the top 10 contributes approximately 85% to the final RBO value. Hence, we can weight how much the similarity between the top d contribution rankings will affect the final RBO value between the overall contribution rankings that we want to measure. In this study, we plan to assign the average contributions of top-ranked features in all local explanations as WRBO to make the contributions of features be reflected in the measure of similarity between contribution rankings. While weighting WRBO as much as the average contribution sizes of the top-ranked features, we compute pairwise RBO values between feature rankings in local explanations of LIME, SHAP, and BreakDown for all defect predictions for each release (e.g. LIME & SHAP, SHAP & BreakDown, and BreakDown & SHAP). Then, the ratios of cases where $RBO < 0.5$ (i.e., both ranked lists are different) are calculated for each release and used to identify how different model-agnostic techniques generate different contribution rankings.

Finally, we also analyze the contribution directions of top-ranked features provided by the model-agnostic techniques in pairwise manner. For example, we find the intersection of a set of top-ranked features from LIME and SHAP. If the intersection has a feature with inconsistent contribution direction, we record that there is a “contradiction” between LIME and SHAP. In this manner, we record the ratio at which contradictions occur in pairs of model-agnostic techniques for each release. We use the ratios to investigate how frequently model-agnostic techniques provide local explanations with contradictory top-ranked features.

IV. RESULTS

In this section, we present the experimental results and explain how they answer the RQ1 and the RQ2 in detail. Prior to presenting the experimental results, we introduce the performance of the RF, GBM, MLP, and XGBoost models. Fig. 3 shows the distributions of the model performance of all ML/AI models for all releases in the defect datasets. We could not find any significant impact of hyperparameter tuning for the models while training, so the models were trained with default hyperparameter settings from *scikit-learn* package.

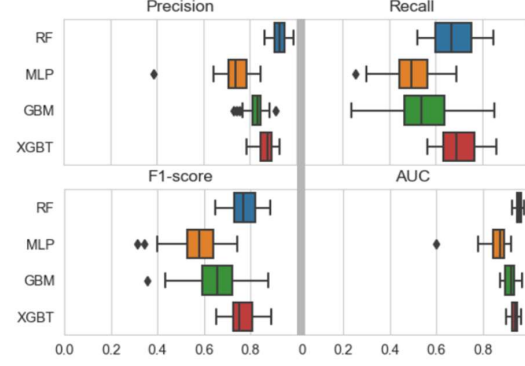


Fig. 3. The distribution of model performance for all releases.

The RF, MLP, GBM, and XGBoost models achieved a median precision of 0.74-0.93, a median recall of 0.5-0.69, a median F1-score of 0.58-0.77, and a median of AUC of 0.87-0.96. This indicates that the defect predictions of the models are accurate enough to avoid biases from being introduced in local explanations due to the inaccuracies of the models. Based on these results, we analyzed local explanations and obtained the following experimental results to answer the RQ1 and the RQ2:

(RQ1) How different are the contribution sizes of top-ranked features in different local explanations?

We focused on the top 5 and the top 10 features that practitioners would be more interested in rather than considering the entire feature ranking to gain actionable guidance for predicted defects. We calculated the average contribution of the top 5 features and the top 10 features per release for each model-agnostic techniques as shown in the violin plots of Fig. 4. Fig. 4 illustrates the distributions of the average contributions of the top 5 and the top 10 features derived from each-model-agnostic technique for 32 releases. The y-axis labels present each model-agnostic technique, the kind of model, and the number of top features. The x-axis labels present the average contribution size of top features for each release in the percentile. The thicker part of each violin plot indicates that the frequency of the corresponding average contribution is high. The heads and tails of the violin plot indicate that the frequency is low and the outliers are present. The white points in the thicker parts of violin plots represent the mean values of the distributions. For instance, the “LIME, RF_5” violin plot shows the distribution of the average contribution size of the top 5 features for 32 releases where the distribution ranges between 0.45 and 0.56 approximately, and the mean is 0.52 approximately. It means that the top 5 features of the local explanations have average 52% of the contributions when LIME explains the predictions of the RF model.

Fig. 4 indicates that there is a clear difference in the average contribution sizes of the top-ranked features between LIME and the other model-agnostic techniques. Although all model-agnostic techniques explain that the top-ranked features have a high average contributions, SHAP and BreakDown explained that the top-ranked features have more information on average than those of LIME. The top 5 features of SHAP, BreakDown, and LIME have average contributions of 64.12%, 61.71%, and 54.80%, respectively where the top 10 features have average contributions of 85.42%, 83.96%, and 77.44%, respectively. This results indicates that the

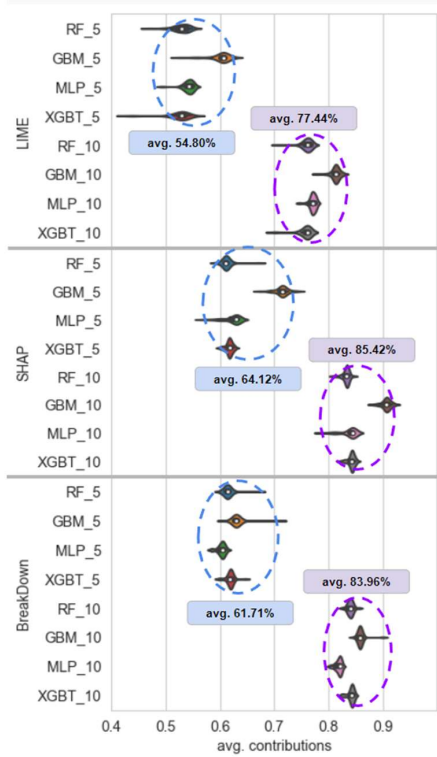


Fig. 4. The distributions of the average contribution sizes of top ranked features for all releases.

average contributions of top-ranked features of SHAP and BreakDown have a difference of less than 3%, whereas the average contributions of the top 5 and the top 10 features of LIME have approximately 6–8% less information than SHAP and BreakDown. This can be interpreted that the LIME evaluates the importance of top features to be lower than the other model-agnostic techniques.

(RQ1) How different are the contribution sizes of top-ranked features in different local explanations?

The average contributions of top-ranked features are different for each model-agnostic technique. In particular, the average contributions of top-ranked features from LIME are evaluated to be 6–8% less when compared to SHAP and BreakDown.

Meanwhile, the average contribution sizes of the top 5 and the top 10 features in all contribution rankings were about 60% and 82%, respectively. According to this result, we assigned the WRBO equal to the average contribution sizes of the top-ranked features to measure the RBOs between contribution rankings of LIME, SHAP, and BreakDown. We assigned a WRBO of approximately 0.60 to the similarity between the top 5 features by adjusting a p of 0.92, and a WRBO of about 0.82 to the similarity between the top 10 features by adjusting a p of 0.912. Therefore, the similarities between the top 5 features and the similarities between the top 10 features influence the derivation of the final RBO values as much as their average contributions.

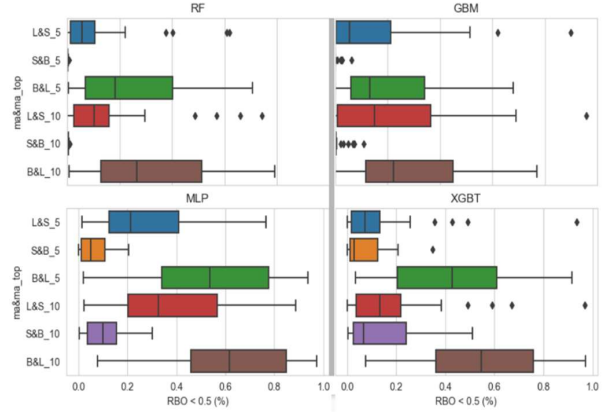


Fig. 5. The distributions of the ratios where RBO < 0.5 for all releases.

(RQ2) How different are the contribution rankings and the contribution directions of features in different local explanations?

A. Contribution Rankings of Features

We measured the similarity in terms of the ratio of the cases where RBO < 0.5 (i.e., both ranked lists are different) between the contribution rankings of features within the local explanations of LIME & SHAP, SHAP & BreakDown, BreakDown & LIME for each release. Fig. 5 illustrates the distributions of the ratios of the cases where RBO < 0.5 according to the pairs of applied model-agnostic techniques for each defect prediction model. The x-axis of the plots indicates the ratio of the cases where RBO < 0.5 and the y-axis represents the pair of model-agnostic techniques with weighted top rank for which the RBO was measured. For instance, the “RF, B&L_5” box plot shows the distribution of the ratios where RBO < 0.5 for 32 releases when calculating the RBO between contribution rankings of BreakDown and LIME. The distribution ranges between 0.07 and 0.4 approximately, and the mean is 0.18 approximately. It means that when comparing contribution rankings in local explanations of LIME and BreakDown for all samples, their explanations are different in approximately 18% of all cases.

Fig. 5 reveals that there are considerable number of cases where different model-agnostic techniques explain same defect predictions with different contribution rankings. In particular, the contribution rankings of LIME are more frequently different when compared to those of others. The ratio of the cases where RBO < 0.5 between LIME & SHAP is 17.34% on average when weighting the top 5 features and 24.29% when weighting the top 10 features. The average ratios of cases where RBO < 0.5 between LIME & BreakDown are higher where it was 35.11% when weighting the top 5 features and 43.96% when weighting the top 10 features. On the other hand, the average ratios of cases where RBO < 0.5 between SHAP and BreakDown are much lower resulting 3.5% and 6.19% on average when weighting the top 5 features and the top 10 features, respectively. In common, it is observed that the ratios of the cases where RBO < 0.5 are increased when the top 10 features are weighted.

B. Contribution Directions of Features.

Meanwhile, after comparing the contribution rankings of different model-agnostic techniques, we measured the ratio at which contradictions occur (i.e. common top features in both local explanations have opposite contribution directions) in

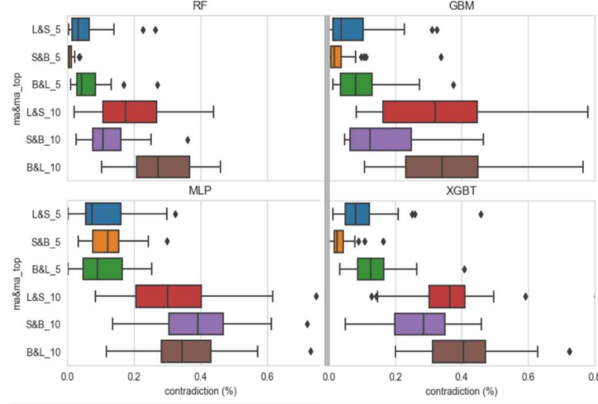


Fig. 6. The distributions of the ratios where contradictions occur in top-ranked features for all releases.

the top 5 and the top 10 features when considering the pairs different model-agnostic techniques for each release. Fig. 6 shows the distributions of the ratios of cases where the contradictions occurred between different model-agnostic techniques of each pair for each defect prediction model.

Fig. 6 indicates that there exist the top-ranked features whose contribution direction are inconsistent in each local explanation of different model-agnostic techniques. The average ratios of the cases where a contradiction exists between the top 5 features of LIME & SHAP, LIME & BreakDown, and SHAP & BreakDown are 8.61%, 9.98%, and 5.26%, respectively where the average ratios of the same cases for the top 10 features are 29.69%, 35.51%, and 24.10%, respectively. In the context similar to the one shown in Fig. 5, the contribution directions of top-ranked features of LIME are more frequently different from those of other model-agnostic techniques. However, it was observed that the ratios of the cases where contradiction occur between SHAP & Breakdown are moderately high despite of their contribution rankings being similar. Hence, it can be deduced that SHAP and BreakDown generate different local explanations where the contribution rankings are similar but the contribution directions of top-ranked features are inconsistent.

(RQ2) How different are the contribution rankings and the contribution directions of features in different local explanations?

The local explanations are different between each model-agnostic technique in terms of contribution rankings by up to 43.96%. Moreover, the ratios of cases with contradictory contribution directions are observed to be up to 35.51%. In particular, the local explanations of LIME are significantly different when compared to others in terms of contribution rankings and directions.

V. IMPLICATIONS OF FINDINGS

A. Summary of Findings and Implications.

In this research, we compared the average contribution sizes, the contribution rankings, and directions centering on top-ranked features between local explanations of heterogeneous model-agnostic techniques. As a result of the experiments, local explanations of each model-agnostic technique showed 3 distinctive differences which can lead

practitioners to take different actions on target defect predictions, and they are as follow.

First, the heterogeneous model-agnostic techniques generated local explanations in which the average contribution sizes of top-ranked features are different. In particular, the average contributions of top-ranked features from LIME were approximately 6% to 8% less than those of SHAP and BreakDown, implying that the contributions were further distributed to the lower-ranked features. It means that practitioners using LIME may have to examine more features with lower contribution ranks to compensate for semantically less stronger information presented by the top-ranked features about target defect predictions. **Therefore, the differences in the contribution of top-ranked features can possibly change how practitioners use local explanations and the actions based on them depending on which model-agnostic technique they use.**

Second, the heterogeneous model-agnostic techniques generated local explanations in which the contribution rankings of features are different. The average ratio of the cases where $RBO < 0.5$ between the contribution rankings of LIME and those of SHAP and BreakDown was considerably high. Since RBO measures the similarity by averaging the overlap ratios, “ $RBO < 0.5$ ” can be interpreted as two different cases which may lead practitioners to take different actions toward target defect prediction. One case is to have the top 5 or the top 10 contribution rankings composed of similar features but the rank of each feature is different. In such case, since the priorities of practitioners highly depends on contribution rankings of features, the aspect of corrective actions for predicted defects can change. The other case is to have the top 5 or the top 10 contribution rankings composed of different features. Where in this case, practitioners can take completely different actions on the defect prediction.

Third, the heterogeneous model-agnostic techniques generated different local explanations in which the contribution directions of top-ranked features are contradictory. **The existence of top features with contribution directions opposite to each other suggests that practitioners can take completely opposite direction of action for defect predictions depending on which model-agnostic techniques used.** For example, imagine a top-ranked feature with a large contribution to a model’s defect prediction are opposite to each other according to the local explanations called “A” and “B” generated by different model-agnostic techniques. A practitioner referring to “A” may take the actions to change the value of the feature which supports the prediction whereas the other practitioner who refers to “B” may leave the value of the same feature as it is and attempt to change the value of other features. If the feature has a low contribution to a defect prediction, there will definitely be less room for practitioners to change the actions. But for the top-ranked feature which has a large proportion of contribution, the situation described in the above example is much more likely to occur.

As mentioned in the related works and motivation section, Rajbahadur et al. [14] and Yang et al. [15] investigated the agreements between model-agnostic techniques using the measurement based on Kendall’s tau where they reported that model-agnostic techniques have a strong agreement. However, they did not consider that the contribution sizes of features can influence the semantical similarities between contribution

rankings. They also did not take the contribution directions of top features into account when investigating the agreements. On the other hand, we reflected them in the measurement of similarities between local explanations of heterogeneous model-agnostic techniques in our experiment. Our observations indicated that there are considerable disagreements in contribution sizes, rankings, and even contribution directions of features. **It means that practitioners can obtain different or even contradictory actionable guides if they use the techniques interchangeably.**

On the basis of the observations and their implications, we **conclude that it can be dangerous for practitioners to conduct maintenance activities based on the local explanations generated by only a single model-agnostic technique.** We also recommend that practitioners should avoid the interchangeable use of different model-agnostic techniques and must perform multi-faceted manual validations when planning actions based on local explanations. Furthermore, we propose that the related studies which drew research conclusions based on local explanations of a single model-agnostic technique should check whether the same research results can be obtained even if they use other model-agnostic techniques.

B. Research Opportunity

Through this experiments, we uncovered a research opportunity that is beyond the scope of this research but deserves a further study. While comparing heterogeneous model-agnostic techniques, we observed that the ratios of the cases where $RBO < 0.5$ and the contradiction occurred in top-ranked features are distributed differently depending on the defect prediction models. More specifically, from Fig. 5 and Fig. 6, we observed that the distributions of the ratios where $RBO < 0.5$ and contradiction occurred in top-ranked features are plotted in the range of higher values in MLP and XGBT when compared with those in RF and GBM.

The aforementioned phenomenon can be observed from "S&B_5" and "S&B_10" plots in Fig. 5. The distributions of the ratios of the cases where $RBO < 0.5$ in both of the RF and the GBM plots are converged to zero, but the same distributions for the MLP and the XGBT are plotted with a higher range of values. Similarly, there is another observation for the top 10 features presented in Fig. 6. The distributions of the ratios of the cases where contradiction occurred for the RF and the GBM are plotted between 0.1 and 0.2 approximately whereas the same distributions for the MLP and the GBM are plotted between 0.3 and 0.4 with less variances.

From these observation, we hypothesized that the similarity between the local explanations of heterogeneous model-agnostic techniques could vary depending on the defect prediction models. We propose possible future studies investigating the correlation between defect prediction models and the internal algorithms used by model-agnostic techniques. Such investigations can serve as a basis for exploring defect prediction models in which different model-agnostic techniques generate consistent local explanations.

VI. THREATS TO VALIDITY

A. Construct Validity

This research uses the dataset labeled ground truths based on the earliest affected releases that were realistically estimated by a software development team for a given defect

[18]. Hence, the bias and distortion may possibly occur in the dataset and can directly be reflected in the results of this research. Jiarpakdee et al. [4] attempted to overcome the bias that may occur in the dataset by removing highly-correlated features using AutoSpearman and constructing a test set using out-of-sample bootstrap sampling. The data preprocessing methods of Jiarpakdee et al. also were applied to our research for it to be in the same context as the studies aimed at evaluating the practical value of applying model-agnostic techniques to the field of defect prediction. However, it cannot be guaranteed that all biases and distortions in the dataset are removed.

B. Internal Validity

In practice, defect prediction models have to predict data that were never observed in experimental environments. To reflect this real-world environment, we used RF, GBM, MLP, and XGBT models to perform predictions on a test set which they had never observed during the training process, and then interpreted these prediction results with LIME, SHAP, and BreakDown. However, this also implies that the research results might vary depending on the test set selected. In order to mitigate this threat, we repeated the same experiment 100 times by sampling the test set 100 times, and there were no significant changes that could overturn the conclusions of this study. However, since this test set does not represent all defects in the real-world, we will revisit this threat in our future studies and confirm whether the results of this research are valid through the data in practice.

Meanwhile, the ratio of defects being present in the dataset is unbalanced. Hence, it is possible for the analysis results of local explanations for samples in which no defects exists might distort the experimental results of this study. In order to mitigate this threat, we used SMOTE [34] to gradually alleviate the imbalance in the dataset and repeated the experiment. From the results of the experiments, we did not observe any significant changes in the conclusions of this research. However, since the distortion of model predictions caused by class imbalance might still occur sufficiently and the resulted possibilities of distortion of local explanations could not be excluded, we plan to revisit this threat in our future studies.

C. External Validity

Although this research draws its conclusions using Yatish et al.'s dataset consisting of the 32 releases of 9 open-source software systems, this dataset cannot represent all defects in practice. Furthermore, since software metrics are the main features in the dataset of Yatish et al., it is not known whether similar results to those of this research would be observed in defect prediction studies that use source codes or other types of features.

Additionally, in this research, the *RandomForestClassifier*, the *Gradient BoostingClassifier*, and the *MLPClassifier* implementations provided in *scikit-learn-1.1.2* package [28], as well as the *XGBoostClassifier* [29] implemented as API version of *scikit-learn* were used to construct RF, GBM, MLP, and XGBT models. However, since these implementations cannot be representative of the models that practitioners would build in practice, the representativeness of local explanations generated for the predictions of the models used in this research might be limited to these implementations.

Lastly, LIME, SHAP, and BreakDown used in this research are the most used model-agnostic techniques not only

in the related studies but also in all ML/AI fields. However, since model-agnostic techniques not covered in this research such as Anchors [35] can also be used, an extensive investigation is required to determine whether the same things as those observed in this study can be observed for local explanations generated by other model-agnostic techniques.

VII. CONCLUSION

This research investigates whether there is a difference in information provided within the local explanations for defect predictions generated by heterogeneous model-agnostic techniques. Initially, we obtained the defect predictions of RF, GBM, MLP, and XGB models for 32 releases of 9 software systems and 323,844 local explanations of LIME, SHAP, and BreakDown. Then, through the repeated experiments, we thoroughly analyzed the contribution distributions, contribution rankings, and contribution directions of features in all local explanations.

As a result, we conclude that (i) different model-agnostic techniques provide practitioners with local explanations where average contributions of the top-ranked features are different; (ii) different model-agnostic techniques provide practitioners with local explanations consisting of different contribution rankings and inconsistent contribution directions of top-ranked features.

Based on our conclusion, we highly recommend that practitioners avoid the interchangeable use of different model-agnostic techniques and must perform a multi-faceted manual validation when planning actions based on the local explanations. Furthermore, we propose that the related studies, which draw research conclusions based on local explanations of a single model-agnostic technique, should check whether the same research results can be derived even with other model-agnostic techniques. As one of the future directions of our research, we plan to devise an algorithm that fuses local explanations of heterogeneous model agnostic techniques and develop a model-agnostic platform that visualizes similarities and contradictions between local explanations and delivers them to practitioners.

ACKNOWLEDGMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-00155885, Artificial Intelligence Convergence Innovation Human Resources Development (Hanyang University ERICA)).

REFERENCES

- [1] H. K. Dam, T. Tran, A. Ghose, "Explainable software analytics," in *Proceedings of the 40th international conference on software engineering: New ideas and Emerging results*, 2018, pp. 53–56.
- [2] C. Tantithamthavorn, J. Jiarapakdee, and J. Grundy, "Explainable AI for Software Engineering," *arXiv preprint arXiv:2012.01614*, 2020.
- [3] C. Tantithamthavorn, J. Jiarapakdee, and J. Grundy, "Actionable analytics: Stop telling me what it is; please tell me what to do," *IEEE Software*, vol. 38, no. 4, pp. 115–120, 2021.
- [4] J. Jiarapakdee, C. Tantithamthavorn, H. K. Dam, and J. Grundy, "An empirical study of model-agnostic techniques for defect prediction models," *IEEE Transactions on Software Engineering*, 2020.
- [5] J. Jiarapakdee, C. K. Tantithamthavorn, and J. Grundy, "Practitioners' perceptions of the goals and visual explanations of defect prediction models," *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, 2021, pp. 432–443.
- [6] S. Wattanakriengkrai, P. Thongtanunam, C. Tantithamthavorn, H. Hata, and K. Matsumoto, "Predicting defective lines using a model-agnostic technique," *IEEE Transactions on Software Engineering*, 2020.
- [7] C. Pomprasit, C. Tantithamthavorn, J. Jiarapakdee, M. Fu, and P. Thongtanunam, "PyExplainer: Explaining the Predictions of Just-In-Time Defect Models," *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2021, pp. 407–418.
- [8] M.T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the Predictions of Any Classifier," in *Proceedings of the International Conference on Knowledge Discovery & Data Mining (KDD)*, 2016, pp. 1135–1144.
- [9] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 4765–4774.
- [10] M. Staniak and P. Biecek, "Explanations of model predictions with live and breakDown packages," *arXiv preprint arXiv:1804.01955*, 2018.
- [11] C. Khanan, W. Luewichana, K. Pruktharathikoon, J. Jiarapakdee, C. Tantithamthavorn, M. Choetkiertikul, C. Ragkhitwetsagul, and T. Sunetnanta, "JITBot: An Explainable Just-In-Time Defect Prediction Bot," *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2020, pp. 1336–1339.
- [12] C. Pomprasit and C. Tantithamthavorn, "JITLine: A Simpler, Better, Faster, Finer-grained Just-In-Time Defect Prediction," in *Proceedings of the International Conference on Mining Software Repositories (MSR)*, 2021, pp. 369–379.
- [13] G. Esteves, E. Figueiredo, A. Veloso, M. Viggiano, and N. Ziviani, "Understanding machine learning software defect predictions," *Automated Software Engineering*, vol. 27, no. 3, pp. 369–392, 2020.
- [14] G. K. Rajbahadur, S. Wang, G. A. Oliva, Y. Kamei, and A. E. Hassan, "The impact of feature importance methods on the interpretation of defect classifiers," *IEEE Transactions on Software Engineering*, vol. 48, no. 7, pp. 2245–2261, 2021.
- [15] X. Yang, H. Yu, G. Fan, Z. Huang, K. Yang, and Z. Zhou, "An Empirical Study of Model-Agnostic Interpretation Technique for Just-in-Time Software Defect Prediction," *International Conference on Collaborative Computing: Networking, Applications and Worksharing*, Springer, 2021, pp. 420–438.
- [16] H. Abdi, "The Kendall rank correlation coefficient," *Encyclopedia of measurement and statistics*, pp. 508–510, 2007.
- [17] W. Webber, A. Moffat, and J. Zobel, "A similarity measure for indefinite rankings," *ACM Transactions on Information Systems (TOIS)*, vol. 28, no. 4, pp. 1–38, 2010.
- [18] S. Yatish, J. Jiarapakdee, P. Thongtanunam, and C. Tantithamthavorn, "Mining Software Defects: Should We Consider Affected Releases?" in *Proceedings of the International Conference on Software Engineering (ICSE)*, 2019, pp. 654–665.
- [19] J. Jiarapakdee, C. Tantithamthavorn, and C. Treude, "AutoSpearman: Automatically Mitigating Correlated Software Metrics for Interpreting Defect Models," in *Proceedings of the International Conference on Software Maintenance and Evolution (ICSME)*, 2018, pp. 92–103.
- [20] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Boston, MA: Springer US, 1993.
- [21] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning*. Springer series in statistics, 2001, vol. 1.
- [22] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "The Impact of Automated Parameter Optimization on Defect Prediction Models," *Transactions on Software Engineering (TSE)*, vol. 45, no. 7, pp. 683–711, 2018.
- [23] C. Tantithamthavorn, A. E. Hassan, and K. Matsumoto, "The Impact of Class Rebalancing Techniques on The Performance and Interpretation of Defect Prediction Models," *Transactions on Software Engineering (TSE)*, vol. 46, no. 11, pp. 1200–129, 2019.
- [24] A. Gupta, S. Sharma, S. Goyal, and M. Rashid, "Novel xgboost tuned machine learning model for software bug prediction," *2020 International Conference on Intelligent Engineering and Management (ICIEM)*, 2020, pp. 376–380.
- [25] Y. K. Saheed, O. Longe, U. A. Baba, S. Rakshit, and N. R. Vajjhala, "An Ensemble Learning Approach for Software Defect Prediction in Developing Quality Software Product," *International Conference on Advances in Computing and Data Sciences*, Springer, 2021, pp. 317–326.

- [26] Pradhan, S., Nanniyur, V., & Vissapragada, P. K. "On the defect prediction for large scale software systems—from defect density to machine learning," *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, pp. 374–381, 2020.
- [27] S. S. Rathore and S. Kumar, "An empirical study of some software fault prediction techniques for the number of faults prediction," *Soft Computing*, vol. 21, no. 24, pp. 7417–7434, 2017.
- [28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and E. Duchesnay *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.
- [29] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system" In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794
- [30] T. Hall, S. Beecham, D. Bowes, D. Gray, and S. Counsell, "Developing fault-prediction models: What the research can show industry," *IEEE software*, vol. 28, no. 6, pp. 96–99, 2011
- [31] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.
- [32] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Benetton, S. Tabik, A. Barbado, and F. Herrera *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information fusion*, vol. 58, pp. 82–115, 2020.
- [33] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation," *Encyclopedia of database systems* vol. 5, pp. 532–538, 2009.
- [34] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.
- [35] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.
- [36] "Online publicly-available dataset of "An Empirical Comparison of Model-Agnostic Techniques for Defect Prediction Models"," <https://figshare.com/s/9bdc0986185983ade12b> [private]