2017 IEEE International Conference on Software Quality, Reliability and Security

Cross-project Defect Prediction Using a Credibility Theory based Naïve Bayes Classifier

Wai Nam Poon
Department of Computer Science,
City University of Hong Kong,
Kowloon, Hong Kong
wnpoon4-c@my.cityu.edu.hk

Kwabena Ebo Bennin
Department of Computer Science,
City University of Hong Kong,
Kowloon, Hong Kong
kebennin2-c@my.cityu.edu.hk

Jianglin Huang
Department of Computer Science,
City University of Hong Kong,
Kowloon, Hong Kong
jianhuang7@cityu.edu.hk

Passakorn Phannachitta
Software Engineering Laboratory,
Nara Institute of Science and Technology,
Ikoma, Japan
phannachitta-p@is.naist.jp

Jacky Wai Keung
Department of Computer Science,
City University of Hong Kong,
Kowloon, Hong Kong
Jacky.Keung@cityu.edu.hk

***Abstract*—Several defect prediction models proposed are effective when historical datasets are available. Defect prediction becomes difficult when no historical data exist. Cross-project defect prediction (CPDP), which uses projects from other sources/companies to predict the defects in the target projects proposed in recent studies has shown promising results. However, the performance of most CPDP approaches are still beyond satisfactory mainly due to distribution mismatch between the source and target projects. In this study, a credibility theory based Naïve Bayes (CNB) classifier is proposed to establish a novel reweighting mechanism between the source projects and target projects so that the source data could simultaneously adapt to the target data distribution and retain its own pattern. Our experimental results show that the feasibility of the novel algorithm design and demonstrate the significant improvement in terms of the performance metrics considered achieved by CNB over other CPDP approaches.**

Keywords—cross-project defect prediction; credibility theory; Naïve Bayes classifier; transfer learning; software engineering; quality assurance

I. INTRODUCTION

Software defects pose a potential threat to the quality of the software by causing unintended behavior in the software [1]. Defect prediction models, which predict buggy software modules, have been adopted and are being used by software quality teams. By using historical data, predicting the defective software module is possible in a more cost-effective way. In this sense, development team could find and fix the defects before the unit testing/ system testing conducted by quality assurance team, as well as allocating resource effectively to defect-prone module for improving the software reliability.

In practice, software projects, especially the newly-developed and small-scale ones, do not possess sufficient historical data to construct predictive models; therefore, the prediction for new projects are hardly possible. Fortunately, researchers have proposed cross-project defect prediction (CPDP) methods to deal with the situation by using another mature project as training set to predict defects successfully in those projects lacking historical data. However, cross-project predictions usually result in poor performance. According to Zimmermann, et al. [5], only 3.4% of 662 cross-project predictions have been successful. Moreover, Turhan, et al. [6] noted that a cross-project prediction yielded an increase in false positive rate with a median of 65% in their studies. They concluded this was due to the difference in data distribution between source and target projects. Turhan, et al. [6] further proposed the NN-filter to suppress the high probability of false alarm caused by cross-project defect prediction.

Several transfer leaning techniques [12], [13], [26] have been proposed to tackle the data distribution differences. These techniques extract knowledge from a source of data and transfers it to another data. The transferred knowledge is employed during model construction.

Considering the inferior performance in cross-project defect prediction and on the basis of transfer learning, we seek to use credibility theory to narrow the difference in feature distribution among datasets to obtain better cross-project prediction performance. This study adopts the credibility theory technique from the field of actuarial science generally used for insurance products pricing. The theory helps an insurer to set reasonable premium price for the insured according to the limited data from other insured and collected data (even less relevant) from its own insurance company [7]. Cross-project software defect prediction could be improved by the heuristic application of credibility theory in insurance product pricing. In the study, the small-scale or newly developed projects are the target data, and mature projects from other companies are source data. By leveraging the credibility theory, we aim to

978-1-5386-0592-9/17 $31.00 © 2017 IEEE
DOI 10.1109/QRS.2017.53

effectively minimize the feature distribution differences between the source and target projects so that we could implement the cross-project defect prediction. Besides, credibility theory is also widely used in machine learning field [8] [9] [7]. These studies applied credibility theory into generalized linear or mixed models, which provides us the incentive to introduce it into Naïve Bayes based cross-project defect prediction. As far as we know, this is the first time that credibility theory is used in software engineering prediction domain.

To test the hypothesis, we conduct experiments and propose a credibility theory based Naïve Bayes classifier to achieve the goal of cross-project defect prediction. The approach aims to reduce the difference of data distribution between source data and target data. The credibility theory provides a credibility factor to estimate the degree of reweighting the source data and target data to determine the extent of knowledge transfer from target data to source data. Such reweighting mechanism can help the source data to absorb the properties of the new or small-scale project to adjust the source data's attribute values, meanwhile, retaining its data distribution.

The main contributions of the paper are twofold:

1. We demonstrate that the credibility theory can help solve data distribution problem in CPDP.
2. We propose the novel credibility theory based Naïve Bayes combination for improved cross-project defect prediction.

The remainder of the paper is structured as follows: In Section II, we discuss related works of cross-project defect prediction, and briefly present the background of the Naïve Bayes classifier and credibility theory. The details of our proposed approach are explained in Section III. We also describe the experimental setup in Section IV and the empirical result in Section V. We discuss and analyze the result in Section VI. The threats to validity of our experiments and results are discussed in Section VII, and we finally conclude with a summary and probable future work in Section VIII.

II. Background

A. Cross-project defect prediction

Many researchers have conducted studies on cross-project defect prediction [10], [11], [5], [6], [12], [13], [14], [15]. Zimmermann, et al. [5] conducted a total of 622 cross-project predictions, but unfortunately, only 3.4% of the prediction proved successful. Turhan, et al. [6] conducted cross-project defect prediction experiments on 10 datasets from NASA and SOFTLAB. They concluded that the high false positive rate that accompanies cross-project predictions makes CPDP impractical. Furthermore, they also proposed NN-filter to aid reduce the high false alarms. The NN-filter uses the k-nearest neighbor method to select training instances, which are similar to the unlabeled test instance and thus, irrelevant instances and information are eliminated for a better cross-project defect prediction. Ma, et al. [12] also proposed Transfer Naïve Bayes (TNB) to reweight the training data instead of removing the noise training instances. The authors simulated Newton's universal gravitational law to assign weighting values to the training data, and then, apply the weights to prior probability and conditional probability to predict defects. TNB was validated on 10 datasets from NASA and SOFTLAB. Nam, et al. [13] proposed TCA+, which is an extension of Transfer Component Analysis (TCA). TCA [16] was proposed earlier as a transfer learning method that is sensitive to normalization selections. Their new proposed algorithm TCA+ provides special rules for selecting the better normalization options for a given pair. The selected normalization option is used to preprocess the data and predict defects.

B. Naïve Bayes classifier

Naïve Bayes [6] is a probabilistic classifier based on the Bayes' theorem, which assumes strong independence among features. The method has been successfully applied to real-world applications. According to the Bayes' theorem, the formula used for Naïve Bayes classifier to calculate the probability of each class is

$$p\left(c\left|X_1,X_2,X_3,...,X_n\right.\right)=\frac{p(c)\,p\left(X_1,X_2,X_3,...,X_n\left|c\right.\right)}{p\left(X_1,X_2,X_3,...,X_n\right)},$$

where c is a class, and $X_1,X_2,X_3,...,X_n$ are the feature values of the predictors for the given test case. For numeric features, Gaussian probability density function is used to compute the conditional probability

$p\left(X_i\left|c\right.\right)=\frac{1}{\sqrt{2\pi\sigma_c^2}}e^{\frac{-(X_i-\mu_c)^2}{2\sigma_c^2}}$, where X_i is the feature value,

μ_c is the mean of the value in feature x associated with class c, and σ_c^2 is the variance of feature value x associated with class c.

Assuming conditional independence between the predictors, the formula is converted into

$p\left(c\left|X_1,X_2,X_3,...,X_n\right.\right)=\frac{p(c)\prod_{i=1}^{n}p\left(X_i\left|c\right.\right)}{p\left(X_1,X_2,X_3,...,X_n\right)}$. We adopt the Naïve Bayes approach and implement it for model construction. The approach estimates the probabilities from the training sample, and outputs the class probabilities (defective or non-defective) for each test case. The output label with the highest class probabilistic value is used to label the predicted class for the sample

C. Credibility theory

Credibility theory for a normal distribution scenario [17] is introduced in our proposed algorithm to reweight the value of each attribute by combining the target and source mean and their respective standard deviations to transfer the target data knowledge into source data. The theory, which is widely used in the actuary and insurance industry provides a mechanism to combine individual and collective claims experience, where the approach belongs to Bayesian statistics. Insurance companies use the theory as a rating method to decide the price of the premium for certain sets of insurance contracts. The formula is written as below, A is the adjusted mean, $\hat{A}$ is the individual mean, and $\bar{A}$ is the collective mean:

$$A = Z\hat{A} + (1 - Z)\bar{A} \tag{1}$$

Credibility factor Z is a number between 0 and 1. Intuitively, if Z is 0, the credibility of individual mean is low so collective mean is used only but individual mean is taken into account; if Z is 1, the credibility of individual mean is high so collective mean is not considered but individual mean is used only. It defines the degree of trust for the individual mean, where N is the number of sample in individual data, σ is the standard deviation of individual data, and τ is the standard deviation of collective data. The formula for normal distribution scenarios is as follows:

$$Z = \frac{N}{N + \left(\frac{\sigma}{\tau}\right)^2} \tag{2}$$

To apply the theory for prediction, the following steps are implemented: compute the credibility factor Z for the source and target data (similar to *Eq* (2)); calculate the weighted mean and standard deviation of each attribute; input them into Gaussian Naive Bayes for prediction so that a more desirable performance in terms of G-mean could be obtained..

III. Approach

In this section, we split our approach into 4 parts and present details for each part. The parts include (1) collection of data information, (2) degree of reweighting estimation, (3) reweighting for training data, (4) prediction. Fig. 1 is the pseudocode of the proposed approach to outline the detail of the approach.

```
Input TRAIN and TEST // both share the same features
LEARNER = [Gaussian Naïve Bayes]
    [POS, NEG] = split TRAIN by class
    [POS_T_MEAN, NEG_T_MEAN] = mean of each feature in
[POS, NEG]
    [POS_T_SD] = sd of each feature in [POS]
    TEST_MEAN = mean of each feature in TEST
    TEST_SD = sd of each feature in TEST
    POS_ADJ_MEAN = []
    NEG_ADJ_MEAN = []
    ADJ_SD = []
    FOR EACH train_mean, train_sd, test_mean, test_sd in
[POS_T_MEAN, POS_T_SD, TEST_MEAN, TEST_SD]
        Z = train_mean / (train_mean + (train_sd/test_sd) ^2)
        adjusted_mean = Z * train_mean + (1 - Z) * test_mean
        Append adjusted_mean into POS_ADJ_MEAN
    END FOREACH
    FOR EACH train_mean, train_sd, test_mean, test_sd in
[NEG_T_MEAN, NEG_T_SD, TEST_MEAN, TEST_SD]
        Z = train_mean / (train_mean + (train_sd/test_sd) ^2)
        adjusted_mean = Z * train_mean + (1 - Z) * test_mean
        Append adjusted_mean into NEG_ADJ_MEAN
    END FOREACH
    FOR EACH train_mean, train_sd, test_mean, test_sd in
[POS_T_MEAN, POS_T_SD, TEST_MEAN, TEST_SD]
        Z = train_mean / (train_mean + (train_sd/test_sd) ^2)
        adjusted_sd = Z * train_sd + (1 - Z) * test_sd
        Append adjusted_sd into ADJ_SD
    END FOREACH
    PREDICTOR = Train LEARNER with [POS_ADJ_MEAN,
NEG_ADJ_MEAN, ADJ_SD]
    PREDICTED_OUTCOME = PREDICTOR on TEST
END
```

Fig. 1. Pseudocode for win/tie/loss evaluation with wilcoxon signed rank test, where A and B are the performance measures

A. Data collection

Assuming that all features are numeric variables, for the training set, we calculate the mean and standard deviation *Train* (μ, σ) of each feature for all training records belonging to the same class label. Then, we compute the mean and standard deviation *Test* (μ, τ) of each feature for all testing records. By adopting credibility theory, we consider the *Train* (μ, σ) as the collective data, while *Test* (μ, τ) *is* considered as the individual data.

B. Degree of reweighting estimation

With a view to transfer the knowledge from test data, the credibility factor Z from credibility theory is introduced to determine the credibility of training data and testing data. If credibility factor Z is 0, the source data is not considered but the target data is used only; if Z is 1, the source data is only used however, the target data is not considered. Credibility factor Z is used to balance the sampling error of test data and modeling error of training data. By adopting the formula from credibility theory, credibility factor Z is calculated as presented in equation 3, where μ is mean of train data, σ is the standard deviation of train data, and τ is the standard deviation of test data:

$$Z = \frac{\mu}{\mu + \left(\frac{\sigma}{\tau}\right)^2} \tag{3}$$

Therefore, the credibility factor Z for each attribute in training data is computed to determine the degree of reweighting for the training data individually.

C. Training data reweighting

For each feature of all the training data belonging to the same class, adjusted mean and adjusted standard deviation are computed as a result of reweighting. The reweighting mechanism is similar to (1), A is the adjusted mean, $\hat{A}$ is the individual mean, and $\bar{A}$ is the collective mean:

$$A = Z\bar{A} + (1 - Z)\hat{A} \tag{4}$$

The difference between the original formula and the proposed one is the swapping of the individual and collective mean. This is done because the training data is large and labeled while the test data is a small and unlabeled. In this study, the defectiveness of the test data remains uncertain which is different from actuarial credibility theory. Such change can help the source data to better understand the structure of test data without sacrificing the known knowledge for defective modules.

D. Prediction

The weighted mean, weighted standard deviation and the value of the test data are used as the input parameters for the Naïve Bayes classifier during model construction and finally used for predicting each instance of test data.

IV. Experimental setup

A. Research questions

To analyze whether credibility theory can work well with Naïve Bayes, we integrate these two distinct concepts to achieve a better cross-project defect prediction result. To evaluate the credibility theory based Naïve Bayes classifier, we formulate two specific research questions (RQs) as follow:

RQ1: Is the credibility theory based Naïve Bayes classifier a feasible solution?

RQ2: Can the credibility theory based Naïve Bayes classifier improve the performance of cross-project defect prediction?

B. Datasets

For the experiment, the existing software defect dataset from tera-PROMISE repository [18] are used. Table I provides the overview of selected datasets. Eleven datasets are selected from 3 different projects: *velocity*, *xalan*, and *xerces*. These datasets were used for similar defect prediction studies [19, 20], [21], [22] and comprise of CK object-oriented metrics [2] and cyclometic complexity. [4] These 11 datasets share the same metric set.

TABLE I. 11 DATASETS FROM 3 PROJECTS

Project	Dataset	# of instance	
		All	***Defective (%)***
velocity	velocity-1.4	196	147 (75%)
	velocity-1.5	214	143 (66.35%)
	velocity-1.6	229	78 (34.06%)
xalan	xalan-2.4	723	110 (15.21%)
	xalan-2.5	803	387 (48.19%)
	xalan-2.6	885	411 (46.44%)
	xalan-2.7	909	898 (97.35%)
xerces	xerces-init	162	77 (47.53%)
	xerces-1.2	440	71 (16.13%)
	xerces-1.3	453	69 (15.23%)
	xerces-1.4	588	437 (74.19%)

C. Performance evaluation

Confusion matrix is usually used to validate and evaluate the performance of defect prediction models. Some terminologies in the matrix are presented below:

True positive (TP): The number of defective modules that has been classified correctly.

True negative (TN): The number of non-defective modules that has been classified correctly.

False positive (FP): The number of non-defective modules that has been misclassified as defective

False negative (FN): The number of defective modules that has been misclassified as non-defective

To evaluate the performance of the approaches, we use three commonly used defect prediction performance metrics: recall (*pd*), probability of false alarms (*pf*), and G-mean. Recall, also known as probability of detection, is used to measure the ability of the model in identifying defective module. It is defined as the probability of true defective modules with respect to the total number of defective module. The measurement ranges from 0 to 1. A higher value of pd is more desirable. The formula is defined as follow:

$$pd = \frac{TP}{TP + FN} \tag{5}$$

False positive rate, which can be called probability of false alarm, is applied to measure the rate of misclassifying non-defective module to defective module. It is defined as the probability of false positive modules with respect to non-defective modules. The model performs better with lower *pf value*, where the value of pf is between 0 and 1.

$$pf = \frac{FP}{FP + TN} \tag{6}$$

Menzies et al. [20] found that precision is less stable than other measures. Therefore, f-measure is not included in this paper to assess the performance of predictors because it measures the tradeoff between precision and recall. Instead, the G-mean [23] performance measure, which is more comprehensive, is used. It is a geometric mean of true positive and true negative. A high G-mean indicates high accuracies on both defective and non-defective class. The formula is presented in equation (7):

$$\text{G-mean} = \sqrt{pd\left(1 - pf\right)} \tag{7}$$

For statistical validation, we use Win/Tie/Loss evaluation and conduct the Wilcoxon signed-rank test [24] ($p < 0.05$) for all the performance values in performance metrics. If the proposed approach outperforms another approach, record a 'Win' is recorded for the proposed approach. Similarly, a 'Loss' is recorded if the approach is underperformed with statistical significance, and record a 'Tie' if there is no difference between two approaches.

```
WIN = 0, TIE = 0, LOSS = 0
if WILCOXON-SIGNED-RANK(A, B) indicates they are the same
  TIE = TIE + 1
else
  if A have outperformed B
    WIN = WIN + 1
  else
    LOSS = LOSS + 1
  end if
end if
```

Fig. 2. Pseudocode for win/tie/loss evaluation with wilcoxon signed rank test, where A and B are the performance measures

D. Experimental procedure

To empirically validate our approach, we conduct extensive cross-project defect predictions by comparing credibility theory based Naïve Bayes classifier to normal Naïve Bayes classifier, and NN-filtering. All versions of each project are combined to form the source data which is used exactly once as the training set. The remaining dataset is used as test data to evaluate the performance of the three approaches.

Each approach is repeated 30 times and the average result is then computed. For all the approaches in every iteration, 90% of source data is selected by stratified sampling to retain the class distribution and allows randomness. In addition, Menzies, et al. [25] noted that the particular attributes used are less important than how the attributes are actually used to construct predictive model. Therefore, all the attributes in the datasets are used as long as they share the same feature.

V. Results

A. Comparison on cross-project prediction performance

Table 2 shows the average prediction performance of G-mean among the three approaches: credibility theory based Naïve Bayes (CNB), normal Naïve Bayes (NB), and NN-filtering with NB (NN-filter+NB). The last row contains the average G-mean of all combinations for each approach. On the other hand, the other rows are the average G-mean in every cross-project prediction combinations. The values highlighted in bold indicate the best performance value among the three approaches.

TABLE II. Cross-project Defect Prediction Result for G-mean (Average)

Source => Target	CNB	NB	NN-filter+NB
velocity => xerces-init	**0.5114499**	0.4139677	0.445485
velocity => xerces-1.2	**0.49224855**	0.366471	0.446657
velocity => xerces-1.3	**0.5719078**	0.50513	0.45269
velocity => xerces-1.4	**0.4956307**	0.3669067	0.448052
velocity => xalan-2.4	**0.6000741**	0.558758	0.449384
velocity => xalan-2.5	**0.5270466**	0.447738	0.44963
velocity => xalan-2.6	**0.5746191**	0.5633498	0.4611789
velocity => xalan-2.7	**0.595594**	0.450763	0.44135
xalan => velocity-1.4	**0.460173**	0.238345	0.2895122
xalan => velocity-1.5	**0.568807**	0.296586	0.293289
xalan => velocity-1.6	**0.6264977**	0.3658755	0.2895122
xalan => xerces-init	0.47429	0.443625	**0.506352**
xalan => xerces-1.2	0.476585	0.446137	**0.50732313**
xalan => xerces-1.3	**0.619027**	0.5261101	0.50812
xalan => xerces-1.4	**0.5455225**	0.38047	0.505782
xerces => velocity-1.4	0.407939	0.374136	**0.473804**
xerces => velocity-1.5	**0.589498**	0.47379	0.47608072
xerces => velocity-1.6	**0.59590344**	0.5524338	0.47800497
xerces => xalan-2.4	0.6521600982	**0.66872**	0.525632
xerces => xalan-2.5	**0.542417559**	0.5316687	0.52638
xerces => xalan-2.6	**0.607400446**	0.558188	0.525196
xerces => xalan-2.7	**0.65468**	0.603394	0.52465
Average	**0.554067**	0.450571	0.455639

From table 2, an improvement is observed in most of the cases, and we observe a significant improvement on average score of CNB (0.554) compared to NN-filter (0.4556) and NB (0.450571). Prediction performance for most projects improved remarkably, an example is the xalan => velocity-1.6, the G-mean increased from 0.3658 (NB) to 0.6264 (CNB).

The G-mean results demonstrate that the reweighting mechanism stemmed from credibility theory can help to reduce the differences in feature distributions between source and target data in order to enhance the cross-project prediction ability. Although there are 4 out of 22 cases that CNB does not yield the best performance, CNB was not the worst approach in any of the cases.

Additionally, the result shows that discarding training instances may affect the performance adversely because some critical data information may be filtered out. NN-filter+NB yielded a worse performance than NB in some situations such as, velocity => xerces-1.3, velocity => xalan-2.4 etc.

In Section VI, we discuss and analyze why the introduction of credibility theory can help reduce the difference between training and test data.

B. Win/Tie/Loss Results

We tabulate the Win/Tie/Loss result of CNB with statistical significance ($p < 0.05$) against NB and NN-filter+NB in Table 3. The experiment conducted 22 prediction combinations because dataset from the same project combines into a large training set so as to generate a more mature source data for test data. In Table 3, CNB does not underperform in any combinations against NB and NN-filter+NB. CNB outperforms NB in all combinations, but it ties with NN-filter in some target sets such as velocity-1.4 and xerces-init.

TABLE III. WIN/TIE/LOSS RESULT IN G-MEAN OF CNB AGAINST NB AND NN-FILTER

Source => Target	Against					
	NB			NN-filter+NB		
	Win	Tie	Loss	Win	Tie	Loss
velocity-1.4	2	0	0	1	1	0
velocity-1.5	2	0	0	1	1	0
velocity-1.6	2	0	0	1	1	0
xalan-2.4	2	0	0	2	0	0
xalan-2.5	2	0	0	2	0	0
xalan-2.6	2	0	0	2	0	0
xalan-2.7	2	0	0	2	0	0
xerces-init	2	0	0	1	1	0
xerces-1.2	2	0	0	1	1	0
xerces-1.3	2	0	0	1	1	0
xerces-1.4	2	0	0	1	1	0
Total	22 100%	0 0%	0 0%	15 68%	7 32%	0 0%

Overall, CNB can achieve better or comparable performance against NB and NN-filter+NB. Against NB, all prediction combinations recorded 'Win' results. Against NN-filter+NB, 15 (68%) prediction combinations are 'Win' results, while the remaining 7 (32%) prediction combinations are 'Tie' results.

VI. DISCUSSION

A. Feasibility of the Algorithm

To discuss the feasibility of the credibility theory based Naïve Bayes classifier (RQ1), we investigate the attributes in training data as the approach reweighted the class conditional probability of each attribute. In Figure 3 and Figure 4, the probability density function graph is used to represent the class conditional probability of the attribute - "loc" for each class. The red one represents the normal distribution of defective instances; while the blue one represents the normal distribution of non-defective instances. The x-axis is the value of the attribute; and the y-axis is the value of likelihood.

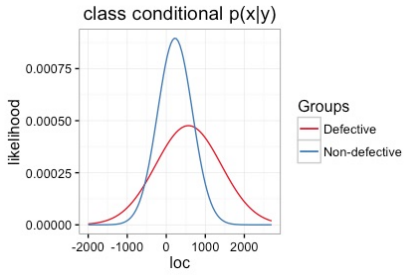

Fig. 3. Class conditional probability of loc in NB from xalan => velocity-1.6 (G-mean = 0.3658)

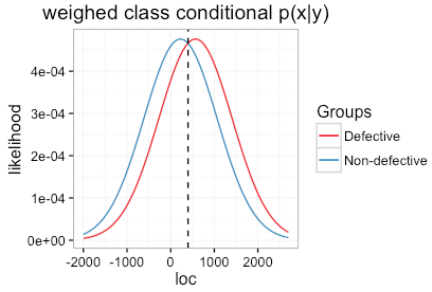

Fig. 4. Class conditional probability of loc in CNB from xalan => velocity-1.6 (G-mean = 0.6264)

Figure 3 shows that the distribution of defective instances is sparse and most of the defective instances overlap with the non-defective instances. In addition, it is observed that the decision boundary is difficult to determine. Hence, the Naïve Bayes classifier can easily get confused with defective instances, and eventually misclassify them as non-defective. The situation in Figure 3 reflects in the case of xalan => velocity-1.6, where Naïve Bayes yielded poor G-mean result of 0.3658.

Nevertheless, the performance is boosted with the support from credibility theory. The reweighting mechanism reshapes the bell curve with the knowledge from test data. In Figure 4, the decision boundary is more ideal for the classifier to identify defective modules. Therefore, CNB scored relatively higher G-mean of 0.6264 in the prediction combination of xalan => velocity-1.6.

Reshaping the class conditional probability density of each feature can help the classifier reconstruct a common data distribution between training data and test data. Such a situation can be observed in other results. The feature's distribution and the experiment result answered RQ1 that this composition is a feasible solution because the approach does not underperform against NB and NN-filter+NB in the 22 cases.

B. Cross-project Prediction Performance

Tables 2 and 3 show prediction result of CNB against NB and NN-filter. By a pair-wise comparison with NB, CNB recorded a significant 'Win' result of 100% in terms of G-mean, and it shows an improvement in cross-project prediction with a higher G-mean in 21 out of 22 combinations. A higher G-mean indicates the model have a more all-round performance to predict both classes correctly as G-mean is a dual assessment of true positive and true negative rates.

For CNB vs NN-filter, the result is less dominating compared to CNB vs NB because NN-filter is one of the existing cross-project prediction approach. However, the result could prove CNB possessed a better cross-project predictability. Its Win results accounts for 68% in all combinations, and the remaining were 'Tie' results. In other words, CNB did not record a 'Loss' result against NN-filter+NB. Hence, we answer RQ2 by demonstrating the improvement in prediction performance of CNB over the two other prediction approaches from the experimental results.

VII. Threats to Validity

We present the conclusion, internal, construct, and external threats to validity in this section. The conclusion validity relates to the ability to draw significant correct conclusions from our experiments and results; regarding which, we appropriately applied statistical tests to show the statistical significance for the obtained results. Moreover, we used relatively large datasets to mitigate the threats related to the number of observations composing the datasets.

The construct validity refers to the agreement between a theoretical concept and a specific measurement. It establishes correct operational measures for the concepts being studied. We made use of different performance metrics and 11 public datasets in the study. The datasets in our study have been previously used in many other empirical studies carried out to evaluate defect prediction.

The Gaussian distribution assumption from Naïve Bayes and credibility theory may be violated in some datasets, and the feature independence assumption from Naïve Bayes ignored the attribute correlations. These violations may pose potential threats to internal validity. The threats to external validity are partially controlled in this study. Datasets from different projects are examined in the study. One particular issue is that we did not consider using more different cross-project prediction methods. This shall be included in a future study.

VIII. Conclusion

In this paper, credibility theory based Naïve Bayes classifier is introduced for cross-project defect prediction. Credibility theory based Naïve Bayes classifier is a novel approach to reweight features' mean and standard deviation with the knowledge from the test data to reshape the class conditional distribution in each feature. The study demonstrates the feasibility of the solution and its usefulness in cross-project defect prediction.

We ran a total of 22 cross-project predictions and found that credibility theory based Naïve Bayes significantly outperforms the standard Naïve Bayes in all 22 combinations. Also, the proposed approach significantly outperforms the NN-filter+NB in 68% of the cases, and recorded ties in the remaining cases. In other words, the proposed approach does not underperform against Naïve Bayes, the commonly used machine learning approaches and NN-filter, a commonly used cross-project defect prediction approach. Additionally, some combinations recorded a remarkable improvement such as velocity => xalan-2.7 (from 0.44135 to 0.595594), xalan => velocity-1.6 (from 0.2895122 to 0.6264977), and xerces => velocity-1.5 (from 0.47379 to 0.589498). Two findings from our experiments are noteworthy:

1. The empirical results show that the credibility theory helps to find a shared and clear data distribution between projects so that defect prediction can work across projects which are different in nature. Additionally, application of the theory yields a better performance in terms of G-mean. Therefore, the credibility theory based Naïve Bayes is a feasible solution.

2. The credibility theory based Naïve Bayes classifier can yield a better prediction performance according to the empirical result. The G-mean result indicated the approach can strike a balance between probability of detection and false positive rate, and most importantly, the approach outperformed NN-filter+NB and Naïve Bayes, a standard machine learner in most of the prediction combinations. Although there is an improvement in cross-project defect prediction, there are several areas we intend to consider improving this work. Comparison with state-of-the-art technique is needed. Some state-of-the-art techniques such as TNB, TCA+ has not been compared to our technique. Therefore, the ability of the approach has not been evaluated completely. We intend to validate our approach on more software project datasets. Three projects from the PROMISE repository were used to evaluate the proposed approach. Several projects could be extracted from the PROMISE repository and other similar repositories to aid generalize our results.

Acknowledgement

This work is supported in part by the General Research Fund of the Research Grants Council of Hong Kong (No. 125113, 11200015 and 11214116), and the research funds of City University of Hong Kong (No. 7004683 and 7004474).

REFERENCES

[1] H. Wang, "Software Defects Classification Prediction Based On Mining Software Repository," ed, 2014.

[2] S. R. Chidamber and C. F. Kemerer, "A metrics suite for object oriented design," *IEEE Transactions on Software Engineering,* vol. 20, pp. 476-493, 1994.

[3] M. H. Halstead, *Elements of software science (Operating and programming systems series)*: Elsevier Science Inc., 1977.

[4] T. J. McCabe, "A complexity measure," *IEEE Transactions on Software Engineering,* vol. 2, pp. 308-320, 1976.

[5] T. Zimmermann, N. Nagappan, H. Gall, E. Giger, and B. Murphy, "Cross-project defect prediction: a large scale experiment on data vs. domain vs. process," in *Proceedings of the the 7th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, 2009, pp. 91-100.

[6] B. Turhan, T. Menzies, A. B. Bener, and J. Di Stefano, "On the relative value of cross-company and within-company data for defect prediction," *Empirical Software Engineering,* vol. 14, pp. 540-578, 2009.

[7] J. Garrido and J. Zhou, "Full credibility with generalized linear and mixed models," *ASTIN Bulletin: The Journal of the IAA,* vol. 39, pp. 61-80, 2009.

[8] J. Nelder and R. Verrall, "Credibility theory and generalized linear models," *Astin Bulletin,* vol. 27, pp. 71-82, 1997.

[9] J. Garrido and J. Zhou, "Credibililty Theory for Generalized Linear and Mixed Models," 2006.

[10] X. Xia, D. Lo, S. J. Pan, N. Nagappan, and X. Wang, "Hydra: Massively compositional model for cross-project defect prediction," *IEEE Transactions on Software Engineering,* vol. 42, pp. 977-998, 2016.

[11] Y. Zhang, D. Lo, X. Xia, and J. Sun, "An empirical study of classifier combination for cross-project defect prediction," in *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, 2015, pp. 264-269.

[12] Y. Ma, G. Luo, X. Zeng, and A. Chen, "Transfer learning for cross-company software defect prediction," *Information and Software Technology,* vol. 54, pp. 248-256, 2012.

[13] J. Nam, S. J. Pan, and S. Kim, "Transfer defect learning," in *the 35th International Conference on Software Engineering (ICSE'13)*, San Francisco, CA, USA, 2013, pp. 382-391.

[14] S. Watanabe, H. Kaiya, and K. Kaijiri, "Adapting a fault prediction model to allow inter languagereuse," in *Proceedings of the 4th international workshop on Predictor models in software engineering*, 2008, pp. 19-24.

[15] J. Nam and S. Kim, "CLAMI: Defect Prediction on Unlabeled Datasets (T)," in *Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on*, 2015, pp. 452-463.

[16] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks,* vol. 22, pp. 199-210, 2011.

[17] H. Bühlmann and A. Gisler, *A course in credibility theory and its applications*: Springer Science & Business Media, 2006.

[18] T. Menzies, R. Krishna, and D. Pryor, "The PROMISE Repository of empirical software engineering data," ed. http://openscience.us/repo: North Carolina State University, Department of Computer Science, 2016.

[19] M. Jureczko and L. Madeyski, "Towards identifying software project clusters with regard to defect prediction," in *the 6th International Conference on Predictive Models in Software Engineering (PROMISE'10)*, Timisoara, Romania, 2010, p. 9.

[20] T. Menzies, A. Dekhtyar, J. Distefano, and J. Greenwald, "Problems with precision: A response to "Comments on 'Data mining static code attributes to learn defect predictors'"," *IEEE Transactions on Software Engineering,* vol. 33, 2007.

[21] M. Jureczko, "Significance of different software metrics in defect prediction," *Software Engineering: An International Journal,* vol. 1, pp. 86-95, 2011.

[22] M. Jureczko and D. Spinellis, "Using object-oriented design metrics to predict software defects," *Models and Methods of System Dependability. Oficyna Wydawnicza Politechniki Wrocławskiej,* pp. 69-81, 2010.

[23] S. Wang and X. Yao, "Using class imbalance learning for software defect prediction," *IEEE Transactions on Reliability,* vol. 62, pp. 434-443, 2013.

[24] L. Chen, B. Fang, Z. Shang, and Y. Tang, "Negative samples reduction in cross-company software defects prediction," *Information and Software Technology,* vol. 62, pp. 67-77, 2015.

[25] T. Menzies, J. Greenwald, and A. Frank, "Data mining static code attributes to learn defect predictors," *IEEE Transactions on Software Engineering,* vol. 33, pp. 2-13, 2007.