Addressing Noise and Class Imbalance Problems in Heterogeneous Cross-Project Defect Prediction:
An Empirical Study

Rohit Vashisht, Jamia Millia Islamia, New Delhi, India & KIET Group of Institutions, Delhi-NCR, Ghaziabad, India*

Syed Afzal Murtaza Rizvi, Jamia Millia Islamia, India

ABSTRACT

When a software project either lacks adequate historical data to build a defect prediction (DP) model or is in the initial phases of development, the DP model based on related source project's defect data might be used. This kind of SDP is categorized as heterogeneous cross-project defect prediction (HCPDP). According to a comprehensive literature review, no research has been done in the field of CPDP to deal with noise and class imbalance problem (CIP) at the same time. In this paper, the impact of noise and imbalanced data on the efficiency of the HCPDP and with-in project defect prediction (WPDP) model is examined empirically and conceptually using four different classification algorithms. In addition, CIP is handled using a novel technique known as chunk balancing algorithm (CBA). Ten prediction combinations from three open-source projects are used in the experimental investigation. The findings show that noise in an imbalanced dataset has a significant impact on defect prediction accuracy.

KEYWORDS

Classification, Cross-Project, Defect, Heterogeneous, Imbalance, Noise, Regression, With-In

INTRODUCTION

Software has become an essential part of everyone's daily life in today's digital era. Even a minor flaw or malfunction in this software might result in financial or even life-threatening losses. Inconsistencies, ambiguities or misinterpretation of the specifications, carelessness or negligence in writing code, insufficient testing, unsuitable or unanticipated use of the software, or other unforeseen issues can all cause software errors. Software testing should be done at the proper time in the early stages of Software Development Life Cycle (SDLC) in order to reduce overall software development cost. The SDLC software testing phase, on the other hand, accounts for 60% of the total cost of software development. As a result, it's vital to do testing on the appropriate modules at the appropriate time.

Software Defect Prediction (SDP) can be broadly split into two classes, according to the state of the art: Within Project Defect Prediction (WPDP) and Cross Project Defect Prediction (CPDP). The available defect dataset is split into two parts in WPDP in order to build the DP model in such a way that one half of the dataset (referred to as labeled observations) is used to train the DP model and the other portion is used to validate the DP model, as illustrated in Figure 1.Finding labels that

are either faulty or non-faulty for unidentifiable instances in the target dataset is how the DP model is tested (Ambros et al., 2012).

Figure 1. With-In project defect prediction
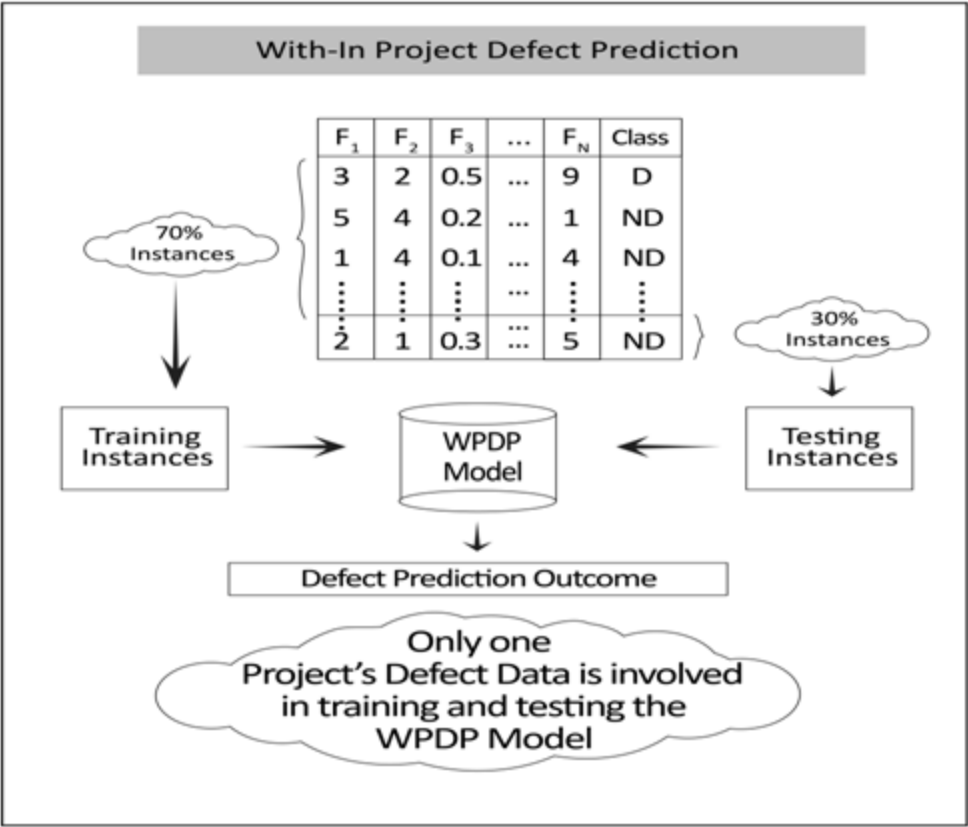
CPDP is a type of SDP in which software projects that lack the required local defect data can develop an accurate and effective DP model using data from other projects. CPDP can also be divided into two subcategories: Homogeneous CPDP (HoCPDP) and Heterogeneous CPDP (HCPDP). HoCPDP collects common software measures/features from both the source (whose defect data is used to train the SDP model) and the target (for which the SDP model is created) applications (He et al., 2014). When using HCPDP, however, there are no uniform metrics between the prediction pair datasets. Uniform features between two applications can be determined by evaluating the coefficient of correlation between all possible software feature combinations. In the case of HCPDP, combinations of feature pairs with a similar distribution in their values are employed as common features between source and target datasets in order to forecast project-wide problems. As shown in Figure 2, correlated feature pairs for the HCPDP category include (A, Q), (B, P), and (D, S). Figure 2 provides more details on both CPDP groups.

Related Content

A Trination Analysis of Social Exchange Relationships in E-Dating
Sudhir H. Kaleand Mark T. Spence (2009). *E-Collaboration: Concepts, Methodologies, Tools, and Applications (pp. 842-856).*
www.irma-international.org/chapter/trination-analysis-social-exchange-relationships/8834

How Likely is Simpson's Paradox in Path Models?
Ned Kock (2015). *International Journal of e-Collaboration (pp. 1-7).*
www.irma-international.org/article/how-likely-is-simpsons-paradox-in-path-models/121624

The Impact of the E-Collaborative and Traditional Learning Styles on Learning Outcomes and Anxiety
Yu Zhonggen (2016). *International Journal of e-Collaboration (pp. 27-47).*
www.irma-international.org/article/the-impact-of-the-e-collaborative-and-traditional-learning-styles-on-learning-outcomes-and-anxiety/159169

E-Mail Reflection Groups as Collaborative Action Research
Robert J. Redmon Jr. (2009). *Handbook of Research on Electronic Collaboration and Organizational Synergy (pp. 349-361).*
www.irma-international.org/chapter/mail-reflection-groups-collaborative-action/20184

Urban Economy and Sources of Its Efficiency as Factors Addressing the Challenges Faced by Urban Economy: The Case Study of Southern African Region
Evelyn Chiloane-Tsokaand Junior Mabiza-ma Mabiza (2018). *E-Planning and Collaboration: Concepts, Methodologies, Tools, and Applications (pp. 729-749).*
www.irma-international.org/chapter/urban-economy-and-sources-of-its-efficiency-as-factors-addressing-the-challenges-faced-by-urban-economy/206032