Noname manuscript No.
(will be inserted by the editor)

Supplemental Material

Online Cross-Project Approach with Project-Level Similarity for Just-In-Time Software Defect Prediction

Cong Teng · Liyan Song · Xin Yao

the date of receipt and acceptance should be inserted later

Abstract This document presents additional information of the submitted paper "Online Cross-Project Approach with Project-Level Similarity for Just-In-Time Software Defect Prediction".

1 Related Work

We report a summary table of related works discussed in Section 2.2 and 2.3 of this paper to assist readers to better track the related works discusses in this paper. In the table, the "CP Method" column lists the CP methods that were proposed or adopted in the literature. The "Project-level Similarity Metric" reports the project-level similarity metrics the paper used. "N/A" means there are no project-level similarity metrics adopted in the paper. The "CP Performance Conclusion" column summarizes related conclusions with respect to the CP methods.

Cong Teng (12132358@mail.sustech.edu.cn)
Research Institute of Trustworthy Autonomous Systems, Southern University of Science and Technology, China and Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, China.

Liyan Song (songly@hit.edu.cn)
Faculty of Computing, Harbin Institute of Technology, Harbin, China.

Xin Yao (xinyao@ln.edu.hk)
School of Data Science, Lingnan University, Hong Kong SAR, China.

* the corresponding authors.

Table 1: Summary of related work in Section 2.2 and 2.3 of the paper.

Paper	CP Method	Project-level Similarity Metric	Online or Offline	#Project	CP Performance Conclusion
Fukushima et al. (2014)	Project-level: Random, Similarity, Ensemble	Spearman Correlation	Offline	11 open source projects, of which 6 projects (Bugzilla, Columba, Eclipse JDT, Mozilla, Eclipse Platform, PostgreSQL) are provided by Kamei et al. (2013) and 5 well-known and long-lived projects (Gimp, Maven-2, Perl, Ruby on Rails, Rhino)	Strong within-project performance of a model does not necessarily indicate that it will perform well in a cross-project context; Similar predictor-dependent variable correlations tend to produce cross-project models that perform well in a cross-project context; Ensemble learning methods tend to produce JIT defect models that perform well in a cross-project context.
Kamei et al. (2016)	Data-level: Global; Project-level: Random, Similarity, Ensemble	domain-agnostic: Spearman Correlation; domain-aware: Company,Intended audience, User interface, Product uses database, Programming language	Offline	The same 11 open source projects as Fukushima et al. (2014)	Using similarity to select JIT models performs worse than WP; Merging all training data into a large pool or combining the predictions of several models can not get a significant better performance than WP.
Cho et al. (2018)	Project-level: Filterd-WP	The number of developers in both sub-projects	Offline	5 open-source projects (Android, Appium, Cordova, Maven, Tizen) that are divided into multiple sub-projects	A JIT-SDP model built with the training data from the same project is preferred over a model built with the training data from other projects; The amount of training data is not the only factor that affects the cost effectiveness of cross-sub-project models; A developer similarity that counts the number of developers in both sub-projects is the most preferred way of filtering out irrelevant sub- projects in building filtered-within-project models.
Shehab et al. (2022)	Project-level: ClusterCommit	Common libraries	Offline	26 Apache Foundation open-source projects written in Java and built using the Maven dependency manager	ClusterCommit yields promising results.

Table 1 continued from previous page

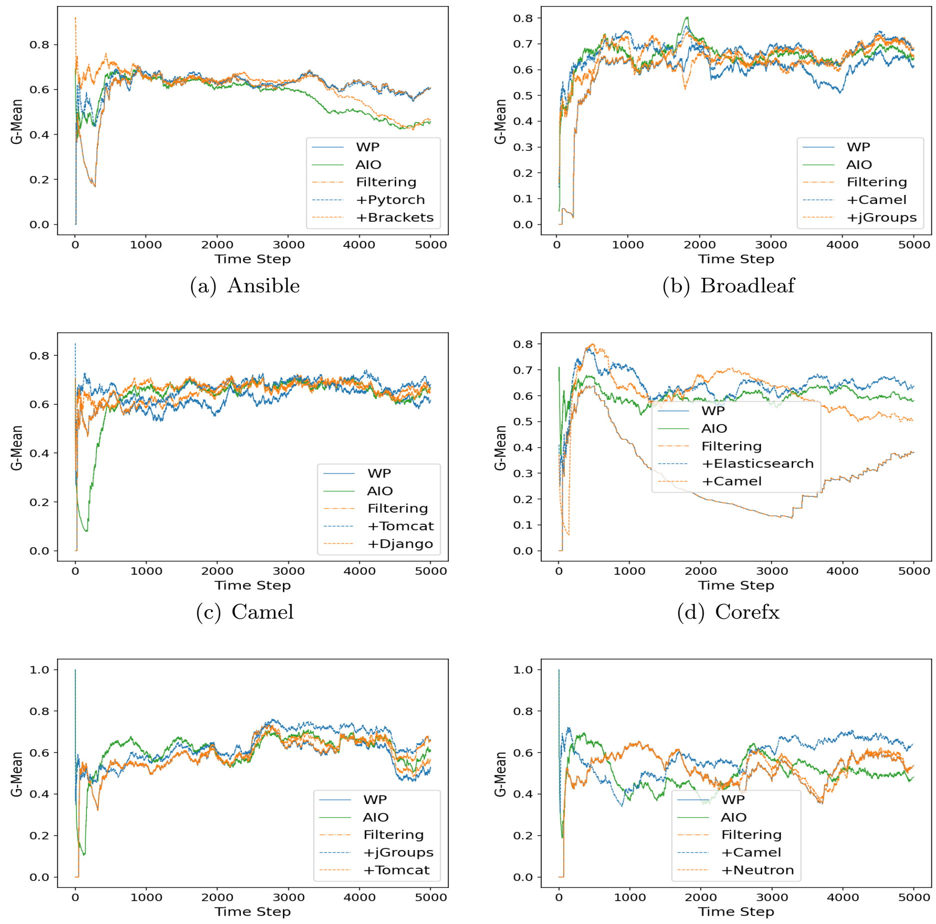
Paper	CP Method	Project-level Similarity Metric	Online or Offline	#Project	CP Performance Conclusion
Zhang et al. (2022)	Data-level: Data-Merging; Project-level: Bellwether+, Random, Similarity, Ensemble, Feature-based ENSEmble modeling approach (FENSE)	Model-level: n_commits_src, model_performance_src, defect_ratio_src; Project-level: project_popularity, project_age, same_owner_type, same_license, same_language, textual_similarity; Social-level: n_core,n_external, core_diff,external_diff, contributor_intersection, entropy_diff; Technical-level: code_size,code_size_diff, n_dependencies, dependency_intersection, dependency_diff.	Offline	113,906 transferring pairs of 338 projects	Project metrics play an important role in cross-project JITDP, especially the programming language and the defect ratio of the source project; The proposed FENSE outperforms other cross-project approaches on 5 out of 6 metrics; The idea of model-level reusing is encouraging because both Bellwether+ and FENSE can select a single model with high performance in a cross-project context.
Yang et al. (2019)	Data-level: Global, Local.	N/A	Offline	6 open source projects (Bugzilla, Columba, Eclipse JDT, Mozilla, Eclipse Platform, PostgreSQL) provided by Kamei et al. (2013)	Local models perform worse in the classification performance than global models in three evaluation scenarios; Local models are significantly better than global models in the effort-aware prediction performance in the cross-validation-scenario and timewise-cross-validation scenario.
Catolino et al. (2019)	Project-level: Ensemble	N/A	Offline	14 apps extracted from the CommitGuru platform (Afwall, Alfresco, Android Sync, Android Walpaper, AnySoftKeyboard, Apg, Atmosphere, Chat Secure Android, Facebook Android SDK, Flutter, Kiwix, Own Cloud Android, Page Turner, Notify Reddit)	Ensemble methods do not provide evident benefits with respect to the single classifiers.

Table 1 continued from previous page

Paper	CP Method	Project-level Similarity Metric	Online or Offline	#Project	CP Performance Conclusion
Zhu et al. (2020)	Project-level: Random	N/A	Offline	6 open source projects (Bugzilla, Columba, Eclipse JDT, Mozilla, Eclipse Platform, PostgreSQL) provided by Kamei et al. (2013)	Only use CP method to compare JIT-SDP models, there are no CP performance conclusion.
Lin et al. (2021)	Data-level: Global; Project-level: Project-Aware, Context-Aware	N/A	Offline	20 open source projects	The interpretation of global JIT models cannot capture the variation of the interpretation for all local JIT models; The project-aware JIT model and context-aware JIT model can provide both a more representative interpretation and a better fit to the dataset than the global JIT model.
Tabassum et al. (2020)	Data-level: All-In-One(AIO), Filtering; Project-level: Ensemble	N/A	Online	10 open source projects (Tomcat, Jgroups, Spring-Integration, Camel, Brackets, Nova, Fabric8, Neutron, Npm, BroadleafCommerce); 3 proprietary projects	AIO and Filtering can improve performance of WP at different periods; The performance of Ensemble is worse than WP.
Tabassum et al. (2022)	Data-level: All-In-One(AIO), Filtering; Project-level: Ensemble	N/A	Online	The same 10 open source projects as Tabassum et al. (2020); 9 proprietary projects	AIO and Filtering can improve performance of WP at different periods; The performance of Ensemble is worse than WP; It is important to update the JIT-SDP model with CP (including WP) data received over time.

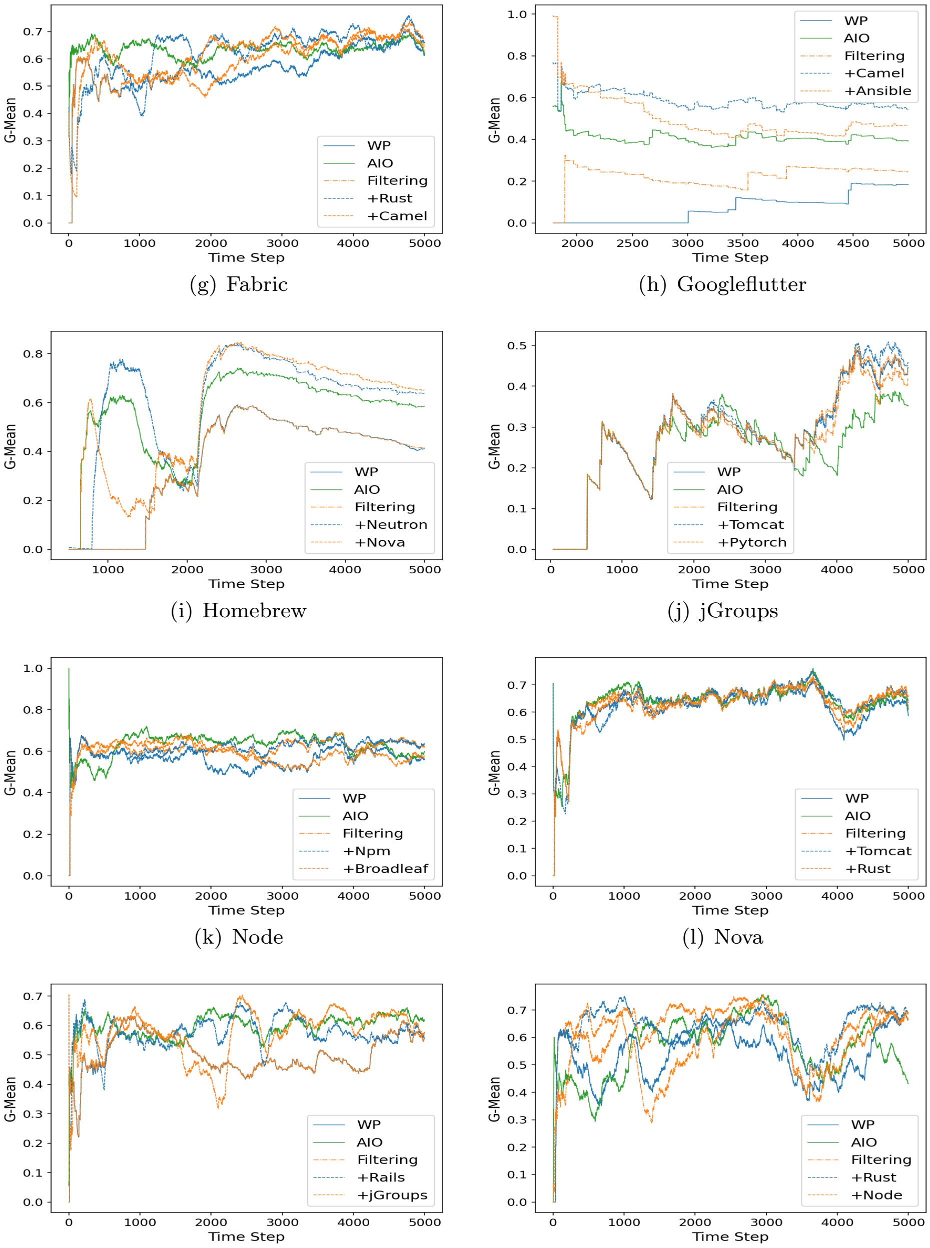
2 Project-Level Online CP JIT-SDP Is Promising

Fig. 1 shows the online predictive performance on different projects in terms of G-Mean. Here the base JIT-SDP model is OOB. The benchmark performance is represented by "WP" using only WP data. The potential of project-level similarity approaches is explored by incorporating the entire dataset of a single cross-project in the learning process of WP, denoted as "+1CP". It shows that for each project, there exist some time steps that the G-Mean values of AIO and Filtering is worse than "+1CP". This suggests that project-level CP method has potential to get better prediction performance than AIO and Filtering.



(a) Ansible (b) Broadleaf (c) Camel (d) Corefx (e) Django (f) Elasticsearch

Fig. 1 Motivation examples to show that the project-level CP method is competitive to the data-level CP methods.



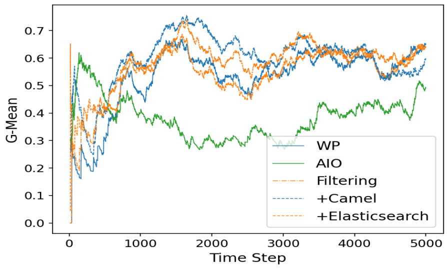
(g) Fabric (h) Googleflutter

(i) Homebrew (j) jGroups

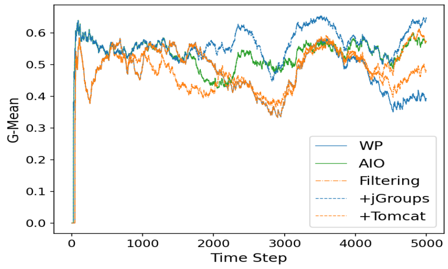
(k) Node (l) Nova

(m) Npm (n) Panda

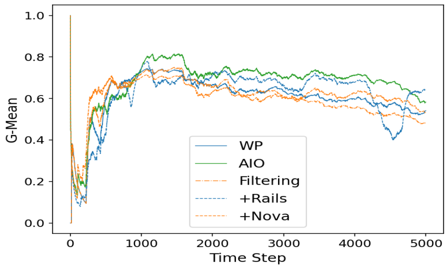
Fig. 1 Motivation examples to show that the project-level CP method is competitive to the data-level CP methods.



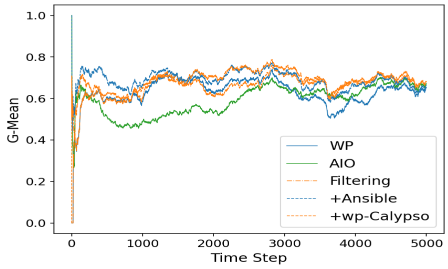
(o) Pytorch



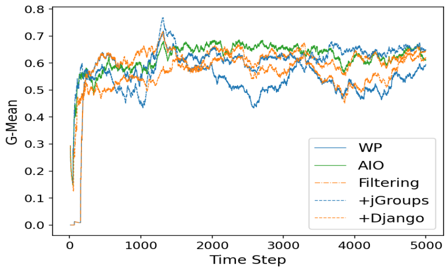
(p) Rails



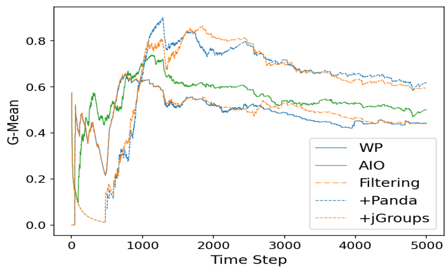
(q) Rust



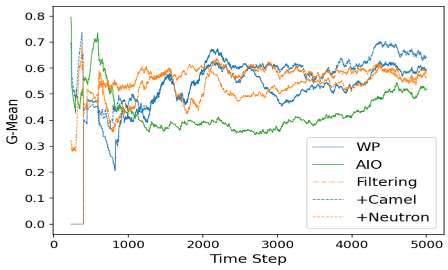
(r) Tensorflow



(s) Tomcat



(t) VScode



(u) wp-Calypso

Fig. 1 Motivation examples to show that the project-level CP method is competitive to the data-level CP methods.

References

Catolino G, Di Nucci D, Ferrucci F (2019) Cross-project just-in-time bug prediction for mobile apps: An empirical assessment. In: 2019 IEEE/ACM 6th International Conference on Mobile Software Engineering and Systems (MOBILESoft), IEEE, pp 99–110

Cho Y, Kwon JH, Ko IY (2018) Cross-sub-project just-in-time defect prediction on multi-repo projects. In: 6th International Workshop on Quantitative Approaches to Software Quality, pp 2–9

Fukushima T, Kamei Y, McIntosh S, Yamashita K, Ubayashi N (2014) An empirical study of just-in-time defect prediction using cross-project models. In: Proceedings of the 11th Working Conference on Mining Software Repositories, pp 172–181

Kamei Y, Shihab E, Adams B, Hassan AE, Mockus A, Sinha A, Ubayashi N (2013) A large-scale empirical study of just-in-time quality assurance. IEEE Transactions on Software Engineering 39(6):757–773

Kamei Y, Fukushima T, McIntosh S, Yamashita K, Ubayashi N, Hassan AE (2016) Studying just-in-time defect prediction using cross-project models. Empirical Software Engineering 21:2072–2106

Lin D, Tantithamthavorn C, Hassan AE (2021) The impact of data merging on the interpretation of cross-project just-in-time defect models. IEEE Transactions on Software Engineering 48(8):2969–2986

Shehab MA, Hamou-Lhadj A, Alawneh L (2022) Clustercommit: A just-in-time defect prediction approach using clusters of projects. In: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), IEEE, pp 333–337

Tabassum S, Minku LL, Feng D, Cabral GG, Song L (2020) An investigation of cross-project learning in online just-in-time software defect prediction. In: Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering, pp 554–565

Tabassum S, Minku LL, Feng D (2022) Cross-project online just-in-time software defect prediction. IEEE Transactions on Software Engineering 49(1):268–287

Yang X, Yu H, Fan G, Shi K, Chen L (2019) Local versus global models for just-in-time software defect prediction. Scientific Programming

Zhang T, Yu Y, Mao X, Lu Y, Li Z, Wang H (2022) Fense: A feature-based ensemble modeling approach to cross-project just-in-time defect prediction. Empirical Software Engineering 27(7):162

Zhu K, Zhang N, Ying S, Zhu D (2020) Within-project and cross-project just-in-time defect prediction based on denoising autoencoder and convolutional neural network. IET Software 14(3):185–195