

[Home](#) > [Communication, Software and Networks](#) > Conference paper

# Empirical Evaluation of Local Model for Just in Time Defect Prediction

| Conference paper | First Online: 28 October 2022

| pp 299–310 | [Cite this conference paper](#)



## Communication, Software and Networks

[Kavya Goel](#) , [Sonam Gupta](#) & [Lipika Goel](#)

 Part of the book series: [Lecture Notes in Networks and Systems](#) ((LNNS, volume 493))

 306 Accesses

## Abstract

Just in time defect prediction abbreviated as JITDP refers to a software that helps to detect instantaneously whether any change made to that software leads to defect or not and that too in no time. It helps us to develop a defect-free software helping both developers and testers to reduce their efforts. We have built a local model for predicting defects just in time for

improved performance. A local model makes clusters of the available training data based on their homogeneity, and then each cluster is trained individually, while global models refer to all those models which use all the available training data as a whole and are used to predict a test instance. In this study, we are focusing on (i) the implementation and (ii) execution performance of local models followed by (iii) the comparison of local and global models in three scenarios such as cross-validation, cross-project validation and time-based chronological scenario. Some of the observations made in this study from the comparison are as follows: Local model performs significantly low in classification. However, local models perform better in effort aware prediction in both cross-validation and cross-project validation scenarios. Also, local models have optimum performance when the value of  $k$  is set to 2 in  $k$ -medoids. Thus, for effort aware prediction in any software, using local models can be a good choice.

 This is a preview of subscription content, [log in via an institution](#)  to check access.

### Access this chapter

[Log in via an institution](#)

 **Chapter**

**USD 29.95**  
Price excludes VAT (USA)

Available as PDF

Read on any device

Instant download

Own it forever

[Buy Chapter](#) 

▼ eBook

USD 219.00

▼ Softcover Book

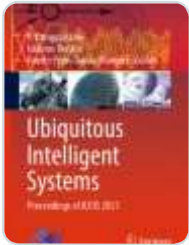
USD 279.99

Tax calculation will be finalised at checkout

Purchases are for personal use only

[Institutional subscriptions](#) →

### Similar content being viewed by others



**Ensemble Based-Cross Project Defect Prediction**

Chapter | © 2022

```
in 1 Pseudocode for classifier building
Classifier(TrainSet, Metrics)
:
set: Training set of changes
r: Metrics (see Table 1) of changes in TrainSet
ff:
ffier: The classifier built on the training dataset
id:
idple TrainSet to balance the number of defective and non-defective chang
idve ND, REXP, LA and LD from Metrics;
idy standard logarithmic transformation to each metric in Metrics except fo
idla classifier Logistic by using logistic regression applied on TrainSet and
idn Logistic;
```

**Revisiting supervised and unsupervised models for effort-aware just-in-time defect...**

Article | 27 October 2018



**Studying just-in-time defect prediction using cross-project models**

Article | 14 September 2015

## References

1. Yang X, Lo D, Xia X, Sun J (2017) TLEL: a two-layer ensemble learning approach for just-in-time defect prediction. Inform Softw Technol 87:206–220

[Google Scholar](#)

2. Yan M, Fang Y, Lo D, Xia X, Zhang X (2017) File-level defect prediction: unsupervised vs. supervised models, pp 344–353. <https://doi.org/10.1109/ESEM.2017.48>
3. Kamei Y, Fukushima T, McIntosh S, Yamashita K, Ubayashi N, Hassan AE (2016) Studying just-in-time defect prediction using cross-project models. *Empir Softw Eng* 21:2072–2106. <https://doi.org/10.1007/s10664-015-9400-x>

[Article](#) [Google Scholar](#)

4. Xu Z, Liu J, Luo X, Yang Z, Zhang Y, Yuan P, Tang Y, Zhang T (2019) Software defect prediction based on kernel PCA and weighted extreme learning machine. *Inf Softw Technol* 106:182–200

[Article](#) [Google Scholar](#)

5. Mockus A, Weiss DM (2000) Predicting risk of software changes. *Bell Labs Tech. J.* 5:169–180

[Article](#) [Google Scholar](#)

6. Kamei Y, Shihab E, Adams B, Hassan AE, Mockus A, Sinha A, Ubayashi N (2012) A large-scale empirical study of just-in-time quality assurance. *IEEE Trans Softw Eng* 39:757–773

[Article](#) [Google Scholar](#)

7. Hata H, Mizuno O, Kikuno (2012) Bug prediction based on fine-grained module histories. In: *Proceedings of the 34th international conference on software engineering, Zurich, Switzerland, 2–9 June 2012*, pp 200–210

[Google Scholar](#)

8. Liu C, Yang D, Xia X, Yan M, Zhang X (2018) Cross-project change-proneness prediction. In: Proceedings of the 2018 IEEE 42nd annual computer software and applications conference (COMPSAC), Tokyo, Japan, 23–27 July 2018, vol 1, pp 64–73

[Google Scholar](#)

9. Yu T, Wen W, Han X, Hayes J (2018) Conpredictor: concurrency defect prediction in real-world applications. *IEEE Trans Softw Eng* 45:558–575

[Article](#) [Google Scholar](#)

10. Tian Y, Li N, Tian J, Zheng W (2020) How well just-in-time defect prediction techniques enhance software reliability? In: Proceedings of the IEEE international conference on software quality, reliability and security (QRS)

[Google Scholar](#)

11. Albahli S (2019) A deep ensemble learning method for effort-aware just-in-time defect prediction

[Google Scholar](#)

12. Kim S, Whitehead EJ, Zhang Y (2008) Classifying software changes: clean or buggy? *IEEE Trans Softw Eng* 34(2):181–196

[Article](#) [Google Scholar](#)

13. Yang X, Lo D, Xia X, Zhang Y, Sun J (2015) Deep learning for just-in-time defect prediction. In: Proceedings of the IEEE international conference on software quality, reliability and security (QRS), pp 17–26, Vancouver, BC, Canada, August 2015

[Google Scholar](#)

14. Madhuri R, Ramakrishna Murty M, Murthy JVR, Prasad Reddy PVGD et al (2014) Cluster analysis on different data sets using K-modes and K-prototype algorithms. In: International conference and published the proceeding in AISC and computing, Springer, vol 249, pp 137–144

[Google Scholar](#)

15. Praneel ASV, Srinivasa Rao T, Ramakrishna Murty M (2019) A survey on accelerating the classifier training using various boosting schemes within casecades of bossted ensembles. In: International conference with Springer SIST series, vol 169, pp 809–825

[Google Scholar](#)

## Author information

---

### Authors and Affiliations

Ajay Kumar Garg Engineering College, Ghaziabad, India

Kavya Goel & Sonam Gupta

Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India

Lipika Goel

### Corresponding author

Correspondence to [Kavya Goel](#).

## Editor information

---

### Editors and Affiliations

Department of Electronics and Communication Engineering, Shri Ramswaroop Memorial College of Engineering and Management (SRMCEM), Lucknow, Uttar Pradesh, India

Vikrant Bhateja

School of Computer Applications, KIIT University, Bhubaneswar, Odisha, India

Jnyana Ranjan Mohanty

Facultad de Ingenieria, Autonomous University of Baja California, Mexicali, Baja California, Mexico

Wendy Flores Fuentes

School of Electronics and Computer Science, University of Southampton, Southampton, UK

Koushik Maharatna

## Rights and permissions

---

[Reprints and permissions](#)

## Copyright information

---

© 2023 The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd.

## About this paper

---

### Cite this paper

Goel, K., Gupta, S., Goel, L. (2023). Empirical Evaluation of Local Model for Just in Time Defect Prediction. In: Bhateja, V., Mohanty, J.R., Flores Fuentes, W., Maharatna, K. (eds) Communication, Software and Networks. Lecture Notes in Networks and Systems, vol 493. Springer, Singapore. [https://doi.org/10.1007/978-981-19-4990-6\\_26](https://doi.org/10.1007/978-981-19-4990-6_26)

[.RIS](#) [.ENW](#) [.BIB](#)

DOI

[https://doi.org/10.1007/978-981-19-4990-6\\_26](https://doi.org/10.1007/978-981-19-4990-6_26)

Published

28 October 2022

Publisher Name

Springer, Singapore

Print ISBN

Online ISBN

eBook Packages

978-981-19-4989-0

978-981-19-4990-6

Engineering

Engineering (R0)

## Publish with us

---

[Policies and ethics](#) 