

Download file | Extraction of | Design Pattern | Download file | Extract Pages

ieeexplore.ieee.org/document/9689174

IEEE Xplore | IEEE SA | IEEE Spectrum | More Sites

IEEE Xplore | Browse | My Settings | Help | Institutional Sign In | IEEE

ADVANCED SEARCH

## Extraction of Prediction Rules of Code Smell using Decision Tree Algorithm

Publisher: IEEE [Cite This](#) [PDF](#)

Pravin Singh Yadav; Seema Dewangan; Rajvart Singh Rao | All Authors

34  
Full  
Text Views

### Abstract

#### Document Sections

- I. Introduction
- II. Related Work
- III. Proposed Framework
- IV. Experimental Results
- V. Conclusion and Future Work

### Abstract:

Code Smell is a set of information of a source code that indicates any serious problem in the software. To detect the code smell prediction rule, we have applied a Decision tree algorithm. For this objective, we have taken two code smell datasets namely Blob-class and Data-class from Fontana et al. These datasets were prepared from 74 open-source systems. To calculate the performance measurement of the decision tree model on each dataset, we applied 5-fold cross-validation that divides datasets into training set and testing set and further splitting the training set into two part, one part is used for training and second part is used for validation. We have applied grid search for hyper-parameter tuning followed by extraction of decision rules to detect code smell instances. We have achieved the highest accuracy of 97.62% in both class predictions.

Published in: 2021 10th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON)

### More Like This

Performance Evaluation of Lazy and Decision Tree Classifier: A Data Mining Approach for Global Celebrity's Death Analysis  
2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)  
Published: 2018

An incremental decision tree learning methodology regarding attributes in medical data mining  
2009 International Conference on Machine Learning and Cybernetics  
Published: 2009

[Show More](#)

ieeexplore.ieee.org/document/9689174

Authors

Figures

References

Keywords

Metrics

Date of Conference: 01-02 December 2021  
INSPEC Accession Number: 21621416

Date Added to IEEE Xplore: 04 February 2022  
DOI: 10.1109/IEMECON53809.2021.9689174

► ISBN Information:  
Publisher: IEEE  
Conference Location: Jaipur, India

### I. Introduction

Nowadays, it is very important to improve the quality of the software while doing the software development process. Generally, functional requirements are highlighted and non-functional requirements like comprehensibility, verifiability, evolution, maintainability, and reusability of software may be weak due to a lack of nonfunctional quality, which increases the density of the code and the complexity of the work. Fowler et al. [2] defined the following metrics by which non-functional quality can be measured and is used to evaluate software.

- Authors
- Figures
- References
- Keywords
- Metrics