

# Research Proposal for Code Smell Algorithm Training Dataset Analysis

Rodger Byrd  
rbyrd2@uccs.edu

## I. INTRODUCTION TO THE PROBLEM

A code smell is a description of a pattern in code that might cause deeper problems. Code smells are not easily found and can vary by programming language and development methodologies. The existence of code smells and antipatterns implies that there are potential problems with software sustainment and imply there are issues with the design. Kent Beck coined the term “code smell” [1] and defined as “certain structures in the code that suggest (sometimes they scream for) the possibility of refactoring”. Machine learning is defined as computers learning to solve problems without being explicitly programmed, although they are “trained”[2]. Arthur Samuel coined the term Machine Learning in 1959[3]. Machine learning is a subset of artificial intelligence. It uses algorithms and statistical models to execute a task without explicitly being programmed. Machine learning is used in a wide variety of applications such as email spam filtering, search engines, video surveillance, and image curation. The big challenge in this field seems to be the best way to identify code smells and how training datasets are labeled and created. There is a lot of inconsistency in training datasets and identifying code smells because they can be subjectively interpreted. Additionally, expert advice and knowledge may be able to help identify code smells, using it to train datasets may not be effective unless it has been experimentally shown to be result in finding code smells.

It is difficult to compare the results of previous research due to the variability of training datasets and differences in machine learning algorithms. This research will compare training datasets and code smell definitions for performance.

Figure 1 show the “potential” results of comparison by algorithm.

## II. THE APPROACH

The approach to this research will be to compare training datasets for performance in detecting code smells. Common existing public datasets will be compared using the highest performing algorithms to determine which has the best performance.

## III. EXPERIMENTAL DESIGN

A formal section on Experimental Design and datasets

## IV. HYPOTHESIS AND STATISTICAL TESTING

A formal section on the hypothesis and Statistical Testing for that hypothesis for your given design and the outcomes of the experiments.

## V. RESEARCH PLAN

### A. Timeline

## REFERENCES

- [1] M. Fowler, *Refactoring: improving the design of existing code*. Addison-Wesley Professional, 2018.
- [2] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.
- [3] A. L. Samuel, “Some studies in machine learning using the game of checkers. Recent progress,” in *Computer Games I*. Springer, 1988, pp. 366–400.

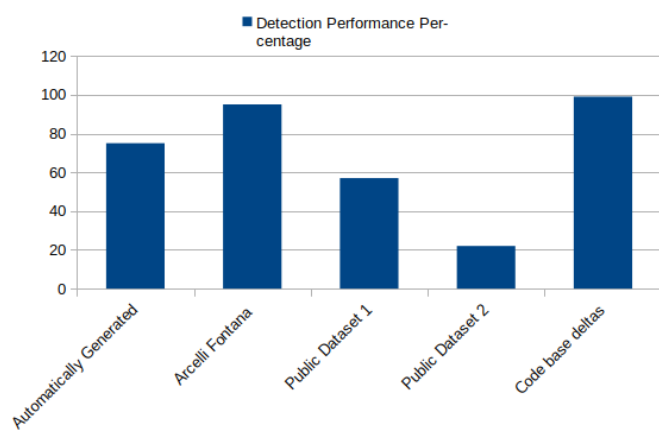


Fig. 1. Training Data Performance