B Control Papers Description

This appendix details the control papers that we used to identify the keywords used for our search, which we described in our methodology (Section 3). These papers – CP01 to CP07 – were read in their entirety to extract common terms (and synonyms) and text describing AD, shaping the search string for our systematic survey. We summarise each paper highlighting its contributions to AD, including ML-specific challenges, new TD types (e.g., AD, Compatibility Debt), and related issues like code smells. This compilation provides insights into our keyword selection process, supporting the empirical foundation for RQ1 to RQ4.

To identify the keywords, the control papers were read in their entirety to identify a list of common words (and their synonyms) that describe and indicate the presence of AD. These papers are summarised below.

Earlier works by CP01 identified frequent Software Engineering (SE) issues around ML systems based on their experiences at Google. The authors argued that ML systems accrue (i) an additional set of ML-specific challenges, and (ii) all of the maintenance issues associated with traditional, and (ii) has a unique tendency for accruing because debt occurs at the system-level rather than the code-level, which could be difficult to uncover. They further highlighted specific TD as being ML-related TD.

iIn CP2, the authors investigated the characteristics and patterns of bugs in ML systems. The authors observed that while developers can easily implement ML functionalities using open-source tools and libraries, there is often a lack of attention to code maintenance and quality. This oversight can lead to significant issues in software reliability. They also provided insights into the evolution of bugs over time. They also discussed issues related to performance. They discussed issues related to memory overflow, which makes a ML/DL program run more memory than the virtual machine provides.

CP03, the authors investigated the prevalence of SATD in DL frameowrks. They analysed seven DL frameworks and discovered that TD is a prevalent issue across these projects. This debt often manifests as code that is difficult to maintain, suboptimal design choices, or incomplete documentation. They extracted approximately 234,000 comments from GitHub to analyse those comments indicating TD. They uncovered two new TD types: AD (being the first publication to mention it), and Compatibility Debt. They argued that AD is prevalent in projects that depend heavily on performance and involve complex algorithms, like game engines.

CP05 manually extracted 327 patches from 26 ML open-source projects, accounting for 4.2 MLOC, and created a hierarchical taxonomy of general and ML-specific refactorings, and the TD they alleviate. Overall, they uncovered 14 new ML-specific refactorings and seven TD categories as a result of their research, with *Configuration Debt* being the largest discovered TD type.

CP06 conducted an empirical investigation on the prevalence of code smells in 74 Python ML open-source projects, aiming to identify the most common ones in ML projects. The authors identified “Code Duplication” as being widely spread, and discovered that substantial challenges to the maintainability and reproducibility of ML projects are linked to Python project dependency management. This paper was selected because they study code smells in ML, and code smells is a manifestation of Code Debt.