Аннотация

Определение авторства программного кода является актуальной задачей для решения вопросов интеллектуальной собственности, поиска плагиата, поиска авторов вредоносного программного обеспечения. Для разных языков программирования различные решения показывают лучшие результаты. Часть работ использует факторы, специфичные для конкретного языка, усложняя перенос результатов между ними. Также на данный момент не проводилось тестирование решений в условиях большого количества данных (тысячи примеров для каждого разработчика), которые могут возникать в практических задачах. Это вызвано отсутствием соответствующих наборов данных. В данной работе предлагается инструмент для сбора данных из проектов с произвольным числом авторов. Он работает с историей проекта, что позволяет получить большее количество данных по сравнению с имеющимися датасетами. С его помощью были собраны 7 датасетов из проекта IntelliJ IDEA, позволяющие тестировать модели для определения авторства в различных условиях. Также в работе представлены две модели для определения авторства, работающие в условиях разного количества доступных данных. Обе модели не используют свойств, специфичных для конкретного языка, что обеспечивает их переносимость на произвольный язык программирования. По сравнению с предыдущими работами в области, одна из предложенных моделей достигает лучших результатов для определения авторства по коду на Java и Python и повторяет результат для C++.

Ключевые слова: определение авторства по исходному коду, стилометрия, абстрактное синтаксическое дерево, машинное обучение, векторные представления кода, случайный лес.

Abstract

Source code authorship attribution is an important problem for resolving plagiarism and copyright issues in the programming field. Modern solutions for authorship identification use features specific to a particular language and can not be easily applied to another one. Also, existing works did not test solutions on large amounts of data due to the lack of an appropriate dataset. In this work, we present a tool for mining the history of project changes that can be used to collect datasets for authorship identification from projects with multiple authors. Considering history instead of static snapshot results in a significantly higher amount of data collected from a single project. Then, we collect 7 datasets from IntelliJ IDEA project to test different aspects of authorship detection models. Also, we design two models achieving state-of-the-art accuracy in authorship attribution for Java, Python and C++ code. The models do not depend on features of a particular language and can be applied to any programming language without modification.

Keywords: source code authorship identification, stylometry, abstract syntax tree, machine learning, path-based representations, random forest.