# Open-Source Android App Dataset*

Jacob Read
jakeread99@gmail.com

## ABSTRACT

Open-source apps play an important role in the Android development field, but research regarding these apps is made difficult due to a lack of data surrounding them. This project was designed to fix this problem by utilizing both Python scripting and manual work to provide a dataset of 400 high quality open-source Android applications, each with at least 1000 downloads on Google Play. For each of these apps, a variety of data is provided, including each language used and their percentage of the total repository, along with whether the app has testcases, and the date of the most recent commit. On top of this, many of the apps have been compiled, with the required build settings outlined in the dataset. For those that could not be built, error logs and general notes were included. More apps could likely be built by optimizing settings, as for this dataset general setups were used to reduce time requirements. The dataset is simple and intuitive to use and could be applied for a variety of research purposes including static analysis and general questions regarding testcases or other provided fields.

## CCS CONCEPTS

• Software and its engineering

## KEYWORDS

Open-Source, Dataset, Android, Android Apps, Compiling, Google Play, Testcases, App Building, Research Tool, Repository Languages

## 1 Introduction

Open-source apps provide a variety of benefits to developers, such as lower costs, improved app accessibility, and a greater likelihood of long-term preservation as new developers join the project [1]. Additionally, they allow for a far greater variety of research due to their directly accessible source code, opening the way for static analysis and trend observation. However, many of these apps never make it off the ground, either due to poor development practices or a lack of interest. F-Droid is the largest collection of open-source apps available for research, but many of these fall into the aforementioned category of unfinished and low quality apps which never make it onto more popular app stores such as Google Play. This raises a question regarding the accuracy of information data mined from F-Droid as it relates to commonly used apps. The solution to this problem would be to conduct research using a set of higher quality apps taken directly from Google Play. Unfortunately, Google Play does not provide the option to sort by open-source, making it difficult to collect such a sample set. This dataset was designed as a solution to this problem, containing 400 apps, each with at least 1000 downloads on Google Play. Additionally, various data is provided for each app, to further speed up the research process for this dataset's users.

## 2 Data Source

The data used in this dataset was collected from multiple sources. For each app, information was gathered from the corresponding Google Play page, along with its F-Droid page (where possible), and finally from its repository page. 98% of the apps in the dataset used a GitHub repository, with the remaining apps using alternatives such as SourceForge and BitBucket. This is likely far removed from the true percentage of Android apps using GitHub vs. alternatives, but since GitHub provides easy access to some data entries in the dataset such as language percentages, apps using GitHub repositories were prioritized where possible.

In order to find new open-source apps for the dataset, a few external sources were used besides direct searches in Google Play and F-Droid. The first of these sources was the Androsec dataset, a collection of over 1000 open-source apps mined from F-Droid in 2015 [2]. Additionally, various websites and forums discussing open-source apps were utilized to search for mentions of new apps.

## 3 Methods

### 3.1 Finding Apps

The first step of the dataset's creation was to find a large collection of open-source Android apps that fit specific standards. For an app to be added to the dataset, it had to be from Google Play and have at least 1000 downloads. These restrictions were set

to avoid flooding the dataset with many unknown or poor-quality applications.

Initially, apps were found via the Androsec dataset [2]. This had the advantage of all apps being open-source, but since this dataset used f-droid exclusively to build its app collection, a vast majority of the apps were not on Google Play or had only a couple of downloads. Additionally, due to Androsec's age, many of the apps were no longer supported or had been broken for some time. Despite this, due to the Androsec dataset's large size, around 250 suitable apps were still found.

When the Androsec dataset was fully searched, new apps were found by searching websites and forums for any mention of open-source apps not yet added to the dataset. Due to the nature of this search, many of the apps found in this stage were often more popular and higher quality, with many having millions of downloads. This method was short-lived however, as it quickly became difficult to find mention of apps not already collected.

The next strategy used was to search Google Play directly using open-source related keywords. Since Google Play does not have an option to sort by open-source, many apps returned in searches were not open source, slowing progress. Since only the 20 or so top results were displayed with each search, it quickly became difficult to find new apps with this method.

The final method of finding apps was to search using F-Droid. Since many apps on F-Droid are not also on Google Play the most successful way found to do this was to find the top few apps in a variety of specific categories of apps; "Firewalls" for example. This was a slow process, but it provided enough apps to fill the final dataset with a total of 400 open-source apps.

## 3.2 Adding Additional Information

Besides app names and links, additional information was added for each entry in the dataset. The first of these was a languages column that lists each programming language used in the program. The dataset also shows the percentage of the total repository that is built using each language. All of this information was retrieved from each app's GitHub site directly. Since other repository types did not provide this information, the information on languages for the small quantity of non-GitHub repos in the dataset is currently incomplete. When all the language related data had been gathered, a Python script was used to sort the language percentage columns based on most common languages, so the more common languages appear further to the left of the dataset.

An additional column was also added for whether or not an app contained test cases. An external tool called SourceGraph was used to scan the code of GitHub repositories for the keyword "test", with the assumption that any testcases in the repository would almost certainly contain this word. The results were then manually checked to ensure there were indeed testcases in the repository. Once again, there were issues with alternative repo types, as they could not be searched by SourceGraph. Due to this, these repositories were searched manually for testcases. This search was reasonably thorough, but it is possible testcases were missed in some cases.

Another column was also added for whether or not a repository had build instructions. This was again collected manually by searching readmes and wikis in each repository. The requirements to be considered build instructions were that a list of instructions for building and developing the app were given in some form. Although some instructions were much more detailed than others, the column simply states Y/N.

The final added column was the date of last commit for each repository. Since this column didn't require information to be manually checked, a Python script was created using GitHub's API to collect the data. **[Link script]**

## 3.3 Building Apps

On top of the other information, an attempt was also made to build each app, with the results listed in a few additional columns. Due to the large number of apps, changing software versions to perfectly suit each individual app would have been too time consuming. Instead, a quantity over quality approach was implemented. Based on testing with a smaller sample-set of apps, it was found that JDK 8 had the highest chance of building an app at random, so a generic setup was created using JDK version "8.0.2020.8" and SDK Platform "Android API 32 (Sv2)" in Android Studio. The Gradle version was left as automatic with Android Studio, although in some instances it was set manually for various reasons which are documented in the dataset.

Using this initial set-up, an attempt was made to build every app in the dataset. Due to the single set-up, multiple apps could be built simultaneously without version changes, greatly speeding up progress, but many apps also failed to build due to improper configuration. A new set-up was then used, replacing the JDK with version "11.0.15.1", which caught some of the builds which had previously failed. In total 154 out of 400 apps were built in these two runs.

# 4 Storage Mechanism

The following is a list of columns in the dataset, the information contained in each, and a detailed description of how the information for each column was obtained:

- App Name – The name of the app as seen on the app's Google Play page
- Google Play Link – The link to the app's Google Play page, found either directly through Google Play or through a redirect from the app's GitHub page
- F-Droid Link – The link to the app's F-Droid page, found by searching Google or through a redirect from the app's GitHub page. Some apps were not available on F-Droid

- Repo Link – The link to the app's source code, found by Google Play app descriptions, direct searches on Google, and through the "Source Code" link on the app's F-Droid page

- Language – A comma delimited list of all languages utilized in the app's source code, found using GitHub's languages section, or in the event of another repo source, by manually searching source code (In such cases one language is stated as "?" to account for languages that may have been missed)

- Has Test Cases? – A Y/N column that states whether the app uses test cases. This information was found using the SourceGraph tool and the following query: "repo:github\.com/[author]/[repo]$ (testcase OR test OR (test AND unit)) fork:yes". Results were then manually browsed to search for uses of testcases

- Has Build Instructions? – A Y/N column that states whether build instructions were provided on the app's source code page. This information was found by browsing app readmes and wikis. For longer readmes, ctrl+f searching was used with keywords "build" and "compile" to speed up results

- Compilable – A Y/N column that states whether the app was successfully compiled using one of two generalized build setups in Android Studio. Builds with warnings or unfatal errors were also considered to have compiled. Unfatal errors were noted in the compiling notes columns.

- JDK Version Used – The exact JDK version used to build the app if applicable. Currently one of two options depending on build setup, JDK 8.0.2020.8 or JDK 11.0.15.1

- SDK Platform Used – The SDK platform used to build the app if applicable. Currently only "Android API 32 (Sv2)" has been used, but this column was included to facilitate further dataset expansion

- Gradle Plugin Version Used – The Gradle plugin version used to build the app. Most builds had automated Gradle versions due to Android Studio, but some were edited or manually forced, which is reflected in this column

- Compiling Notes JDK 8/JDK 11 – Various notes regarding the build process of each app. Error logs were included from failed builds and were gathered from Android Studio's build terminal. In the event that settings were manually changed for a build to complete, the exact steps are listed in this column

- Latest Commit (YYYY/MM/DD) – The latest commit to the app's source code at the time it is added to the dataset. Found using Python and GitHub's API

- Various language columns – Each column in this group represents a language appearing somewhere in the dataset. The more often the language appears, the closer to the left the column is in the dataset. Each column records the percentage of its respective language that makes up the apps total source code. This information was found in GitHub's languages section. This information is not provided for repos using GitHub alternatives.

## 5 Originality

This is one of the first datasets of open-source Android applications, with the only other dataset focusing on this topic being the Androsec dataset, published in May 2015 [2]. This is the original publication of the new dataset. On top of being newer than the Androsec dataset, the apps are also of a higher quality, as whereas Androsec pulled its apps directly from F-Droid, this new dataset contains only apps with at least 1000 downloads on Google Play. It is also the first dataset to include set-ups to build many of the apps, along with information such as the percentage each language makes up of a total repository, and whether or not an app contains testcases.

## 6 Usage and Potential Research

At the time of writing, this dataset is brand new and has not yet been used for any research, but it is designed to provide a variety of useful data. The dataset consists of 400 open-source apps, more than 150 of which have information on JDK, SDK, and Gradle versions required to build them. For the apps that failed to build using a generic set-up, error logs were taken for both the JDK 8 and 11 build attempts. These errors and other general notes can be found in the compiling notes columns and should aid in successfully building a higher percentage of the apps by hinting at the optimal settings for each failed build.

In general, the Gradle versions used in the build were the ones automatically set by Android Studio upon loading a project, but there are a couple of exceptions. If a cell states, "Android Studio prompted Gradle version", this means the build initially failed, but Android Studio offered a Gradle upgrade prompt in the error log which fixes the issue. If a cell states, "Auto with Android Studio recommended project upgrades", this means that before building, Android Studio was allowed to run a project update in order to fix version issues. There were also some cases where the Gradle

plugin version was changed manually, in which case the used version is provided.

Besides this, usage of the dataset is straightforward, and information can be understood with minimal required knowledge. Since there exists no other such collection of high quality open-source apps, this dataset should prove invaluable for many research purposes. Since links to source code are included for each app, a wide array of research can be done using static analysis with this dataset as a sample. Other questions that could be asked include, "What languages are the most common amongst popular Google Play apps", or "How frequently are testcases utilized in open source apps?"

Some data that should not be taken from this dataset are questions such as "What percentage of open-source apps use GitHub Repos", or "What percentage of open-source apps can be compiled", as GitHub repos were preferred when finding new apps, and many uncompiled apps in the dataset could likely be built using more precise settings.

## 7 Limitations and Further Improvements

The major limitation for this dataset was time. Preferably, additional care could have been taken to ensure builds were using correct versions in order to minimize build failures, but to save time generic build settings were selected instead to allow multiple builds to run simultaneously.

Due to less than 2% of the dataset's apps using non-GitHub repositories, time restraints made it unfeasible to find new ways to gather information for some of these app's columns. Where possible, data was manually gathered by searching repositories in these cases, meaning that for some instances such as the testcases and languages column, there may be a degree of inaccuracy for these apps. Additionally, language percentages were not found for these repos.

All the issues outlined in this section could be resolved in future improvements to the dataset. Additionally, new apps could be added, although finding uncollected apps that fit the requirements for the dataset has become difficult.

## 8 Conclusion

In conclusion, the dataset is the first of its kind, with a broad collection of data, including build requirements and compiling notes, for a large number of open-source Android applications, all of which have at least 1000 downloads on Google Play. The dataset provides Google Play, F-Droid, and Google Play links, as well as a list of each repository's languages and what percentage of each was used to create the app. Data on testcases, build instructions in a repo, and the date of each repo's most recent commit are also provided, enabling research into these areas. This data could be used for a variety of research using static analysis. The dataset could be improved by taking a more careful and time intensive approach to building the apps in order to successfully build a higher percentage. Some data could also be added for apps that do not use a GitHub repository

## REFERENCES


[1] Edward M. Corrado. 2005. The Importance of Open Access, Open Source, and Open Standards for Libraries. *Issues in Science and Technology Librarianship* 42 (2005). DOI:http://dx.doi.org/10.29173/istl2002

[2] Daniel E. Krutz, Mehdi Mirakhorli, Samuel A. Malachowsky, Andres Ruiz, Jacob Peterson, Andrew Filipski, and Jared Smith. 2015. A dataset of open-source Android applications. *In Proceedings of the 12th Working Conference on Mining Software Repositories (MSR '15).* IEEE Press, 522–525.