**Empirical Study with Boa:**

To validate the findings of Pingzer et al. and to achieve the research questions stated in the previous section, we propose a study that aims to construct developer-module networks for several large, open-source projects on GitHub.

***Dataset:***

Due to the many inherent limitations in the GitHub API, the study conducted in this project does not directly access project data from GitHub. As a result, one of the first

***Methodology:***

To

***Software Tool:***

To efficiently carry out the methodology as described in Section 4.1, a software tool called *NetworkMine* was developed that automates the majority of the process. Written in Java, the *NetworkMine* tool is comprised in two different parts; a Boa miner which pulls all software repository details from the Boa website, and a social network builder which constructs the developer-module network and computes all relevant metrics. This division is important since, in future iterations of the tool, different GitHub mining repositories can be swapped for Boa. The remainder of this section will explain the two sections of the tool in detail.

*Boa Miner:*

The Boa miner component is responsible for pulling project data from Boa’s servers into a local SQLite3 database. The tool allows for users to search for projects based on the number of contributors and files and can run the specified search of any of Boa’s three different GitHub datasets. Once a collection of projects is found that match a user’s desired search parameters, the user can download any number of the returned projects. Figure 1 shows a screenshot of the tool allowing users to select projects to download.

The Boa miner component is written in Java and requires the SQLite JDBC driver and Boa API developed by Iowa State.

*Social Network Builder:*

The social network builder component carries out the brunt of the methodology. Using the project repository data downloaded from the Boa miner component, the social network builder constructs a developer-module network for a specified project and then computes the betweenness, closeness, and degree centrality metrics for that graph. Once complete, the user is then able to see the Spearman correlations for each of those social network metrics and the results of the logistic regression run on the graph. Unlike the Boa miner component, this part of the tool is accessed via the command line. The output of the program is a comma-separated values (CSV) file that contains the results of the Spearman correlation and logistic regression functions.

While the tool was developed during this study, it relies on several open-source libraries including Java Universal Network/Graph Framework (JUNG) and Apache Spark. JUNG is used to develop the graph and compute centrality metrics while Apache Spark, while normally used as a cluster computing framework, is used to perform the logistic regression.

**Results:**

This section describes the results that were obtained after running

***Selected Projects:***

While the Boa dataset has a large number of GitHub software projects that could be used in this study, we decided to limit our scope by focusing on a subset of these projects. While this project aims to test whether developer-module networks are able to predict failures over a large number of GitHub projects, too many projects would take far too long to compute and also dilute the results presented in this paper. As such, we selected fifteen of the largest projects contained in the large GitHub dataset on Boa as a good representative sample. Table 1 shows statistics for each of the projects that were used in this study including the number of commits and number of commits used to fix bugs.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Project Name** | **# of Files** | **# of Contributors** | **# of Commits** | **# of Bug Fixes** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

***Spearman Correlation***

Similar to Pingzer et al., for each project that we tested, we computed the Spearman correlation between each of the metrics