Friedrich-Alexander-Universität Erlangen-Nürnberg
Technische Fakultät, Department Informatik

Weixin Wang
MASTER THESIS

Paid vs. Volunteer Job of Open Source in China

Submitted on 2018 Juli 25th

Supervisor: Prof. Dr. Dirk Riehle, M.B.A.
Professur für Open-Source-Software
Department Informatik, Technische Fakultät
Friedrich-Alexander University Erlangen-Nürnberg

Versicherung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, 26.07.2018

License

Erlangen, 26.07.2018

Abstract

China is now playing a larger and larger role in the global economy and technology. With the support from the chinese government since the early 2000s, Open-Source-Software in China has become a field of interest to study about the current and future development of chinese IT industry. This thesis reveals the statistics about the Open-Source-Software growth in China using data gathered from a Chinese code management platform using exploratory data analysis. The result indicates that there is a growing amount of commitment and participation being devoted in this field. It also shows that around 46% of the work is done in the normal working hours, which reveals some characteristics of the working pattern of chinese software developers. Furthermore, several approaches have been made to classify the developers into different groups by their working patterns, in order to draw a picture of the difference between volunteer work and one being paid or supported.

Contents

1 Introduction

It may sound like a contradictory at the beginning when terms "commercial activity" and "open source" are mentioned at the same moment. However, research from Dirk et al. on the ratio of paid work to volunteer job in open source development showed that around 50% of all contributions to the target population had been paid work. Furthermore, this number varies less than 10% within several other projects (Riehle, Riemer, Kolassa & Schmidt, 2014). On the other side, it is well known that big companies like Google, Facebook and Microsoft are open-sourcing many of their projects in quite many fields including cutting edges such as big data and machine learning. That suggests that volunteer work and commercial support are more and more integrated into each other in open source development.

Since the existing researches are mainly focused on software development in western countries (Europe & North America), a similar analysis is conducted in this thesis with a population of chinese open source development, and several new approaches are proposed to classify different working conditions.

Goals of this thesis are:

1. To survey the status of open-source software industry in China in the past decade and gain knowledge about the growth in both general and individual statistics.
2. To reveal the distribution of different working patterns (paid or volunteer) of chinese software developers who use Gitee.com for their code management by analyzing their commit information, such as commit time and commit numbers.
3. To compare the situation in China with those in western countries by comparison with statistics from Linux Kernel and Ohloh Projects
4. After comparison and analysis of those different working patterns, a better understanding of their working condition can be gained as a result, which includes how likely a developer is to perform developments in different projects as a career or just out of interest.

In Section 2, a background survey is performed with statistic and literature. Section 3 describes our data for research along with term definitions. The results of data analysis are listed in Section 4. Section 5 contains our discussion of the findings we gained. The related works that can be referenced are reviewed in Section 6. Section 7 is our final conclusion to this thesis.

Furthermore, several new approaches are made and evaluated as a supplement to the thesis in elaboration chapter.

2 Current Status of Chinese Open-source Industry

In spite of the prevailing opinion that Chinese software developers take significant fewer activities in the Open-Source community, the spark of Open-Source was lighted in China 2 decades ago. According to a research paper published in 2007, the Open-Source Software Development in China took its first step in 1999, where Red Flag Software Co., Ltd released red Flag Linux 1.0. Along with Red Flag Linux, several open-source software like Linux Virtual Server, Smart Boot Manager and MiniGUI were also developed and recognized by the community. (Pan & Bonk, 2007)

However, term as Open-Source remained out of the horizon of most people till the middle of the first decade of 21st Century. Since 2006, large Chinese enterprises like Huawei, Alibaba and Baidu started to take part in the Open-Source community: Sina Corporation open-sourced storing system Memcachedb, cache system Ncache, Xweibo etc. Huawei Technologies Co., Ltd participated in the Hadoop project. Alibaba Group held the open-sourced projects such as TFS, TAIR and OceanBase, etc.(Liu, 2013) In the meantime, the number of open-source communities had reached over 200. Many of them have a specific user group, like the Leadership of Open Source University Promotion Alliance (LUPA), CSDN.NET, PHP China. (Pan & Bonk, 2007)

According to the statistics collected from Gitee (a Github-like platform for Chinese users), commit numbers produced by developers started to bloom after 2006. Which is illustrated in the Figure 1.

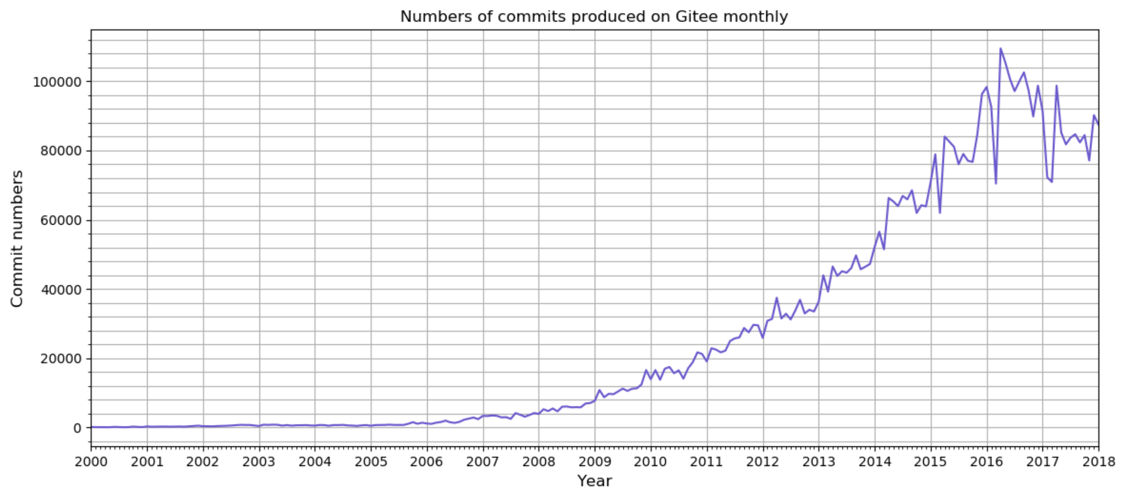


Figure 1. Numbers of commits produced on Gitee by year

Following figure shows the year-to-year growth of the commit number produced on Gitee, which increased significantly from 12,559 to 98,221 since the year 2006. Which indicates, the amounts of both time and manpower devoted to the open-source development in China rocketed since 2006, and accordingly, Chinese software developers are making more and more progress in this field.

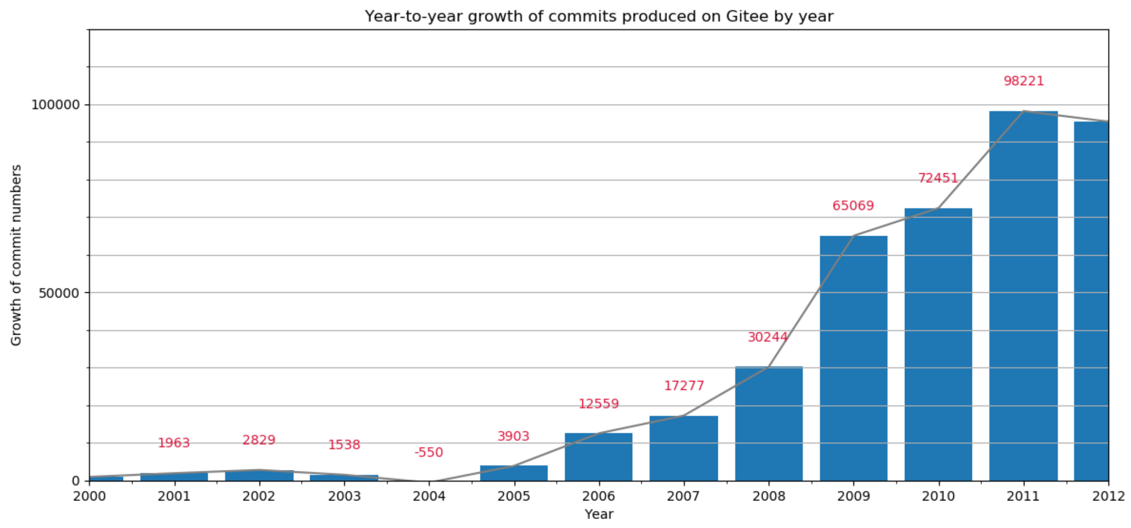


Figure 2. Year-to-year growth of commits produced on Gitee by year

After reviewing a discussion thread with the title “How is the current situation of Open-Source-Software in China?” on Zhihu.com . Till the time of writing this paper, 84 answers were posted in total. Among which 29 answers have clearly stated their opinions. There are 14 answers approving of the progress made by Chinese companies, 12 answers assessing negatively about the contribution from Chinese companies, while 3 answers stand neutral ("How is the current situation of Open-Source-Software in China?", 2018).

Although the comments of ordinary developers conveyed both positive and negative opinions about the contribution and purposes of those enterprises, these actions of reaching out to Open-Source marked the turning point of how Open-Source being accepted and recognized in China. Thus, the year 2006 has been selected as the starting point for the data of study in this thesis.

A recent report pointed out, “in September 2016 Baidu announced that it would license machine learning platform PaddlePaddle under Apache, an open source license (Moore, 2018). In December 2014 China’s Ministry of Industry and Information Technology declared its support for OpenStack for state-owned enterprises. Not long after, Tencent embraced the Open Daylight open source software project’s software-defined network, instead of developing its own proprietary solution. Today, Alibaba, Baidu, China Mobile and Tencent are all silver members of the Open Daylight Foundation. Taken together, these developments suggest an ongoing shift in how China thinks about software and how its companies position themselves with regard to global trends” (Keith Bergelt, 2018).

In summary, there are reasons to believe that Chinese software developers, both working in companies and programming in their free time, are playing more and more critical role in the global open-source community.

3 Researched Data

3.1 Term Definitions

Gitee is a Github-like code management platform mainly serving the Chinese software developers. A large number of Chinese software developers manage their projects on Gitee for reasons such as language, network speed, stability and usage habit ("Feature comparison between Gitee and GitHub", 2018).

Following definitions are used throughout this thesis:

- A developer is a person who devotes himself to open-source software development.
- A user is a developer who uses Gitee as his code management platform.
- A committer is a user who uploads his change of code to Gitee.
- An author is the original creator of a certain piece of code.
- An active project is a repository that has more than one commit (initial commit) till the time when data is collected.
- An active committer is a committer who has committed more than one commit till the time when data is collected.

We introduce the following relations between a developer, a use and a committer to make these terms intuitive and simplified:

- Different different developers are represented by different users according to the registration rules of Gitee, however different users can be one developer.
- Correspondingly, different users will not be identified as the same committer. One user can show several identities of committers depends on his working environment, in detail, where and on which computer he works.

The standard working hour is defined using the same term in the western case study:

- Normal working hour for a specific developer is the time from 9 am to 5 pm in his local time zone, from Monday to Friday
- Ordinary spare time is the rest time out of one’s normal working time span

Although there is no regulation from Chinese Government about the exact working hour for software developers and it varies from company to company (Zhang, 2018), which somehow leads to a further problem in this research later, most of the Chinese enterprises share the similar working hour system with western enterprises due to the international cooperation and organization.

Furthermore, legislation in China also agrees with westerns, according to the Article 36 of Labour Law of the PRC, “ The State shall practice a working hour system wherein laborers shall work for no more than eight hours a day and no more than 44 hours a week on the average” (*Labour Law of the People’s Republic of China*, 1995).

3.2 Data Sources

Different from the original study about the western countries conducted by Riehle et al., data sources such as public configuration management data found at Kernel.org and a 2008 snapshot of the Ohloh open source project database are not introduced in this thesis. The reason that altered data sources are needed is that the result in the original study shows that software developers which located in time zone from UTC +2:00 to UTC +10:00 (Chinese software developers are mainly located in UTC +8:00 time zone) showed significantly less participation in these two open-source projects (Riehle, Riemer, Kolassa & Schmidt, 2014). Furthermore, most developers are working in North America, Western and Northern Europe according to a research on the Geography of Open Source Software through Github (Takhteyev & Hilts,2010).

Factors such as language gap, culture difference along with the Internet censorship in China lead to this situation (Protalinski, 2018). There is a variety of software developments systems being created and used in China including coding.net, csdn.net, Gitee.com, etc. Many large software companies also have their git-based platform. To focus on the behavior of Chinese software developers and learn some facts about the working patterns specific to China, all the data used in this thesis are collected from Gitee.

Gitee is a product of OS-China forum, which can be considered the first group of pioneers of open-source community in China (Liu,2013), it currently holds around 2,000,000 projects according to the introduction of the website, which is a quite large base number for data collection ("Feature comparison between Gitee and GitHub", 2018). As a Git-based platform, it also provides us with a similar API system compare to Github, through which we can quickly gather the information by a crawler. Therefore Gitee is considered as an available data source for this thesis.

3.3 Data Quality and Quantity

3.3.1 Active and Inactive Projects

At the end of the data collection phase, we managed to identify 622,186 repositories with their project names and owner information out of 2,000,000 repositories that are claimed to be stored on Gitee. The Web API Crawler managed to retrieve data from 210,000 repositories. The unreachable (private) projects, inactive projects caused the reduction of repository numbers. Even though the repository numbers have been cut to one-tenth of the original, there are still some inactive projects left in the final data set, which will be ignored in the following processing. Together with the ranking function of the Gitee search engine, it is safe to consider our dataset containing all the active projects and is representative.139,197 projects are finally left for analysis after cleaning of the inactive projects.

3.3.2 Time Zones

Gitee system logs every activity including commits overseas in the form of the time zone of where its server is located, that is UTC +8:00. In our case, differences caused by time zones do not make the data more complicated in comparison to the original research, since the Chinese Government legislated that all the provinces use “Peking Time” (UTC +8:00) as standard time (Guo,2001). However, due to a vast territory, the working time may still vary in different longitudes. People who live in the east part of China may start their work 1 hour earlier than those living in the Qinghai-Tibet Plateau or Xinjiang Desert. According to a research on the geographical distribution of IT industry production in China, more than 70% of IT industry production is generated near the east coast of the Chinese mainland (Song & Zhang,2015). It is evident that more than 90% of the IT industry production is generated around the longitude of Peking (showed with red line) from the picture.

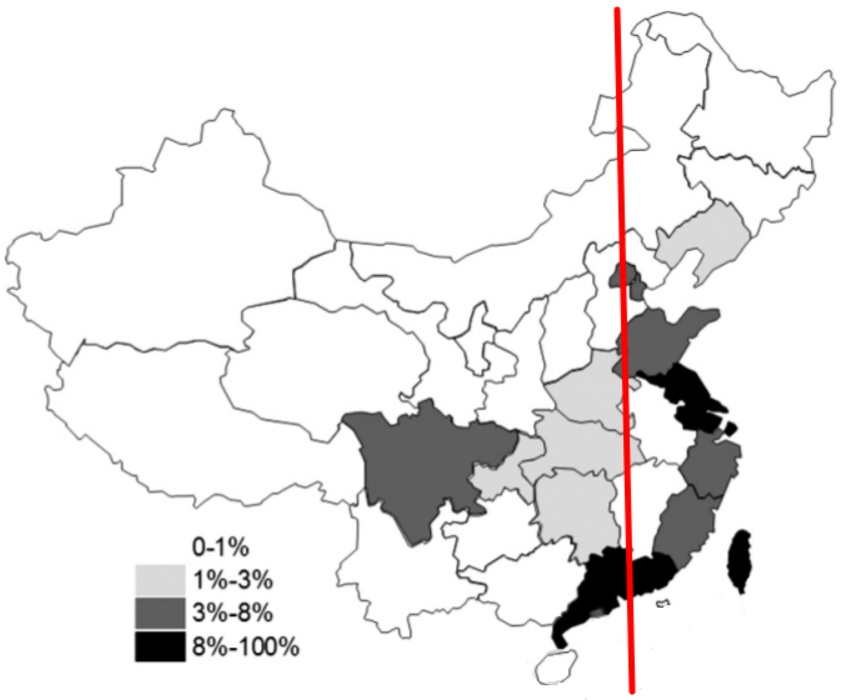


Figure 3. Geographical distribution of IT industry production (Song & Zhang,2015)

Meanwhile, Gitee is designed to serve Chinese software developers mainly, thus we consider all commits produced by overseas users or working when traveling as noises to our data, those noise data are presented in the form of a time zone shift compare to the representative data. To recognize and suppress the influence of these abnormal data, we sorted every committer by the percentage of his work which is made during a particular time window. Here we choose from 0:00 to 6:00 UTC +8:00, because not only it is over midnight in China, but also the corresponding local times are around from 18:00 to 0:00 in Europe and about from 12:00 to 18:00 in the United States. Those time periods are peak hours of committing in western countries (Riehle, Riemer, Kolassa & Schmidt, 2014). Therefore, if the commit numbers of a committer in this period exceeds 50% of his total commits, his data will not be counted in the later analysis.

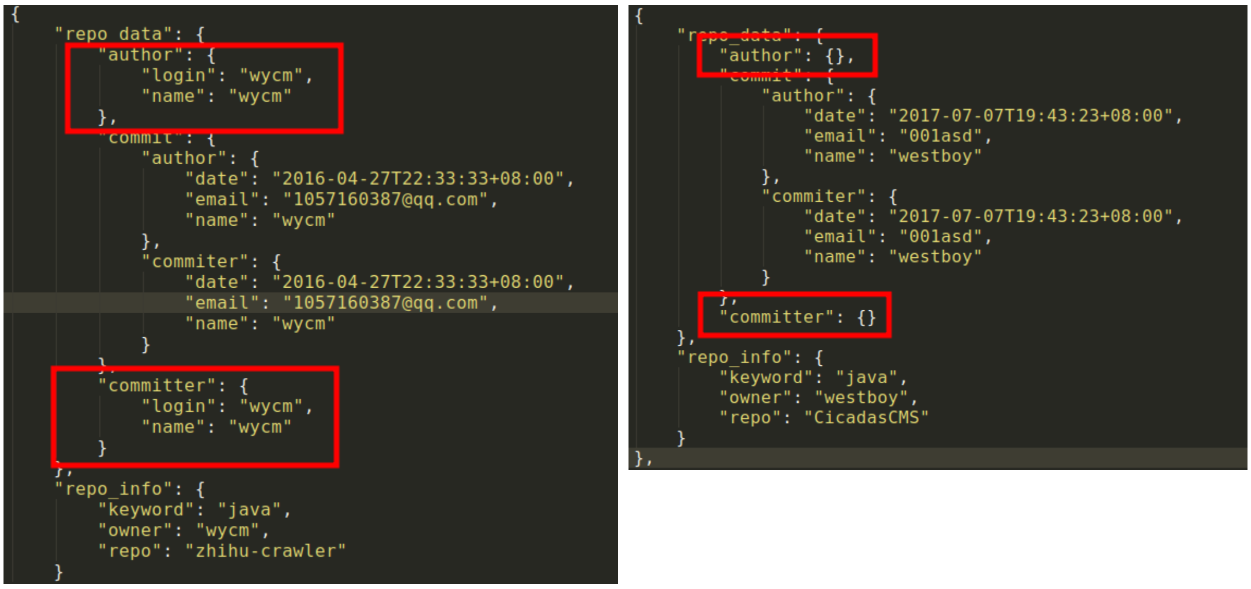
In the meantime, several other nearby time windows are also used to classify the abnormal data, and it does not show an enormous difference. Finally, 89,887 committers are erased from the data set, which is about 22.4% of the original data set.

3.3.3 Unlinked Identities

During calculation of the total user number, another problem has been raised when the data is initially processed and sorted:

Gitee system logs the committer information not based on their user account information, but the Single Sign-On token along with the name or email address they passed to the server.

That leads to a large number of committer identities which not sufficiently linked with a specific user account. In another word, one user may show as different committers depending on where, which computer or even which computer account he used to make a specific commit. All these committer identities will be linked to his user account only when he always used the same name or email address by generating SSO tokens. The following picture shows how it appears in the data when a committer identity is linked or unlinked to a user account.



```
{
    "repo_data": {
        "author": {
            "login": "wycm",
            "name": "wycm"
        },
        "commit": {
            "author": {
                "date": "2016-04-27T22:33:33+08:00",
                "email": "1057160387@qq.com",
                "name": "wycm"
            },
            "commiter": {
                "date": "2016-04-27T22:33:33+08:00",
                "email": "1057160387@qq.com",
                "name": "wycm"
            }
        },
        "committer": {
            "login": "wycm",
            "name": "wycm"
        }
    },
    "repo_info": {
        "keyword": "java",
        "owner": "wycm",
        "repo": "zhihu-crawler"
    }
```

```
{
    "repo_data": {
        "author": {},
        "commit": {
            "author": {
                "date": "2017-07-07T19:43:23+08:00",
                "email": "001asd",
                "name": "westboy"
            },
            "commiter": {
                "date": "2017-07-07T19:43:23+08:00",
                "email": "001asd",
                "name": "westboy"
            }
        },
        "committer": {}
    },
    "repo_info": {
        "keyword": "java",
        "owner": "westboy",
        "repo": "CicadasCMS"
    }
},
```

Figure 4. Difference between linked and unlinked identities

The solution to this problem is either finding the connection between unlinked identities and possible user accounts or considering these unlinked identities as independent samples. We do not know which committers are actually which user in reality, yet the commit data of single one committer can be considered as a subset of the whole data. Besides, these subsets are also disjoint and complement sets because every commit can only have one committer. In the first stage of the data analysis in this thesis, the commit data are counted as a whole. Therefore it does not hurt the analysis when those unlinked committers are treated as independent samples. Later in the elaboration chapter, some inference will be made to ensure that the feasibility still stands.

As a result, 73,722 committers are recorded in the final data set. It is already a sufficiently large number comparing to the western case study.

4 Data Analysis

4.1 Classification Boundaries

Before we start analyzing data, a boundary to classify different working conditions (paid or volunteer) needs to be defined. Intuitively, we can use regular working hours from 9:00 to 17:00 during weekdays. That means any commit generated within this time window is considered paid work. Otherwise, it is treated as a volunteer job. Someone may argue that developers like students also work in this time window. However, they are not being paid by universities or companies. Or in another case, employed software developers can work overtime or have flexible working hour regulations by the company to participate in the work out of this time window. Which then leads to an interesting question that does someone needs to be physically paid to do a paid job?

In this part of the thesis, we initially focus more on the working time adapted by individual software developers instead of the question if they receive a salary from specific companies or organizations. In fact, those who don't get any physical reward for their work in the standard working hours such as students, are to be considered sponsored by themselves to compensate the opportunity cost due to the time and effort they spend. Similarly, when employees work at night or on the weekends, they choose to trade their free time for working progress, which is a voluntary activity.

Therefore we use the similar classification boundary of the working time to the western case study (Riehle, Riemer, Kolassa & Schmidt, 2014):

- Paid work time window: from 9:00 to 18:00 local time, weekdays. Here one hour is added for lunch break comparing to the western case.
- Volunteer work time window: the rest time.

4.2 Overall Statistics of Paid and Volunteer Work in Open-source

In this part, every single commit is counted alone and considered independent from each other. Figure 5 shows several facts about the general working patterns of Chinese software developers:

- A significant percentage of commits are produced in the daytime, i.e. from 8:00 to 19:00. No matter which day in the week.
- Notable more commits are produced on the weekdays than in the weekends.
- There is a production gap between 12:00 and 13:00 on weekdays, which is most likely a result of lunch breaks. However, dinner times do not have such a significant impact on the production.

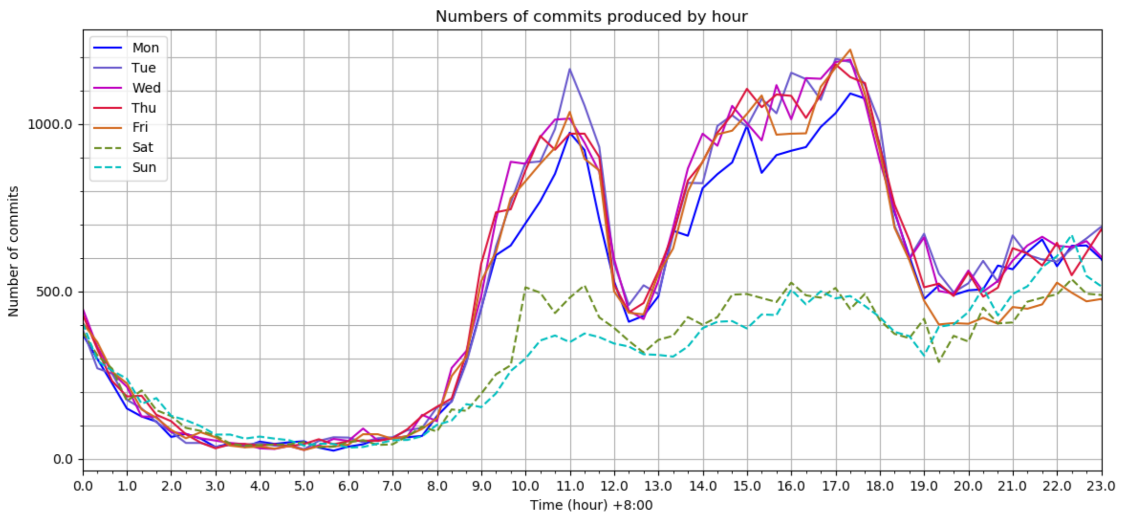


Figure 5. Numbers of commits produced on Gitee by hour, sampled in every 20 minutes

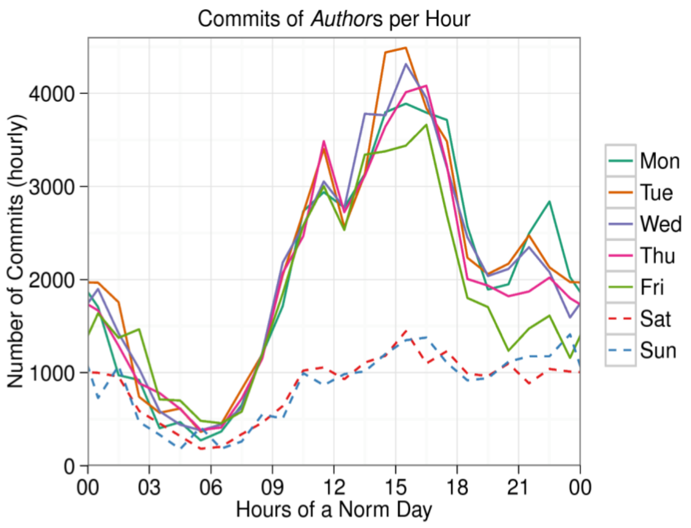


Figure 6. Number of commits by authors (Riehle, Riemer, Kolassa & Schmidt, 2014)

If we take a close a look at both chinese and western cases, some interesting differences also indicate different working habits between different cultures (figure 6). The most obvious is lunchtime: the impact of the lunch break on production in China is far intenser than in western countries. It is not a secret that food and meals play an essential role in chinese culture, having lunch together with friends or colleagues is often a way to establish and maintain one's social relations (Ma,2015). As a result, it is no wonder that the curve forms a deep valley in the figure above, while it just reduces production in western countries to a limited extent.

Another interesting fact is that chinese software developers are almost equally productive in the morning and the afternoon. On the other hand, western software developers trend to reach their best in the afternoon. This is also related to the commit habits of different developers, especially when a developer is used to committing his work in a particular time of the day, like in afternoons. This can also be an interesting subject to study.

Table 1 shows the overall statics about the distribution of commits produced in different time windows. The result reveals the fact that the production of chinese software developers is general evenly divided into these two time windows with around 6% more in spare time window than paid one.

	Paid	Volunteer
Commit numbers	2,609,115	3,032,881
Percentage	46.2%	53.8%

Table 1. Numbers and percentage of work performed during working hours

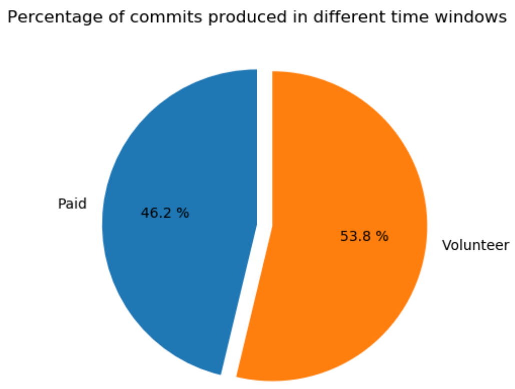


Figure 7. Percentage of work performed in different time windows

Meanwhile, the percentages of commits generated in working time window are:

- 45% of authors and 51.36% of committers for Linux Kernel
- 47.3% of known committers and 55.4% of extended committers for Ohloh Projects

(Riehle, Riemer, Kolassa & Schmidt, 2014).

Here we notice that our result aligns with the statistic of authors from Linux Kernel for the first time.

4.3 Trends of the Distribution over Years

In the previous chapter, we have gained an overview of how many commits are generated in different time windows. Now we try to find out how this distribution changed over the years in the past decade.

Firstly, we approached from the perspective of the total numbers. The statistic contains quite many fluctuations, and it is hard to find any regular patterns such as seasonal upsurge or downturn year by year. However, we managed to distinguish several patterns which do follow a certain consistency. The figure below reveals the total number of commit made over months, where we marked the date of Chinese New Year and Chinese National Day:

1. October 1st is Chinese National Day and always followed with a statutory holiday of 7 days. Blue dotted line highlight the sharp decrease in total numbers during Octobers from 2010 to 2017. It needs to be noticed that the number not only collapse in October but also rocket to peaks shortly before October. This phenomenon did not appear just in the year 2015 and 2017, yet the numbers in end of the September and October still make a significant contrast, which suggests that many software developers were trying to work more or even work overtime before the holiday and make up for the shutdown in holidays.
2. Another festival with such importance or even more in China is the Spring Festival, i.e., Chinese New Year. In contrast to the western celebrations, traditional Chinese festivals are all counted in the lunar month, which causes the date of the festivals shifts back and forth when using Gregorian Date. After looking up the exact date of the Chinese Spring Festival (Appendix B), several patterns similar to 1. can be found in the curve. Red dotted line mark these patterns.

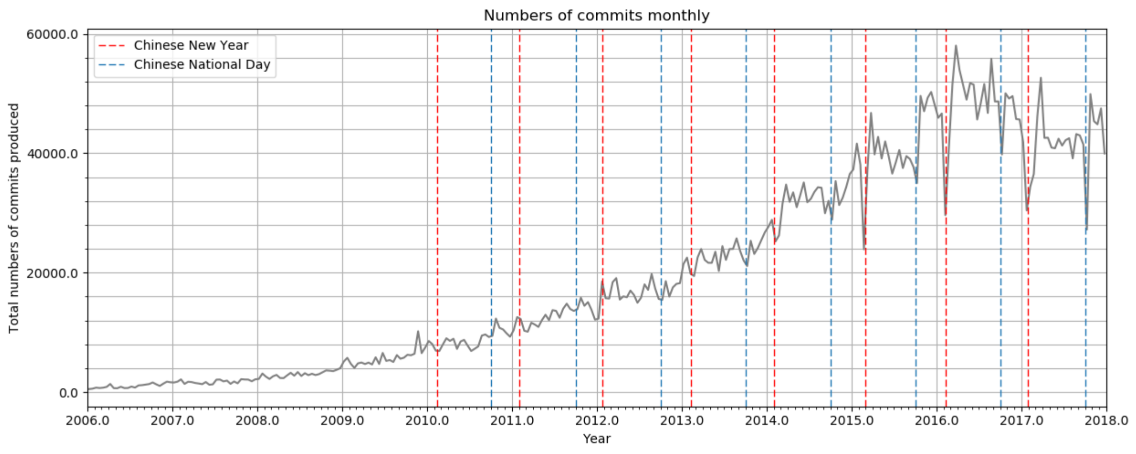


Figure 8. Growth of commits on Gitee monthly (sampled with double week time span)

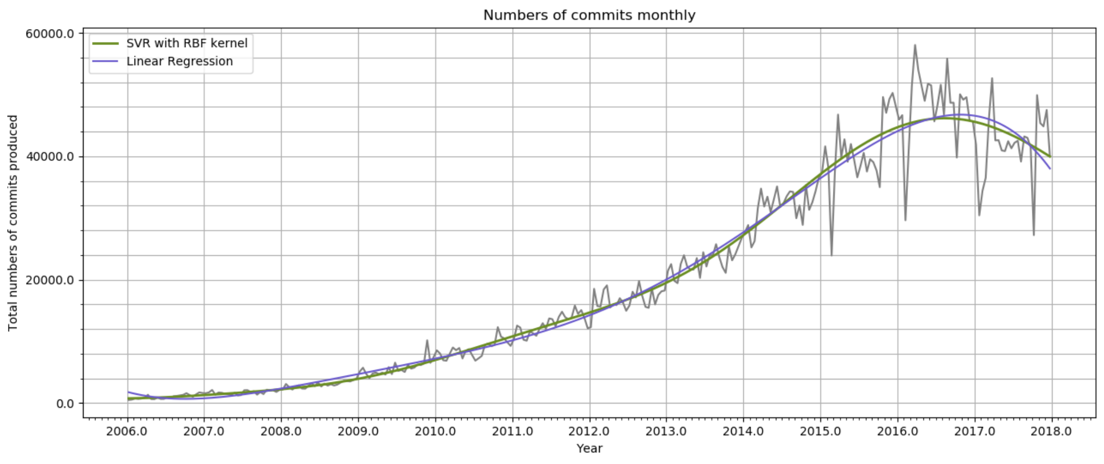


Figure 9. Regression results of growth of commits on Gitee monthly

To make a better presentation and visualization, we used support vector regression(SVR) with a kernel function of radial bias function together with linear regression with a polynomial model to fit a curve to the data, which is shown in the following figure. The results from two regression methods presented high similarity with each other. Following formula expresses the fitting result of linear regression:

$$Y = -3.7 * X^5 + 90.9 * X^4 - 760.5 * X^3 + 2991.8 * X^2 - 3287.2 * X - 1762.1$$

Where Y is the percentage of commits generated in working hours, X is half month number since from Jan. 1st 2016. This result aligns with another research conducted by M. Godfrey et al., which revealed that Linux kernel grew at a polynomial rate (Godfrey & Tu,2000).

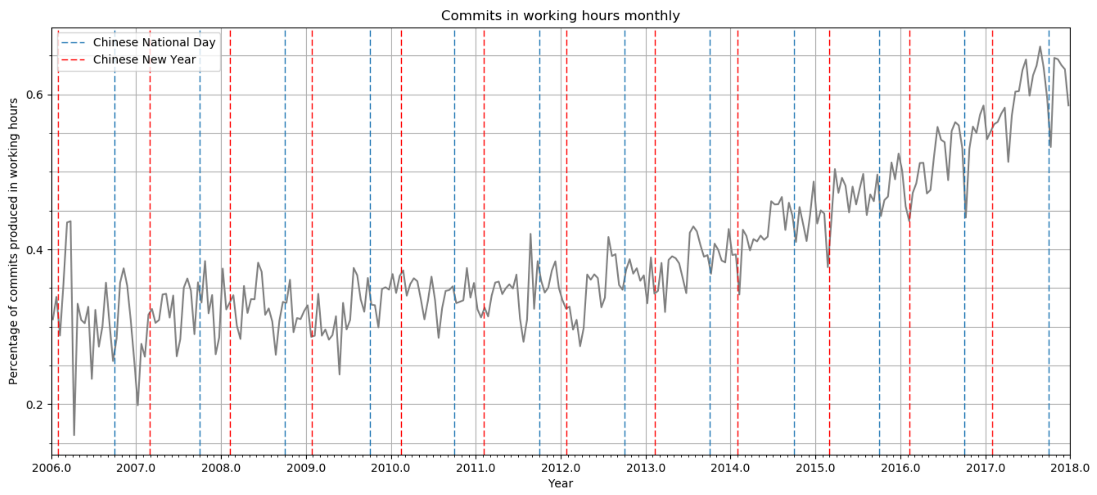


Figure 10. Percentage of commits produced in working hours monthly (sampled with double week time span)

Similar to the curve of total commit numbers, fluctuations make it hard to find any regular patterns at first glance. But we found some consistency when we focus on the term of traditional or statutory holidays. As can be read from the figure, in the 12-year period we surveyed, the percentage of commits made in working hours endured a sharp drop during

- 9 National Day holidays, which are from the year 2010 to 2017 and 2006.
- 8 Spring Festival holidays, which are from the year 2011 to 2016, 2006 and 2009.

Furthermore, there are no decreases detected near 3 out of 24 national holidays.

Using the same procedure as before, an analysis of the percentage of commits produced in working hours, which represents the amount of the paid work among the entire contribution made over the years. Figure 11 reveals a long-term increase in the percentage of commits produced during the working hours from 2006 to 2017. Similarly, support vector regression and linear regression are used here for better representation. Here we use the result of linear regression because of its simplicity.

$$Y = 0.00028 * X^3 - 0.0017 * X^2 + 0.0064 * X + 0.31$$

Where Y is the percentage of commits produced in the working hours and X is half month number since Jan. 1st 2016. The original research of western countries, however, revealed two different development patterns of Linux Kernel and the Ohloh Projects. Statistics OF Linux Kernel shows an increasing proportion of effort devoted in the working hours while those of the Ohloh Projects remained constant between the year 2006 and 2012 (Riehle, Riemer, Kolassa & Schmidt, 2014).

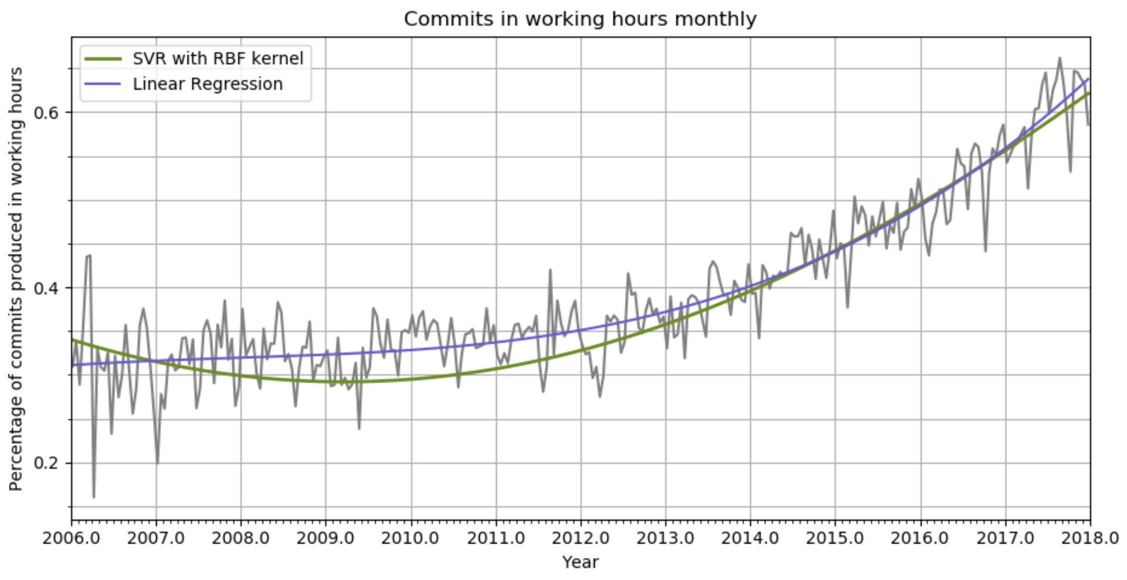


Figure 11. Regression results of percentage of commits produced in working hours

After comparison, we can find many shared characteristics between our statistics gathered from Gitee and the statistics of Linux Kernel. The percentage of both datasets rose from about 40% to 60% during the middle of the researched period. The difference between lies on both time and amplitude axis, the number of chinese software developers swung in a broader range. This process also took more extended time, 12 years in China and 6 years for Linux Kernel. From the trend we predict that the growth of commits generated in working hours will keep rising before converge after several years.

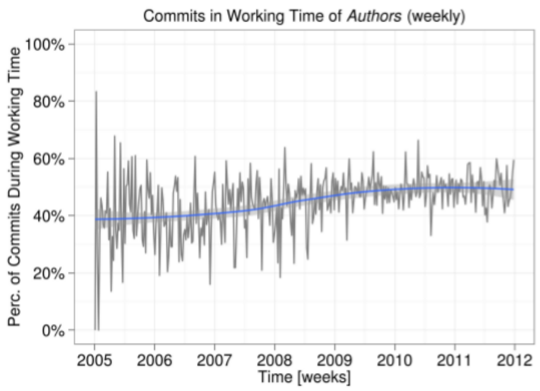


Figure 12.Data and trend line for percentage of commits made by authors to the Linux Kernel during working time for a given week (Riehle, Riemer, Kolassa & Schmidt, 2014)

The analysis on total commit numbers and percentage of commits produced in working hours deliverers us several insights and facts about chinese open-source software industry:

1. Exact time interval intensively influences both the amount and condition (paid or volunteer) of production in the year, such as holidays. Which helps to prove that the dataset used for the study are mainly composed of chinese software developers after purge out the abnormal data from other time zones or working while traveling.
2. Both the amount and condition show more similarity to Linux Kernel than the Ohloh Projects. With more and more participation and commercial support of IT companies and government in China, it is not surprising for chinese open-source software industry to find itself on the path parallel to the development of Linux Kernel. It is also important to notice that, this process took far longer in chinese open-source communities.
3. Despite the longer researched period than that of Linux Kernel, the percentage of commits produced in working hours has not converge yet till year 2018, showing a ongoing commercialization of chinese open-source software industry.

4.4 Developer Classification

After analyzing how the statistics of the whole industry changed over time, we focus on the individual developers in this chapter to find out some detailed facts. In the beginning, we calculated the percentage of commits produced in working hours for each developer and stacked the numbers together. The result is showed in Figure below, and it needs to be noticed that the Y-axis is displayed in log-scale. Here we also cite the result of research on western countries for better comparison.

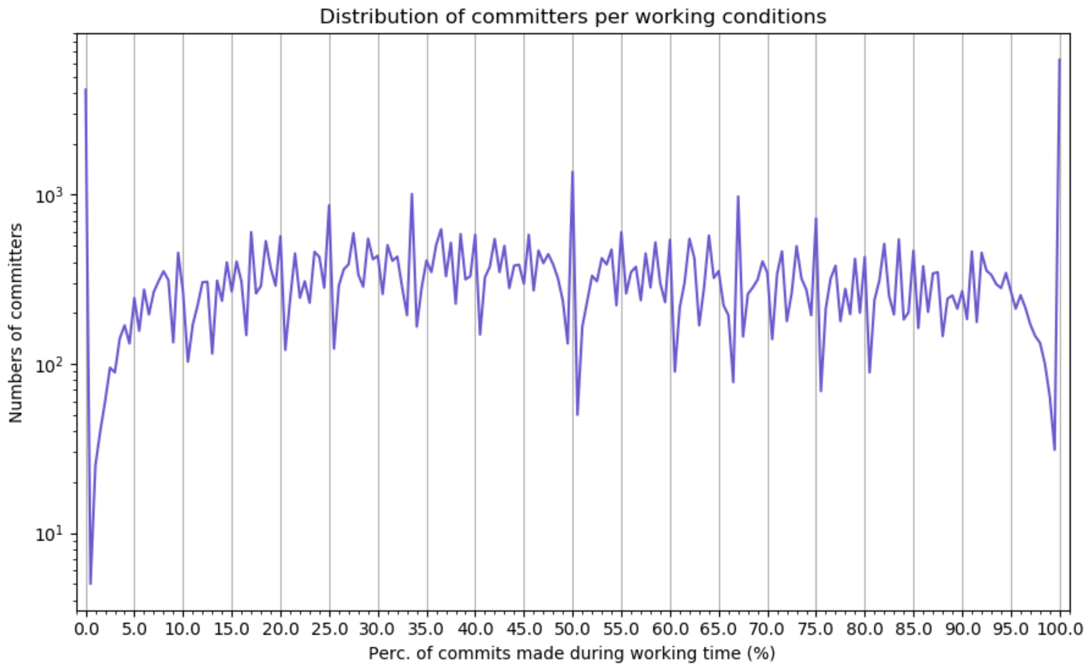


Figure 13. Numbers of committers with a given percentage of paid work on Gitee during years 2006-2017

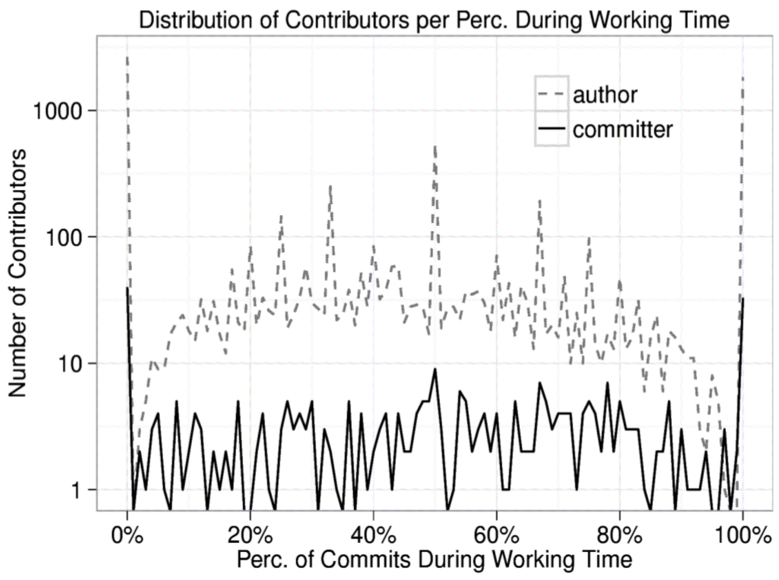


Figure 14. Numbers of committers with a given average percentage of paid work for the years 2006-2017 for the Linux Kernel (Riehle, Riemer, Kolassa & Schmidt, 2014)

If we pay more attention to both ends of the curve, we can see that the number of developers which work in extreme patterns dominates the distribution with 8.5% only commit in the working hours and 5.6% never do so. Detailed statistics are listed in Table 2. As following the classification boundaries in the western case research, we categorize developers who made their 95% of commits in the working hours as paid or employed developers and those with 5% or less commits as the volunteer or amateur developers.

Table 2. Distribution of volunteer to paid developers over years 2006-2017

Developer Types	Volunteer		Mixed	Paid	
Perc. of Commits in Working Hours	0%	0.01%-5%	5.01%-94.99%	95%-99.99%	100%
Perc. of Developers	5.6%	1.0%	82.2%	1.7%	8.5%

The result shows 10.2% of developers are most likely to be paid for their job, while 6.6% just make open-source development in their spare time. When we compare the statistics to those of western countries, the curve matches to the data from authors of Linux Kernel and comitters of Ohloh Projects. Yet, the exact numbers are not likely to categorized into any of the four group in the research of Riehle et al.. Since we revealed in last chapter that the distribution of paid and volunteer work in chinese open-source development is still changing with the time, a more precise conclusion needs to be made in the future.

Table 3. Distribution of volunteer (spare time) to paid (working time) developers, binned, over years (Riehle, Riemer, Kolassa & Schmidt, 2014)

Developer Types			Volunteer		Mixed	Paid	
Perc. of Commits in Working Hours			0%	0.01%-5%	5.01%-94.99%	95%-99.99%	100%
Perc. of Developers	Linux Kernel	Author	33.06%	0.35%	43.45%	0.17%	22.98%
	Ohloh Projects	Extended Committers	7.04%	0.4%	74.58%	0.41%	17.56%

But there still exists several differences between individual developers like the total commit number of developers. Figure 15 shows how this affects the distribution we calculated before: the two extreme cases no longer dominate the distribution when a population of committers with more than 100 commits is selected, which is also intuitive that the more one developer works, the more likely he is to not only commit in the working hours but also outside them. The more and longer a developer works the more his working are affected by random incidents, causing his data to show more diversity. In the later elaboration chapter, several new methods to classify those developers in mixed patterns will be introduced.

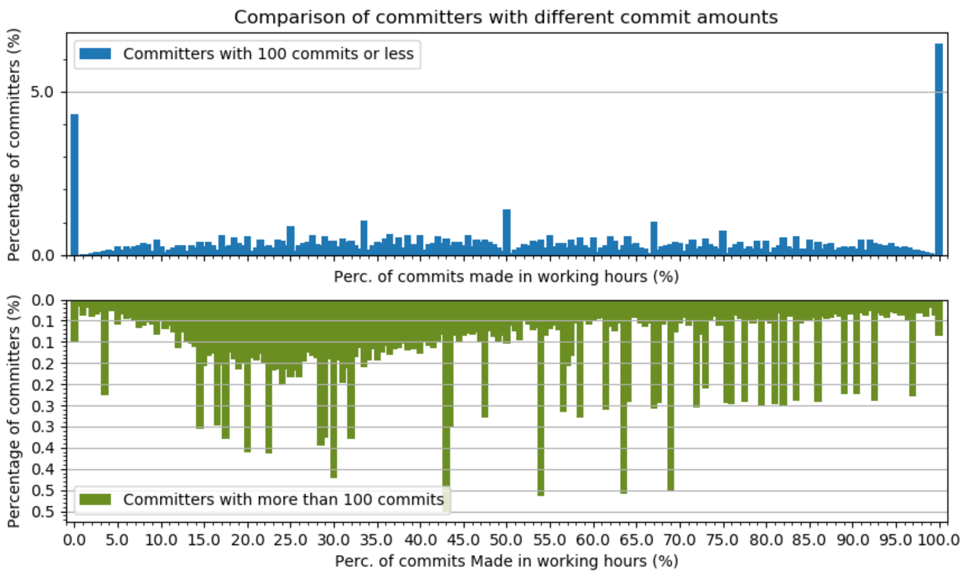


Figure 15. Comparison of committers with different commit amounts

4.5 Project Classification

Now we have analyzed about the confrontation between paid and volunteer development patterns with the overall distribution over time and statistics of the individual developer. There is another dimension left where a single project is considered as one sample. Through analysis of how each project was developed and maintained, we can not only gain a more precise answer of our research question but also use the statics as a tool to assess the health state of open-source projects (Riehle, Riemer, Kolassa & Schmidt, 2014).

We analyzed data from 127,365 repositories which is the same magnitude as the data quality researched in the western countries cases (more than 9,192 repositories). We counted the numbers of three different categories of developers in each project using previously defined boundaries that developers who make 95% of their commits or more in the working hours are paid developers and those making the same amount of commits in their spare time are volunteer developers. We finally get two almost mirrored curves with a similar distribution to previously researched distribution of individual developers showed in the figure below. Notice that to better display huge numbers, Y-axises are again showed in log-scale. That reveals the fact that both individual developers and individual projects are dominated by two extreme patterns, fully worked by paid developers or completely composed of volunteer ones, while handful rest of the projects scattered between.

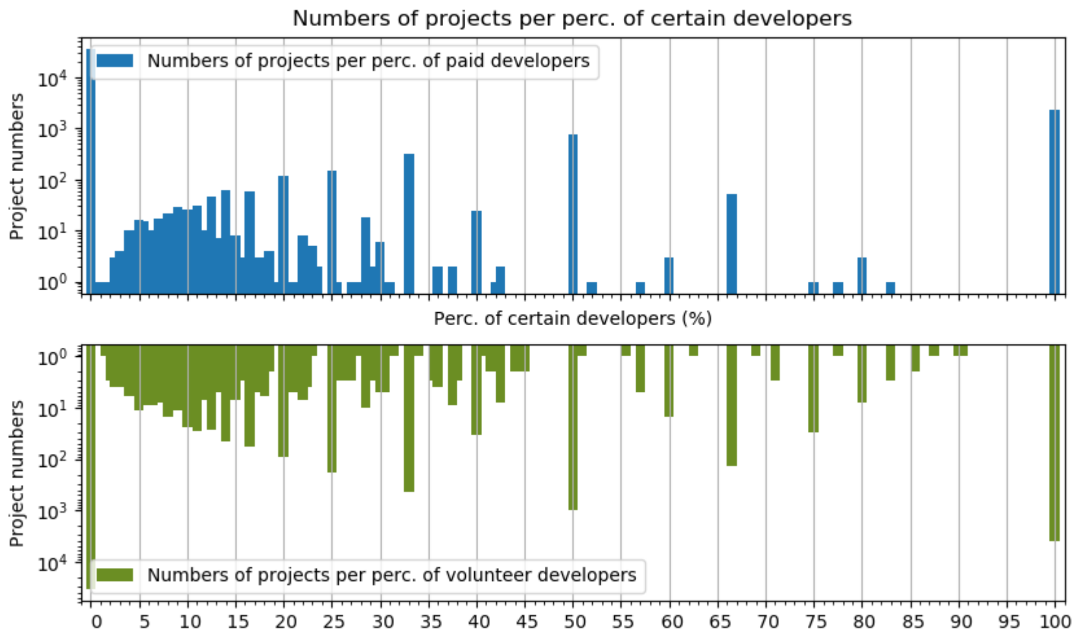


Figure 16. Numbers of projects with a given percentage of paid or volunteer developers

We already know that the distribution of two development patterns changes when a population of developers with more commits is selected, likewise it varies with the different sizes of projects. Developer numbers of both extreme patterns decrease significantly and do not dominate anymore when only repositories with more than 1,000 commits are inspected, which is the result of more developers working in mixed pattern.

This consistency of the distribution and how it changes throughout the analysis of individual developer and project indicates an intuitive casual relation: A project with larger size tends to be accomplished and maintained by developers with more commits. Meanwhile they trend to work both inside and outside the working hours when they devote more time into development.

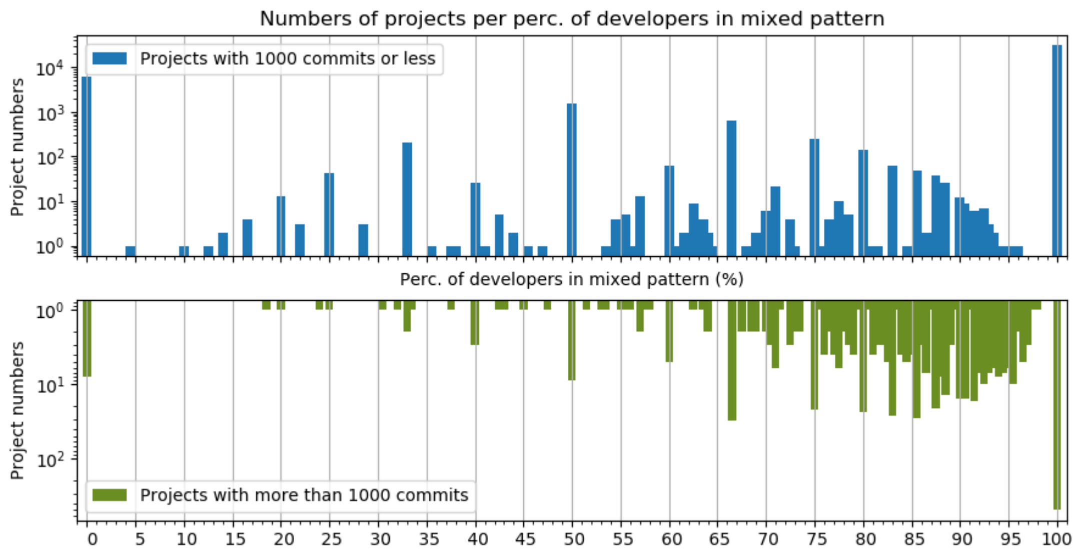


Figure 17. Numbers of projects with a given percentage of developers working in mix pattern

We managed to identify several projects in both extreme patterns after taking an in-depth look at the data. The top 4 projects developed only by paid developers are Buession, Hetao, VideO_transmission and iLOG3. Some of the statistics are listed in Table 4. Easy to notice that most of these projects are developed and maintained by one or two developers. Two out of four projects are even created by the same developer.

Table 4. Statistics about several fully paid projects

Projects Name	Contributors	Commit Numbers	Total Commit Numbers
Buession	Yong.teng	213	213
Hetao	calvinwilliams	173	173
VideO_transmission	SWFighter	84	130
	Altium	46	
iLOG3	calvinwilliams	96	96

As an echo of the western countries case research, here we also manually inspected the detailed commit information from these four projects. We found that:

- The developer with name calvinwilliams often used commit name like "Update to V 1.0" and put a large chunk of code in it. In the project Hetao, he made all the commits in this way.

- In the project named VideO_transmission, two contributors used no consistent naming rules for commit name. However, chunks of code are still frequently found in both their commits.
- By project named Buession, its contributor used to commit quite many times in one day using precisely same commit name such as “fix bug”. The code segments in these commits were usually several lines of code and dedicated to the single objective.

On the other hand, there are not many world-wide popular open-source projects held by chinese developers, most of which are run on the Github for global collaboration. But Gitee.com has its ranking and recommendation system which evaluates the project according to the number of the followers, the number of the stars it gets from the developers, etc. Here we use the most starred project recommendation to pick the projects with most influence and popularity (Each star means that the project has been added to a users favorites).

Table 5. Statistics of 6 top ranking projects according to Gitee

Project Name	Contributor Number	Watcher Number	Star Number	Fork Number	Total Commits	Perc. of Paid Developers
Zheng	13	5,900	12,300	5,700	1228	16.7%
SpringBoot	3	3,500	9,500	3,800	254	0%
iBase4J	10	2,800	6,500	3,300	1223	33%
JFinal	5	3,000	6,300	3,200	242	0%
MCMS	22	2,200	4,300	2,200	476	40%
OSChina Andorid	9	1,500	4,700	3,600	1006	16.7%

These projects are all watched and forked by quite many developers, all of them have been donated from 20 to 123 times. Although the exact donation number is private data, differences can still be identified when the four previously surveyed projects worked by paid developers had received no donate at all and were significantly out-numbered in every aspect. Thus, there are reasons to believe that those two kinds of projects are developed in two different ways:

- Fully paid projects usually worked by a small group of developers who only commit in the working hours. They use simple or sometimes same commit information to describe their work and commit mostly with chunks of code.
- Projects with grander scale or popularity show more diversity on the distribution of different kinds of developers. However, in this region, it is unusual to find a fully paid project or project with more than half of its developers being paid.

Compare to the situation in western countries. We again found many common characteristics, which is not a surprise after the consistency between revealed in previous chapters.

5 Discussion of Findings

In this thesis, we mainly used data gathered from Gitee.com which is a popular code management platform in China. Since Gitee is developed by one of the earliest open-source communities in China (Liu,2013) and provides us with a significant amount of data (around 139,197 projects developed and maintained by approximately 73,722 developers), any bias that exits in this dataset will also occur in other data sources. It is reasonable to believe that our data are sufficient for this research.

The data we collected from Gitee contained both author and committer information, however, during the initial part of the data analysis we found that in our case, these two parts of data are always identical, which may be a result of Gitee's log mechanism. Thus, we only analyzed committer data in this thesis.

In the next phase, we worked on the possible bias caused by different time zones. There is only one standard time (Peking Time) is used in China since 1949 (Guo,2001). We surveyed the provinces where software industry cluster, and the results indicate more than 90% of chinese software industry located around the UTC +8:00 time zone (Peking Time). Therefore, biases caused by people living in different longitude are not to be expected in this thesis. In addition, all the committers who made 50% or more of their commits from 0:00 to 6:00 in chinese local time are treated as noise and not counted in later analysis anymore. Since this time window is usual sleep time in China but highly productive hours in western countries (Riehle, Riemer, Kolassa & Schmidt, 2014), committers with dominating the number of commits in this time window are suggested to be overseas or traveling developers. The effect of this step is confirmed through later analysis where we found a sharp decrease of both total numbers of commits and commits during working hours near every national holiday in China.

We chose to start our analysis from the year 2006 because statistic and literature indicate that chinese open-source communities and developers had been significantly more active since 2006. After manual inspection of the data, many of the projects on Gitee which can be traced back before 2006 are actually not created by chinese developers. They are mostly old Github projects which were cloned to Gitee with their original information being kept. As a matter of fact, there are still many foreign projects cloned by chinese developers after 2006. Our solution to this bias are time zone examination for overall data and manual inspection when focused on a single project. On the other hand, the new data after the time we run data collection (February 2018) is not be found in our dataset. However, we stop analyzing data with a timestamp that later than December 31st 2017, in order to not disturb the annual analysis. The result of monthly total commit number analysis shows an identical trend to Linux Kernel, which indicates that the time span we chose is suitable.

As for the classification boundaries for paid and volunteer work, we followed the same definition used in the western countries case research for better result comparison, but we extended it for 1 hour to compensate the lunch break. As discussed in that paper (Riehle, Riemer, Kolassa & Schmidt, 2014), we also consider this boundaries conservative because overtime working is far more usual in China than in western countries (Wang, 2016; Zheng & Lu, 2006).

6 Related Work

This thesis is inspired by research on the distribution of paid and volunteer work in open-source by Riehle et al.. Data form Linux Kernel and Ohloh Projects are used in the research, the results revealed, “about 50% of all contributions to projects in our sample population have been paid work” (p.9) , and how percentage of paid developers can be used to evaluate the health state of an open-source project, “larger projects have healthy mixture of paid and volunteer work in the 10-20% range as well” (p.9). Due to the lack of data from UTC +8:00 time zone, similar research with data gathered in Asia can be used to support the results or reveal more findings.

There are many pieces of research on both the development and market of chinese open-source software conducted by local and foreign researchers. Keith Bergelt (2017) did a survey about how open-source operation systems are utilized and developed in China and Japan, which stated limited contributions to the global open-source community in the past and the current changes of the situation, especially in the telecoms industry. He also mentioned about one of the reasons why open-source was facing hard times in China, “ However, the so-called ‘Great Firewall of China’ essentially, a closed Internet – has clearly had a dampening effect on global collaboration between developers within China and elsewhere in the world. ” (p.46) Earlier in the year 2007, Pan and Bonk had pointed out that “The open-source software movement is gaining increasing momentum in China. Of the limited numbers of open-source software in China, Red Flag Linux stands out most strikingly, commanding 30 percent share of the Chinese software market” (p.1). They focused on the Linux and educational purpose software to predict an increasingly growth of chinese open-source software. In contrast, it is not easy to find any researches that cut in from commercialization or uses data science methods. We have also not found any other database or projects that monitor the statistic of chinese developers’ daily work. It is obvious that this field has not gone interested for both the scholars, government or companies, which suggests that chinese open-source industry is currently not so mature compared to Europe or North America.

7 Conclusion

This thesis is dedicated to analyze and reveal the statics and facts about commercialization in chinese open-source software industry. We used data gathered from chinese local code management platform Gitee and conducted similar research referring to the research focused on western countries by Riehle et al.. A classification boundary that a commit uploaded during regular working hours (from 9:00 to 18:00 local time, weekdays) are considered as paid work and the rest are treated as volunteer work. After surveyed around 139,197 projects with around 73,722 developers, from the year 2006 to 2017, we calculated that 46.2% of total commits are paid work. By comparing the numbers of commit generated hourly to western countries, we revealed the fact that most people pause their work during lunch break in China.

When we examine the data over the years, we found out that both the statistics of total commit number and percentage of paid work in it can be fitted with polynomial functions. They both experienced a long-term growth since 2006. Similar to lunch breaks, national holidays also have an enormous impact on both statistics.

In the next phase, a more precise analysis is made on individual committers. Statistics show that two extreme patterns dominate the distribution in the whole dataset. However, developers tend to work in a more mixed pattern when he produces more commits. The results are compared to both Linux Kernel and Ohloh Projects and our statistics are relatively more matched with Linux Kernel in each aspect, which suggested that chinese open-source software industry may share more other common characteristics with the development of Linux Kernel.

In addition, we ran an analysis on individual projects to see how they were composed of. It delievered an identical result of the previous study on individual developers. Projects are dominated by those accomplished by fully paid or volunteer developers in general, but their developers show more diversity when the size of the projects increases. We also took a more in-depth look at some typical projects with manual inspection. It appeared that these different kinds of projects were developed in different ways. Fully paid projects are usually in small scale and held by handful developers while larger projects with more popularity contained a percentage of paid developers under 50%.

Since we have a relatively large dataset which contains lots of uninterested and noise data, we are always looking forward to finding more information from the data and eliminating bias or noise in the data. In the next elaboration chapter, we tried to analyze the data using machine learning techniques to discover new features or validate our gained results.

8 Elaboration

8.1 Introduction

In the previous sections, we treat every single commit as one individual sample, that means we assume every commit is independent of each other. Even when those samples belong to one committer and dedicated to one projects, we would still classify them into different groups according to their time-stamps. It will simplify the data analysis significantly and is feasible when we focus on the macroscopic distributions. However, it still left a fuzzy classification of those developers and projects who do not follow extreme patterns, more precisely, those contained both commits in working hours or spare time simultaneously. Furthermore, the ignorance of the consistency between each commits causes massive bias when we focus more on the specific developer or project. Thus in this elaboration section, we are devoted to finding several new approaches that calculate the distribution of paid and volunteer work regard with the consistency between commits.

8.2 Population, Samples and Features

First of all, it is intuitive to consider that, a single developer would maintain a specific work condition (paid or volunteer) in his one specific project. In chapter 3.3.2 we stated that our data contained many related but unlinked committer identities, due to the different email addresses they used as names of their Single-Sign-On keys. Following Bayesian Network illustrated the causal relations between each variable in our system.

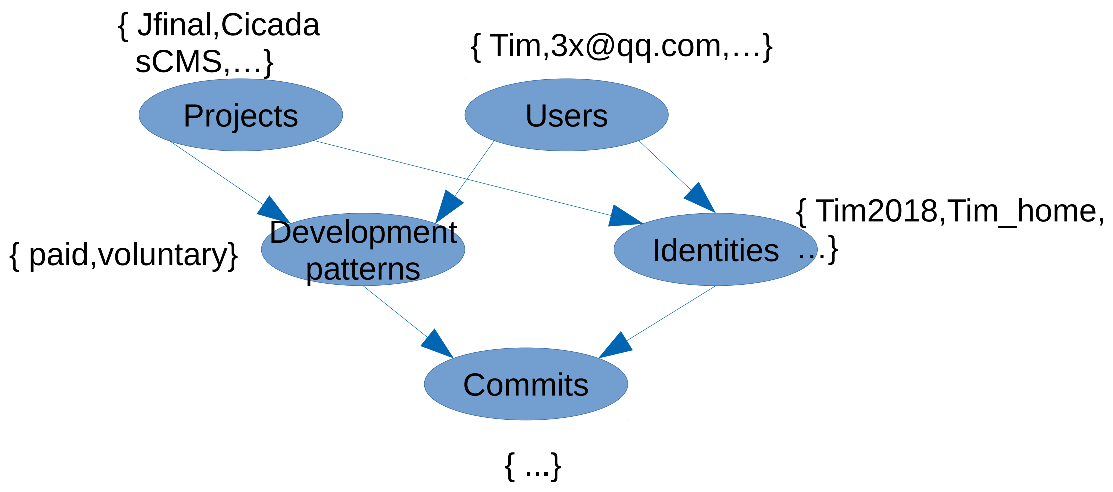


Figure 18. Bayesian Network of variable in the data

As we know about the mechanism of Git-based code management platform, the SSO key is a unique signature of one user. Which means if two commits came with the same SSO key, they must belong to one developer. Here we don't consider the situation where one developer does the work and commit on an account and Git settings of others, because that is too complicated for us in this thesis to identify. And from the assumption above, we can infer that those commits with same SSO key must follow the same pattern, here to be precise, paid or volunteer.

Thus we treat commits with the same SSO key (committer identity) which dedicated to a specific project as one sample. And the numbers of commit produced in particular time spans are used as the features of this group of commits.

The original data set is also too hardware and time consuming for quick implementation and validation of our classification approaches. So we randomly chose data from 1,000 projects out of the whole 139,197 projects to form a new data set. From the previous chapters, we established a way to classify committers and projects into three groups: fully paid, fully volunteer and mixed. Now we use fully paid and volunteer samples as the training set to find a model that classify the mixed samples into these two groups, but only more harsh. We cleaned out every sample that contains commits in the time span from 2:00 am to 5:00 am. Besides, we only consider samples with more than 99% of commits in specific paid or volunteer time window as our training samples. We chose our training samples so strict to make sure that less bias would be introduced as possible. Detailed statistics of our dataset are listed in the following table.

Table 6. Statistics of training set and data set

	Previous classified groups	Sample number	Total commit number
Training set	Paid	3,576	19,340
	Volunteer	3,366	13,717
Data set	Mixed	5,366	124,789

After selection of the training and data sets, we need to define which statistics as features of these samples. We have the time-stamp of each commit in our training and data sets. Technically, the number of commits produced in each time span from year 2006 to 2017 depending on how long it is chosen, is a unique signature of a committer. However, it is going to not only introduce too many random factors which we are not interested like the mood of that committer, his family status or even the weather of that day, but also make the calculation too complicated to accomplish. Thus, we stack the data of one sample together according to the days in a week and we sampled the number once an hour. That means for each sample, there are 24*7 features.

Figure 19 and 20 illustrated how the statistics of training set look like.

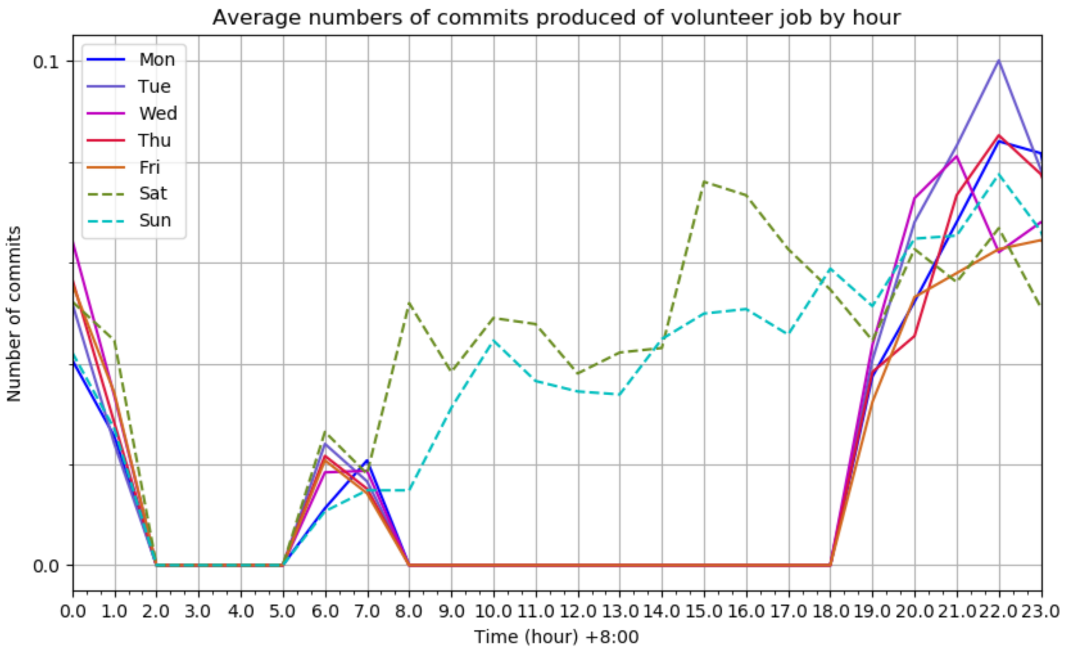


Figure 19. Average statistics of training set of volunteer job

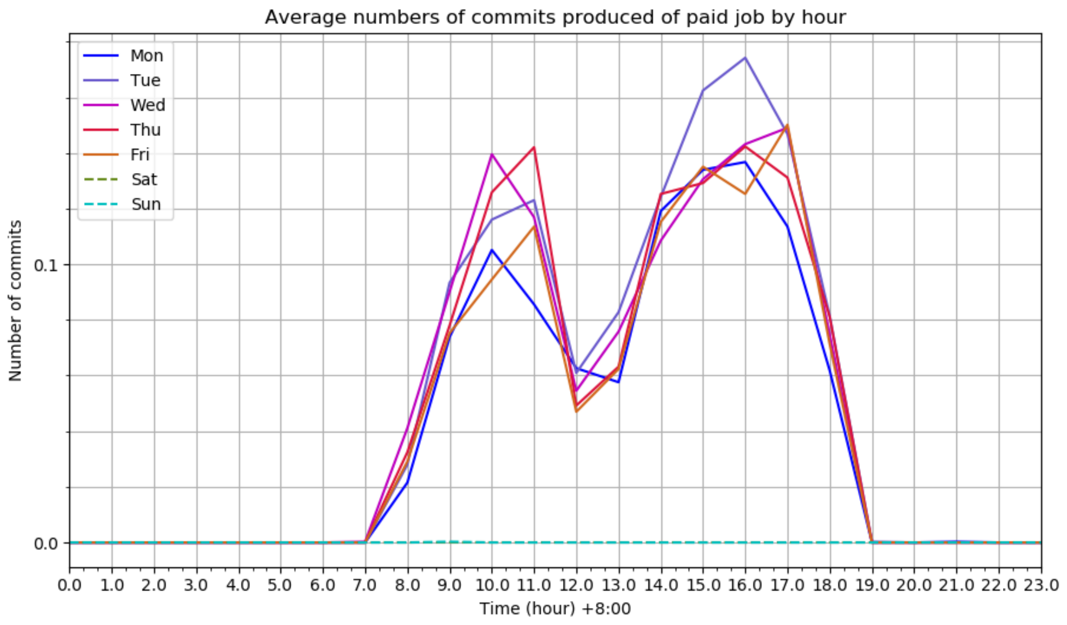


Figure 20. Average statistics of training set of paid job

8.3 Logistic Regression Approach

To start simple with this binary classification problem, we use Logistic Regression as classification algorithm and assume that the classification boundary is a straight line first. We used the open-sourced machine learning package Scikit-learn and trained a Logistic Model with our training set. Then we used this model to classify the data set. The results are shown in Figure 21.

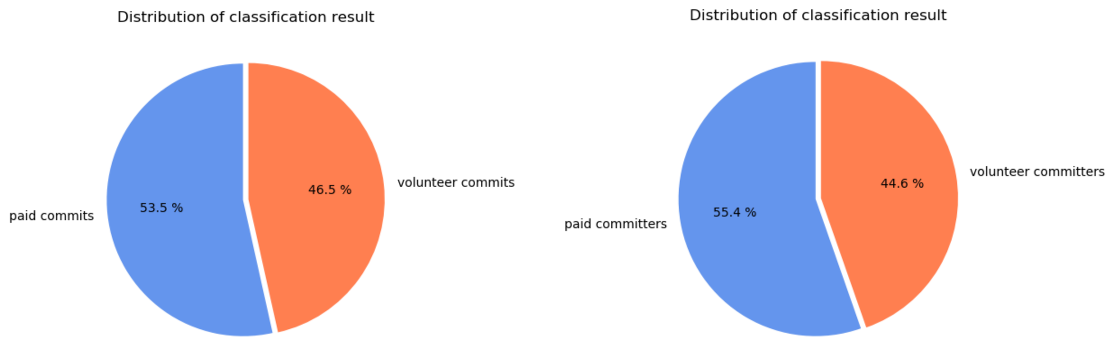


Figure 21. Classification results of Logistic Regression

Results show that both committer and commit numbers are almost divided equally into two classes, with slightly more in paid category. Furthermore, Logistic Regression also deliver us the possibility of each sample to be paid or volunteer group. For example, Figure 22 shows two samples we randomly picked in the data set. The sample in left figure with a significantly high commit number on Monday from 10:00 to 13:00 and low activity on weekends, Logistic Regression returns a high possibility with 95.26% that it appears to be paid committer. In the meantime, the sample in right figure with more diversity has only 22.4% possibility to be paid committer. The high amount of commits on early mornings and in late afternoons push it more to the volunteer side.

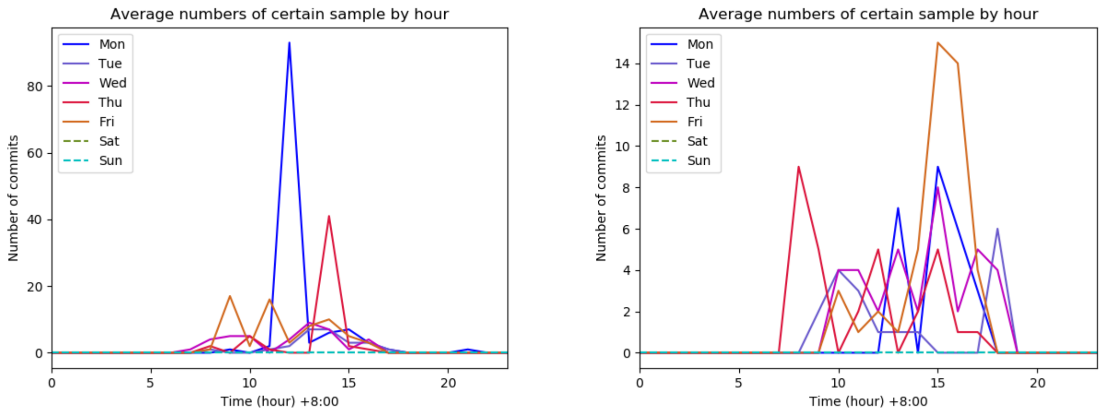


Figure 22. Statistics of a random samples

8.4 Discussion of Findings

In this short part of elaboration chapter, we proposed a new perspective to classify our data into paid and volunteer groups. We believe that by calculate the commit from single committer to specific project as an entirety, a more microscopic distribution of data can be gained. We aimed to classify every single committer using a Logistic Regression Model trained with two pre-defined training sets. We composed the training sets with harsh conditions such as no commits should be in the regular sleeping hours to avoid possible bias.

The result aligns with the previous ones, but this time we can tell how likely a committer with both commits in paid and volunteer time windows to be each specific group. It opens a new path to the original purpose of this thesis and leaves a broad wide space for improving the results with different machine learning algorithms and preprocessing methods.

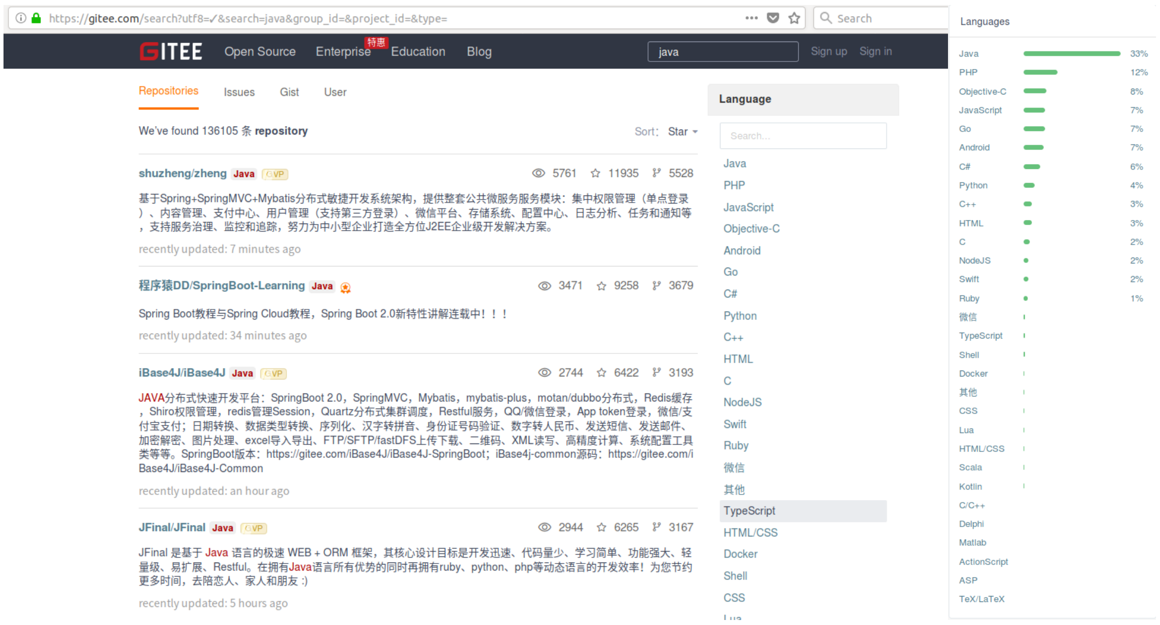
Appendix A Data Collection

After selection of the data source, the task of this phase of research is to retrieve data from the Website. However, usage of the Gitee API requires necessary pieces of information as the parameter of the request. Since Gitee only provides us API via Web Services for data query while no public index files or databases that stores the metadata of the repositories, the names and owner information of all the repositories are not available at the very beginning of data collection.

The path through this first maze is the search engine provided by Gitee. The search engine accepts several keywords of the search, such as:

1. Partial Name of the projects or users
2. Words used in the description of the projects including their applications or purpose.
3. Types of programming languages adopted by the projects.

We currently have no knowledge about (1) and (2), yet there are not countless many types of programming languages. Also, Gitee has listed some programming languages as recommended keywords. By searching with these keywords, we can get an overview of all the public repositories, which is showed in the following picture. Moreover, the search engine automatically ranks the results according to its activity level. The detailed algorithm being used cannot be reviewed, but it is obvious that factors like last commit date, total commit numbers and project size, etc. are taken into calculation from the perspective of the result sequence. Which also simplified the data collection because the inactive projects, e.g., projects that only have initial commits, will be located in the back of the queue. These inactive projects are less interested to us because they don't contain representative data of the samples.



We've found 136105 条 **repository**

shuzheng/zheng Java GVP 5761 ☆ 11935 5528

基于Spring+SpringMVC+Mybatis分布式敏捷开发系统架构，提供整套公共微服务服务模块：集中权限管理（单点登录）、内容管理、支付中心、用户管理（支持第三方登录）、微信平台、存储系统、配置中心、日志分析、任务和通知等，支持服务治理、监控和追踪，努力为中小型企业打造全方位J2EE企业级开发解决方案。

recently updated: 7 minutes ago

程序猿DD/SpringBoot-Learning Java 3471 ☆ 9258 3679

Spring Boot教程与Spring Cloud教程，Spring Boot 2.0新特性讲解连载中！！！

recently updated: 34 minutes ago

iBase4J/iBase4J Java GVP 2744 ☆ 6422 3193

JAVA分布式快速开发平台：SpringBoot 2.0，SpringMVC，Mybatis，mybatis-plus，motan/dubbo分布式，Redis缓存，Shiro权限管理，redis管理Session，Quartz分布式集群调度，Restful服务，QQ/微信登录，App token登录，微信/支付宝支付；日期转换、数据类型转换、序列化、汉字转拼音、身份证号码验证、数字转人民币、发送短信、发送邮件、加密解密、图片处理、excel导入导出、FTP/SFTP/fastDFS上传下载、二维码、XML读写、高精度计算、系统配置工具类等等。SpringBoot版本：https://gitee.com/iBase4J/iBase4J-SpringBoot；iBase4j-common源码：https://gitee.com/iBase4J/iBase4J-Common

recently updated: an hour ago

JFinal/JFinal Java GVP 2944 ☆ 6265 3167

JFinal 是基于 Java 语言的极速 WEB + ORM 框架，其核心设计目标是开发迅速、代码量少、学习简单、功能强大、轻量级、易扩展、Restful。在拥有Java语言所有优势的同时再拥有ruby、python、php等动态语言的开发效率！为您节约更多时间，去陪恋人、家人和朋友 :)

recently updated: 5 hours ago

Languages

Java	33%
PHP	12%
Objective-C	8%
JavaScript	7%
Go	7%
Android	7%
C#	6%
Python	4%
C++	3%
HTML	3%
C	2%
NodeJS	2%
Swift	2%
Ruby	1%
微信	
TypeScript	
Shell	
Docker	
其他	
CSS	
Lua	
HTML/CSS	
Scala	
Kotlin	
C/C++	
Delphi	
Matlab	
ActionScript	
ASP	
TeX/LaTeX	

Figure A1. Searching repository information using search engine

Like many other data science researches nowadays, the scale of the data that need to be collected is often quite large. For example, Gitee returns 136,105 entries of results when keyword ‘Java’ is used. In this case, a search engine crawler has been written to log these data automatically, and the result is stored in a JSON file, which is a standard format for data exchange on the Internet. The collection procedure itself took less than an hour and generated files of 22 MB.

After having the repository information, the next phase is to gather corresponding data which including numbers of contributors, the time-stamp of the commits, etc. Which is accomplished by another Web API crawler similar to the former one. Following graph (Figure A2) illustrates how these components work together. It took about two weeks to finish data collection in this step due to the significant larger data scale. As a result, it generated files with a total size of 42 GB. The collection finished in March 2018. Therefore the data generated after is not taken into analysis.

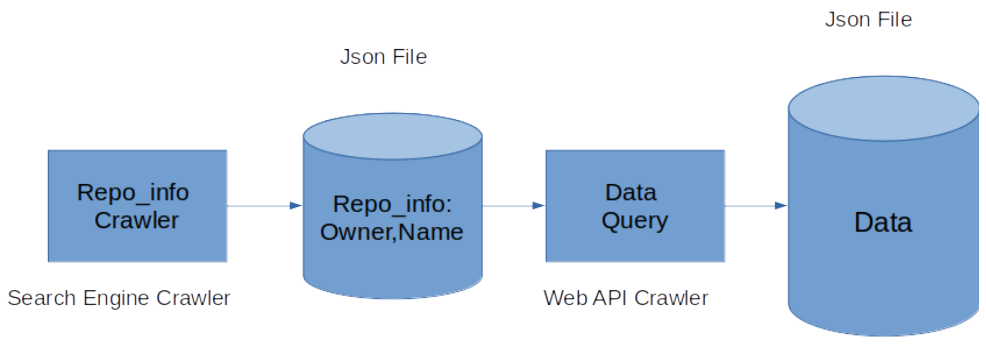


Figure A2. Structure of data collection programm

Appendix B Data Cleaning and Storage

Same as the initial output of both crawlers, the JSON format is used to store the raw data gathered by the crawlers. If we look into the data more thoroughly, it's not hard to find out that it is organized in a graph, more precisely tree data structure. Which is illustrated in the following figure.

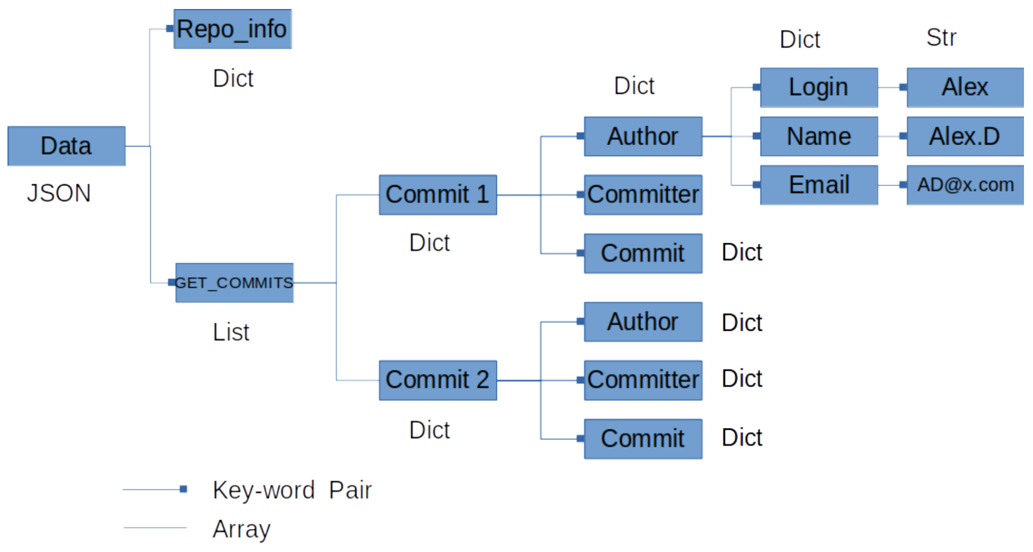


Figure B1. Data structure of stored the data

The raw data collected from the response of the web service requests sent by the crawler contains a significant amount of irrelevant information to our study. To process the data both time and memory efficiently, we need to filter out the irrelevant information and extract the desired features.

To search in a graph, the two most common algorithms are Depth-first Searching and Breadth-first Searching. Here the Depth-first searching is used to achieve the balance between time and memory efficiency. Since every single entry we searched does not contain a large scale of depth but a large scale in breadth, using BFS will result in a lower memory need yet slightly longer execution time against BFS.

The detailed process is being carried out layer by layer, a list of keys is pre-defined and stored for each layer which is in the form of a dictionary. The searching function runs recursively to reach the leaf node (a dictionary that its words are in String format). The underlying figure shows how the searching runs stepwise.

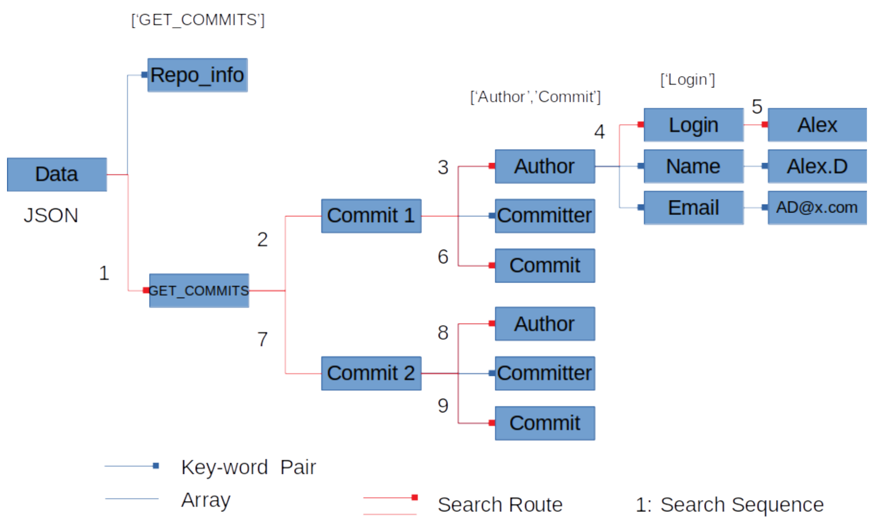


Figure B2. Search sequence of stored data

Appendix C Calender Date of Chinese New Year

Year	Calender Date
2006	January 29th
2007	February 18th
2008	February 7th
2009	January 26th
2010	February 14th
2011	February 3rd
2012	January 23rd
2013	February 10th
2014	January 31st
2015	February 19th
2016	February 8th
2017	January 28th

References

Riehle, D., Riemer, P., Kolassa, C., & Schmidt, M. (2014). Paid vs. Volunteer Work in Open Source. *2014 47Th Hawaii International Conference On System Sciences*. http://dx.doi.org/10.1109/hicss.2014.407

Moore, S. (2018). Is Open Source Alive in China? – Inside Machine learning – Medium. Retrieved from https://medium.com/inside-machine-learning/is-open-source-alive-in-china-3f606aafbd3b

Pan, G., & Bonk, C. (2007). The Emergence of Open-Source Software in China. *The International Review Of Research In Open And Distributed Learning*, *8*(1). doi: 10.19173/irrodl.v8i1.331

Liu, K. (2013). *Research on the Development Status and Trend of Open Source Software Industry in China* (Master). Huazhong Normal University. Retrieved from http://www.wanfangdata.com.cn/details/detail.do?_type=degree&id=Y2351647

How is the current situation of Open-Source-Software in China?. (2018). Retrieved from https://www.zhihu.com/question/21965679/answer/19882451

Keith Bergelt, K. (2017). *The state of open source in Asia* (p. 46). Intellectual Asset Management | May/June 2017. Retrieved from http://www.iam-media.com

Feature comparison between Gitee and GitHub. (2018). Retrieved from https://gitee.com/contrast

Protalinski, E. (2018). GitHub Blocked in China. Retrieved from https://thenextweb.com/asia/2013/01/21/the-chinese-government-appears-to-have-completely-blocked-github-via-dns/

Zhang, C. (2018). Working Time System in China « Employment Law in China. Retrieved from http://www.attorneycz.com/2011/04/06/working-time-system/

Chinese people's Congress. Labour Law of the People’s Republic of China (1995).

Takhteyev, Y., & Hilts, A. (2010). Investigating the geography of open source software through GitHub. *Manuscript submitted for publication*.

Guo, Q. (2001). A Study on the Standard Time Changes for the Past 100 Years in China. *China Historical Materials Of Science And Technology (In Chinese)*, *22*(3).

Song, H., & Zhang, M. (2015). The Distribution of China's IT industry and its Determinants. *Zhejiang Academic Journal*, *2*.

Ma, G. (2015). Food, eating behavior, and culture in Chinese society. *Journal Of Ethnic Foods*, *2*(4), 195-199. doi: 10.1016/j.jef.2015.11.004

Godfrey, M., & Tu, Q. (2000). Evolution in open source software: a case study. *Proceedings International Conference On Software Maintenance ICSM-94*. doi: 10.1109/icsm.2000.883030

Wang, T. (2016). System Structure and Legislation of Working Time Benchmark. *Journal Of Northwest University Of Political Science And Law*. doi: DOI:10.16290/j.cnki.1674-5205.2016.01.013

Zeng, X., & Lu, L. (2006). Dual challenges of standardization and flexibilization. *Journal Of Renmin University Of China*. Retrieved from http://xsqks.ruc.edu.cn/Jweb_rdxb/CN/article/downloadArticleFile.do?attachType=PDF&id=9184