The Open-Closed Principle of Modern Machine Learning Frameworks

Houssem Ben Braiek
SWAT Lab., Polytechnique Montréal
houssem.ben-braiek@polymtl.ca

Foutse Khomh
SWAT Lab., Polytechnique Montréal
foutse.khomh@polymtl.ca

Bram Adams
MCIS, Polytechnique Montréal
bram.adams@polymtl.ca

ABSTRACT

Recent advances in computing technologies and the availability of huge volumes of data have sparked a new machine learning (ML) revolution, where almost every day a new headline touts the demise of human experts by ML models on some task. Open source software development is rumoured to play a significant role in this revolution, with both academics and large corporations such as Google and Microsoft releasing their ML frameworks under an open source license. This paper takes a step back to examine and understand the role of open source development in modern ML, by examining the growth of the open source ML ecosystem on GitHub, its actors, and the adoption of frameworks over time. By mining LinkedIn and Google Scholar profiles, we also examine driving factors behind this growth (paid vs. voluntary contributors), as well as the major players who promote its democratization (companies vs. communities), and the composition of ML development teams (engineers vs. scientists). According to the technology adoption lifecycle, we find that ML is in between the stages of early adoption and early majority. Furthermore, companies are the main drivers behind open source ML, while the majority of development teams are hybrid teams comprising both engineers and professional scientists. The latter correspond to scientists employed by a company, and by far represent the most active profiles in the development of ML applications, which reflects the importance of a scientific background for the development of ML frameworks to complement coding skills. The large influence of cloud computing companies on the development of open source ML frameworks raises the risk of vendor lock-in. These frameworks, while open source, could be optimized for specific commercial cloud offerings.

KEYWORDS

Machine Learning, Open Source, Framework, Technology adoption

ACM Reference Format:
Houssem Ben Braiek, Foutse Khomh, and Bram Adams. 2018. The Open-Closed Principle of Modern Machine Learning Frameworks. In *MSR '18: MSR '18: 15th International Conference on Mining Software Repositories , May 28–29, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3196398.3196445

MSR '18, May 28–29, 2018, Gothenburg, Sweden
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5716-6/18/05…$15.00
https://doi.org/10.1145/3196398.3196445

1 INTRODUCTION

Nowadays, data is everywhere and its volume is growing exponentially everyday. The International Data Corporation (IDC) estimates that 40 zettabytes (40 billion terabytes!) of data will be created in the next two years alone, with the business value of big data projected to surpass $203 billion by 2020 [8]. Because of these impressive figures, data is considered to be the oil of the digital era. However, unlike oil, the value of data does not merely stem from its volume, but rather from the rich insights that can be generated from it through analytics. Explanatory and predictive data analytics is an inter-disciplinary field that brings together computer science, statistics and mathematical modeling to generate a useful explanation of an observed phenomenon, and make predictions or draw insights based on patterns identified within data [9].

The main workhorse of data science is Machine Learning (ML). ML allows developing intelligent systems that give computers the ability to learn hidden patterns without being explicitly programmed [16]. The produced computer programs encapsulate complex algorithms and sophisticated mathematical models that can perform classification, regression, clustering as well as recommender system tasks [4]. The ability to learn-on-the-job, then to automate prediction of future pattern occurrences enables ML to drive value without ongoing human intervention.

As the community strives to ready ML for prime time, we are witnessing a spike in open source ML frameworks. In 2015, Google released its powerful framework *TensorFlow* [3] to the public, under the Apache 2.0 open source license, stating "we hope this will let the machine learning community - everyone from academic researchers, to engineers, to hobbyists - exchange ideas much more quickly, through working code rather than just research papers." Other companies, including Baidu, Facebook, Microsoft and Amazon, and research labs, such as Yoshua Bengio's MILA lab, quickly followed suit or had done so already. Thanks to these open source frameworks, companies and individuals can now leverage state-of-the-art ML algorithms with minimal overhead.

However, given what's at stake for modern ML, as well as the many stakeholders involved (companies and universities, engineers and scientists, volunteers and paid employees) the question is how "open" open source ML framework development really is. Can anyone join and contribute, or does "open-ness" degenerate to "cost-free source code"? Do open source ML frameworks experiment with new kinds of development processes, are they run like other hybrid open source projects such as the Linux kernel or Android [5], or are they run like traditional open source projects?

This paper aims to understand the role played by open source in the democratization of ML, and identify key actors behind this phenomenon. To achieve this goal, we analyzed 598 core contributors of 20 top open source ML frameworks in depth and also analyzed 4,099

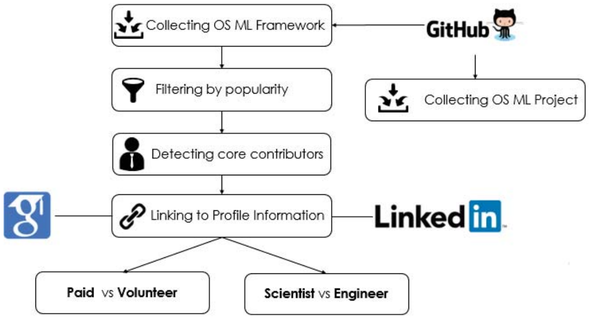

Figure 1: Data collection process.

ML-related open source projects in order to answer the following three research questions:

- **RQ1. What is the state of adoption of open source ML Frameworks?** The goal is to determine at what stage we have arrived in the adoption of machine learning technologies, by measuring the growth in number of frameworks, their adoption as well as the number of ML-related applications created on GitHub.
- **RQ2. Who drives the democratization of ML: Company or Community?** We compare contribution measures of companies and of the ML community to understand who is the major player in the diffusion of ML technology. The measures considered are the proportion of committed code, and the number of community-driven frameworks (resp. company-driven frameworks) that are dependent on company-driven frameworks (resp. community-driven frameworks). This can tell us more about the development process used by open source ML frameworks.
- **RQ3. Who drives the development of Open Source ML Frameworks: Scientist or Engineer?** We mined GitHub for data on contributors, then linked this data with LinkedIn (most popular business social network) and Google Scholar (academic search engine) data. The obtained data allows to compare the contributions of scientist contributors with those of engineer contributors. We aim to understand the degree of multidisciplinarity required for ML framework development.

The remainder of this paper is organized as follows. Section 2 describes our data collection process and provides background information about the technology adoption lifecycle. Section 3 presents and discusses the results of our study. Section 4 discusses threats to the validity of our study. Section 5 summarizes the related literature, while Section 6 concludes the paper.

2 CASE STUDY SETUP

In this section, we describe the data collection process used to address the research questions presented in the introduction and explain the technology adoption lifecycle.

2.1 Data Collection

The data used for this study were collected from the GitHub ecosystem following the process illustrated in Figure 1. We detail below its corresponding steps.

2.1.1 Collecting Open Source Repositories. Project data was collected from GitHub. First, the most popular machine learning frameworks available were extracted using GitHub's search API. This search used technical terms related to Machine Learning such as "machine learning", "deep learning", "statistical learning", "neural network", "supervised learning", "unsupervised learning" and "reinforcement learning", and known keywords such as "toolkit", "tool", "framework" and "library". We then manually filtered the resulting search results based on the license file, README, list of contributors (and affiliations) and GitHub's built-in wiki, eventually arriving at a list of 104 projects.

To estimate the state of adoption of these machine learning frameworks, it is difficult to determine which ML project in GitHub uses which specific framework, so we track the evolution of the numbers of followers of the framework as measure of its adoption, but we also extend our data by a set of ML projects in general as estimation of global interest in ML technology. To identify open source projects that implement machine learning algorithms (either using a generic-purpose framework or from scratch), we use GitHub API to search for projects about machine learning. Specifically, we use a new feature in GitHub search API which allows exploring repositories by a label titled "topic". This label is added by the project team and its role is to create subject-based connections between GitHub repositories. In this study, we retrieved all projects that defined machine learning as their topic. In total, we extracted 4,099 repositories that define "machine learning" as a topic

2.1.2 Filtering by Popularity. To compare the role of companies and data science communities in the development of ML technology using qualitative analysis, we first sampled a representative sub-set of the collected frameworks as follows: we computed the number of stargazers of each framework, and selected the set of frameworks which cumulative number of stars represent 80% of the total number of stars attributed to ML projects On GitHub, the number of stars of a given project captures the appreciation of the community towards that project. This yielded a total of 29 popular frameworks.

Second, we qualitatively analyzed these projects' descriptions on GitHub, their corresponding organizations, and their official Websites to determine whether they are owned by a company, which released and primarily supports the project, or by the data science community. Out of the 29 selected frameworks, 12 projects appeared to be driven by a company and 17 by the community. Therefore, to obtained a balanced dataset and a reasonable number of frameworks for the qualitative analysis, we selected the top-10 company-driven and the top-10 community-driven projects. These 20 projects represents 70% of the total number of stars attributed to ML projects on GitHub. Table 1 presents some descriptive statistics about the 20 projects used in our study.

2.1.3 Detecting Core Contributors. In order to understand the composition of teams developing these frameworks, we needed to determine the contributors who participate in the development of the frameworks' core. So, we defined those core contributors as the smallest set of contributors whose total contributions in the source code repository accounted for 90% or more of the total contributions. We selected a threshold of 90% here (instead of 80%) to obtain a reasonable number of contributors, since (as explained in the next subsection) obtaining the profile information of contributors is not straightforward. Table 1 presents the resulting number of analyzed core contributors.

2.1.4 Linking to Profile Information. Afterwards, we extracted the user profile of each contributor of ML frameworks on GitHub using his git username, which is a unique identifier for a user in GitHub. We found that many contributors on GitHub do not have public information about their functions and their companies. Hence, we developed a Web Scraper to explore data on LinkedIn and Google Scholar, to obtain contributors' information via their full name, if available on GitHub. LinkedIn is the most popular and the largest business social network in the world. As of January 2018, LinkedIn had 530 million members in 200 countries [1]. Google Scholar is a web search engine of scholarly literature. While Google does not publish the size of Google Scholar's database, third-party researchers estimated it to contain roughly 160 million documents as of May 2014 [13].

In particular, we aimed to determine if a contributor is a scientist working in the R&D lab of a company, a software engineer or an academic researcher. By comparing the name of the company or organization where the contributor works and the one owning a framework, we can also determine if the person is a paid employee or a volunteer.

For community-driven frameworks, we also identified external developers who are paid by third-party companies ("company-sponsored developers"). To do that, we grouped the contributors of each framework by their employer in order to compute the total commit count of contributors who work in each company. We then try to detect the third-party companies that sponsor the community-driven framework by fixing a threshold of 50% for the percentage of their contributors and their corresponding contributions count. More specifically, we consider that a framework is sponsored by a company if more than 50% of commits are submitted by employees of the company. In order to reduce the number of false positives in our results, we manually verified on the frameworks' official Websites that these frameworks are officially sponsored by the companies (by looking for "powered by" or logo amongst the official sponsors). Table 1 reports the number of core contributors of each framework, and the number of contributors for which we successfully identified the profile. We share our datasets in our on-line appendix at: https://github.com/hoss-bb/msr-2018.

2.2 Technology Adoption Life Cycle

The technology adoption life cycle curve describes the process of adoption of a new technology over time [7]. It is typically illustrated as a classical normal distribution (Figure 2) representing a sociological model that indicates the demographic and psychological characteristics of defined adopter categories. Each category has been determined to have traits that affect their likelihood to adopt an innovation. The model shows that the first individuals to adopt an innovation are "innovators" who have the closest contact to scientific sources and other innovators, followed by "early adopters" who have the highest degree of opinion leadership among the other adopter categories.

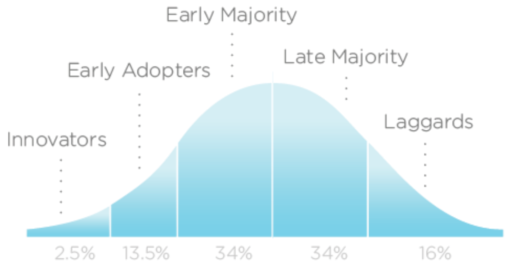

Figure 2: Technology adoption life cycle curve. [2]

Next comes the "early majority" category who adopts an innovation only after a varying degree of time, followed by "late majority" who will adopt an innovation after the average member of the society (since they take less risks). The last group to eventually adopt a technology are the so-called "laggards" who are very conservative and typically tend to be tied to traditions. While the passage of time is necessary for innovations to be adopted, the period of time required to move from one category to another is not predictible in advance. This period varies according to how potential adopters evaluate an innovation on its perceived advantages, its compatibility with the pre-existing technologies, its complexity, its testability and its potential for reinvention.

We estimate the adoption cycle of ML frameworks by considering the moment at which stargazers starred each framework, since stargazers represent the users who keep track of projects they find interesting and want to try out later . While this does not guarantee that the stargazer eventually did adopt a technology, as long as the number of stars is sufficiently large it provides an indication of the interest of GitHub developers in the framework.

3 CASE STUDY RESULTS AND DISCUSSION

This section presents and discusses the results of our three research questions. For each research question, we present the motivation behind the question, our analysis approach and a discussion of our findings.

RQ1. What is the State of Adoption of Open Source ML Frameworks?

Motivation. Sonnenburg et al. [17] have argued that using the open source model of sharing information and software implementations would be highly beneficial for the machine learning field, because it would ease adoption by researchers and professionals from other disciplines and various industries. This RQ aims to study progress in the development of open source ML frameworks as well as the adoption of this technology by GitHub users, including data scientists, developers, researchers, etc.

Approach. First, we measure the evolution of the number of ML frameworks introduced each year on GitHub. Second, we evaluate the evolution of the number of new ML adopters of these popular frameworks, by identifying users that bookmarked the framework in GitHub (i.e., stargazers). Starring is used in GitHub to show

Table 1: Basic information on the top-10 company-driven and the top-10 community-driven frameworks. "ccc" represents the total number of core contributors, while "iccc" the number of core contributors whose personal profile was identified.

Company						Community					
Name	ccc	iccc	stars	created	License	Name	ccc	iccc	stars	created	License
tensorflow	82	70	78208	2015	Apache-2.0	Scikit-learn	61	56	23033	2010	BSD-3
CNTK	29	23	13104	2015	1bit-SGD	keras	121	107	21836	2015	MIT
deeplearning4j	10	9	7672	2013	Apache-2.0	caffe	28	25	21295	2013	BSD-2
spaCy	3	3	7062	2014	MIT	incubator-mxnet	76	67	12058	2015	Apache-2.0
caffe2	34	27	6293	2015	Apache-2.0	pytorch	49	45	9251	2016	BSD-3
Paddle	20	17	5815	2016	Apache-2.0	torch7	32	29	7455	2013	BSD-3
sonnet	4	4	5555	2017	Apache-2.0	Theano	35	33	7297	2011	BSD-3
deeplearnjs	8	8	4710	2017	Apache-2.0	tflearn	35	31	7101	2016	MIT
amazon-dsstne	9	8	3947	2016	Apache-2.0	pattern	1	1	5842	2011	BSD-3
neon	19	18	3306	2014	Apache-2.0	ntlk	23	17	5567	2009	Apache-2.0

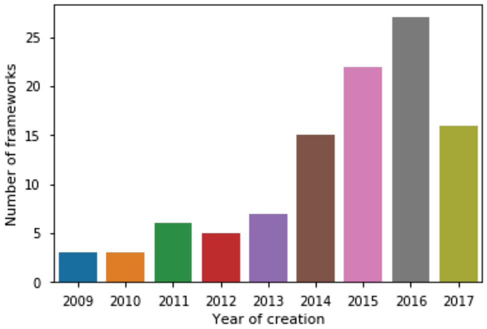

Figure 3: Number of frameworks per year of creation

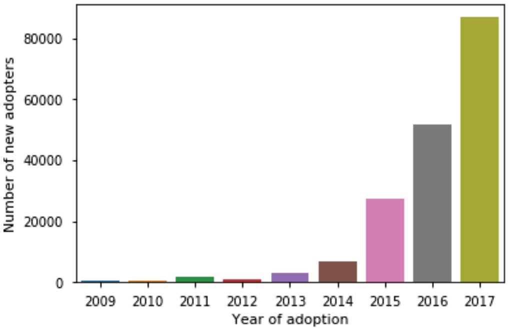

Figure 4: Number of new adopters per year of adoption

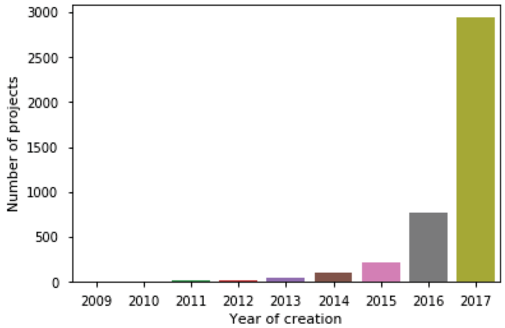

Figure 5: Number of projects per year of creation

interest in projects. Third, we examine the evolution of the total number of open-source projects that focus on machine learning in GitHub (including frameworks and regular software applications). We separate these projects by programming languages with the aim of comparing the popularity of different programming languages within the ML domain.

Findings. Figure 3 shows a substantial increase in the number of ML frameworks published since 2014. This result is not surprising given the big investments made by ML firms to promote the use and development of machine learning tools.

There was a spike in the number of new adopters of ML in 2015, according to Figure 4. We attribute this spike to the release of several deep learning frameworks by major ML players: Tensorflow by Google, CNTK by Microsoft and caffe2 by Facebook, and two popular community-driven frameworks: incubator-mxnet and keras. After 2015, the number of new adopters continue to increase exponentially, reflecting the growing popularity of these key frameworks. The number of regular machine learning projects on GitHub also increased steadily, years after years since 2015 (see Figure 5). In addition to the availability of open-source frameworks, another potential contributor to the expansion of ML is the availability of massive data and the effectiveness of ML algorithms in extracting useful information from these data. By projecting data about new adoptions of ML and new ML project creation on the adoption curve presented in Figure2, which is a Gaussian distribution, we can see that we are currently on the rising slope of the curve, moving towards the maximum. Hence, if we assume that GitHub users represent the population of ML users and that ML's open-source frameworks and applications represent ML technology, we can conclude that we are between the stage of early adoption and the stage of early majority. Looking at the distribution of programming languages used in the studied open source ML projects, we observe that Python is the most used language, followed by Java, then Matlab, then R (see Figures 6 and 7). Since we were analyzing projects that develop machine learning algorithms or use standard machine learning libraries, we were expecting to see R dominate other programming languages (given its popularity in the scientific community).

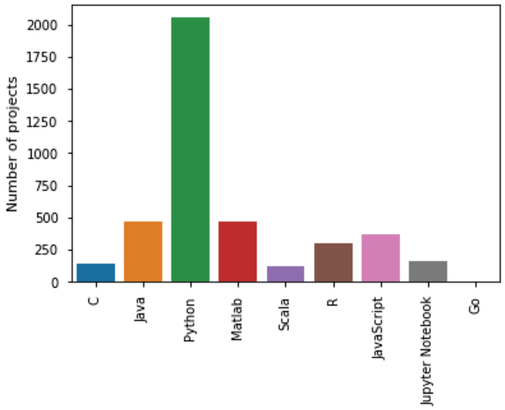

Figure 6: Number of projects by programming language

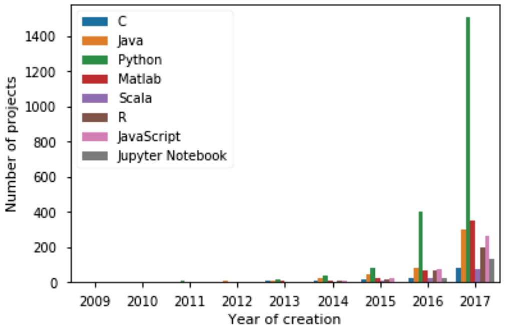

Figure 7: Number of projects by programming language per year of creation

However, this substantial increase in the popularity of Python can be explained by its use by industry professionals for scientific computation and machine learning, because it simplifies the integration of ML models to web applications and production databases. Indeed, with Python, one can encapsulate scientific code for performing large-scale mathematical calculations inside a software system's business logic. There are excellent Python libraries in a variety of disciplines directly related to data science, such as statistics, machine learning, data processing, data visualization and much more in fields adjacent to it, such as image processing and web development. This rich ecosystem reduces the collaboration gap between the team of scientists and engineers in the development of these hybrid applications (i.e., ML applications), since they share the same language even though they work with different APIs.

> *According to the technology adoption lifecycle, ML is currently between the stages of early adoption and early majority and Python is dominating ML software development.*

RQ2. Who Drives the Democratization of ML: Company or Community?

Motivation. In **RQ1**, we observed a spike in machine learning frameworks promoted by companies after 2015. These frameworks led to a large adoption of ML by developers on GitHub. However, we observed that although these company-driven ML frameworks are released under open-source licenses, their development and maintenance is controlled by company employees. Also, other companies are supporting community-driven ML open-source software projects by paying developers to contribute in the projects. This wide influence of corporate in the development of ML is raising suspicions about the "real" openness of open source ML framework development. This RQ aims to clear out these suspicions.

Approach. To answer this research question, we consider the top-10 community-driven frameworks and the top-10 company-driven frameworks on GitHub. To quantify the contribution of companies and the community in the development of these frameworks, we proceed as follows. For both company-driven frameworks and community-driven frameworks, we extract the list of core contributors as described in Section 2.1.3 and examine their profiles to identify the motivations behind their participation to the projects (i.e. if they are paid participants or volunteer). We achieve this by verifying where the employer of a contributor to a framework is a sponsor or an owner of the framework. We considered a contributor to be volunteer when his employer has no relation with the framework. We classify core contributors in three categories: company employee, company-sponsored contributor, and community volunteer. To capture the influence of corporates on a project, we calculate the percentage of its core contributors who are working for a company or sponsored by a company. We also calculate the percentage of commits submitted by these company employees and company-sponsored contributors.

In addition to tracking contributors, we also examine the code source of the studied frameworks and extract information about library used. We want to know if some of our studied community-driven frameworks build on–or are dependent of some company-driven frameworks and vice versa.

Findings. The development of open source ML frameworks was initially driven by the community, as shown on Figure 8. The first company-driven ML frameworks in our dataset were released in 2013. Since then, the number of ML framework backed by companies have surpassed the number of community-driven ML frameworks. Figure 9 shows the number of new adopters generated each year by each category of ML framework. As expected, the growth of the number of adopters of ML frameworks supported by tech giants largely surpass the number of adopters of community-driven ML frameworks. This phenomenon can be attributed to the large spendings made by tech giants like Google and Facebook to promote their frameworks. We believe that marketing campaigns and public conferences related to newly created company-driven frameworks have played an important role in their advantage over community-driven frameworks.

Figures 16 and 17 show the distribution of ML community-driven frameworks that are sponsored by companies. From these figures, it appear that beside releasing their own ML frameworks, companies actively contribute financially to community-driven ML frameworks by providing and-or supporting skilled professionals.

Figures 10 and 11 show the distribution of community-driven ML frameworks that build on company-driven frameworks. We did not find any ML company-driven framework in our data set that builds on a ML community-driven framework. This is another evidence

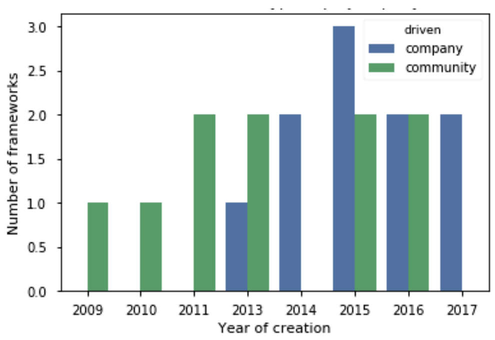

Figure 8: Number of frameworks for each type of project (i.e. community-driven or company-driven) per year of creation

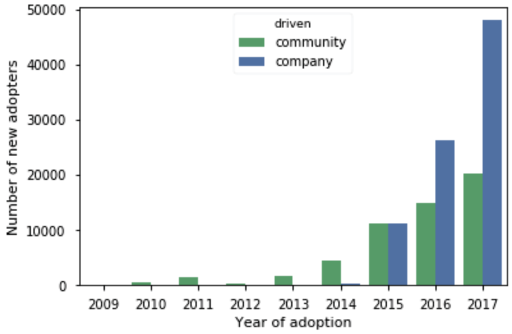

Figure 9: Number of adopters for each type of project (i.e. community-driven or company-driven) per year of adoption

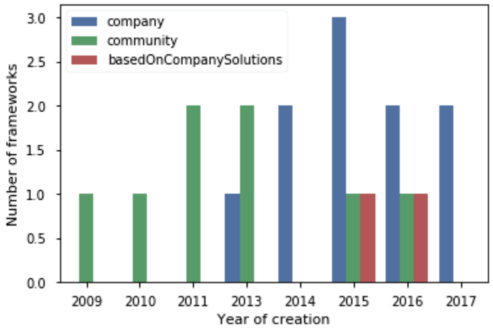

Figure 10: Number of frameworks for each type of project (adding the new category "based on company solution") per year of creation

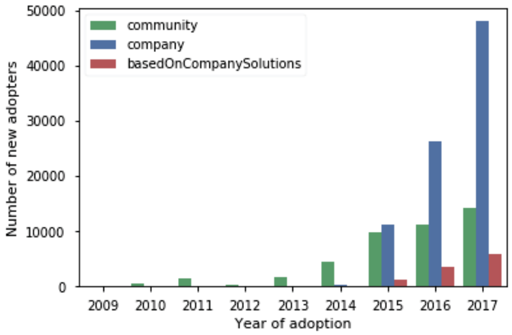

Figure 11: Number of adopters for each type of project (adding the new category "based on company solution") per year of adoption

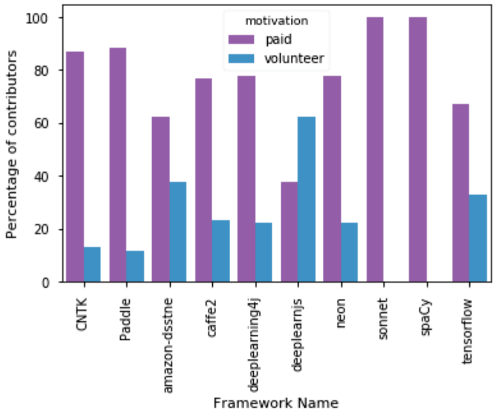

Figure 12: Percentage of contributors by motivation for each company-driven framework

that corporates are playing an important role in the democratization of ML.

Four notable community-driven ML frameworks that build on company-driven frameworks are *PyTorch*, *Torch*, *incubator-mxnet* and *Keras*. In these projects, the number of commits submitted by company-sponsored contributors exceeds 50% of the total number of commits of the projects (see Figures 13). In all these projects except *Keras*, company-sponsored contributors also make more than half of the total number of contributors to the project. In the case of *Keras*, the scientist who created the project on GitHub is now a Google employee, and Google is a supporter of the project. The majority of contributions to *Keras* are made by Google developers despite the fact that they represent only a small fraction of *Keras* contributors.

In the majority of analyzed frameworks, we observed fairly large numbers of volunteer contributors (see Figure 12). However, the contributions of these volunteers to the projects (in terms of number of commits) is significantly lower (see Figure 13) than the contributions on paid contributors (i.e., company employee, company-sponsored contributor).

We noticed that the frameworks named *spaCy* and *sonnet* contain respectively 3 and 4 contributors who are responsible for more than 90% of the total contributions. But, they contain in total respectively 117 and 16 contributors. So, although these two projects contain many volunteers. The contributions of most the these volunteers are minimal. However, the low number of contributors observed in the repository of *sonnet* is attributable to its young age. In fact, *sonnet* was created in 2017.

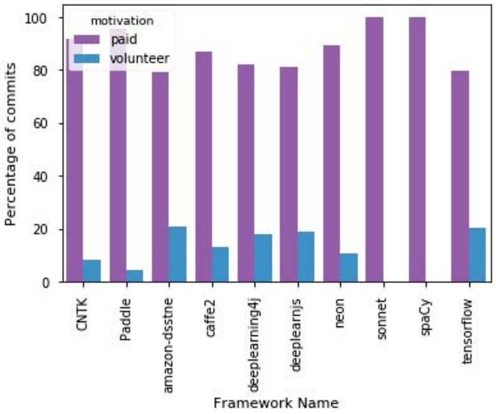

Figure 13: Percentage of commits by motivation of the author for each company-driven framework

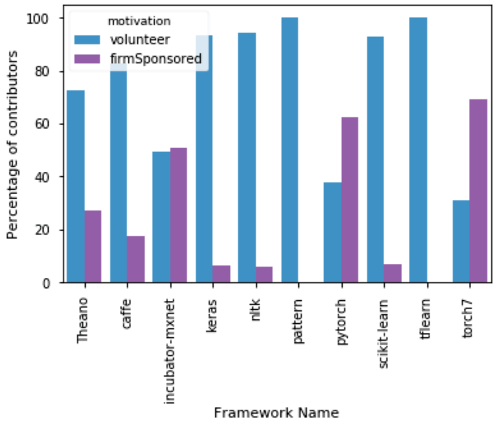

Figure 14: Percentage of contributors by motivation for each community-driven framework

> *Although initially driven by the academic community, the democratization of ML is now dominated by companies. Beside releasing and promoting their own ML frameworks, these companies actively contribute financially to community-driven ML projects, by supplying skilled professionals.*

These findings are in line with previous results obtained by Bird et al. on two widely used open source projects [6]: Firefox and Eclipse. They observed that IBM is responsible for more than 90% of the commits to Eclipse, while the Mozilla Corporation has contributed more than 50% of the source code of Firefox. Which suggests a possible subversion of open source ideals by large organizations.

To better understand the implications of this wide corporate involvement in the promotion of ML and their implicit motivations, we examine three important aspects of ML applications: data, computing power and algorithms. Apart from frameworks that implement ML algorithms, scientists need good processing power and large amounts of data to build effective and viable ML models.

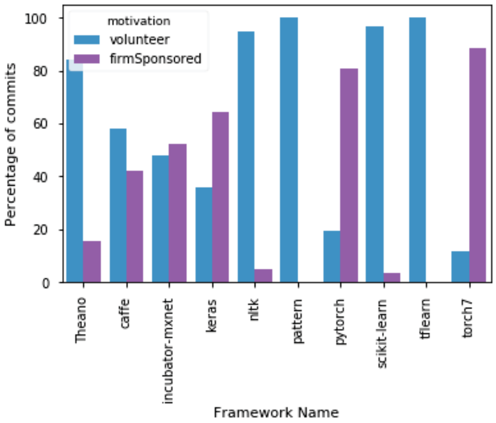

Figure 15: Percentage of commits by motivation of the author for each community-driven framework

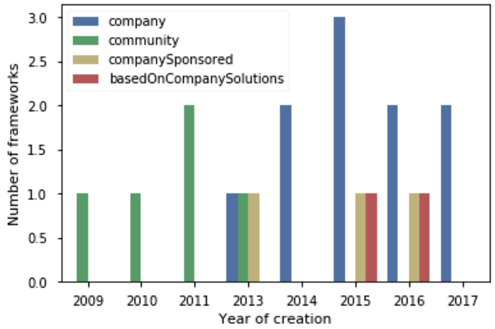

Figure 16: Number of frameworks for each type of project (adding the new category "company-sponsored") per year of creation

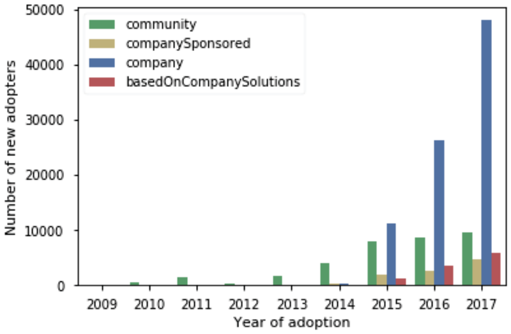

Figure 17: Number of adopters for each type of project (adding the new category "company-sponsored") per year of adoption

1) The need for processing power: Since cloud computing has become a strategic business unit for IT companies, which rent access to the processing power of mutualized computing resources via Internet, dominating the development of open source machine learning core frameworks can strengthen a cloud infrastructure offering. Tech giants release machine learning platforms that work best into their broader SDK or cloud-platform strategy in order to minimize IT infrastructure costs and offer great computing performance to their customers. This guarantees that all open developments with their official tool will be uploaded to their cloud platforms. The most striking example is the tensor processing unit (TPU) which is an ML accelerator chip developed by Google [11]. Those Cloud TPUs were designed from the ground up to accelerate machine learning workloads, and more specifically designed for models built with Google's TensorFlow framework.

Regarding this dependence between the tool used and the model trained, we believe that the unification and standardization of the machine learning models, which are defined as an outcome of the process of fitting and evaluating ML algorithms on training and testing data, can make it possible to transfer models among all the frameworks available on the ecosystem. This would eliminate the technical dependency to a cloud service provider or a deployment platform. Because each software or infrastructure platform have advantages and disadvantages, users have to make hard choices, which could be ease by a framework transfer solution.

2) The need for massive training data: Referring to supervised learning techniques, training a model to learn a desired behavior requires fitting algorithms on a large dataset of similar instances from different contexts. Since the richest data sit inside the servers of large companies like Google and Facebook, data scientists are often forced to use the APIs offered by these companies to access their vast quantity of high quality data in order to train ML models. However, generative models [15] represent an interesting alternative to this lock-in situation. After being trained on a significant amount of data from a domain, a generative model can produce synthetic data and use it to train a second model, based on supervised learning. Leveraging this free synthetic data, could reduce the dependence to tech giants' datasets and APIs. Generative models is a strategic direction of research in the ML community nowadays.

RQ3. Who Drives the Development of Open Source ML Frameworks: Scientist or Engineer?

Motivation. Google, one of the leading developers of ML (and other) frameworks, has a tradition of using a "hybrid approach" for its research endeavours [18] in order to marry innovation with rapid delivery of products. This approach follows an iterative process and usually involves writing production, or near-production, code from day one. In order to generate scientific and engineering advances, long-term, challenging research projects are split into discrete, shorter-term steps, combining both mathematical reasoning and engineering, monitored through empirical analysis.

Unfortunately, there are risks associated with the close integration of research and development activities, in particular the concern that research might take a back seat in favor of shorter-term projects. Hence, measures should be taken to avoid this, such as making some researchers work with engineers to rapidly iterate on existing products, while others are focusing on forward-looking projects. Since the latter affects directly the composition of the teams developing the company-driven ML frameworks, this RQ aims to study the composition of those ML teams in terms of engineers and scientists to identify the degree of "hybrid" teams in use and the potential for specialization to engineering or research.

Approach. To answer this research question, we define three profiles for contributors: *ResearcherPro* who are scientific researchers currently working in the industry, *Engineer* who represents software engineers and computer science engineers in general, and *ResearcherAC* who are academic researchers (that includes researchers working in scientific laboratories, university professors, and Ph.D. students). After identifying the profiles of contributors, we calculate the percentage of contributors by profile for each framework as well as the percentage of commits submitted by them.

Findings. Professional researchers and engineers contribute equally in company-driven frameworks. In 5 out of 10 projects, the contribution of researchers is superior to that of engineers (see Figure 18 and Figure 19). Except for few cases (i.e., deeplearning4j, deeplearnjs, caffe2, and sonnet), professional researchers and engineers are generally the two main contributors in company-driven frameworks. The contribution of academic researchers in these projects is generally low.

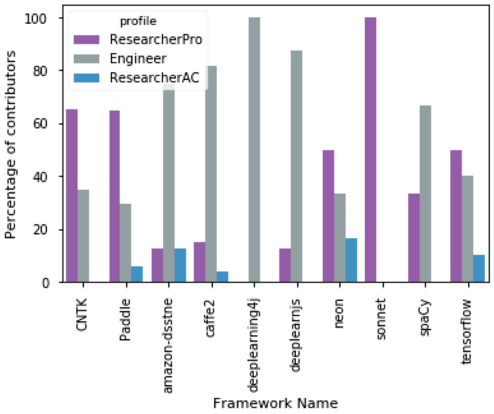

Figure 18: Percentage of contributors by profile for each company-driven framework

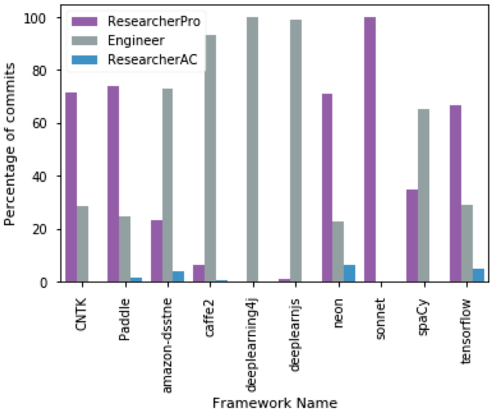

Figure 19: Percentage of commits by profile of the author for each company-driven framework

We now discuss the few cases in which professional developers play a negligible role. The first case is deeplearning4j, which is a

ML framework that offers a Java implementation of deep learning optimized for scientific computing. It is an industrial oriented framework which may explain why its development team is mostly composed of deep learning engineers with significant ML experience. Secondly, we have deeplearnjs which is an open-source library that brings performant machine learning building blocks to the web, allowing users to train neural networks in a browser. The core team of this ML framework is compose of web development engineers. Thirdly, we have caffe2 which is an open source framework published by Facebook. It follows in the steps of the original caffe project started at the University of California, Berkeley [7] Caffe2 offers allows developers to build and deploy high-performance products efficiently. The fact that Caffe2 builds on the caffe project may explain why its development teams is mostly composed of engineers. Most of the learning algorithm implemented in this framework are reused from the original Caffe and the main task of engineer is to scale them up.

Finally, Sonnet is a TensorFlow-based neural network library that is fully developed by Researchers from DeepMind in order to build higher-level frameworks for TensorFlow that fits their research needs (i.e., It has some features specifically designed around their research requirements). It is therefore not a surprise that researchers constitute the essential of its development team. By making Sonnet public, the DeepMind team hope that other models created within DeepMind will be shared easily with the community. They also hope that the community will use Sonnet to take their own research forwards.

As discussed in 3, the emergence and use of python in the implementation of scientific code allows these researchers to contribute directly to the writing of the production code. Also, we believe that the knowledge acquired by scientists through their experience in hybrid research teams and their collaboration with software engineers allow them to be more and more autonomous and independent in the production of matured software, as shown by the example of *sonnet*. On the other side, many engineers who work with data scientists become very curious about the discipline and look to develop their ML skills. This is the case for deeplearning4j and deeplearnjs framework teams . Hence, we believe that fewer boundaries between the teams can help develop data scientists and engineers who can work on a full ML project i.e., both building models and producing code.

Academic and professional researchers are the main contributors in community-driven frameworks. In 50% of our studied community-driven ML projects the main contributors are academic researchers. In the other 50% the main contributors are professional researchers. In all of these projects, the contributions of engineers is low (see Figure 20 and 21). It is only in incubator-mxnet, pytorch, and scikit-learn that we have a good mix of academic, professional researchers and engineers. These projects are among the community-driven frameworks sponsored by companies. Which may explain the strong presence of engineers in the project.

A close look at the profile of professional researcher who have a significant presence in either company-driven or community-driven frameworks, reveals that they typically hold a Ph.D. in Data Science and have work in either a R&D lab or an innovation product team. These scientists exhibit strong knowledge of software development processes.

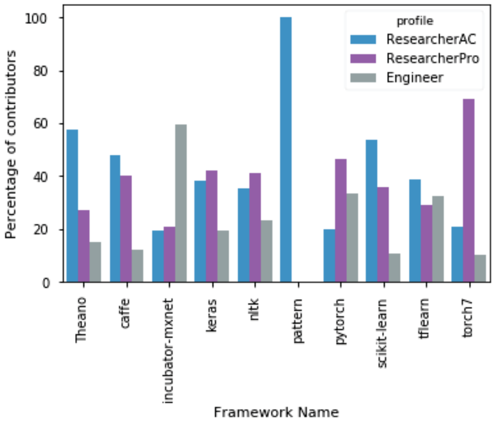

Figure 20: Percentage of contributors by profile for each community-driven framework

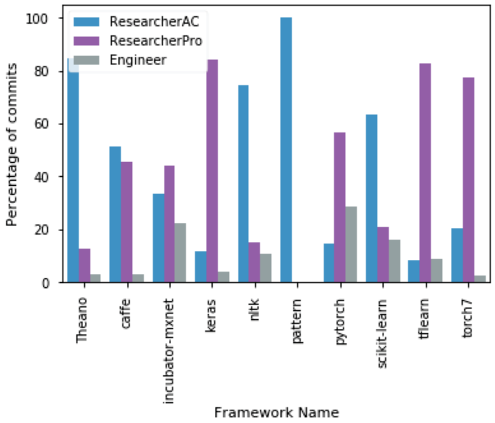

Figure 21: Percentage of commits by profile of the author for each community-driven framework

Most ML development teams are hybrid. However, the contribution of the members of these teams are unequal in general. In company-driven ML projects, professional researchers often contribute equally with engineers, while academic researchers seldom contribute to the production of code. In community-driven ML projects, the roles are reversed; with professional and academic researchers writing almost the totality of the code.

4 THREATS TO VALIDITY

We now discuss the threats to validity of our study following common guidelines for empirical studies [20].

Construct validity threats concern the relation between theory and observation. In this work, this threat is mainly due to the fact that the diffusion model of a new technology does not allow to estimate the time required to pass from one phase to another. Indeed, various criteria have an impact on the acceleration or delay of the transition from one category of adopters to another. We are able to show through our GitHub data analysis that we have reached the rising slop of the adoption curve, which indicates that we are between the stages of early adoption and early majority. Although we cannot estimate the period of time or the number of adopters required to reach the early majority stage, our analysis provides useful insights about the speed of adoption of ML frameworks in the GitHub community. We determined the affiliation of project contributors using LinkedIn and Google Scholar. However, it is possible that some of the obtained information were obsolete because some contributors didn't updated their profiles.

Threats to internal validity concern our selection of subject systems, tools, and analysis method. We relied on contributors job affiliations to judge their motives to participate to the projects. However, some academic researchers may have contributed as volunteers in some community-driven frameworks before being sponsored or hired by a company wiling to use and support this community-driven open-source solution. In such case, using our method, this researcher will be classified as company-sponsored developer i.e., we only consider the current information and do not take into account the whole job history of contributors. Nevertheless, in the context of our study, the past job history of contributors is not important, because we are interested in the current state of the open-source ML applications ecosystem. A researcher who changed from an academic career to a professional or from volunteer to paid, does not change the fact that the most active researchers currently contributing to ML frameworks are professional researchers who work in collaboration with engineers and that companies lead the development of ML frameworks, by supporting or supplying skilled researchers or professionals. We defined a threshold to identify community frameworks sponsored by companies. Our choice was to consider the frameworks for which more than 50% of commits were contributed by developers of a company as being sponsored by a company. Increasing this threshold only affects the classification of the *incubator-mxnet* framework because it has almost an equal distribution between commits from sponsoring companies and the community. The other frameworks that are considered as company-sponsored frameworks, will remain so even when increasing the threshold to 80%.

5 RELATED WORK

To the best of our knowledge, this study is the first attempt to empirically study the machine learning software ecosystem using open-source software repositories mining.

In this section, we discuss some of the studies that relate to the scope of our work, i.e., dealing with the phenomenon of open source software (OSS), considering Open source as open innovation, commercial involvement in OSS projects, and paid developers in OSS.

The open source software phenomenon Krogh and Spaeth [12] examined the characteristics of open source software that promote research. They argue that the strong proximity between open source software and science makes it very attractive for researchers coming from different fields and disciplines. They claim that open source development can open up an interesting dialog between researchers from different disciplines.

Open source as open innovation West and Gallagher [19]consider that open source software is a great exemplar of open innovation, because of the freedom to use the resulting technology as well as the collaborative development of the technology. However, they also reflect on how the fierce competition in the domain of IT threatens this innovative collaborative model. They conclude on suggestions that open source are great to foster innovation within companies.

Commercial Involvement/Paid Developers in OSS Projects There is an increase in the participation of companies in OSS. The number of employees paid to work on open source projects is also on the rise [14]. By studying OpenStack, Docker, and Android, Zhou et al. [21] analyzed how commercial involvement in OSS communities influences the onboarding of new developers. Through a survey with Linux developers, Homscheid and Schaarschmidt [10] examined the role of external developers who are paid by third-party companies ("company-sponsored developers"). Riehle et al. [14] analyzed more than 5,000 active open-source projects, from 2000 to 2007, and found that around 50% of all contributions have been paid work. Their perspective is that any contribution made from Monday to Friday, between 9 AM and 5 PM are paid contributions.

Bird et al. [6] examined the contributions of IBM and Mozilla Corporation in two large open source projects (i.e., Eclipse and Firefox) and observed that IBM is responsible for more than 90% of the commits to Eclipse, while the Mozilla Corporation has contributed more than 50% of the source code of Firefox.

6 CONCLUSION

In this paper, we investigate the role of open source development in the democratization of ML, through an examination of the growth of open source ML ecosystem on GitHub. Our results show that ML is currently in between the stages of early adoption and early majority, and that Python is the dominating language in ML projects. The contribution of companies to the development of ML frameworks have now surpassed the contribution of the academic community. Companies are also actively supporting community-driven ML projects through sponsoring and supply of skilled professionals. Also, most ML development teams contain a mix of academic researchers, professional researchers, and engineers. Professional researchers and engineers are the main contributors of company-driven ML projects, while professional and academic researchers are often the only active contributors in community-driven frameworks. Despite the large influence of corporates on current open source ML developments, there seem to be a real openness in the ecosystem. All the projects examined in our study enjoyed a large number of volunteer contributors, even though their concrete contributions (in terms of commits) were small in general. However, this openness can hide a more insidious issue, which is vendor lock-in. The open-source frameworks released by tech giants are often deployed and optimized for their commercial cloud offerings. Therefore, the community should be mindful of the risk that these open source frameworks serve as baits to trap data scientists in commercial platforms.

ACKNOWLEDGEMENT

This work is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

[1] [n. d.]. About LinkedIn. https://news.linkedin.com/about-us#statistics. ([n. d.]). Accessed: 2018-01-23.

[2] 2011. Innovation adoption lifecycle. (2011). https://en.wikipedia.org/wiki/File:DiffusionOfInnovation.png

[3] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J. Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A. Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B. Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *CoRR* abs/1603.04467 (2016).

[4] Ethem Alpaydin. 2010. *Introduction to Machine Learning* (1st ed.). Adaptive Computation and Machine Learning.

[5] Nicolas Bettenburg, Ahmed E. Hassan, Bram Adams, and Daniel M. German. 2015. Management of community contributions. *Empirical Software Engineering* 20, 1 (01 Feb 2015), 252–289. https://doi.org/10.1007/s10664-013-9284-6

[6] Christian Bird and Nachiappan Nagappan. 2012. Who? where? what? examining distributed development in two large open source projects. In *Mining Software Repositories (MSR), 2012 9th IEEE Working Conference on*. IEEE, 237–246.

[7] Joe M. Bohlen and George M. Beal. 1957. The Diffusion Process. *Agriculture Extension Service, Iowa State College* 18, 1 (1957), 56–77.

[8] International Data Corporation. 2016. Double-Digit Growth Forecast for the Worldwide Big Data and Business Analytics Market Through 2020. https://www.idc.com/getdoc.jsp?containerId=prUS41826116. (October 2016).

[9] Shmueli Galit and R. Koppius Otto. 2011. Predictive Analytics in Information Systems Research. *MIS Quarterly* 35, 3 (2011), 553–572.

[10] Dirk Homscheid and Mario Schaarschmidt. 2016. Between organization and community: investigating turnover intention factors of firm-sponsored open source software developers. *WebSci '16 Proceedings of the 8th ACM Conference on Web Science* 16 (2016), 336–337.

[11] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. 2017. In-Datacenter Performance Analysis of a Tensor Processing Unit. *SIGARCH Comput. Archit. News* 45, 2 (June 2017), 1–12.

[12] Georgvon Krogh and Sebastian Spaeth. 2007. The open source software phenomenon: Characteristics that promote research. *The Journal of Strategic Information Systems* 16, 3 (2007), 236–253.

[13] Enrique Orduna-Malea, Juan M. Ayllón, Alberto Martín-Martín, and Emilio Delgado López-Cózar. 2015. Methods for Estimating the Size of Google Scholar. *Scientometrics* 104, 3 (Sept. 2015), 931–949.

[14] D. Riehle, P. Riemer, C. Kolassa, and M. Schmidt. 2014. Paid vs. Volunteer Work in Open Source. *HICSS* 14 (2014), 3286–3295.

[15] Ruslan Salakhutdinov. 2015. Learning Deep Generative Models. *Annual Review of Statistics and Its Application* 2, 1 (2015), 361–385.

[16] A. L. Samuel. 1959. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3, 3 (1959), 210–229.

[17] Soren Sonnenburg, Mikio Braun, Cheng Soon Ong, Samy Bengio, Leon Bottou, Geoffrey Holmes, Yann Lecun, Klaus R Muller, Fernando Pereira, Carl E Rasmussen, Gunnar Ratsch, Bernhard Scholkopf, Alexander Smola, Pascal Vincent, Jason Weston, and Robert Williamson. 2007. The Need for Open Source Software in Machine Learning. 8 (01 2007), 2443–2466.

[18] Alfred Spector, Peter Norvig, and Slav Petrov. 2012. Google's Hybrid Approach to Research. *Commun. ACM* 55, 7 (July 2012), 34–37.

[19] Joel West and Scott Gallagher. 2006. Challenges of open innovation: the paradox of firm investment in open-source software. *R&D Management* 36, 3 (2006), 319–331.

[20] R. K. Yin. 2002. *Case Study Research: Design and Methods - Third Edition*. SAGE Publications.

[21] M. Zhou, A. Mockus, X. Ma, L. Zhang, and H. Mei. 2016. Inflow and retention in oss communities with commercial involvement: A case study of three hybrid projects. *ACM Transactions on Software Engineering and Methodology* 25, 3 (2016).