

# Open sourcing AI: intellectual property at the service of platform leadership

by Carlos Muñoz Ferrandis and Marta Duque Lizarralde\*

**Abstract:** Artificial Intelligence (AI) is one of the most strategic technologies of our century. Consequently, tech companies are adopting intellectual property strategies to protect their investment in the field, which encompasses copyright, patents, and trade secrets. While the number of AI-related patent applications is increasing, the number of open-source AI projects sponsored by major AI patent holders is also on the rise. This article explores the commercial and policy strategic reasons behind the growing adoption of open-source licensing in the AI space. More precisely, it assesses how IP rights are

articulated around “openness” as a competitive factor in ecosystem competition, and how some players are using open-source licensing successfully to attract a critical mass of users and build an ecosystem around their AI platforms. Moreover, this article integrates the debate on the protectability of AI features by IP rights to assess the potential implications for open-source. Finally, it analyses the most used open-source licenses in AI projects and highlights existing and future challenges from an IP and contractual law perspective.

Keywords: AI; platforms; open-source; licenses; copyright; patents

© 2022 Carlos Muñoz Ferrandis and Marta Duque Lizarralde

Everybody may disseminate this article by electronic means and make it available for download under the terms and conditions of the Digital Peer Publishing Licence (DPPL). A copy of the license text may be obtained at <http://nbn-resolving.de/urn:nbn:de:0009-dppl-v3-en8>.

Recommended citation: Carlos Muñoz Ferrandis and Marta Duque Lizarralde, Open sourcing AI: intellectual property at the service of platform leadership, 13 (2022) JIPITEC 224 para 1.

## A. Introduction

1 Artificial Intelligence (AI) is transforming the world while “becoming one of the most strategic technologies of the 21st century”.<sup>1</sup> Nevertheless, AI tech-

nology is nothing new. The concept of AI was first introduced as an academic discipline in 1956, subsequently suffering ups and downs until the current boom, caused by the growth in computing power, connectivity, and the greater availability of data.<sup>2</sup>

2 Although there is no universal definition of AI, it can be regarded as “a discipline of computer science that is aimed at developing machines and systems

\* Carlos Muñoz Ferrandis was at the time of the paper’s acceptance PhD Researcher at the Max Planck Institute for Innovation and Competition, and member of the Global Innovation Policy & Law Research Group (Alicante Univ.). Marta Duque Lizarralde is Doctoral Candidate and Research Associate at the Technische Universität München. Both authors have equally contributed to this paper. The views expressed herein are those of the authors alone and do not necessarily represent the views of their respective organizations.

the European Economic and Social Committee and the Committee of the Regions, “Artificial Intelligence for Europe” (2018) 1 <<https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52018DC0237&from=EN>>.

1 All links last accessed on the 25<sup>th</sup> January 2022. European Commission, Communication from the Commission to the European Parliament, the European Council, the Council,

2 Josef Drexl, Reto M. Hilty et al., ‘Technical Aspects of Artificial Intelligence: An Understanding from an Intellectual Property Law Perspective, Version 1.0’ (2019) <<https://ssrn.com/abstract=3465577>>; WIPO, ‘WIPO Technology Trends 2019’ (2019) 58, 79 <[https://www.wipo.int/edocs/pubdocs/en/wipo\\_pub\\_1055.pdf](https://www.wipo.int/edocs/pubdocs/en/wipo_pub_1055.pdf)>.

that can carry out tasks considered to require human intelligence”.<sup>3</sup> There are many ways to achieve AI, machine learning (ML) being one of them. ML is a subfield of AI that is “limited to predicting a future that looks mostly like the past”.<sup>4</sup> It involves pattern recognising systems that “learn” by adjusting to previous data, in order to make predictions about new data.<sup>5</sup> Three main types of ML exist: supervised<sup>6</sup>, unsupervised<sup>7</sup> and reinforcement.<sup>8</sup> Some well-known applications of AI are machine vision, object and speech recognition, and detection and language translation.<sup>9</sup>

- 3 Against this background, many companies have understood the need to protect their investments in the creation of AI systems by means of Intellectual Property Rights (IPRs). This may explain the drastic increase in AI-related patent applications in recent years. Statistics compiled by the World Intellectual Property Organisation (WIPO) show that although approximately 340,000 patent applications for AI-related inventions have been filed since the emergence of AI, more than half of these applications are from 2013 onwards.<sup>10</sup>
- 4 On the other side of the spectrum, there is a continuous increment in the number of open-source soft-

ware (OSS) projects related to AI.<sup>11</sup> According to the OECD, since 2014 the number of OSS repositories related to AI has grown about three times more than the rest of OSS.<sup>12</sup> This is partly due to the roots of AI in academia, which has been at the origins of collaborative software development projects and tended to be reluctant to participate in projects with access restrictions due to IP.<sup>13</sup> Nowadays, however, some of the most relevant OSS AI projects are governed by large tech companies<sup>14</sup>, such as Google and Facebook (now Meta) with their respective ML frameworks: TensorFlow<sup>15</sup> and PyTorch<sup>16</sup>. Despite owning the largest patent portfolios in the AI sector, these companies also share their source code and provide open-source licenses for their AI-related patents.<sup>17</sup>

3 WIPO (n 2).

4 Matt Taddy, ‘The Technological Elements of Artificial Intelligence’ (2019) NBER Working Paper 24301 <[https://www.nber.org/system/files/working\\_papers/w24301/w24301.pdf](https://www.nber.org/system/files/working_papers/w24301/w24301.pdf)>.

5 Mohri Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar, *Foundations of Machine Learning* (MIT Press, 2018) 1,2.

6 Anthony Man-Cho So, ‘Technical Elements of Machine Learning for Intellectual Property Law’, in J.-A. Lee, K.-C. Liu, R. M. Hilty (eds.), *Artificial Intelligence & Intellectual Property* (Oxford University Press, 2020): In supervised learning the system is trained with labelled data and must be able to apply this knowledge to recognize the labels in a new dataset.

7 Mohri et al (n 5): In unsupervised learning the training data samples do not have any labels and the goal is to cover hidden structure underlying the data.

8 Anthony Man-Cho So (n 6): In reinforcement learning the system must achieve a certain goal and receives penalties or rewards for its performance, the goal being to maximise the total reward.

9 WIPO (n 2).

10 WIPO (n 2).

11 Open-source is a software collaborative innovation and development model based on the freedoms to access, run, study, re-distribute the used software and distribute derived one, while respecting the terms of the open-source license. For the purpose of this paper the definition proposed by the Open-source Initiative (OSI) is used, according to which each license must comply with the 10 OSI criteria. See <<https://opensource.org/osd>>.

12 Stefano Baruffaldi et.al. ‘Identifying and measuring developments in artificial intelligence: Making the impossible possible’ (Organisation for Economic Co-operation and Development, 2020) 32.

13 Ibrahim Haddad, *Open-source AI Projects, Insights, and Trends* (The Linux Foundation, 2018) 104; Danish Contractor et al., ‘Behavioral Use Licensing for Responsible AI’ (arXiv - Computer and Society, 2020) 1; assessing opposing views, see Knut Blind et.al. *The impact of Open-source Software and Hardware on technological independence, competitiveness and innovation in the EU economy* (European Commission, 2021) 306,307.

14 Tom Simonite, ‘Despite Pledging Openness, Companies Rush to Patent AI Tech’ (31 July 2018, WIRED) <<https://www.wired.com/story/despite-pledging-openness-companies-rush-to-patent-ai-tech/>>; WIPO (n 2). There are, however, some OSS AI projects which maintainers are research organisations (e.g., UC Berkeley) or OSS institutions (e.g., the Apache Software Foundation).

15 TensorFlow <<https://github.com/tensorflow/tensorflow>>.

16 Pytorch <<https://github.com/pytorch/>>.

17 Nathan Calvin, Jade Leung, ‘Who owns artificial intelligence? A preliminary analysis of corporate intellectual property strategies and why they matter’, (2020) 7,8 <<https://www.fhi.ox.ac.uk/wp-content/uploads/GovAI-working-paper-Who-owns-AI-Apr2020.pdf>>; Patrick Shafto, ‘Why big tech companies are open-sourcing their AI systems’ (2016, The Conversation) <<https://theconversation.com/why-big-tech-companies-are-open-sourcingtheir-ai->

- 5 Patenting and open-source commercial strategies are not alien to each other in the ICT realm. Both are considered core innovation and competition factors in isolation. Having an efficient IP proprietary strategy allows companies a direct return on investment, to avoid free-riding, and to establish a competitive advantage.<sup>18</sup> Nevertheless, literature has recently highlighted the articulation of open-source as a strategic competitive move in contexts that depend on strong network effects, such as standardisation.<sup>19</sup> Interestingly, and in line with the aforementioned, the AI sector shows how the combination of patents and open licensing schemes towards hybrid IP strategies might have a strategic impact on the market.
- 6 This article aims to give insight into the objectives of tech companies when adopting open-source and proprietary strategies. It seeks to illustrate how OSS is contributing to the rapid development of AI technologies, but also to highlight the risks that stakeholders may face if they do not comprehend the licensing terms before contributing to AI open-source projects.
- 7 The structure of this article is as follows: Section B outlines how open-source licenses are used as strategic competitive elements in the quest to build ecosystems in the AI field. Then, Section C explores the IP rights involved in the protection of AI systems, before examining the most commonly used open-source licenses in AI projects according to the data collected from the scrutinised 60 open-source AI projects. The authors have taken an inductive approach, with the research criteria when selecting an open-source AI project for analysis being: (i) the open/public-access platform hosting the software (e.g., repositories such as GitHub); (ii) the platform's sponsors; (iii) the release under an OSS license; and (iv) the ecosystem around the OSS. The analysis of these data allows a better understanding of the rationale behind the use of a specific open-source license for an AI function, and to draw practical conclusions from it. In particular, the pervasiveness of permissive licenses over restrictive ones highlights

---

systems-54437>.

- 18 See Alfonso Gambardella, 'The functions of patents in our societies: innovation, markets, and new firms' (2021) <[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3789554](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3789554)>.
- 19 Jorge L. Contreras, 'Patent Pledges' (2015) 47(3) *Arizona State Law Journal* 546; Eli Greenbaum, 'Puzzles of the Zero-Rate Royalty' (2016) 27(1) *Fordham Intellectual Property, Media and Entertainment Law Journal* 13; Liza Vertinsky, 'The Hidden Costs of Free Patents' (2017) 78(6) *Ohio State Law Journal*.

the expected business strategies behind the choice of licenses such as Apache 2.0 or MIT; this will be explained in Section D.<sup>20</sup>

## B. Open-source dynamics and their strategic impact in the AI space

- 8 Taking a strategic approach to OSS, IP assets might be conceived as attraction and control mechanisms. OSS licenses, especially permissive ones, are legal tools for software mass market adoption (B.I.), and play a core role in the development and market leadership of software platforms (B.II). Firms compete to capture the network effects derived from the adoption of OSS tools and/or platforms by trying to be the first in releasing specific OSS (B.III.). In addition, some companies use 'open' patent strategies complementary to OSS in order to leverage their IPRs as attractive instruments (B.IV.).

### I. A non-traditional use of exclusivity rights

- 9 In general, open-source uses IP as a tool aimed at maximising the diffusion of innovation through licenses designed around the concept of distribution.<sup>21</sup> Hence, it represents a shift from a direct reward via licensing of IP to a focus on distribution and attraction as means to compete in markets. Companies commercially leveraging the potential of OSS might extract their return on investment at different points of the value chain (vertical approach)<sup>22</sup> and/or from adjacent connected markets (horizontal

---

20 For the sake of clarity, informational purposes, and transparency the list of all the assessed OSS AI projects is attached in Annex A. It is thus expected to inform the reader on the AI specific licensed feature, the chosen OSS license, and the stakeholder behind the project.

21 Steven Weber, 'The Success of Open-source Groups' (2005) *Harvard University Press* 1,86; Van Lindberg, 'OSS and FRAND: Complementary Models for Innovation and Development' (2019) 20 *The Columbia Science and Technology Law Review* 254.

22 For instance, OSS business models might be based on dual licensing or open core, where aside from the OSS a commercial version is offered, either with a license enabling more flexibility to the user than the OSS one (dual licensing, e.g., MySQL); or technically optimized to better perform on an enterprise environment by adding extra closed software features (open core. e.g., MongoDB). Moreover, a classic example is one of RedHat's business models monetizing open-source by means of support, educational, and security services related to the OSS feature.

approach).<sup>23</sup> For instance, by open sourcing TensorFlow (an ML framework), Google enables developers to access ML capabilities and consequently generates demand for cloud computing and data centre provision.<sup>24</sup>

- 10 Companies relying on traditional IP strategies generally enforce their right to exclude others to protect their inventions from imitators or free riders, or/and to secure a direct return on investment from the monetisation of the IPR. Contrarily, open-source licenses implement both dissuasive and passive exclusion. With dissuasive exclusion, those licensees not complying with the terms of the license will lose the benefit of using the software.<sup>25</sup> Passive exclusion neutralises licensees' enforcement rights by compelling them not to enforce certain IPRs infringed within the OSS project. This can be done by means of reciprocity, non-assertion, and retaliation clauses.
- 11 Open-source licenses are *de facto mass-market licenses*<sup>26</sup>, which means that the licensees are presented with a given set of standard and non-negotiable terms.<sup>27</sup> This is known as frictionless distribution<sup>28</sup>, as the users only have the option of joining the contract, contrary to other existing licensing practices where the terms of the agreement are negotiated by the parties.<sup>29</sup> Moreover, actions such as using, reproducing or distributing the software are sufficiently indicative of the acceptance of the terms of the licenses.<sup>30</sup>

23 A 'modern' or not so explored angle of OSS business models are the ones targeting platform and market control by means of (not so) 'open' source strategies, such as Google's Android, analysed below.

24 Blind et al. (n 13) 89.

25 David McGowan, 'Legal Implications of Open-source Software' (2001) 241 *Illinois Law* 34.

26 Steven Weber (n 21) 212.

27 Van Lindberg (n 21) 254.

28 Greg R. Vetter, 'Open-source Licensing and Scattering Opportunism in Software Standards' (2007) 48(1) *Boston College Law Review* 247,248.

29 According to Weber, the open-source licenses are contrary to the adversarial legal dynamic in which each one tries to obtain the most advantageous terms for its side. Steven Weber (n 21) 179.

30 Some open-source licenses are more explicit than others regarding which actions trigger "acceptance", see Eclipse Public License v2 <<https://opensource.org/licenses/EPL-2.0>>; GPL v3 Section 9 <<https://opensource.org/licenses/>

- 12 Due to the aforementioned characteristics, open-source licenses might reduce transaction costs, since both the licensor and the licensee are not forced to engage in a lengthy negotiation process. Besides, these licenses might promote faster adoption and a wider scope of innovation due to network effects, conversely to what happens in a static situation where the allocation of IPRs depends on individual negotiations, e.g., Linux. However, potential costs derived from OSS quality, licensing compliance and enforcement should not be overlooked.

## II. Sided markets and ecosystem creation

- 13 From a market competition perspective, open-source can be a double-edged innovation tool. On the one hand, it may facilitate a broader access to technology, making its use easier and promoting participation. On the other hand, firms involved in the innovation process usually compete in terms of achieving network effects and market tipping<sup>31</sup>, since this can have a positive indirect effect on adjacent component markets from which they seek to extract revenues.<sup>32</sup> In words of Blind et al. "Open-source has a multi-faceted role for competition."<sup>33</sup>

---

GPL-3.0>; Apache 2.0 Definition of the term "License" <<https://opensource.org/licenses/Apache-2.0>>; from a literature standpoint, see Andrew M. S. St. Laurent, *Understanding Open-source and Free Software Licensing* (O'Reilly, 2004) Chap. 6; Lawrence Rosen, *Open-source Licensing Software Freedom and Intellectual Property Law* (Prentice Hall, 2004) 54,55; Andrés Guadamuz, 'The License/Contract Dichotomy in Open Licenses: A Comparative Analysis' (2009) 30(2) *University of La Verne Law Review* 8; Van Lindberg, 'OSS and FRAND: Complementary Models for Innovation and Development' (2019) 20 *The Columbia Science and Technology Law Review* (n 21) 255,256.

- 31 Weber holds that free software counters opportunistic behaviours by reducing barriers to entry and avoiding potential lock-in. Steven Weber (n 21) 221. However, lock-in may also appear in open-source settings, despite competitors benefiting from low barriers to entry and the freedom to fork.

- 32 Michal S. Gal, Daniel L. Rubinfeld, 'The Hidden Costs of Free Goods: Implications for Antitrust Enforcement' (2016) 80(3) *Antitrust Law Journal* 523,535; Stephen M. Maurer, Suzanne Scotchmer, 'Open-source Software: The New Intellectual Property Paradigm' (2006) *NBER Working Paper* 12148.

- 33 Blind et al. (n 13) 337: "OSS is not an obstacle, but rather a facilitator for companies to enter competitive markets also based on AI. However, the large platform providers challenging competition policies and authorities also make use of OSS contributions for the development of software they use for developing their platform

- 14 A firm might seek to invest in an OSS project in order to benefit from it in other markets where network externalities are decisive.<sup>34</sup> For instance, one of the incentives for stakeholders to compete in the Market A with an OSS product may be to exclude competitors relying on proprietary business models. The latter strategy will allow them to gain an advantage in Markets B and C where they also compete by offering proprietary components vis-à-vis the same participants from Market A.<sup>35</sup>
- 15 Namely, in order to compete in the mobile operating system market, Google chose to first develop and control the formation of a de facto standard, Android, by means of an industry consortium, the Open Handset Alliance, and with an “open” approach towards the technology.<sup>36</sup> Google then developed an ecosystem around Android in which it leaves some parts open for development from tier developers, and closes other parts that are developed and periodically released with new versions by Google. With Android, Google embraces openness as a means to an end, but not as an end in itself.<sup>37</sup> The end goal is to create an ecosystem around the platform, using the latter as an element of attraction for developers

---

*architectures and ecosystems. Consequently, open-source has a multi-faceted role for competition. Therefore, it is recommended to explicitly consider open-source in the further discussion and development of competition policies in general and platform policies in particular.”*

- 34 Elad Harison, *Intellectual Property Rights, Innovation and Software Technologies: The Economics of Monopoly Rights and Knowledge Disclosure* (Edward Elgar Publishing, 2008) 106; Josh Lerner, Jean Tirole, ‘The Scope of Open-source Licensing’ (2002) NBER Working Paper <<https://www.nber.org/papers/w9363>>.
- 35 However, this is just an over-simplified scenario focused on price as an essential competition parameter. The market can be more or less price-sensitive, and thus other parameters such as quality might play a relevant role. See Ramon Casadeu-Masanell, Pankaj Ghemawat, ‘Dynamic Mixed Duopoly: A Model Motivated by Linux vs. Windows’ (2006) 52(7) *Management Science* 1072.
- 36 Ron Amadeo, ‘Google’s iron grip on Android: Controlling open-source by any means necessary’ (2018, arsTECHNICA) <<https://arstechnica.com/gadgets/2018/07/googles-iron-grip-on-android-controlling-open-source-by-any-means-necessary/>>; Michele Herman, ‘Sensible Open-source Licenses For Standards Development Organizations’ (2020) <[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3717031](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3717031)>.
- 37 Alan Cunningham, ‘Open-source, Standardization, and Innovation’ in Noam Shemtov, Ian Walden (eds.) *Free and Open-source Software: Policy, Law, and Practice* (Oxford University Press, 2013) 366.

as well as hardware manufacturers. A similar strategy is being pursued today by open-source ML platforms.<sup>38</sup>

### III. The race-for-release

- 16 The “release early and release often” dynamic stemming from the Linux development project has become a ‘maxim’ in the highly competitive and fast-growing ICT field. As a result of it, some companies compete fiercely by means of OSS products<sup>39</sup>, aiming to attract a critical mass of users, composed of customers and developers, to consolidate an ecosystem around the released OSS tool.<sup>40</sup> By launching a product promptly the company seeks to benefit from the “first-mover advantages”<sup>41</sup>, especially if it has considerable financial power to invest in terms of marketing policy and strategy.<sup>42</sup> Conversely, the introduction of a new OSS tool may be a response to a competitor’s first move, or to its strong influence in a given market.<sup>43</sup> Moreover, a company can decide to release OSS to avoid potential competitors’ attempts to patent a technology which is fundamental for the market.
- 17 Examples of the “race for release” can be found within markets related to autonomous vehicles. Clear illustrations are the open-source releases of
- 
- 38 Ibrahim Haddad (n 13) 36.
- 39 Sandeep Krishnamurthy, ‘An Analysis of Open-source Business Models’, in Joseph Feller, Brian Fitzgerald, Scott Hissam and Karim Lakhani (eds.) *Making Sense of the Bazaar: Perspectives on Open-source and Free Software* (Workshop 2001) 17,18.
- 40 Stephen M. Maurer, Suzanne Scotchmer, (n 32); Josh Lerner, Jean Tirole, (n 34): “IBM released half-a-million lines of its Cloudscape program, a simple database that resides inside a software application instead of as a full-fledged database program, to the Apache Software Foundation. Hewlett-Packard released its Spectrum Object Model-Linker to the open-source community to help the Linux community write software to connect Linux with Hewlett Packard’s RISC computer architecture. This strategy is to give away the razor (the released code) to sell more razor blades (the related consulting services that IBM and Hewlett Packard hope to provide)”.
- 41 In markets relying on network effects, companies seeking to be the first to launch a product/service want to capture and consolidate them to be able to lock-in demand and render more difficult market entry for potential competitors.
- 42 Steven Weber (n 21).
- 43 Stephen M. Maurer, Suzanne Scotchmer (n 32).

Uber<sup>44</sup> and Lyft.<sup>45</sup> While one might think that some of these companies are active only in certain specific ridesharing markets, the reality is that the released OSS tools may also be useful for them in other markets<sup>46</sup>, such as ML tools applications and related markets.<sup>47</sup>

#### IV. Hybrid strategies

**18** The predominant strategy of the leading AI companies is to simultaneously accumulate patents and heavily invest in the OSS community.<sup>48</sup> The debate on the need for AI-related patents can be assimilated with the debate on software patents. On the one hand, some national and regional strategies seek to reinforce the protection of IPRs and to ensure the patentability of AI-related inventions in order to foster research and investment.<sup>49</sup> They argue that AI-related patents encourage innovation and diffusion of AI technology via the disclosure of the technology in exchange of its protection.<sup>50</sup> On the other hand,

others claim that patents on fundamental AI techniques with broad applications discourage innovation because the privatisation of the basic elements of AI can be used to exclude third parties from competition.<sup>51</sup> They fear that the increase of AI-related patents could lead to an unsustainable level of litigation, which is claimed to be extremely costly, might discourage innovation and hamper the growth of the AI sector.<sup>52</sup>

**19** While AI-related patents are barely litigated so far<sup>53</sup>, the IP strategy of the patent holders cannot be described as purely defensive. AI-related patents are being used to gain influence in other spheres, as seen in the patent sharing agreement concluded between Google and Tencent, which “is paving the way for Google’s entry into the Chinese market”.<sup>54</sup> Furthermore, as most of the AI-related patents granted are very recent, not enough time has passed as to assess the level of litigation in this area, which will only become visible when more AI applications and products are commercialised. Once this stage is reached, some believe that the number of AI patent lawsuits may increase.<sup>55</sup> Another view considers that patent holders may be hesitant to enter into disputes since the qualification of AI core inventions as patentable subject matter is still uncertain and this could lead to the invalidation of some of their patents in court.<sup>56</sup> Furthermore, AI related patents may be difficult to enforce due to the technical complexity of the inventions in question.<sup>57</sup>

**20** Nevertheless, it is far from accurate to assert that the existence of AI-related patents will have a negative impact on the market and lead to further restrictions on AI’s openness. Some companies engage in heavy R&D investments because of their trust on the IPR

44 Kyle Wiggers, ‘Uber open-sources Manifold, a visual tool for debugging AI models’ (2020, Venturebeat) <<https://venturebeat.com/2020/01/07/uber-open-sources-manifold-a-visual-tool-for-debugging-ai-models/>>.

45 Kyle Wiggers, ‘Lyft releases Flyte, a platform for maintaining AI workflows’ (2020, Venturebeat) <<https://venturebeat.com/2020/01/07/lyft-releases-flyte-a-platform-for-maintaining-ai-workflows/>>.

46 Thomas R. Eisenmann, Geoffrey Parker, Marshall Van Alstyne, ‘Opening Platforms: How, When and Why?’ in Annabelle Gawer (ed.), *Platforms, Markets and Innovation* (Edward Elgar Publishing, 2011) 16,17.

47 Jesús Rodríguez, ‘Uber Has Been Quietly Assembling One of the Most Impressive Open-source Deep Learning Stacks in the Market’ (2020) Datasource.ai <<https://www.datasource.ai/en/data-science-articles/uber-has-been-quietly-assembling-one-of-the-most-impressive-open-source-deep-learning-stacks-in-the-market>> .

48 Nathan Calvin and Jade Leung (n 17) 2.

49 *Ibid*; See China, the USPTO, the EPO and the Singapore Patent Office; Rogier Creemers, Graham Webster, Paul Tsai, Paul Triolo, Elsa Kania, ‘Translation State Council Notice on the Issuance of the Next Generation Artificial Intelligence Development Plan’ (2017) <https://d1y8sb8igg2f8e.cloudfront.net/documents/translation-fulltext-8.1.17.pdf> >.

50 Nick Bostrom, ‘Strategic Implications of Openness in AI Development’ (2017) Global Policy 2; IPO, ‘Artificial Intelligence A worldwide overview of AI patents and patenting by the UK AI sector’ (2019) <[https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attach-](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attach-ment_data/file/817610/Artificial_Intelligence_-_A_worldwide_overview_of_AI_patents.pdf)

[ment\\_data/file/817610/Artificial\\_Intelligence\\_-\\_A\\_worldwide\\_overview\\_of\\_AI\\_patents.pdf](https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attach-ment_data/file/817610/Artificial_Intelligence_-_A_worldwide_overview_of_AI_patents.pdf)>.

51 Raphael Zing, ‘Foundational Patents in Artificial Intelligence’ in J.-A. Lee, K.-C. Liu, R. M. Hilty (n 6) 74,98.

52 Nathan Calvin and Jade Leung, (n 17) 4,5; Tom Simonite (n 14).

53 WIPO (n 2) 111,117: less than 1% of the 340,000 AI-related patent families have faced litigation so far.

54 Nathan Calvin and Jade Leung (n 17) 4.

55 WIPO (n 2) 141.

56 Patent Strategy, ‘Machine yearning: AI and patents’ (2019, ManagingIP) <<https://www.managingip.com/pdfsmip/Machine-yearning-AI-and-patents.pdf>>.

57 See Tabrez Ebrahim, ‘Artificial Intelligence Inventions & Patent Disclosure’ (2020) *125(1) Penn State Law Review* 149,220.

system and the possibility to obtain an adequate reward. Thus, a system lacking patents could discourage further R&D investments, leading to less innovation and negatively impacting the market in the mid- to long-term.<sup>58</sup>

- 21 Regarding the articulation of patent portfolios and OSS platform investment, it should be emphasised that when large tech companies use this hybrid strategy<sup>59</sup>, the aim in the short run might be to gain traction by means of an “open” AI platform. In the long run, however, they seek to standardise and commoditise the technology, and ultimately to control essential software layers, and by extension their markets.<sup>60</sup>
- 22 In the software sector, for example, the major patent holders, IBM and Microsoft, instead of enforcing their IPR, have adopted policies to license them on a royalty free (RF) basis to users, provided the latter grant parallel access to their own IPR.<sup>61</sup> In this way, these companies managed to create and consolidate large “IP-neutralised” areas.<sup>62</sup> Defensive patent strategies and open-source dynamics might well complement each other to achieve market tipping and innovation control in a given market or software layers. Either in proprietary-based or open-source, IPRs are used as dissuasive instruments securing a non-assertion zone in which the sponsor could both avoid costly litigation and gain access to others’ patents through a reciprocal ‘patent pledge’.<sup>63</sup> The pledge may have a narrow scope devoted to a specific market use, to enable the sponsor a considerable margin of manoeuvre exploiting their patents for different uses and markets.

58 See Alfonso Gambardella (n 18).

59 Blind et al,(n 13) 38.

60 E.g., IBM’s strategy with the x86 OS. See John C. Koenig, ‘Seven Open-source Business Strategies for Competitive Advantage’ (2006) *IT Manager’s Journal* 5.

61 Anne Layne-Farrar, David S. Evans, ‘Software Patents and Open-source: The Battle Over Intellectual Property Rights’ (2004) <[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=533442](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=533442)>

62 *Ibid*; Ronald J. Mann, ‘Do Patents Facilitate Financing in the Software Industry?’ (2006) 83(4) *Texas Law Review* 1005,1007.

63 On common characteristics of patent pledges and their functioning see Jorge L. Contreras, ‘Patent Pledges’ (2015) 47(3) *Arizona State Law Journal* 546; Eli Greenbaum, ‘Puzzles of the Zero-Rate Royalty’ (2016) 27(1) *Fordham Intellectual Property, Media and Entertainment Law Journal* 13; Liza Vertinsky, ‘The Hidden Costs of Free Patents’ (2017) 78(6) *Ohio State Law Journal*.

- 23 Previous experiences have shown that the use of OSS in some emerging technologies brings positive effects.<sup>64</sup> For small players having access at zero cost to the code and patented technology of the largest players can be a great opportunity and at the same time a significant risk, since RF access does not mean unconditional access.<sup>65</sup> In view of this, even if a high degree of openness in AI is desirable, and OSS can help to achieve this aim, contributors of AI OSS platforms should be aware of the licensing terms before committing to such projects.

## C. IPR protection of AI features: implications for open-source licenses

- 24 Open-source licenses are characterised as conditional copyright licenses. That is, they grant all copyrights subject to the compliance with certain conditions for their exercise.<sup>66</sup> If these licenses apply to something that is not protected by copyright or related rights, they will not be triggered.<sup>67</sup> In addition, some open-source licenses contain patent grants and defensive termination provisions, so clarification is likewise needed as to which elements of AI systems may also be protected by patents.

## I. Copyright

- 25 The software code and its preparatory design material are considered literary works protectable by copyright in the US and the EU. It follows that the copyright holder has the exclusive rights to

64 Bill Briggs, Stefan Kircher, Mike Bechtel, ‘Open for business, How open-source software is turbocharging digital transformation’ (2019, Deloitte Insights) <<https://www2.deloitte.com/us/en/insights/industry/technology/how-open-source-software-is-turbocharging-digital-transformation.html>>; Eseosa Ehioghae and Sunday Idowu, ‘Open-source Software in Emerging Technologies for Economic Growth’ (2021) 7(27) *ITEGAM-JETIA, Manaus* 63,69.

65 See Jianan Wang and Xiaobao Peng, ‘A Study of Patent Open-Source Strategies Based on Open Innovation: The Case of Tesla’ (2020) <[https://www.scirp.org/html/31-1763645\\_101900.htm](https://www.scirp.org/html/31-1763645_101900.htm)>.

66 Heather Meeker, *Open-source for Business: A Practical Guide to Open-source Software Licensing* (Last Mile Publishing, 2020) 77,88.

67 Begoña Gonzalez Otero, ‘Machine Learning Models under the copyright microscope: is EU Copyright fit for purpose?’ (2021) *GRUR International* 1043,1055.

authorise or prohibit the reproduction, translation, adaptation, arrangement, and any other alteration of the software, as well as its distribution.<sup>68</sup> It must be emphasised that copyright only protects the form in which the underlying ideas and principles of the software are expressed, i.e., its code, but its functional aspects are not covered.<sup>69</sup>

- 26 The algorithms composing AI systems are not by themselves protectable by copyright. However, these training algorithms are encoded in a programming language and embedded in software.<sup>70</sup> This software code, if meets the originality requirement, is copyrightable.<sup>71</sup> Under the same condition, the code provided in ML frameworks for training the models may also be protected.<sup>72</sup> As for the protection of ML models, Gonzalez Otero argues that even if they are expressed in coded form, and therefore can be qualified as computer programs, they may not meet the originality requirement.<sup>73</sup> In the same vein, it has also been pointed out that while simple linear ML models do not meet the requirements for protection under sui generis database right, it is debatable whether complex, dynamic ML models would be eligible for such protection.<sup>74</sup> Some have also proposed to introduce a new sui generis right for ML models.<sup>75</sup> Further research is needed on this subject and on how lack of IP for models would affect

investment in their creation.<sup>76</sup> Finally, those parts of the overall AI application that are provided in the form of code may also be protected by copyright.<sup>77</sup>

- 27 A hot topic today is what IPRs protect training datasets. Many training datasets include data that although publicly accessible and freely available, are protected by copyright or related rights.<sup>78</sup> In addition, some training datasets may be susceptible to copyright or sui generis database rights protection.<sup>79</sup> Even when raw data and datasets are not protected by IPR, companies often restrict access to them through contractual restrictions or technical protection measures, creating de facto control.<sup>80</sup>

## II. Patents

- 28 *AI-related inventions* can be divided between *AI-core* and *AI-applied* inventions. AI-core inventions are those characterised by mathematical or statistical-information-processing technology that improves the performance of the AI itself. Some examples are the algorithms composing the AI system, or improved ML methods.<sup>81</sup> AI-applied inventions are those resulted from applying AI-core inventions to individual technical fields. For instance, a ML

68 Art. 4 Directive 2009/24/EC (Software Directive); 17 U.S.C. §§ 101-103.

69 Art. 4 WIPO Copyright Treaty 1996; Art. 1 Software Directive; and 17 U.S.C. §§ 101. See *SAS Institute v World Programming Ltd*, CJEU (2012) C-406/10, ECLI:EU:C:2012:259.

70 Stefano Baruffaldi, et.al. (n 12) 26. Peter R Slowinski 'Rethinking Software Protection', in J.-A. Lee, K.-C. Liu, R. M. Hilty (n 6) 341,361.

71 Peter R Slowinski (n 70); Katarina Foss-Solbrekk, 'Three routes to protecting AI systems and their algorithms under IP law: The good, the bad and the ugly' (2021) 16(3) *Journal of Intellectual Property Law & Practice*, 246, 258.

72 Peter R Slowinski (n 70) 354.

73 Begoña Gonzalez Otero (n 67).

74 Josef Drexl, Reto M. Hilty et.al. 'Artificial Intelligence and Intellectual Property Law Position Statement of the Max Planck Institute for Innovation and Competition of 9 April 2021 on the Current Debate' (2021) <[https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3822924](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3822924)>.

75 Intellectual Property Owners Association Artificial Intelligence and Emerging Technologies Committee, 'Sui Generis Right for Trained AI Models' (2020) <<https://ipo.org/wp-content/uploads/2020/11/SG-model-rights-committee-paper-pub.pdf>>.

76 Blind et al.(n 13) 340.

77 Peter R Slowinski (n 70) 356.

78 Benjamin Sobel 'A Taxonomy of Training Data: Disentangling the Mismatched Rights, Remedies, and Rationales for Restricting Machine Learning' in Reto Hilty, Jyh-An Lee, Kung-Chung Liu (n 6).

79 Communication from The Commission to The European Parliament, The Council, The European Economic and Social Committee and The Committee of The Regions, 'Towards A Common European Data Space', COM(2018) 232 final [2018] 6.

80 Josef Drexl, 'Designing Competitive Markets for Industrial Data – Between Propertisation and Access' (2017) 8 *JIPITEC*, para 6,12; Catarina Arnaut, Marta Pont, Elizabeth Scaria, Arnaud Berghmans, Sophie Leconte, 'Study on data sharing between companies in Europe' (2018) <<https://op.europa.eu/en/publication-detail/-/publication/8b8776ff-4834-11e8-be1d-01aa75ed71a1/language-en>>.

81 Japan Patent Office, 'Recent Trends in AI-related Inventions – Report' (2020) <[https://www.jpo.go.jp/e/system/patent/gaiyo/ai/document/ai\\_shutsugan\\_chosa/report-2020.pdf](https://www.jpo.go.jp/e/system/patent/gaiyo/ai/document/ai_shutsugan_chosa/report-2020.pdf)>; Kimberley Bayliss, 'Drafting AI patent applications for success at the EPO – eligibility and claim formulation' (iam, 2021) <<https://www.iam-media.com/patents/ai-epo-patent-drafting-eligibility-claim-formulation-hlk-co-published>>.

model can be applied to image recognition, speech recognition, diagnosis, or prediction.<sup>82</sup>

29 When examining AI-related inventions, the European Patent Office (EPO) applies the two-hurdle approach of computer-implemented inventions (CII).<sup>83</sup> According to the patent-eligibility requirement, the invention cannot be *excluded subject matter*. To be patentable, AI-related inventions must be described and claimed in the context of an operation in a technical system, or in control of a technical process.<sup>84</sup> Subsequently, the EPO will analyse, as in any patent application, whether the AI-related invention meets the requirements of novelty, inventive step and industrial application.<sup>85</sup> Regarding the inventive step, the EPO will only consider the features of the technical character of the invention.<sup>86</sup>

30 In the US, AI-related inventions must pass the “two-part test” implemented by the Supreme Court in *Alice v. CLS Bank*.<sup>87</sup> Following to the ruling, claims must be directed to a “process, machine, manufacture or composition of matter”<sup>88</sup>, but not to an abstract idea such as an algorithm or method of calculation.<sup>89</sup> Nevertheless, as the Court clarified, even if the claims are directed to an abstract idea, the invention may be patentable if it comprises an “inventive concept”,

meaning that “the implementation of the idea is not generic, conventional or obvious”.<sup>90</sup>

31 AI in general and ML in particular are based on algorithms and computer models, which are of an abstract mathematical nature.<sup>91</sup> They are therefore excluded from patentability when claimed as such.<sup>92</sup> The same applies to some parameters, such as the weights, biases and evaluation mechanisms used in the training of the system. However, when all these features are applied in a specific technical use, they can be protected as elements of a broader invention, but only for that specific application.<sup>93</sup>

### III. Trade secrets

32 The ideas or principles underlying the software, the programming language, the algorithms, models, and the aforementioned parameters can be protected by TS<sup>94</sup> if they are secret, have commercial value because of it, and the person lawfully in control of the information has taken reasonable steps to preserve their secrecy.<sup>95</sup> Nonetheless, since it is generally considered difficult to reverse engineer AI systems, maintaining the secrecy of AI innovation could prevent collaboration and integration among

82 *Ibid.*

83 The term “computer implemented inventions” covers claims which involve “computers, computer networks or other programmable apparatus, whereby at least one feature is realised by means of a program”. EPO Guidelines for Examination (2021) <[https://www.epo.org/law-practice/legal-texts/html/guidelines/e/f\\_iv\\_3\\_9.htm](https://www.epo.org/law-practice/legal-texts/html/guidelines/e/f_iv_3_9.htm)>.

84 EPO, ‘Guidelines for Examination, Mathematical methods’ (2018) <[https://www.epo.org/law-practice/legal-texts/html/guidelines/e/g\\_ii\\_3\\_3.htm](https://www.epo.org/law-practice/legal-texts/html/guidelines/e/g_ii_3_3.htm)>.

85 Art. 52.1 European Patent Convention (EPC).

86 EPO, ‘Guidelines for Examination, Artificial Intelligence and machine learning’ (2018) <[https://www.epo.org/law-practice/legal-texts/html/guidelines/e/g\\_ii\\_3\\_3\\_1.htm](https://www.epo.org/law-practice/legal-texts/html/guidelines/e/g_ii_3_3_1.htm)>.

87 US Supreme Court, *Alice Corp. v. CLS Bank International*, 573 U.S. 208 (2014).

88 35 U.S.C. Code §101.

89 Supreme Court: *Assoc. for Molecular Pathology v. Myriad Genetics, Inc.*, 569 U.S. 576 (2013); US District Court Northern D. Illinois E.D.: *Neochloris, Inc. v. Emerson Process Mgmt. LLP*, 140 F. Supp. 3d 763 (2015), wherein one claim recited “an artificial neural network module” and the Court found that “it is not even clear [from the specification or claim itself] what [that term] refers to besides a [generalized] central processing unit – a basic computer’s brain”.

90 USPTO, ‘Patent Subject Matter Eligibility [R-10.2019]’ (2019) <<https://www.uspto.gov/web/offices/pac/mpep/s2106.html>>; James H. Ortega, ‘Clarifying the Distinction Between the “Inventive Concept” and “Patentability” Requirements When Determining Patent-Eligible Subject Matter’ (21 October, 2016, C&C Insights) <<https://cclaw.com/2016/10/21/clarifying-distinction-inventive-concept-patentability-requirements-determining-patent-eligible-subject-matter/>>.

91 EPO (n 86).

92 Art. 52.2(c) and 3 EPC.

93 Peter R Slowinski (n 70) 355; Josef Drexl, Reto M. Hilty et al. (n 74), Katarina Foss-Solbrekk (n 71).

94 Andrew Rapacke, ‘Using Trade Secret Protection for AI IP’ (2018) Rapacke Law Group <<https://arapackelaw.com/trade-secrets/trade-secret-ai-ip/>>; Jessica M. Meyers, ‘Artificial Intelligence and Trade Secrets’ (2019, American Bar Association) <[https://www.americanbar.org/groups/intellectual\\_property\\_law/publications/landslide/2018-19/january-february/artificial-intelligence-trade-secrets-webinar/](https://www.americanbar.org/groups/intellectual_property_law/publications/landslide/2018-19/january-february/artificial-intelligence-trade-secrets-webinar/)>.

95 According to Art. 2.1 Directive 2016/943 (Trade Secrets Directive), “trade secret” means information which meets all of these requirements; Josef Drexl, Reto M. Hilty et al (n 74); Peter R Slowinski (n 70) 356.

AI developers.<sup>96</sup> Conversely to reciprocal open-source licenses, permissive open-source licenses might work well with the use of non-disclosure agreements related to TS and know-how of the AI system.<sup>97</sup>

#### IV. Impact on the enforceability of OS licenses

33 There is no clear-cut answer for IPR protection of AI features. These might be subject to different interpretations, coming from the substance of the object of protection. Without IP rights the question arises whether the object of the license is missing. If potential implementers had to undertake the assessment of copyrightability and patentability of open-source AI features, they would incur an additional cost. Not all implementers would be willing, or have the legal expertise and financial resources, to do so. Also, the implementation of an IP clearance system in OS repositories carried out by the sponsor could have the effect of discouraging contributions to these repositories, as it is a large cost as well. The scenario is challenging. However, before embarking on a possible solution, the first step in this debate is to determine whether some AI features, such as ML models and datasets, are indeed protectable or not, given their wide availability under OSS licenses.

34 From the IPR holders' perspective, the enforceability of their IP rights is crucial. Traditionally the enforcement of OSS licenses has been conducted under the so-called "community enforcement", in which a warning letter or a report notifying the non-compliance is reportedly sufficient for overcoming the problem.<sup>98</sup> Nevertheless, even if voluntary compliance remains predominant, commercial litigation around OSS is not alien in the field. Consequently, IPR holders may also enforce their rights by claiming IPR infringement<sup>99</sup> and/

or<sup>100</sup> contractual breach<sup>101</sup>, depending both on the jurisdiction and the facts at the origin of the claim.<sup>102</sup> It is worth noting that unfair competition laws might also be a pertinent instrument in some instances.<sup>103</sup>

35 Until now, this article has explored the strategic use of open-source licenses as core competitive factors, and the implications of the IPR protection of AI features for open-source licensing. The next step is

---

this option. Among them see, in the EU, *Welte v. Sitecom Deutschland GmbH*, Munich District Court (*Landgericht München*) Case No. 21 O 6123/04 (19 May 2004). In the US, see Court of Appeals for the Federal Circuit *Jacobsen v. Katzer, inc.* 535 F.3d 1373, 1379 (Fed. Cir. 2008); *Free Software Foundation v. Cisco*, District Court Sth. D. New York (11 December 2008), the case ended with a settlement. < <https://www.fsf.org/news/2008-12-cisco-suit>>.

100 Regarding accumulation of IPR infringement and breach of contract claims, not every jurisdiction accepts accumulating both contractual and IPR infringement claims. In the US, see *Artifex Software, Inc. v. Hancom, Inc.*, Case No. 16-cv-06982-JSC, (N.D. Cal. 2017). Contrarywise, in France, civil liability law is based on the principle of non-cumulation of criminal and contractual liability. Thus, an IPR holder will have always to claim either breach of contract or IPR infringement, but not both; See also Heather Meeker, 'Open-source and the Age of Enforcement' (2012) 4(2) *Hastings Science and Technology Law Journal* 275,276.

101 In the EU, see *Entre'Ouvert v Orange & Orange Business Services* Paris Court of Appeal, Pôle 5 Ch. 2, 19<sup>th</sup> March 2021, n°19/17493, where the Court held that: "lorsque le fait générateur d'une atteinte à un droit de propriété intellectuelle résulte d'un manquement contractuel, le titulaire du droit ayant consenti par contrat à son utilisation sous certaines réserves, alors seule une action en responsabilité contractuelle est recevable (...)".

102 The conundrum relies on discerning whether IP law or contract law applies when enforcing an open-source license. Notwithstanding the latter, from a holistic approach, see CJEU C-666/18 *IT Development SAS v Free Mobile SAS* (2020) ECLI 1099. In this case, the CJEU held that regardless of the national applicable legal framework, an IPR holder will always be able to benefit from the warranties stemming from the provisions of the Directive 2004/48/CE (IPR Enforcement Directive).

103 The *Entre'Ouvert v Orange & Orange Business* case involved a breach of the GPLv2, the Court held that the licensee had taken an unfair competitive advantage stemming from the use of the software without complying with the licensing conditions imposed by the GPLv2, leading the company to be selected in a public procurement process before the French public administration (i.e., "parasitisme"). See *Entre'Ouvert v Orange & Orange Business Services* (n 96). Also, on the enforcement of unfair competition law by OS distributors see Till Jaeger, 'Enforcement of the GNU GPL in Germany and Europe', (2010) 1 *JIPITEC* 35.

96 EPO, 'Patenting Artificial Intelligence Conference summary' (2018) <[https://e-courses.epo.org/pluginfile.php/23523/mod\\_resource/content/2/Summary%20Artificial%20Intelligence%20Conference.pdf](https://e-courses.epo.org/pluginfile.php/23523/mod_resource/content/2/Summary%20Artificial%20Intelligence%20Conference.pdf)>; Katarina Foss-Solbrekk (n 73).

97 Blind et al. (n 13) 191,192, see fig 6.5. For instance, permissive licenses might provide the IPR holder with an opportunity to offer a custom proprietary premium license attached to the OS core feature where sensitive information for the use of the OS core AI feature is disclosed.

98 Eben Moglen, 'Enforcing the GNU GPL' (2001, GNU Operating System) <<https://www.gnu.org/philosophy/enforcing-gpl.html>>.

99 Several judicial decisions have already pointed towards

to examine which open-source licenses are the most widely used in the AI space, and why. The choice of an open-source license might define a company's IPR strategy.

## D. Open-source dynamics: a legal approach

### I. Most used open-source licenses for AI: rationale and legal assessment

- 36 Open-source strategies play a key role in the development and control of AI ecosystems.<sup>104</sup> To gain a better understanding of these dynamics in AI settings, the authors scrutinised 60 OSS AI projects and their licenses (see Annex I).<sup>105</sup> The main points of assessment were the predominant licensing terms; whether the project has a sponsor or has been *community-driven* from the beginning; and the existence of platform strategies in terms of ecosystem creation.<sup>106</sup> 42 projects have been released individually by a firm; 8 have been jointly released by a partnership of several firms/institutions; 8 from consortia or OSS organisations (Apache Foundation); and 2 by research centres.
- 37 While 56 of the 60 analysed OSS AI projects use permissive open-source licenses (42 chose Apache 2.0; 8 MIT, 3 selected BSD 2-clause and 3 BSD 3-clause), only 4 AI OSS projects use restrictive licenses.<sup>107</sup>

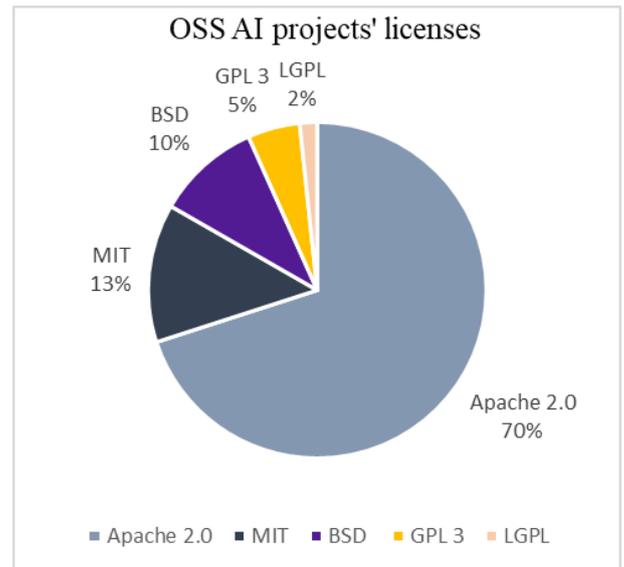


Figure 1 – Most used OSS licenses in 60 analysed AI projects

- 38 Our finding goes in line with a recent report sponsored by the European Commission, in which a survey of 441 respondents places permissive open-source licenses as the most used strategy for “the protection” of organisations’ know how.<sup>108</sup>
- 39 The authors believe that the preference for permissive licenses in AI projects seems to be mainly due to three strategic business factors. The first one is the possibility for software to be sublicensed under different terms and to be incorporated into proprietary applications. This possibility of combining permissive licenses with restrictive licenses, and even with proprietary ones, provides the necessary flexibility for adopting hybrid licensing models<sup>109</sup>, which are present in AI markets. For instance, in the field of ML and data analytics, companies such as H2O.ai<sup>110</sup> or TIBCO<sup>111</sup>, use open-source licenses tailored for commercial purposes, like MIT or Apache 2.0.<sup>112</sup>

104 Ibrahim Haddad (n 13) 8, 104; Gartner, ‘Magic Quadrant for Machine Learning and Deep Learning Platforms’ (2020). <<https://www.gartner.com/doc/reprints?id=11Y4BB6P-M&ct=200110&st=sb.html&status=200>>.

105 For project selection criteria, see section A. Taking a technical approach, although we focused on ‘AI software tools’ as a general framework including a non-exhaustive list of core technical features (libraries, ML frameworks, programming languages, etc), we specially focus afterwards on the platforms offering an AI toolkit or framework.

106 For this paper, an ecosystem is a network of interconnected systems, in this case interconnected software features, each of them potentially representing a product/service market.

107 3: GPLv3; 1: Lesser GPL (2.1).

108 Blind et al. (n 13) 192, fig 6.5.

109 For OS licenses’ compatibility, see Heather Meeker (n 66); Thomas F. Gordon, ‘Report on Problem Scope and Definition about OSS License Compatibility’ (2009) Quality Platform for Open-source Software <<https://www.osscc.net/pdf/QualipsoA1D113.pdf>>.

110 See H2O.ai <<https://www.h2o.ai>>.

111 See TIBCO <<https://www.tibco.com>>.

112 Even so-called restrictive open-source licenses might in given circumstances allow combination with other licenses. For instance, in the case of KNIME’s platform, the OSS license GPLv3 integrates an additional exception that allows the use of an Application Programming Interface (API) to add proprietary extensions. Henceforth, the fact that GPL-family licenses integrate ‘copyleft’ clauses do not

- 40 The second business factor is based on the complexity of GNU General Public License (GPL-style) licenses and the lack of harmonisation on the interpretation of some specific terms and their scope.<sup>113</sup> This makes the license an ambiguous set of legal terms which might be seen as a deterrent for firms willing to release their software under an open-source license.<sup>114</sup> Although GPL-style licenses have been used on a marginal and strategic vein with the advent of commercial OSS, the increasing frictions between big cloud service providers and smaller companies (SMEs) on the use of open-sourced software has reinvigorated its use.<sup>115</sup>
- 41 The third factor is that permissive licenses are designed to ensure mass adoption of a technology, as implementers feel more confident if they are allowed to build any kind of project, open-source or not, on top of the licensed code. Therefore, permissive licenses are a pertinent option when sponsors aim for their software tool to become a de facto standard in a given market, and subsequently build an ecosystem around it. As for the use of a permissive license to build an ecosystem, the best examples are the ML

frameworks<sup>116</sup>, such as TensorFlow<sup>117</sup> and Paddle Paddle sponsored by Google and Baidu respectively under the Apache.2.0 license, or PyTorch, sponsored by Facebook and licensed under BSD-3.<sup>118</sup> Some of these actors, like Google and Facebook, are proving to be very successful with such a strategy. For example, from the projects analysed, several are compatible with both TensorFlow and PyTorch—e.g., features built on top. More tellingly, there are some specific projects that seek interoperability between tools and frameworks to train models<sup>119</sup>, such as ONNX, as well as to use models trained in diverse ML frameworks, such as Neuropod.<sup>120</sup> In addition to this, it should be noted that some companies in the hardware market are also building AI-related microprocessors that aim to be compatible with these current predominant ML frameworks.<sup>121</sup>

---

literally imply that subsequent commercial strategies are foreclosed. It will depend on the affected software module, on the license and on the interpretation of its scope. See KNIME's open-source record <<https://www.knime.com/knime-open-source-story>>.

- 113 On contractual interpretation of OS licenses and their terms/clauses see also Andrés Guadamuz (n 30); and, Eli Greenbaum, 'Open-source Interpretation' (2021) *12(1) Journal of Open Law, Technology, & Society*.
- 114 The latter statement might also be true for permissive licenses in some cases, although these are simpler and more user-friendly than GPL-family ones.
- 115 More tellingly, the trend for SMEs nowadays in cloud infrastructure markets is steering towards the adoption of restrictive open-source licenses and a new type of open software license called 'source available' license. See Heather Meaker, 'Elastic License 2.0 and the Evolution of Open-source Licensing' (2021, COSS.community) <<https://www.coss.community/coss/elastic-license-2-0-and-the-evolution-of-open-source-licensing-3jb3>>.

---

116 We provide a definition which might also serve as justification for us to refer to these frameworks as 'platforms': Caffe2, 'Caffe2 and PyTorch join forces to create a Research + Production platform PyTorch 1.0' (2018) Caffe2: "In practice, any deep learning framework is a stack of multiple libraries and technologies operating at different abstraction layers (from data reading and visualization to high-performant compute kernels)." <[https://caffe2.ai/blog/2018/05/02/Caffe2\\_PyTorch\\_1\\_0.html](https://caffe2.ai/blog/2018/05/02/Caffe2_PyTorch_1_0.html)>

117 TensorFlow DL framework is licensed under an Apache 2.0 license, it has received more than 41,000 commits from 1,600 distinct contributors, and over 68,000 forks have been made (copy of the code for further modification). See Stefano Baruffaldi et al. (n 12) 26.

118 Ibrahim Haddad (n 13) 98: "Most AI platforms are the results of years of investment and talent acquisition, and the open-source spinoff is a consequence of wanting to build an ecosystem versus a desire to collaborate with others on constructing a platform."

119 See Open Neural Network Exchange (ONNX) "a common set of operators - the building blocks of machine learning and deep learning models - and a common file format to enable AI developers to use models with a variety of frameworks, tools, runtimes, and compiler". <ONNX | Supported Tools>.

120 See Neuropod, "a library that provides a uniform interface to run deep learning models from multiple frameworks in C++ and Python" <GitHub - uber/neuropod: A uniform interface to run deep learning models from multiple frameworks>.

121 Devin Coldewey, 'Mac-optimized TensorFlow flexes new M1 and GPU muscles' (2020, TechCrunch) <<https://techcrunch.com/2020/11/18/mac-optimized-tensorflow-flexes-new-m1-and-gpu-muscles/>>.

## II. Common open-source licenses in AI settings

### 1. Permissive licenses

42 Permissive licenses allow users to freely copy, distribute and modify the software.<sup>122</sup> By not imposing restrictive conditions on the redistribution of the software, they allow licensees to profit from their modifications of the underlying OSS.<sup>123</sup> However, in the decision whether to embrace permissive licenses the following should be considered: as with the rest of open-source licenses, it is mandatory to maintain the copyright and license notice when redistributing the source code.<sup>124</sup> Some permissive licenses, such as Apache 2.0<sup>125</sup>, also require the distributor to add notices regarding the modification of the files.<sup>126</sup> Subsequently, it is important to understand the exact scope of the license, especially if patents are involved, and to be aware that the program is provided by the licensor without any warranty and with an exclusion of liability. Instead, those using the licensed software are responsible for obtaining grants for third-party IP rights in case they are infringed.<sup>127</sup>

43 Although there are many permissive licenses, the most popular ones in AI projects are the BSD 2 and 3 Clause, the MIT, and Apache 2.0.

#### a) BSD 2 and 3 Clause

44 The BSD 2 and 3 Clause licenses are short and at first sight simple to understand. They allow for the “redistribution and use in source and binary form”

of the software, “with or without modification”.<sup>128</sup> Among the rights conferred on the copyright holder listed in section C, only the right to “redistribute” is expressly mentioned. Nevertheless, the rights of transformation and reproduction are implicitly granted, as the redistribution may be of a modified or unmodified copy.<sup>129</sup>

45 The other explicitly authorised action, i.e., the use of the software, is an exclusive right of the patent holders. This license ‘language’ raises doubts as to whether an implicit patent license is also granted, and if so, what would be the scope.<sup>130</sup> It should also be observed that the term “sublicensing” does not appear in the text of the license. Thus, to establish whether a sublicense is possible and, if so, what would be its scope, it is necessary to analyse the principles of contract interpretation and the practice of the OSS community.<sup>131</sup>

46 To conclude, the BSD may be an attractive option for ML platform sponsors, since it offers the licensors the flexibility to design their own patent statement.<sup>132</sup> Yet, one should be cautious when combining the BSD with other license terms, as illustrated by the example of the Facebook React Project.<sup>133</sup> The project was issued under the BSD-3 Clause license text plus a Facebook’s own custom-written patent declaration, under which those suing Facebook for patent rights, even those not related to the project, would face an automatically revocation of the royalty free patent license. Since the added patent clause received strong criticism by stakeholders, Facebook had to re-license it under MIT.<sup>134</sup>

122 Ayala Goldstein, ‘Open-source Licenses Explained’ (2010, WhiteSource) <<https://resources.whitesourcesoftware.com/blog-whitesource/open-source-licenses-explained>>.

123 David J. Kappos, ‘Open-source Software and Standards Development Organizations: Symbiotic Functions in the Innovation Equation’ (2017) 18 *The Columbia Science & Technology Law Review* 263, 264.

124 Matt Mecoli, ‘A Data Scientist’s Guide to Open-source Licensing’ (2018, towards data science) <<https://towardsdatascience.com/a-data-scientists-guide-to-open-source-licensing-c70d5fe42079>>.

125 See license text at <<http://www.apache.org/licenses/LICENSE-2.0.html>>.

126 Clause 4.d).

127 Lawrence Rosen, (n 30) 77,80.

128 See the license in <<https://opensource.org/licenses/BSD-3-Clause>>.

129 Andrew Sinclair, ‘License Profile: BSD’ (2010) 2(1) IFOSS L. Rev 2,4.

130 Lawrence Rosen (n 30) 83,84.

131 *Ibid.*

132 Aner Mazur, ‘Apache license 2.0, MIT license or BSD license: Who is the fairest of them all?’ (2017, snykblog) <<https://snyk.io/blog/mit-apache-bsd-fairest-of-them-all/>>.

133 Jenn Schiffer, ‘Over React? Open-source licensing, Facebook, WordPress, and Patents’ (2018, Medium) <<https://medium.com/glitch/over-react-open-source-licensing-facebook-wordpress-and-patents-efeece333f12>>; Martin Husovec, ‘Standardization, Open-Source, and Innovation: Sketching the Effect of IPR Policies’, in Jorge Contreras (ed.) *Cambridge Handbook of Technical Standardization Law* (Cambridge University Press, 2019).

134 Quincy Larson, ‘Facebook just changed the license on React. Here’s a 2-minute explanation why’ (2017, freeCodeCamp)

**b) MIT**

- 47 The MIT license shares the principles of, but it is more comprehensive than, the BSD license. The MIT gives permission free of charge to “use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software.”<sup>135</sup> Therefore, it refers to all the economic rights of copyright holders and, except for the right to “make”, targets almost all the exclusive rights under patent law. Then, under a broad interpretation, the MIT implicitly includes a patent license, whose scope is nevertheless uncertain.<sup>136</sup> As stated previously, this is relevant for stakeholders who might not be aware of which patents are granted, for what purpose, and whether sublicenses are permitted.<sup>137</sup> In the event that the patent license does not cover the derivative works, licensees must obtain directly from the original licensor of the software an explicit grant of the patent rights that are required to use its modified versions.<sup>138</sup>
- 48 MIT is also a highly flexible license that leaves significant freedom in designing the scope of patent grants. Nevertheless, clear and explicit patent grants entitle the licensee to use, modify, distribute and—under some open-source licenses—sublicense software covered by the patent with greater certitude.<sup>139</sup> Consequently, although it is clearer in its terms than the BSD, some other licenses, as the Apache 2.0., seem to be more aligned with the interests of the ML platform’s sponsors.

**c) Apache 2.0**

- 49 Apache 2.0 is a permissive “perpetual, worldwide, non-exclusive, no-charge and royalty free” license for copyright and patents.<sup>140</sup> Whilst it has similar principles to the BSD and MIT licenses, Apache

2.0 is much more detailed and thus provides more certainty to its adopters.

- 50 Apache 2.0 includes a comprehensive copyright grant and includes the right to sublicense and distribute in source or object both original and derivative software.<sup>141</sup> In addition, there is an explicit grant of any patents of the contributor that other collaborators of the project governed by the Apache license automatically infringe by using its contribution; as well as of any patents infringed by the resulting combination on the date of submission of such contribution with the Apache 2.0 licensed software to which it was provided.<sup>142</sup> Licensable patent claims include those that may be acquired in the future, “as long as they read on the original contribution as made at the original time”.<sup>143</sup> However, the license does not extend to patents that would be infringed by an intermediate contribution altering the upstream code or combining it with the work in a new way.<sup>144</sup>
- 51 The most sensitive element of this license for a patent holder is its patent retaliation clause. This clause provides that any patent rights granted under the Apache 2.0 will be immediately revoked against a contributor that initiates a patent infringement litigation regarding the work or a contribution incorporated in the work.<sup>145</sup> The purpose of patent retaliation is to discourage any licensee from suing for patent infringement over the Apache licensed software.<sup>146</sup>
- 52 Apache 2.0 is the predominant license used in AI OSS projects due to its specificity in terms of licensees’ obligations. The clarity, especially regarding the granting of patents, helps to attract the organisations that are most concerned about lack of access to software patents.<sup>147</sup> Yet, being

---

<https://www.freecodecamp.org/news/facebook-just-changed-the-license-on-react-heres-a-2-minute-explanation-why-5878478913b2/>.

135 See license in <https://opensource.org/licenses/MIT>.

136 Anna Haapanen, ‘Free and Open-source Software & the Mystery of Software Patent Licenses’ (2015) 7(1) *International Free and Open-source Software Law Review* 20.

137 *Ibid.*

138 Lawrence Rosen (n 30) 88,90.

139 Andrew M. St Laurent (n 30) 14,24.

140 Clauses 2 and 3.

141 Clause 2.

142 Clause 3.

143 See FAQ about Apache Licensing, ‘What is the scope of patent grants made to the ASF?’ <http://www.apache.org/foundation/license-faq.html#PatentScope>.

144 Andrew Sinclair, ‘License Profile: Apache License, Version 2.0’ (2010) 2(1) *IFOSSL Rev.* 109,110.

145 Clause 3.

146 Jay P. Kesan, ‘The Fallacy of OSS Discrimination by FRAND Licensing: An Empirical Analysis’ (2011) *Illinois Public Law Research Paper No. 10-14* 6; Eli Greenbaum (n 109).

147 Joseph Morris, ‘Which License Should I Use? MIT vs. Apache vs. GPL’ (2016, Exygy) <https://exygy.com/blog/which-license-should-i-use-mit-vs-apache-vs-gpl/>.

aware from the beginning of the scope of the patents covered by the license and the potential risk of a patent retaliation clause is crucial for adopting an adequate OSS strategy.

- 53 However, companies choosing permissive licenses must be aware of the possibility of competitors' appropriation and improvement of the released software tool. A recent example that illustrates both the complexities of license compatibility and its articulation with companies' business models can be found in Elastic. The company launched two projects under Apache 2.0, Elasticsearch and Kibana<sup>148</sup>, but has recently changed its licensing model apparently due to some frictions with Amazon Web Services (AWS) products.<sup>149</sup> Elastic decided that future versions of these two programs would be dual-licensed, allowing users to choose between Elastic's own license<sup>150</sup> and the Server-Side Public License (SSPL).<sup>151</sup> Both licenses impose stricter conditions than Apache 2.0 on the use and modification of derivative works. Hence, by their adoption Elastic has rendered future versions of its projects incompatible with other licenses that allow the distribution of modified software as commercial services. In response, AWS<sup>152</sup> and other companies<sup>153</sup> have announced that they will create and maintain an Apache 2 licensed fork of Elasticsearch and Kibana.<sup>154</sup>

---

148 Elasticsearch is a database manager designed for enterprise search, and Kibana is a data visualisation tool. See their respective webpages at <<https://www.elastic.co/de/elasticsearch/>, <https://www.elastic.co/de/kibana>>.

149 Steven J. Vaughan-Nichols, 'Elastic changes open-source license to monetize cloud-service use' (2021) ZDNet <<https://www.zdnet.com/article/elastic-changes-open-source-license-to-monetize-cloud-service-use/>>.

150 See license at <<https://www.elastic.co/licensing/elastic-license>>.

151 See license at <<https://www.mongodb.com/licensing/server-side-public-license>>.

152 Carl Meadows, Jules Graybill, Kyle Davis, and Mehul Shah, 'Stepping up for a truly open-source Elasticsearch' (2021, AWS Open-source Blog) <<https://aws.amazon.com/blogs/opensource/stepping-up-for-a-truly-open-source-elasticsearch/>>.

153 Tomer Levy, 'Truly Doubling Down on Open-source' (2021, logz.io) <<https://logz.io/blog/open-source-elasticsearch-doubling-down/>>.

154 Steven J. Vaughan-Nichols, 'AWS, as predicted, is forking Elasticsearch' (2021, ZDNet) <<https://www.zdnet.com/article/aws-as-predicted-is-forking-elasticsearch/>>.

#### d) Permissive licenses' allocation in ML frameworks

| “Project”                   | License    | Datasets | Models | STK + Complementary Material <sup>155</sup> | Interfaces |
|-----------------------------|------------|----------|--------|---|------------|
| Tensorflow                  | Apache 2.0 | X        | X      | X   | X          |
| Pythorch                    | BSD 3      | X        | X      | X   | X          |
| ParlaAI                     | MIT        | X        | X      | X   | X          |
| Microsoft Cognitive Toolkit | MIT        |          | X      | X   | X          |
| Paddle Paddle               | Apache 2.0 | X        | X      | X   | X          |
| Keras                       | Apache 2.0 | X        | X      | X   | X          |

Table 1. Examples of ML Frameworks: technical components' licensing

54 Relevant ML frameworks are released under permissive open-source licenses. Although Apache 2.0 predominates, and is used in Tensorflow, Padle Padle, Keras; BSD-3 is used in another of the most relevant frameworks, Pytorch, as well as MIT in the Microsoft Cognitive toolkit and Parla AI.

55 It is worth noting that many platforms, in addition to the software toolkit for model training and the code that incorporates the different ML algorithms<sup>156</sup>,

155 STK means software tool kit. Complementary material might be composed by the tools provided in addition to the software development kit needed to run and/or train the model, and training algorithms

156 See TensorFlow <<https://github.com/tensorflow/models>> <Libraries & extensions | TensorFlow> <Tools | TensorFlow>; Catboost <GitHub - catboost/catboost: A fast, scalable, high performance Gradient Boosting on Decision Trees library, used for ranking, classification, regression and other machine learning tasks for Python, R, Java, C++. Supports computation on CPU and GPU.>; ParlaAI <Standard Agents — ParlaAI Documentation>; Microsoft Cognitive Toolkit <GitHub - microsoft/CNTK: Microsoft Cognitive Toolkit (CNTK), an open-source deep-learning toolkit>; Paddle Paddle <GitHub - PaddlePaddle/Paddle: PArallel Distributed Deep LEarning: Machine Learning Framework from Industrial Practice (『飞桨』核心框架·深度学习&机器学习高性能单机、分布式训练和跨平台部署)>; In addition to the code provided in these platforms, we can also find other AI libraries, such as Scikit learn: ML library for ML basics <<https://scikit-learn.org/stable/>>; AI Fairness 360: “includes a comprehensive set of metrics for datasets and models to test for

offer other tools, such as datasets<sup>157</sup>, APIs<sup>158</sup> and models.<sup>159</sup> Two different open-source licensing practices should be considered: tool-by-tool licensing and umbrella licensing. Under umbrella licensing, which is most used<sup>160</sup>, all the software tools under the ML framework are embedded under a single license. Conversely, under the tool-by-tool

biases” <<https://github.com/Trusted-AI/AIF360>>, and AI Explainability 360: “The AI Explainability 360 Python package includes a comprehensive set of algorithms that cover different dimensions of explanations along with proxy explainability metrics”. <<https://ai-explainability-360.org/>>.

157 See TensorFlow <Models & datasets | TensorFlow>; ParlaAI <Tasks — ParlaAI Documentation>; and Pythorch <<https://pytorch.org/vision/0.8/datasets.html>>.

158 See TensorFlow <<https://www.tensorflow.org/versions>>; Keras <<https://keras.io>>; Pythorch <[https://pytorch.org/cppdocs/api/library\\_root.html](https://pytorch.org/cppdocs/api/library_root.html)>; Padle PAdle <<https://github.com/PaddlePaddle/Paddle-Lite>>; Catboost <catboost/CatboostModelAPI.md at master · catboost/catboost · GitHub> and Microsoft Cognitive Toolkit: <<https://docs.microsoft.com/en-us/cognitive-toolkit/cntk-library-api>>.

159 ParlaAI <Model Zoo — ParlaAI Documentation>; TensorFlow <Models & datasets | TensorFlow> <TensorFlow Hub>; Pythorch <<https://pytorch.org/vision/stable/models.html>>; Microsoft Cognitive Toolkit <CNTK/PretrainedModels at master · microsoft/CNTK · GitHub>; and Paddle Paddle <<https://github.com/PaddlePaddle/PaddleHub>>.

160 This is the case of, for instance, Tensorflow, Paddle Paddle, and Keras.

licensing, each software tool of the ML framework has its own license.<sup>161</sup>

- 56 In practical terms, it might seem pertinent to ask whether there is any difference between the modular approach of tool-by-tool licensing, and the holistic approach of umbrella licensing. At a first glance, as the target is the same, it might look indifferent to use either when releasing each feature or all the features of the framework in an OSS repository. Even more, umbrella licensing might streamline the licensing of the entire framework and avoid transactions costs and time investment integrating individual licenses, although being the same, in each tool of the framework. Nonetheless, it must be further explored whether the adoption of a single license for an entire ML framework might also have the effect of *prima facie* covering tools for which IP protection is uncertain, such as APIs and algorithms, by an open-source license.
- 57 It must also be observed that further contributions to the various projects may be released under different licenses. In the same vein, the datasets used for training the model may have a different license than the framework with which they interact. Interoperability between frameworks and elements is therefore essential for AI development. It is equally important to ensure compatibility between the different open-source licenses covering each feature.<sup>162</sup> There are no drawbacks in this regard in the cases under review, since they are covered by permissive licenses, and they impose no restrictions on what code is added to the program or how it can be distributed. However, if it is intended to combine components that have a permissive license with a restrictive one, the situation becomes more complicated, as copyleft provisions in some restrictive licenses might be incompatible with permissive licenses' scope.<sup>163</sup>

161 OpenAI have many repositories with different licenses (mainly Apache 2.0 and MIT), and models are released in different ways. For instance, GPT-2 is licensed under a “modified MIT” `sgpt-2/LICENSE` at `master · openai/gpt-2 · GitHub`; the dataset of GTP-2 outputs under MIT as well, but GPT 3 not, and actually has been exclusively licensed to Microsoft.

162 Even if many platforms also provide APIs for this purpose, it is likewise possible to find projects that seek interoperability between tools and frameworks to train models, as well as to use models trained in diverse ML frameworks, such as ONNX and Neuropod, mentioned above.

163 For OS licenses' compatibility, see Heather Meeker (n 61) 63; See this post listing which licenses are compatible with GPL at: <https://www.gnu.org/licenses/license-list.html#GPLCompatibleLicenses>; Richard Stallman, ‘License Compatibility and Relicensing’ (20 November, 2020, GNU

## 2. Restrictive licenses: GPL family

- 58 Restrictive—also called hereditary<sup>164</sup> or reciprocal<sup>165</sup>—licenses, impose strict distribution requirements on the recipient. In principle, the distribution<sup>166</sup> of the modified software must be carried under the same license.<sup>167</sup> This idea is secured by so-called ‘copyleft’ clauses, which guarantee that those who wish to enjoy the freedom related to the licensed software have to give back to the community the same that they received from it in the first place.<sup>168</sup>

### a) GPL as a strategic competitive tool

- 59 Despite initially having access to the core software feature, implementers might be forced to disclose follow-on innovation under the same license, benefiting the sponsor. Furthermore, the same action might lead in the mid/long run to the commoditisation of a given software layer and to the exclusion of any price competition. As a result, competitors for whom price competition is an essential parameter to remain competitive in the market could be affected.<sup>169</sup> Quality and innovation are thus going to be the leading competition parameters, which might not be affordable for every market actor. In a different setting, a company willing to “over throne” a competitor whose software product is becoming the standard in the market might release a competing GPL alternative.

---

operation system) <https://www.gnu.org/licenses/license-compatibility.html>.

164 Heather Meeker, *The Open-source Alternative: Understanding Risks and Leveraging Opportunities* (Wiley, 2008) 57.

165 Ronald R. Mann, ‘The Commercialization of Open-source Software: Do Property Rights Still Matter?’ (2006) 20(1) *Harvard Journal of Law & Technology* 15.

166 For discussions around the scope of the term ‘distribution’ under the GPL see Steven Weber (n 21) 180; Ross Gardler, ‘Open-source and Governance’, in Noam Shemtov, Ian Walden (n 37) 74.

167 Josh Lerner, Jean Tirole (n 34) 22; Elad Harison (n 34) 90.

168 Steven Weber (n 21) 180; Ross Gardler (n 166) 73.

169 See Mingqing Xing, ‘The effect of competition from open-source software on the quality of proprietary software in the presence of network externalities’ (2015) *Journal of Industrial Engineering and Management*; Terrence August, Wei Chen, Kevin Zhu, ‘Competition Among Proprietary and Open-source Software Firms: The Role of Licensing in Strategic Contribution’ (2020) 67(5) *Management Science*; Blind et al. (n 13) 43.

With this move the company aims to attract a mass of users by facilitating ‘open’ zero price access to the software, and beyond, block the competitor’s proprietary use of its software.<sup>170</sup>

## b) Copyleft effect on the output of the ML system

- 60 In the context of ML techniques, such as natural language processing, models are trained to generate weights. The weights can be considered as the output of the process and might take the form of a machine-readable codified dataset from which interpretations are extracted.
- 61 In a context where some of the ML material, such as the trained model based on which weights are produced, is released under a GPL-style license it might be pertinent to ask whether the output result of running the model should be considered either a “derivate work”<sup>171</sup> or a “covered work” and “work based on the program”<sup>172</sup>, depending on the version of the GPL. For instance, companies such as OpenAI expressly modify the open-source license in order to clarify that there is no claim of ownership on the *content* created with GPT-2.<sup>173</sup> Nonetheless, without those disclaimers there is uncertainty on the scope and effect of copyleft on the weights generated by the trained model.

## c) Two examples of AI business models and GPL provisions

- ML-as-a-service and the limits of Affero GPL

- 62 Running an ML system might be offered as a cloud service, by which the user accesses the ML system by means of an API, such as OpenAI’s GPT-3<sup>174</sup> and

Amazon SageMaker.<sup>175</sup> If not yet, ML-as-a-service has the potential to become a standard practice, thus cloud-native licensing implications should be considered in the context of OS. GPL licenses, even Affero GPL<sup>176</sup> (AGPL), do not efficiently address remote server use, mainly due to the uncertainty around a lack of definition of terms essential for the triggering of the copyleft, e.g., “user”, “interacting remotely through a computer network”.<sup>177</sup> More precisely, it is doubtful whether the copyleft clause would be triggered in case the AGPL software is indirectly used, e.g. infrastructure-as-a-service where the AGPL software is just a module comprised in a software infrastructure, and thus it can be argued that the user does not directly interact with the AGPL software (i.e., a finetuned commercial application of the model).

- GPL3’s flexibility and commercial compatibility

- 63 The GPL3 qualifies as a ‘strong copyleft’ license due to the broad restrictions required for the distribution of works derived/based on the licensed program.<sup>178</sup> Yet, there is an interesting section of GPL3 bringing flexibility to both the IPR holder willing to implement the license and potential licensees: Section 7. Section 7 allows the IPR holder, either the sponsor of the software or a company having created a new derived version of it, to add further “additional permissions” which are described as “exceptions from one or more of its conditions”. “Additional permissions” may be freely removed from downstream licensees at their choice when conveying the work. However, for the latter to be integrated within the GPL3, it has to be made by the company holding IPRs related to the additional permissions, and not by any third party.<sup>179</sup>

- 64 This section brings flexibility in terms of potential combination of the license with other OSS licenses, such as Apache 2.0, or even allowing subsequent proprietary extensions. A clear example of its use

170 Heather Meeker (n 164) 231.

171 See GPL2 license.

172 See GPL3 license.

173 See OpenAI’s GPT-2 Github repository <<https://github.com/openai/gpt-2/blob/master/LICENSE>>; Another example, although not in the field of AI, is the one of GNU Image Manipulation Program, where the software is licensed under GPL3 but the artwork generated by it is free from GPL3 restrictions <<https://www.gimp.org/docs/userfaq.html#can-i-use-gimp-commercially>>.

174 OpenAI API <<https://openai.com/blog/openai-api/>>.

175 Free Machine Learning Services on AWS <[https://aws.amazon.com/free/machine-learning/?nc1=h\\_ls](https://aws.amazon.com/free/machine-learning/?nc1=h_ls)>.

176 Affero GPL <<https://www.gnu.org/licenses/agpl-3.0.en.html>>.

177 See Heather Meeker (n 164) 168; Jakub Mencl, W Kuan Hon, ‘Copyleft in the Cloud’, in Noam Shemtov, Ian Walden (n 37) 345.

178 See more in Luke McDonagh, ‘Copyright, Contract, and FOSS’, in Noam Shemtov, Ian Walden (n 37) 82; Clark D. Asay, ‘The General Public License Version 3.0.: Making or Breaking the FOSS Movement?’ (2008) 14 *Michigan Telecommunications and Technology Law Review* 274.

179 Free Software Foundation, ‘Opinion on Additional Terms’ (2006) <<https://gplv3.fsf.org/additional-terms-dd2.html/>>.

is the case of KNIME<sup>180</sup>, which provides its KNIME Analytics Platform under a GPL3 license with a specific extension of it granting additional permission for licensees to use a standard API enabling the adding of proprietary node extensions.<sup>181</sup> Thus, if an implementer develops new software nodes based on KNIME's platform, it has the certainty under the extension granted by KNIME beyond the GPL3, that these nodes are not covered works of KNIME Analytics Platform. This can be perceived as sharp strategy from the sponsor's side. While a GPL family license is used to restrict possible private derivations of its platform, there is also flexibility to develop proprietary extensions by using a standard API, potentially provided by KNIME. This allows KNIME to keep control over the platform and over which kind of commercial extensions are created, as well as the restriction of some others.

## E. Conclusion

- 65 There are several reasons for tech companies to employ open-source strategies in AI development. Some of them include achieving a competitive advantage in adjacent component markets from which they seek to derive revenue, gaining “first-mover advantages,” or preventing a competitor from patenting a core technology. Foremost, the main goal of certain market players is to attract a critical mass of users in order to create an ecosystem around their ML platforms. This is facilitated by the use of permissive licenses.
- 66 Employing open-source in the development of some emerging technologies has proven to create positive effects. Open-source licenses can reduce transaction costs and promote faster adoption of the technology. In addition, OSS platforms serve as a free testing area where bugs and risks can be corrected. Nonetheless, while understanding that participating in OSS projects could open great opportunities for small players, OSS should not equate free of charge with unconditional access. Thus, contributors to AI OSS platforms must be aware of the licensing terms before committing to such projects. For instance, an open-source license might oblige the licensee not to enforce certain infringed IPRs within the OSS, e.g., through reciprocity, non-assertion and retaliation clauses. Therefore, companies seeking a direct return on investment from the monetisation of their IPRs should have a clear understanding of the scope of the OSS license in question, especially

when it involves patents, and be sure that they are not granting more than what would be detrimental to their business model.

- 67 However, it should be stressed that for an OSS license to be effective, IPRs must exist. The protection by different IPRs of several elements essential to the development of AI systems, such as datasets, algorithms, ML models and APIs, is currently hotly debated. This is an issue of great importance that needs to be deeply analysed.
- 68 Nowadays, aside private R&D efforts carried by big tech and governments, the AI technology race is primarily taking place in open-source platforms and ecosystems.<sup>182</sup> Moreover, open-source is also experiencing a tough competition for future disruptive technologies.<sup>183</sup> Consequently, governments around the globe are recognising the importance of open-source in the success of these AI developments.<sup>184</sup> Derived from it, long term open innovation policies are trying to align with

180 KNIME is a company focused on data science and analytics <<https://www.knime.com/knime-open-source-story>>.

181 KNIME Analytics Platform license <<https://www.knime.com/downloads/full-license>>.

182 Ministry of Industry and Information Technology (n 70). There is also a trend on opening hardware infrastructure design for AI purposes, see Blind et al. have found opposed views for ML code, see Blind et al., (n 13) 309,310.

183 See Will Douglas Heaven, ‘Google is making it easier to develop quantum machine-learning apps’ (2020) MIT Technology Review <<https://www.technologyreview.com/2020/03/09/905420/google-software-tensorflow-quantum-machine-learning-apps-ai-computing/>>; Kyle Wiggers, ‘Baidu open-sources Paddle Quantum toolkit for AI quantum computing research’ (2020) Venturebeat <<https://venturebeat.com/2020/05/27/baidu-open-sources-paddle-quantum-toolkit-for-ai-quantum-computing-research/>>.

184 Blind et al., (n 13); Alexandra Theben, Laura Gunderson, Laura López Forés, Gianluca Misuraca, Francisco Lupiáñez Villanueva, *Challenges and limits of an open-source approach to Artificial Intelligence*, (European Parliament, 2021) Study for the Special Committee on Artificial Intelligence in a Digital Age (AIDA), Policy Department for Economic, Scientific and Quality of Life Policies; European Commission, Communication from the Commission, “Open-source Software Strategy 2020 – 2023, Think Open” (2020) 7149 final <[https://ec.europa.eu/info/sites/default/files/en\\_ec\\_open\\_source\\_strategy\\_2020-2023.pdf](https://ec.europa.eu/info/sites/default/files/en_ec_open_source_strategy_2020-2023.pdf)>; National Institute of Standards and Technology, “U.S. Leadership in AI: A Plan for Federal Engagement in Developing Technical Standards and Related Tools Prepared in response to Executive Order 13859. (2019) <[https://www.nist.gov/system/files/documents/2019/08/10/ai\\_standards\\_fedengagement\\_plan\\_9aug2019.pdf](https://www.nist.gov/system/files/documents/2019/08/10/ai_standards_fedengagement_plan_9aug2019.pdf)>; Ministry of Industry and Information Technology - Informatization and Software Services Division (n 70); Chen Du, ‘Chinese AI lab challenges Google, OpenAI with a model of 1.75 trillion parameters’ (2021, PingWest) <<https://en-pingwest-com.cdn.ampproject.org/c/s/en.pingwest.com/amp/a/8693>>.

innovation phenomena like open-source. Therefore, beyond the scope of this paper, it remains to be seen (and further explored) which role open-source is going to play in geopolitical innovation strategies.

**Annex I – Scrutinised OSS AI projects**

|                                | <b>AI related feature</b>                       | <b>OSS License</b> | <b>Further information</b>  |
|--------------------------------|---|--------------------|---|
| Acumos H2O Model Builder       | Model building and export                       | Apache 2.0         | <a href="https://github.com/acumos/model-builder-h2o-model-builder">https://github.com/acumos/model-builder-h2o-model-builder</a>       |
| Adlik                          | Optimising framework for DL models              | Apache 2.0         | <a href="https://github.com/Adlik/Adlik">https://github.com/Adlik/Adlik</a>   |
| Adversarial Robustness Toolbox | ML Python library                               | MIT                | <a href="https://github.com/Trusted-AI/adversarial-robustness-toolbox">https://github.com/Trusted-AI/adversarial-robustness-toolbox</a> |
| AI Explainability 360          | ML Python library                               | Apache 2.0         | <a href="https://github.com/Trusted-AI/AIX360">https://github.com/Trusted-AI/AIX360</a>   |
| AI Fairness 360                | ML Python/R library                             | Apache 2.0         | <a href="https://github.com/Trusted-AI/AIF360">https://github.com/Trusted-AI/AIF360</a>   |
| Amundsen                       | Metadata engine                                 | Apache 2.0         | <a href="https://github.com/amundsen-io/amundsen">https://github.com/amundsen-io/amundsen</a>   |
| Angel                          | ML and graph/computing platform                 | Apache 2.0         | <a href="https://github.com/Angel-ML/angel">https://github.com/Angel-ML/angel</a>   |
| Apache Singa                   | Distributed DL Library                          | Apache 2.0         | <a href="https://github.com/apache/singa">https://github.com/apache/singa</a>   |
| Apache Mahou                   | Distributed linear algebra framework            | Apache 2.0         | <a href="https://github.com/apache/mahout">https://github.com/apache/mahout</a>   |
| Apache Spark                   | Analytics engine                                | Apache 2.0         | <a href="https://github.com/apache/spark">https://github.com/apache/spark</a>   |
| Apache MXNet                   | DL framework                                    | Apache 2.0         | <a href="https://github.com/apache/incubator-mxnet">https://github.com/apache/incubator-mxnet</a>                                       |
| Apache PredictionIO            | ML server                                       | Apache 2.0         | <a href="https://github.com/apache/predictionio">https://github.com/apache/predictionio</a>   |
| Apache SystemDS                | ML system for end-to-end data science lifecycle | Apache 2.0         | <a href="https://github.com/apache/systemds">https://github.com/apache/systemds</a>   |
| BERT                           | Pre-trained language model(s)                   | Apache 2.0         | <a href="https://github.com/google-research/bert">https://github.com/google-research/bert</a>   |
| CatBoost                       | ML Method                                       | Apache 2.0         | <a href="https://github.com/catboost/catboost">https://github.com/catboost/catboost</a>   |
| Caffe                          | DL Framework                                    | BSD-2              | <a href="https://github.com/BVLC/caffe">https://github.com/BVLC/caffe</a>   |
| CLIP                           | Trained neural network                          | MIT                | <a href="https://github.com/openai/CLIP">https://github.com/openai/CLIP</a>   |
| Dagli                          | ML Framework                                    | BSD-2              | <a href="https://github.com/linkedin/dagli">https://github.com/linkedin/dagli</a>   |
| DeepDetect                     | ML API and server                               | GPL3               | <a href="https://github.com/jolibrain/deepdetect">https://github.com/jolibrain/deepdetect</a>   |
| DeepLearning4J                 | DL framework                                    | Apache 2.0         | <a href="https://github.com/eclipse/deeplearning4j">https://github.com/eclipse/deeplearning4j</a>                                       |
| DeepMind Lab2D                 | 2D platform for ML                              | Apache 2.0         | <a href="https://github.com/deepmind/lab2d">https://github.com/deepmind/lab2d</a>   |
| Delta                          | DL language/speech processing platform          | Apache 2.0         | <a href="https://github.com/Delta-ML/delta">https://github.com/Delta-ML/delta</a>   |
| Determined                     | DL training platform                            | Apache 2.0         | <a href="https://github.com/determined-ai/determined">https://github.com/determined-ai/determined</a>                                   |
| Egeria                         | Metadata and governance framework               | Apache 2.0         | <a href="https://github.com/odpi/egeria">https://github.com/odpi/egeria</a>   |

|                             |  |            |   |
|-----------------------------|--|------------|---|
| Elastic Deep Learning       | Cloud training and inference of DL models        | Apache 2.0 | <a href="https://github.com/elasticdeeplearning/edl">https://github.com/elasticdeeplearning/edl</a>                           |
| Fair Learn                  | Python toolkit for AI fairness assessment        | MIT        | <a href="https://github.com/fairlearn/fairlearn">https://github.com/fairlearn/fairlearn</a>                                   |
| Fairseq                     | Sequence modelling toolkit                       | MIT        | <a href="https://github.com/pytorch/fairseq">https://github.com/pytorch/fairseq</a>   |
| Feast                       | Feature store for ML                             | Apache 2.0 | <a href="https://github.com/feast-dev/feast">https://github.com/feast-dev/feast</a>   |
| ForestFlow                  | ML model server                                  | Apache 2.0 | <a href="https://github.com/ForestFlow/ForestFlow">https://github.com/ForestFlow/ForestFlow</a>                               |
| Gym                         | Reinforcement learning Python library            | MIT        | <a href="https://github.com/openai/gym">https://github.com/openai/gym</a>   |
| Horovod                     | DL training framework                            | Apache 2.0 | <a href="https://github.com/horovod/horovod">https://github.com/horovod/horovod</a>   |
| H2O                         | In-memory ML platform                            | Apache 2.0 | <a href="https://github.com/h2oai/h2o-3">https://github.com/h2oai/h2o-3</a>   |
| Keras                       | DL API   | Apache 2.0 | <a href="https://github.com/keras-team/keras/blob/master/LICENSE">https://github.com/keras-team/keras/blob/master/LICENSE</a> |
| Klio                        | Audio data pipelines                             | Apache 2.0 | <a href="https://github.com/spotify/klio">https://github.com/spotify/klio</a>   |
| KNIME Analytics Platform    | Data analytics platform                          | GPL3       | <a href="https://www.knime.com/knime-open-source-story">https://www.knime.com/knime-open-source-story</a>                     |
| Kubeflow                    | ML toolkit                                       | Apache 2.0 | <a href="https://github.com/kubeflow/kubeflow">https://github.com/kubeflow/kubeflow</a>                                       |
| LinkedIn Fairness Toolkit   | Fairness measurement and bias mitigation library | BSD2       | <a href="https://github.com/linkedin/LiFT">https://github.com/linkedin/LiFT</a>   |
| Ludwig                      | DL framework                                     | Apache 2.0 | <a href="https://github.com/ludwig-ai/ludwig">https://github.com/ludwig-ai/ludwig</a>   |
| Marquez                     | Metadata service                                 | Apache 2.0 | <a href="https://github.com/MarquezProject/marquez">https://github.com/MarquezProject/marquez</a>                             |
| Microsoft Cognitive Toolkit | DL Framework                                     | MIT        | <a href="https://github.com/microsoft/CNTK">https://github.com/microsoft/CNTK</a>   |
| Milvus                      | Vector database                                  | Apache 2.0 | <a href="https://github.com/milvus-io/milvus/">https://github.com/milvus-io/milvus/</a>                                       |
| ML Agents                   | ML agents toolkit                                | Apache 2.0 | <a href="https://github.com/Unity-Technologies/ml-agents">https://github.com/Unity-Technologies/ml-agents</a>                 |
| ML Flow                     | ML dvp platform                                  | Apache 2.0 | <a href="https://github.com/mlflow/mlflow/">https://github.com/mlflow/mlflow/</a>   |
| ML Kit samples              | Code samples                                     | Apache 2.0 | <a href="https://developers.google.com/ml-kit/guides">https://developers.google.com/ml-kit/guides</a>                         |
| Monai                       | Healthcare DL framework                          | Apache 2.0 | <a href="https://github.com/Project-MONAI/MONAI">https://github.com/Project-MONAI/MONAI</a>                                   |
| Neuropod                    | Interface library                                | Apache 2.0 | <a href="https://github.com/uber/neuropod">https://github.com/uber/neuropod</a>   |
| NNStreamer                  | Neural network streamer                          | LGPL2.1    | <a href="https://github.com/nstreamer/nstreamer">https://github.com/nstreamer/nstreamer</a>                                   |
| ONNX                        | Software format for AI models                    | Apache 2.0 | <a href="https://github.com/onnx/onnx">https://github.com/onnx/onnx</a>   |
| Opacus                      | ML training library                              | Apache 2.0 | <a href="https://github.com/pytorch/opacus">https://github.com/pytorch/opacus</a>   |
| Paddle Paddle               | DL Framework                                     | Apache 2.0 | <a href="https://github.com/PaddlePaddle/Paddle">https://github.com/PaddlePaddle/Paddle</a>                                   |
| ParlAI                      | Model testing framework                          | MIT        | <a href="https://github.com/facebookresearch/ParlAI">https://github.com/facebookresearch/ParlAI</a>                           |

|                           |   |            |   |
|---------------------------|---|------------|---|
| Pyro                      | Probabilistic programming language                | Apache 2.0 | <a href="https://pyro.ai">https://pyro.ai</a>   |
| OpenAI Baselines          | Reinforcement learning implementations            | MIT        | <a href="https://github.com/openai/baselines">https://github.com/openai/baselines</a>   |
| Scikit Learn              | ML Python module                                  | BSD3       | <a href="https://github.com/scikit-learn/scikit-learn">https://github.com/scikit-learn/scikit-learn</a>                         |
| Sparklyr                  | Scale interface for data science and ML workflows | Apache 2.0 | <a href="https://github.com/sparklyr/sparklyr">https://github.com/sparklyr/sparklyr</a>   |
| Streamlit                 | Datascience and ML app framework                  | Apache 2.0 | <a href="https://github.com/streamlit/streamlit">https://github.com/streamlit/streamlit</a>                                     |
| TensorFlow                | ML framework                                      | Apache 2.0 | <a href="https://github.com/tensorflow/tensorflow">https://github.com/tensorflow/tensorflow</a>                                 |
| TensorLy                  | Tensor Python library                             | BSD3       | <a href="https://github.com/tensorly/tensorly">https://github.com/tensorly/tensorly</a>   |
| Torch                     | ML library  | BSD3       | <a href="http://torch.ch">http://torch.ch</a>   |
| Zero-shot Object Tracking | Object tracking implementation                    | GPL3       | <a href="https://github.com/roboflow-ai/zero-shot-object-tracking">https://github.com/roboflow-ai/zero-shot-object-tracking</a> |