Google's expansion into security

Platforms, infrastructure, (in)visibility

van Hardeveld, A.A.

Publication date
2025

CHAPTER 3

Machine Learning and the Platformization of the Military: A Study of Google's Machine Learning Platform TensorFlow

This chapter is co-authored by Marijn Hoijtink. A version of this chapter is published as: Hoijtink, M., & Planqué-van Hardeveld, A. (2022). Machine Learning and the Platformization of the Military: A Study of Google's Machine Learning Platform TensorFlow. *International Political Sociology*, *16*(2), 1-19. https://doi.org/10.1093/ips/olab036.
Both authors contributed equally to the article.

Introduction

In 2017, the US Department of Defense (DoD) announced the established the Algorithmic Warfare Cross-Functional Team, code-named Project Maven, "to accelerate DoD's integration of big data and machine learning" (US Deputy Secretary of Defense, 2017).The work of Project Maven focused on the development of machine learning (ML) algorithms to automate the extraction of "objects of interests" across US surveillance footage (Pellerin, 2017a). In the summer of 2018, Project Maven became a site of public controversy when it was revealed that the US DoD was using Google's ML platform TensorFlow, based on deep learning, to help military analysts detect objects in images. A Google spokesperson defended Project Maven stating that "[t]his specific project is a pilot with the Department of Defense, to provide open source TensorFlow APIs [Application Programming Interfaces] that can assist in object recognition on unclassified data", and that is "for non-offensive uses only" (Conger & Cameron, 2018). However, a petition against Google's involvement in the business of war, signed by over 4,000 Google employees, pointed out that "the technology is being built for the military, and once its delivered it could easily be used to assist in these [offensive] tasks" (Google Employees, 2018).

The case of Project Maven is exemplary for how the military is increasingly experimenting with ML-based technologies for military decision-making. Understood as "a branch of artificial intelligence that allows computer systems to learn directly from examples, data, and experience" (The Royal Society, 2017, p. 5), ML is increasingly deployed by the military to determine unknown patterns and relations across large data sets, make recommendations or predictions, and accelerate decisionmaking (e.g. NATO, 2019). Relevant applications that already play a significant role in the decision to use lethal force include pattern recognition, natural language processing, sentiment analysis and facial and image recognition (ICRC, 2019). Such applications can inform a wide range of decisions, from lethal decisions about who or what to attack and when (such as in the case of Project Maven), to decisions about military strategy and specific operations(International Committee of the Red Cross, 2019). The implications for civilian populations in armed conflict are equally broad, as these technologies are connected to the spread of digital surveillance, push asymmetry on the battlefield, and constitute and justify their own form of violence (Bellanova et al., 2021).

How can we account for the growing role of ML in the military domain and the (violent) effects that it bears? A growing body of scholarship within critical security studies has begun to explore these questions from the perspective of "algorithmic warfare" (Amoore, 2009a; Amoore & Raley, 2017; Bellanova et al., 2021; Suchman, 2020; Wilcox, 2017). For example, Louise Amoore and Rita Raley (2017, p. 6) argue that although there is a longer history of the deployment of algorithmic techniques in the context of war, "the historical significance of the increased use of generative machine learning algorithms cannot be overstated". According to them, ML algorithms are constitutive of a distinct security logic through which threats and targets are abductively generated "via the recognition of patterns in vast volumes of data" (Amoore & Raley, 2017, p. 6). This way, advances in ML and neural network computation are radically reshaping "existing practices of knowledge production, decisionmaking and securitization" (Amoore & Raley, 2017, p. 8). In her discussion of algorithmic warfare, Lauren Wilcox (2017, p. 15) attends to the algorithmic pattern-of-life analysis that enables the production of "targetable, killable bodies" in drone assemblages. At the same time, she highlights that algorithmic targeting cannot be separated from gendered and racialized assumptions about who constitutes a target in the first place. In a similar argument, Lucy Suchman (2020, p. 176) argues that current investments in algorithmic decisionmaking should be understood as a recent illustration of the recurring promise of a technological solution to the military problem of "situational awareness". However, as Suchman points out, despite claims that algorithmic warfare enhances accuracy, there is no technological solution to this problem. Instead, claims about accuracy and precision in algorithmic warfare obscure and conflate "the difference between the precision with which a weapon, once fired, will strike its target, and the acts of identification of legitimate threats that targeting presupposes" (Suchman, 2020, p. 183).

Our chapter builds on these contributions and the political-ethical questions raised by this literature, but also suggests that we need to shift the focus of our attention away from ML algorithms, and toward the broader technical-material infrastructure that drives algorithmic warfare, which is materialized by the ML *platform*. A ML platform is a programmable, modular ecosystem of tools, libraries and community resources that mobilize pre-existing ML expertise, and that provides the necessary infrastructure to code and deploy ML models. These models are what powers specific ML applications that are used in real world situations. Put differently, "[t]he model is the "thing"

that is saved after running a machine learning algorithm on training data and represents the rules, numbers, and any other algorithm-specific data structures required to make predictions" (Brownlee, 2020). ML algorithms are important in the creation of ML models, but it is the model that gets implemented to make predictions about future data. We thus propose a focus on the ML platform, both as a space of experimentation with ML algorithms needed for the development of ML models and as the infrastructure needed to deploy these models into real world applications.

Our analytical focus on the ML platform enables us to make two broader arguments. First, it allows us to shed light on ongoing efforts across the military to build platforms that enable new practices of decentralized and experimental algorithm development. We conceptualize these developments as part of a broader "platformization" of the military, with which we refer to the rise and permeation of the ML platform as the infrastructure for doing, facilitating, and experimenting with machine learning for decisionmaking (following Helmond, 2015; van Dijck et al., 2019). Second, a focus on the ML platform allows us to draw attention to the deepening relationship between the military and the tech sector, taking place against the background of this platformization. Specifically, we are highlighting two elements of these deepening military-technology assemblages (Gilbert, 2019). On the one hand, we focus on the growing role of Big Tech in building scalable ML platforms and providing platform expertise. On the other hand, we point out how – through open-source ML platforms provided by Big Tech – an open-source community of software developers becomes implicated in the making of military technology. Ultimately, through a focus on platforms, we explain how practices of algorithmic warfare are pulled closer toward the center of military organizations, but also proliferate 'outwards', across units, forces, and battlefields.

In conceptualizing the role and implications of platforms and the platformization of the military, we build on the burgeoning platform literature within new media studies and bring this body of literature to critical security studies. Specifically, we are taking inspiration from scholarship within new media studies that has focused on the computational aspects of platforms(e.g. Bogost & Montfort, 2009; Langlois & Elmer, 2019; Mackenzie, 2018), adopting a "material-technical perspective" on platforms that highlights their programmability and modularity in relation to their political, economic and infrastructural expansion (Helmond, 2015, p. 2). In developing such a perspective, this does not mean that we take for granted the wider imaginaries

associated with the concept of "the platform" and the rhetorical work that platforms do (e.g. Gillespie, 2010). To the contrary, we find that a focus on the performative effects of the rhetoric around platforms remains key, specifically within the field of security, where the notion of the platform – as it refers to weapon systems – bears a strong resonance. At the same time, we argue that those imaginaries should be situated in relation to the actual practices and technologies of machine learning (see also Elish & boyd, 2018). In focusing on those, we side with recent scholarship within critical security studies that has called for a fine-grained focus on what technologies "actually *do*" and on how technological affordances affect security practices (Bellanova et al., 2021, p. 125 emphasis in original, 2020; Hoijtink & Leese, 2019).

Empirically, our paper draws on a close study of Google's ML platform TensorFlow, which is one of the world's most widely-used open-source ML platforms. TensorFlow provides us with a relevant case for two reasons. First, while the platforms that the military is developing are often surrounded by secrecy and difficult to access, there is a lot more public information available about commercial platforms, such as TensorFlow, and their technical architecture and specificities. In studying TensorFlow, we build on a close analysis of different technical sources, including the TensorFlow Guide, Glossary, and Blog, video recordings of TensorFlow conferences, and TensorFlow tutorials. Additionally, we followed the Coursera course "AI for everyone" by Andrew Ng, an expert in the field of ML and founder of Google Brain.

Second, the case of TensorFlow is specifically relevant, because of how it served as a model that the US military has sought to replicate. While to date, actual details about the role of Google and TensorFlow in Project Maven remain unclear, what we *can* trace is the wider performative role that TensorFlow has had across the US military. As we argue in our analysis, Project Maven and the deployment TensorFlow should be considered as a decisive moment for the broader platformization of the military that we describe, as well as for the growing involvement of Big Tech and their platforms within the military domain. This is not to say that the relationship between the US DoD and Google/TensorFlow is seamless. If anything, the public controversy surrounding Google's involvement in Project Maven and Google's subsequent decision not to renew its contract illustrate how Big Tech need to carefully navigate between their role as providers of digital services and platforms, and their emerging role in the military domain.

In tracing the emergence of ML platforms in the military domain, we combine a detailed study of TensorFlow with a broader analysis of a variety of publicly available, defense-related sources, including official policy documents, speeches, blogs, defense magazine articles, project descriptions, company websites and descriptions, and newspaper articles. Our main focus is on the US, which is the main driver of the platformization of the military domain. Our argument develops as follow. First, we discuss the relevant literature on platforms within the field of new media studies, focusing in particular on those contributions that have highlighted the technical-material "work that platforms do"(Helmond, 2015, p. 2). Next, taking up this perspective, we explore TensorFlow and the level of control that Google has over this open-source platform. Third, we link these insights to the emerging role of ML platforms within the military domain, exploring the contours of "platform war" and the defining role that Big Tech companies play in this context. We conclude by summarizing what is politically at stake in this platformization of the military.

Platforms and platformization

This paper builds on a growing body of scholarship at the intersection of critical security studies and Science and Technology Studies (STS) that focuses on the role of digital technologies and algorithmic decisionmaking in security and warfare (Amoore, 2009a; Aradau & Blanke, 2015; O'Grady, 2021; Suchman, 2020; Suchman et al., 2017). In recent years, this literature has taken a specific interest in conceptualizing the wider socio-material context in which digital technologies emerge and get to shape and temporally stabilize security practices. This has led to a focus on the study of security devices (Amicelle et al., 2015), algorithms (Amoore & Raley, 2017), infrastructures (Aradau, 2010; Bellanova & de Goede, 2022b) and technological and material agency (Hoijtink & Leese, 2019; Salter, 2015). To date, the study of platforms has been largely absent from this literature.[5] This absence is somewhat remarkable given the prominence of platforms in related disciplines, but also because the concept of the platform traditionally bears a strong resonance within International Relations (IR) and the subfields of security and strategic studies. Indeed, the study of military platforms (i.e. weapons systems) has been a key concern within this literature, specifically within rationalists

5 Relevant exceptions are Aradau et al. (2019), Egbert (2019), Lagerwaard (2020), and Westermeier (2020).

approaches, where platforms are primarily discussed as determinants of state power.

Our understanding of platforms is informed by new media studies, where the concept of the platform has gained prominence to draw attention to the ways in which digital platforms – and social media platforms in particular – are profoundly transforming how "societies are organized and publics are shaped" (van Dijck & Poell, 2015). From a more narrow perspective, the (social media) platform can be defined as "a programmable digital architecture designed to organize interactions between users", which is based on "the systematic collection, algorithmic processing, circulation, and monetization of data" (van Dijck et al., 2018, p. 4). As such, the concept of the platform carries particular social, cultural and political meaning. In one of the most central discussions on platforms, Tarleton Gillespie (2010) argues that it is precisely this ambiguous meaning that makes the concept of the platform politically relevant as it allows companies, such as YouTube, to stage themselves as neutral intermediaries, while navigating conflicting interests between users, content developers and advertisers. This allows these companies to maintain "the fantasy of a truly 'open' platform", even if all platforms "must, in some form of another, moderate" and always steer social activity in certain directions (Gillespie, 2018, p. 5).

While Gillespie highlights the discursive work that is performed by the concept of the platform, another relevant strand of platform literature has adopted a more techno-economic outlook on platforms. José Van Dijck (2013, p. 24) argues that platforms can be taken apart "into their techno-cultural components (coding technologies, users, and content) and their socio-economic elements (ownership status, governance, business models)". Similarly, for Paul Langley and Andrew Leyshon (2017, p. 19), platforms are "particular comings together of code and commerce" that are geared toward the accumulation of data and the coordination of network effects. Understood as, at once, a sociotechnical construct and a business model, the platform has been put at the center of broader transformations in the global economy, such as the rise of "platform capitalism" (Pasquale, 2016; Srnicek, 2017), "platform surveillance" (Wood & Monahan, 2019; cf. Zuboff, 2019b), and the "platform society" (van Dijck et al., 2018). Central to all these contributions is a focus on the pivotal role of Big Tech that develop and own these social platforms, and, increasingly, on the ways in which these platforms operate as part of a wider "platform ecosystem"(Andersson Schwarz, 2017; van Dijck, 2021).

This brings us to a third body of literature on platforms, which highlights their technical-material dimensions and zooms in on how digital platforms work technically, which warrants a more computational perspective of the platform as put forward by the field of "platform studies"(e.g. Baldwin & Woodard, 2009; Bogost & Montfort, 2009; Langlois & Elmer, 2013). According to social media scholar Anne Helmond (Helmond, 2015), a defining feature of digital platforms is their *programmability,* which allows platforms to be "adapted to countless needs and niches that the platform's original developers could not have possibly contemplated" (Andreessen, 2007).

Inquiring into the programmability of social media platforms, Helmond highlights the role of Application Programming Interfaces (APIs), which can be defined as "a set of rules or protocols that enables software applications to communicate with each other to exchange data, features and functionality"(Goodwin, 2024). A key element of APIs is that third-party developers can interact with the interface to gain access to the data and functionality without having to know or even understand the underlying process. For Helmond (2015, p. 4), APIs are what "turns it [a website] into a platform that others can build on" and what allows third-party developers to build on the platform's data and functionalities. APIs, however, function as a two-way street. As explained by Jean-Christophe Plantin, Carl Lagoze and Paul Edwards (2018, p. 10) "APIs constitute specific constraints (defined by the platform that creates and controls them), which determine who can access data, in which forms, and under which conditions". For example, in the context of Facebook, the APIs, social plug- ins, and Open Graph enable Facebook to make external web data compatible with its existing platform features and "platform ready" so that data can be collected (Helmond, 2015, p. 1). Plantin, Lagoze and Ewards (2018) hence conceptualize APIs as "a technology of governance"(following Bucher, 2013).

Another defining feature of platforms is their *modularity*. Platforms consist of core and complementary components or modules, which function as ready-to-use, interoperable building blocks that users can add, take out, and/or replace to serve their objectives (Baldwin & Woodard, 2009). Taken together, the platform's programmability and modularity are "enabling the *decentralization* of platform functionality and data produced on the platform and the *recentralization* of data produced outside of the platform" (Helmond, 2015, p. 8 emphasis added). According to Helmond (2015, p. 1), this dynamic between decentralization and recentralization is what is driving "the

platformization of the web", with which she refers to the extension of social media platforms into the rest of the web and the broader rise of the platform "as the dominant infrastructural and economic model of the social web". It is this dual logic of platformization that connects the study of platforms to the study of infrastructures and processes of infrastructuralization. In a recent contribution, Jean-Christophe Plantin and his colleagues argue that digital platforms are increasingly entangled with infrastructures (Plantin et al., 2018). In addition to a "'platformization' of infrastructure", we are witnessing an "'infrastructuralization' of platforms" (Plantin et al., 2018, p. 295). This notion bears witness to the ways in which digital platforms are increasingly operating as infrastructures and have acquired properties that are generally considered to be typical of infrastructures, such as scale, ubiquity and temporality. For example, Helmond, Nieborg and van der Vlist (2019, p. 141) draw out how Facebook has gradually expanded from a social networking site to a "platform-as-infrastructure", gaining infrastructural properties over time by "embedding itself in other markets and industries to render technical and business operational more widely and immediately available". For them, and other scholars in new media studies who are emphasizing the infrastructural properties of platforms, the concept of the platform *as* infrastructure warrants a closer focus on the political economy of social media platforms as connected to their technical features.

Bringing this body of scholarship on platforms to critical security studies, we are adopting a similar emphasis on the platform architecture and technical properties of platforms in relation to their political, economic and infrastructural expansion. In so doing, we are advancing the notion of a "platformization" of the military, which we broadly define as the rise and permeation of the ML platform as the technical infrastructure for doing, facilitating, and experimenting with ML for decisionmaking (following Helmond, 2015; Poell et al., 2019). This definition takes into account digital platforms and their programmability and modularity, but also allows us to look at the broader discursive and material infrastructures that support these developments, which includes a focus on the pivotal role of Big Tech corporations, in our case Google.

Our argument diverges from the existing platform literature in the sense that we analyze a different type of platform than usually considered by this literature. To date, the platform literature has primarily focused on the bigger social media platforms and their infrastructural expansion, and, to a lesser

extent, on “sectoral” or “connective” platforms (some of which owned or partly owned by Big Tech), such as Uber or Airbnb (van Dijck et al., 2018). While TensorFlow relies on dynamics of (de)centralization that are typical of social media platforms, as well as on an API-based architecture, there is a significant difference between ML platforms and social media platforms. The latter has a clear commercial objective, which it achieves by organizing exchanges between users, content providers, and developers/advertisers (Rieder & Sire, 2014). ML platforms bring open-source machine learning expertise and work (in the form of layers, models, labeled datasets etc.), developed by researchers across the world, to others who have less resources or who are looking for a convenient platform to build and deploy their ML application, *free of charge*. Examining ML platforms therefore warrants a specific focus on practices of data sharing, code writing, building of layers etc., and the cultivation of an open-source community of ML developers. A focus on ML platforms also highlights how (in this case) Google has been investing in new ways to build up expertise and technical capabilities in the area of ML, thereby “remaking itself as ‘machine learning first’ company” (Steven, 2016).

A study of machine learning platforms: the case of TensorFlow

This section provides a detailed analysis of the example of TensorFlow to understand the specific technological architecture of ML platforms. TensorFlow, which replaced DistBelief as Google’s “internal deep learning infrastructure”, was initially released by the Google Brain Team in 2015 (Dean & Monga, 2015). In short, TensorFlow provides users with a programmable and modular platform that enables them to build ML models for a range of applications, including voice and sound recognition, text analysis, image recognition, time series algorithms and video detection. TensorFlow offers pre-labelled training datasets, pretrained ML layers and ML models, as well as access to a long list of external code libraries that are made accessible through various APIs such as the Keras API. As explained in the above, APIs function as an “an intermediary between two disconnected applications” (Goodwin, 2024) which allows users to code on a higher level of abstraction; the underlying code that is needed to instruct the program how to execute that layer is already embedded in the platform.

This way, TensorFlow provides the necessary infrastructure to efficiently code and develop ML models. It all starts with the availability of labeled training data, which is the first stumbling block that many (including the US military) face when they want to adopt ML. While the military might have access to significant amounts of decentralized and unlabeled datasets (e.g. satellite images or video streams from drones), for a supervised ML model to categorize new inputs correctly, it needs to be trained on a centralized and carefully labeled dataset. The collection, assembling, and then labeling of training datasets, however, is very labor intensive. While users can use their own datasets, TensorFlow allows them to access public pre-assembled and labeled datasets through its API, some of which are published by Google. This means that users can skip these steps of assembling and labeling data and do not have to write their own code "to download and prepare every dataset they work with, which all have different source formats and complexities" (TensorFlow, 2019).

TensorFlow, however, is more than a data management platform. By publishing pre-made machine learning layers, pre-trained models, and by integrating APIs that make the process of coding quicker and easier, TensorFlow makes ML less resource-dependent and labor intensive. Traditionally, it has been within well-funded research communities and corporations that ML was first explored. This is because, in addition to creating training datasets, the training of models demands substantial amounts of computing power and time. As illustrated by the TensorFlow team, Google Brain's NASNet, a model for large scale image classification and object detection "took thousands of GPU [graphic processing unit] hours to train", which translates into months of work depending on the amount of GPU's one has available (Gordon, 2018). To put this into context, the cost a of a single GPU ranges from a thousand to ten thousand euros, depending on the level of optimization for machine learning. Furthermore, realistically you need a cluster of GPUs to train such a model as NASNet, which in turn has significant "space, power and cooling needs" (Gupta, 2013). TensorFlow removes the need to train ML models from scratch. Instead, it allows users to access tested ML layers and models made by other researchers through APIs. And because TensorFlow is fully modular, users can decide how to mix and combine these building blocks to serve their specific objectives.

TensorFlow also simplifies and optimizes the process of putting ML models into production by integrating these into an application or system – which to a large extent explains its popularity. For example, TensorFlow includes various ways to optimize the ML model and to scale the model down so that it can run on mobile devices that are limited in processing power, battery life and memory. It has ready-made optimizers in place that smooth out and shorten the code of the model to minimize the amount of latency. TensorFlow also assists users in scaling up their model, for instance by having optimizers available that will tell the model how to make the most efficient use of multiple GPUs, so that these can distribute the workload.

To advance these optimization efforts, TensorFlow has developed a range of external complementary components. These include visualization tools, such as TensorBoard and TensorFlow Playground. TensorFlow has also integrated so-called fairness and privacy optimizers, which promise a technical "solution" to bias and questions of privacy. To implement these optimizers correctly, supposedly, "no expertise in privacy or its underlying mathematics should be required [...]; it is often sufficient for you to make some simple code changes and tune the hyperparameters relevant to privacy" (Radebaugh & Erlingsson, 2019). If more help is needed, users can rely on TensorFlow Extended, a paid service which assists businesses in building and managing an automated ML pipeline. More generally, the TensorFlow team regularly publishes a variety of educational resources, such as user guides, presentation videos, and tutorials, which are all meant to invite new users into the ML community. All of this sits nicely with the mission of Google's CEO Sundar Pichai "to *democratize access to the technology* [AI]" (2017).

Google's ambition to democratize ML via TensorFlow links up with its aim to foster – and tap into – an open-source community of developers that share data sets, models and code. Both TensorFlow and the tools that surround it are open source, released under the Apache License 2.0. This license holds that users can download TensorFlow to locally develop ML models and that they can tweak the source code of the platform as they see fit without exchanging data with Google. Users can also decide to run the platform directly in their browser, making use of TensorFlow Colab. This allows them to use Google's TPU (tensor processing unit), which is designed to train models quickly and which runs various major Google products. However, this also means that users automatically share their data with Google as TensorFlow Colab runs on the company's servers. Another option is to use TensorFlow Research

Cloud, which allows selected researchers to use a cluster of a thousand of Google's Cloud TPU's (GPU's optimized for ML) (Stone, 2017). The sharing of code, however, is a requirement to use this service, which means that users will return code back to Google.

Users of TensorFlow are also encouraged to contribute to the platform by sending in so-called pull requests if they think of code changes that should be included in the TensorFlow code-base (the main repository). Before users can contribute actual code, they first need to sign the Google Individual Contributor License Agreement to show that they understand that:

> this project and contributions to it are public and that a record of the contribution (including all personal information [they] submit with it, including [their] full name and email address) is maintained indefinitely and may be redistributed consistent with this project, compliance with the open-source license(s) involved, and maintenance of authorship attribution. (Google Open Source, n.d.)

Furthermore, contributors need to agree with the TensorFlow Code of Conduct and their work is reviewed by a maintainer, contributors who are elevated to maintainer status "on evidence of previous contributions and established trust" (TensorFlow, n.d.-a). If the project users want to contribute to is initiated by Google, a Google employee will be in charge of the review process. And if users want to submit a new general TensorFlow feature, they need to go through a Request for Comment (RFC) process, led by Google developers. In view also of the fact that the TensorFlow team continues to invest in TensorFlow by pushing new versions of the platform (the current version is v.2.5.0.), we can see how Google continues to exercise a certain level of control over TensorFlow, while cultivating an open-source community of ML developers.

Building on this open-source community of ML developers via TensorFlow is one way in which Google "expands itself infrastructurally" (Nieborg & Helmond, 2019, p. 197), and reaches out to new customers. TensorFlow is powering major Google services such as Google Search, Translate and Gmail, but it has also been the preferred platform for a range of other companies and services. For example, PayPal, uses TensorFlow for fraud detection. Airbnb deploys TensorFlow to automatically categorize listing photos and GE Healthcare uses TensorFlow to improve MRI scans of the brain. Other users

of TensorFlow include Twitter, Swisscom, Spotify, Qualcomm, Lenovo, Intel, China Mobile, Airbus, and, as mentioned, the US DoD.

Finally, by relying on open-source ML development, this also allows Google to establish itself as a key player in the rise of deep learning, a specific way of doing ML facilitated by TensorFlow. Models built through deep learning, also referred to as neural networks, are made up out of a composition of multiple hidden layers that all extract particular parameters from the dataset to link the input (the data) to the output (the prediction): this is how the model learns. The development of deep learning models normally requires a significant investment in terms of data collection, computing power and training hours so that multiple hidden layers can learn from the data, especially when faced with complex problems (Shead, 2020). However, TensorFlow allows users to build on the work of others who have invested in the development of the various ML layers that are necessary to create a deep learning model. In fact, its entire programmable and modular architecture is aimed at developing multi-layered neural networks, with its main API, Keras, being a deep learning code library.

Yet, deep learning is a contested approach to ML. The core of this contestation revolves around the multiple hidden layers that make deep learning what it is: the more layers, the "deeper" the network is and the harder it is to reconstruct what exactly the model has learned about the data. While TensorFlow has introduced various tools to account for the possible privacy and fairness issues of ML, the specific limitations of deep learning are hardly addressed. And when they are addressed, they are often presented as technical problems for which ready-made technical fixes exists. For instance, in one of the TensorFlow Core tutorials, it is acknowledged that in situations when "AI applications [...] are safety-critical (e.g., medical decision making and autonomous driving) or where the data is inherently noisy (e.g., natural language understanding)" deep learning might have certain limitations (TensorFlow, n.d.-b). However, a technical solution quickly follows: "[t]he deep classifier should be able to be aware of its own limitations and when it should hand control over to the human experts" which is based on a quantification of the model's uncertainty (TensorFlow, n.d.-b).

This technical exploration of TensorFlow as an example of ML platforms has brought forward three main insights. First, we can see how, despite the (depoliticized) imaginary of an open-source platform as a glass box,

ML platforms are far from neutral tools. TensorFlow facilitates users with a space for experimentation with ML, thereby steering ML developments and the imaginaries that surround it. Second, our discussion illustrates how TensorFlow supports Google in positioning itself as a "machine learning first" company and in its infrastructural expansion in the field of AI. Third, studying TensorFlow has shown us what the development of a ML model requires and why, as well as some of the challenges that users face when deploying the model into their application. These insights will now help us to contextualize the growing role of ML in the military domain more generally, and understand the challenges faced by the US military during Project Maven more specifically.

Project Maven and the rise of platform war

In April 2017, then US Deputy Secretary of Defense Robert Work announced the establishment of Project Maven. The objective of Project Maven was to "turn the enormous volume of data available to the DoD into actionable intelligence and insights at speed" (US Deputy Secretary of Defense, 2017), using deep learning and computer vision algorithms to "detect, classify, and track objects within Full Motion Video (FMV) images" (Office of the Secretary of Defense, 2019). To date, exact details about the categories under analysis and the technologies of data analysis used in Project Maven are absent. As Suchman (2020, p. 182) writes, the case of Project Maven "overwhelmingly begs the question of the criteria by which 'objects' are identified as imminent threats". Similarly, further details about Google's role in Project Maven have not been disclosed. A news story by *The Verge* suggests that Google did not only offer TensorFlow ML software, but that it was also actively configuring TensorFlow for use in Project Maven. However, whether this means that Google was "simply walking DoD researchers through TensorFlow's manual" or that Google engineers were "being paid to develop algorithms using drone video" is unclear (Vincent, 2018).

Irrespective of these unknowns, what *can* be traced are the wider imaginaries associated with Project Maven and the performative role that it served within the US military. From the very beginning, US DoD officials emphasized that Project Maven served an important purpose beyond the development of computer vision algorithms, because it was supposed to accelerate (and justify) further investments in AI and ML applications across the US military.

As the person formerly in charge of Project Maven, Air Force Lt. Gen. Jack Shanahan, explained, “Maven was designed to be that pilot project – that pathfinder – that spark that kindles the flame front for artificial intelligence across the rest of the department” (cited in: Corrigan, 2017).

Particularly relevant for this paper is how Project Maven also underlined the need for a common platform for ML development, ultimately giving rise to a renewed focus on the need of a common platform infrastructure, what we refer to here as the “platformization” of the military. This platformization of the military domain is rooted in the emergence of ML-based technologies in the military and their associated imaginaries of automation and precision, but it should also be linked to Project Maven and the experiences gained as part of it. More specifically, we can trace how several initial setbacks experienced in Project Maven led to the deployment of TensorFlow and a growing focus on ML platforms as the necessary infrastructure for doing ML.

Let us unpack these setbacks, starting with a problem Project Maven faced related to the quality, usability and labeling of data. While it is common among defense specialists to argue that there is a wealth of data “out there”, early experiences within Project Maven indicated that much of that data was of limited use. According to a US intelligence officer involved in intelligence, surveillance, and reconnaissance operations during the counter-ISIS campaign, “the type of data that is ‘out there’ for the Defense Department” is often “not thoroughly cleaned, organized, and wrangled [...] for a specific use case” (Cook, 2020). As an example, he mentions how, in one project, MQ-9 Reaper drone data had position data and telemetry burned into the imagery. As a consequence, the “algorithm would be continually confused by the irrelevant numbers and letters etched into the images, making the data significantly more difficult to use for AI development” (Cook, 2020).

Project Maven also exposed an issue related to data access and the lack of a centralized platform for handling that data. As another US intelligence officer writes, military data is often neither usable nor accessible as it is restricted and siloed away from other data sources (Perkins, 2020). In the case of Project Maven, data access and data sharing proved to be particularly difficult and laborious, as described by the following passage:

> Once returned to Tampa, the data passed through a laborious process. Handlers would transfer the data from hard drives to a dedicated repository platform. Reviewers would curate the contents to create suitable data sets. Handlers would again transfer the data to a separate platform. Labelers would clean and catalogue the data to feed algorithm design and development. Handlers would transfer the data back to transportable media. The data would move to a lab for algorithm development. (Clarke & Shultz, 2020)

This way, Project Maven uncovered the lack of a dedicated data management platform that could cut across classification levels. However, that realization also raised a new set of questions about the military's ability to make sense of "the explosion of data made available by such a platform" (Clarke & Shultz, 2020). ML technologies were presented as the solution here, yet the initial algorithms provided by Project Maven proved to be far from successful:

> While the AI could place a boundary box around vehicles, buildings, and people, and display them on a map, the algorithms were rudimentary with many false detections [...]. Indeed, accuracy of detections was only around 50 percent. Determining the difference between men, women, and children was difficult. (Clarke & Shultz, 2020)

Similarly, early attempts by Project Maven to retrain algorithms based on inputs provided by the US special operations forces that were experimenting with the technology were fraught with uncertainties. While the Project Maven team started out thinking they only had to deploy the best algorithms, they soon found out that they needed a centralized platform that would enable them to adjust and retrain their algorithms and that would allow them to respond to user feedback (Cook, 2020). After all, their algorithms needed to be constantly optimized and reconfigured. As Shanahan explains:

> The first time you use it [the machine learning algorithm] in Afghanistan you realize you never trained it against data that had women wearing full-length black burkas, it didn't know what those were. Interesting little problem. So you get real-world data, feed it back into the algorithms and it performs better and better. A second, third time, so on, and so on. It was all about the user defining what success looks like. (Blank, 2020)

It is in this context that Google was contracted and that TensorFlow was presented as a way of centralizing surveillance data, while also allowing for the decentralization of ML development and experimentation.

While Google announced in June 2018 that it would not renew its contract with the US DoD, the idea that a common platform was needed to engage in ML had already gained traction across the US military. In 2019, the US DoD Joint Artificial Intelligence Center (JAIC) began working on its own platform, the Joint Common Foundation, which Shanahan described as "a platform that will provide access to data, tools, environments, libraries and to other certified platforms to enable software and AI engineers to rapidly develop, evaluate, test and deploy AI-enabled solutions to warfighters" (cited in: Shanahan, 2019). This platform would "lower the barriers of entry, democratize access to data [and] eliminate duplicative efforts", and "reside on top of an enterprise cloud infrastructure" (cited in: Shanahan, 2019). Launched early in 2021, the Joint Common Foundation has been described as a "lending library" or "cloud-based development environment", which – in ways very similar to TensorFlow – brings together data sets, code libraries and other open platforms under a single infrastructure (cited in: Heckman, 2021). Likewise, the army and US Air Force are building their own platforms, respectively Coeus and Platform One. Eventually, the US DoD wants to stitch these platforms together into a "fabric of platforms" or a "platform-of-platforms" architecture (Heckman, 2021), which testifies to the broader platformization of the military.

What is this ongoing platformization about? A key justification is that such platforms will put the right tools "in the hands of AI developers" (Perkins, 2020), and allow for decentralized development and experimentation with ML and deep learning (Sherman, 2020). Similar to the imaginaries that surround TensorFlow, these platforms are presented as tools that will allow "people without deep experience in machine learning techniques and software programming to build a suite of predictive algorithms", which will, hence, "democratize AI for the US DoD" (Cisco, 2021). The decentralization of ML development is key, so the argument goes, to "dynamically retrain the algorithm and to conduct test and evaluation" in a context that is supposedly also constantly changing (Cook, 2020). A key figure that embodies these developments is the "soldier coder", "a new type of soldier, trained in software development, data science and AI, [who can] work and rework algorithms on, or very near, the frontlines" (Tucker, 2020). But that soldier coder also needs "systems that allow them to use computational tools and to constantly adjust,

or retrain, their algorithms to a changing battlefield" (Cruickshank, 2020), which the platform provides.

In summary, this section has shown how Project Maven and TensorFlow served as a model for platform development within the US military. The next section will describe how Project Maven also functioned as a vehicle for new investments in the platform and the growing role of Big Tech as platform providers or integrators in the military domain. We also point out how through the deployment of open-source ML platforms by the military, an open-source community of ML developers becomes enrolled in the development of military technology.

Platform extensions

In this section, we point out how Maven has been exemplary for the deepening relationship between the US military and Big Tech and how that relationship revolves around investments in, and competition over, the ML platform. As Emily Gilbert points out, "as militaries propel and take up new scientific innovations [...], [t]hey also propel new kinds of private-sector partnerships" (Gilbert, 2019, p. 106). Obviously, as shown by the example of Project Maven, these new partnerships are not without contestation. While in recent years, we can see a more overt commitment among Big Tech to engage in larger contract bids (some of these being highly mediatized) and to collaborate with the US military, these companies also remain deeply concerned about how their involvement is publicly perceived. In the context of Google, this has led to a specific strategy of operating indirectly through start-ups, which are less subject to public scrutiny as their services are not embedded into the fabric of our daily digital lives.

An example of such a start-up is Labelbox, which entered into a contract with the US Air Force in May 2020 (Strout, 2020). As its name suggests, Labelbox offers the military a platform that supports the labeling and managing of training data, most of which is done by human labelers. A year before it showed up on the radar of the military, Labelbox received $10m from Gradient Ventures, a venture capital fund owned by Google (Shieber, 2019). With this funding, Labelbox self-reportedly aimed to "continue to double down on bringing data labeling infrastructure to the machine learning teams with powerful automation, collaboration, and enterprise-grade features" (cited

in: Shieber, 2019). And Google's involvement in Labelbox is no exception. In fact, after Google decided that it would not extend its involvement in project Maven, *The Intercept* revealed that it "is providing financial, technological, and engineering support to a range of startups", some of which are providing ML technology to the military (Fang, 2019d). This technological and engineering support includes access to "vast swaths of training data that Google has accumulated to train their own AI systems" as well as "the opportunity to receive advanced AI trainings from Google" (Fang, 2019d).

Coming back to Project Maven, we can see that Maven has been exemplary for how Big Tech have been competing over larger defense contracts. For Google, Project Maven was supposed to secure new contracts with the US DoD and provide a competitive advantage over its competitors. While Google tried to downplay Project Maven by claiming that it was just providing open-source software and that the contract was worth "only" $9 million, internal emails published by *Gizmodo* revealed that Google's senior management expected Project Maven to grow as high as $250 million (Conger, 2018b). The leaked emails also revealed that Google considered Project Maven to be "directly related" to a much larger cloud computing contract that Google and the other Big Tech companies were competing to win (Fang, 2018). After months of protests from its employees, Google announced that it would not renew its contract for Project Maven and withdrew its bid for the Joint Enterprise Defense Infrastructure (JEDI) cloud computing contract with the US DoD. However, after the public controversy surrounding its involvement in Project Maven died down, Google quickly reasserted its eagerness to continue working with the DoD (Tucker, 2019).

Since then, Google has worked its way back into the US DoD cloud environment by securing a multimillion-dollar contract with the DoD Defense Innovation Unit (DIU), a US DoD liaison in Silicon Valley with a mandate to experiment with new technologies. As part of this contract, Google offers the DoD ML applications such as "Speech-to-Text On-Prem", which powers speech recognition models made possible by TensorFlow. This way, Google is capitalizing on the contributions made by the open-source community that surrounds the platform.

Indeed, we can see how, through the US DoD's use of TensorFlow, the open-source community of software developers that uses TensorFlow becomes enrolled in the making of military technology. This enrolment

can be indirect, such as when Google is using TensorFlow as part of other services such as Google Cloud, but there is also a direct use by the US DoD. As Chief of Strategy and Communications at JAIC, Greg Allen, points out in a Changelog podcast, the US DoD acknowledges the relevance of involvement in TensorFlow and PyTorch communities:

> [T]he basic thinking there is that open source tools are popular for a reason. They work well, they have been tested by users who have pretty significant requirements and needs, and we want to make sure that when you're developing machine learning software in the DoD environment we don't want you to have to use baby software, or software that has been largely disabled because so many of the features were not approved…So the goal very much is to give our communities of developers access to the same types of things that they would be using if they were doing so in a commercial industry environment, but doing so in a way that gives us confidence about the security of that operation. (Benson & Whitenack, n.d.)

Others within the US military are similarly calling for using open-source ML platforms like TensorFlow, in addition to the US DoD's own platforms (e.g. Cook, 2020; Cruickshank, 2020; Perkins, 2020). The availability of third-party ML platforms would enable US military inhouse AI developers, once they gain access to the right military data, to use state-of-the-art tools to analyze and use that data. While this might raise questions of security, as the fact that these ML platforms are open source could mean that adversaries can detect faults in the source code of the platform and act on these, some military officials are pointing out that the deployment of open-source platforms will instead *increase* security. Indeed, the fact that they are open source makes it unlikely that a vulnerability "would be detected in a review, yet somehow missed by millions of other users" (Perkins, 2020).

Conclusions

Our chapter has argued for an analytical focus on the ML platform to understand the emerging role of ML-based technologies within the military. Taking inspiration from scholarship within new media studies that has focused on the computational aspects of platforms, we proposed to take seriously the technical work that platforms do. Through an in-depth study of Google's

ML platform TensorFlow, we developed two broader arguments. First, we have argued that we are witnessing a platformization of the military domain with which we referred to the permeation of ML platforms as the broader infrastructure for doing and experimenting with ML for military decisionmaking. Second, we have shown how this platformization is accompanied by new connections between the military and commercial actors in the technology sector, particularly Big Tech, which rely on their platform expertise and ability to scale.

While Google defended (and tried to depoliticize) its involvement in Project Maven by claiming that it was merely providing open-source software to the US military, we suggest that there are key political-ethical developments at stake in this broader platformization of the military. Through an analytical focus on the ML platform, we can see algorithmic practices in warfare accelerate as these platforms provide the military with the infrastructure and expertise needed to design, optimize, and, most importantly, put algorithms into practice across the military. Specifically, ML platforms are pulling algorithmic practices even closer to the center of military organizations, while also allowing for their circulation outwards, across different units, forces, and battlefields. This dual move of *centralizing* and *decentralizing* algorithmic practices in warfare that platforms enable amplifies, both in space and time, the violent effects of algorithmic warfare as described by existing critical security studies literature.

We should also attend to the relationship between the military and technology corporations, which is becoming more deeply entangled. While Big Tech has traditionally taken pride in its supposed independence from public government, the platformization of the military appears to change this relationship. This is not to suggest, again, that the relationship between the military and the tech sector is seamless, but the military ambitions of Google and the other Big Tech is unmistaken and requires further analysis. Lasty, we need to draw further attention to how Google profits from the open-source community that organizes itself around TensorFlow and other digital platforms, and that makes this community, to some extent, complicit in the making of military technology. While the open-source community benefits from the conveniences and expertise that is centralized and generated through and around the platform, their involvement raises various ethical questions that need to be addressed in conversation with the ML community.