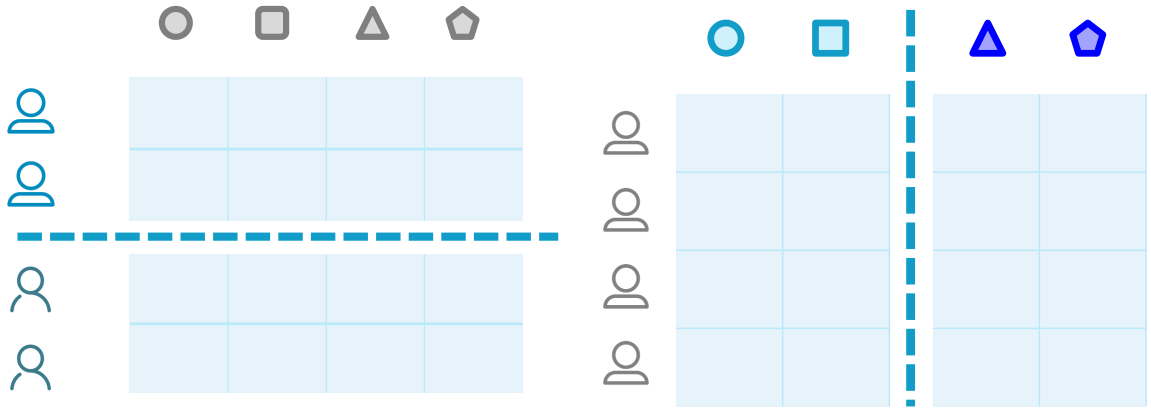
VANTAGE6: an open source priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange

Arturo Moncada-Torres, PhD[1], Frank Martin, MSc[1], Melle Sieswerda, MSc, MD[1], Johan Van Soest, PhD[2], Gijs Geleijnse, PhD[1]
[1]Netherlands Comprehensive Cancer Organization (IKNL), Eindhoven, NL;
[2]Maastricht University Medical Centre+, Maastricht, NL

Abstract *Answering many of the research questions in the field of cancer informatics requires incorporating and centralizing data that are hosted by different parties. Federated Learning (FL) has emerged as a new approach in which a global model can be generated without disclosing private patient data by keeping them at their original location. Flexible, user-friendly, and robust infrastructures are crucial for bringing FL solutions to the day-to-day work of the cancer epidemiologist. In this paper, we present an open source priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange, VANTAGE6. We provide a detailed description of its conceptual design, modular architecture, and components. We also show a few examples where VANTAGE6 has been successfully used in research on observational cancer data. Developing and deploying technology to support federated analyses – such as VANTAGE6 – will pave the way for the adoption and mainstream practice of this new approach for analyzing decentralized data.*

INTRODUCTION

Answering many of the questions in the field of cancer informatics (and in health care in general) often requires incorporating data that are located at different sources. Distinct parties could have the same features of different patients (i.e., horizontally-partitioned data, Fig. 1a) or could have different features of the same patients (i.e., vertically-partitioned data, Fig. 1b). Integrating these types of data can increase the volume of patients and the number of features available for analysis, respectively. Typically, this is done by generating a copy of each dataset at its source and delivering them to a trusted party (such as the researcher or principal investigator). The latter is responsible for storing, merging, and analyzing the combined data. However, this centralized approach is undesirable due to several organizational, operational, and political challenges[1], such as loss of data control, logistics of data transmission, and data governance. More importantly, centralizing data poses heavy concerns regarding the protection of patient data privacy. Regulatory bodies have started implementing laws ruling responsible data use and management, such as the General Data Protection Regulation (GDPR, 2018) in Europe[2,3] and the California Consumer Privacy Act (CCPA, 2020) in California, USA[3].



(a) Horizontally-partitioned data: parties have the same features of different patients

(b) Vertically-partitioned data: parties have different features of the same patients

Figure 1: Graphical representation of data partition in FL

Federated learning (FL) has emerged as a technology with the potential to overcome these limitations[4] The general idea behind it is that instead of sharing sensitive (patient) data, sites run computations on their local data, yielding either aggregated parameters or encrypted values. These are then shared to generate a global (statistical) model. This way, different parties can collaborate while making sure that the original data are kept undisclosed and safe at their original location[5–7]. In other words, instead of bringing the data to the algorithms, FL brings the algorithms to the data.

To date, FL has been used successfully in numerous cases[5,7–9]. However, in order to increase the adoption and mainstream practice of FL solutions, they required a proper infrastructure to support them. While there are a few open-source infrastructures that support FL systems, they present a few shortcomings[8,10]. For example, although DataSHIELD was developed with the needs of the cancer researcher in mind[11], it restricts him/her to a single language (R) and to a pre-defined library of functions/algorithms. Similarly, Facebook's CrypTen bounds the researcher to PyTorch (i.e., Python) and does not support all operating systems (e.g., Windows). Initiatives such as Open Mined's PySyft[12] and Google's TensorFlow Federated[13] do not support vertically-partitioned data. A robust infrastructure capable of overcoming these challenges while at the same time remaining flexible and user-friendly is a key aspect for bringing FL to real-life scenarios.

In this paper, we present our priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange – VANTAGE6, an open source platform for FL. First, we provide a description of the considerations that we followed during VANTAGE6's development, including autonomy, heterogeneity, and flexibility, the latter which allows VANTAGE6 to deal with horizontally- *and* vertically-partitioned data. Then, we describe in detail its architecture, emphasizing the structure and functionality of its building elements, which allow the users to easily implement functions or algorithms in any (open source) programming language of their choice (capable of sending an HTTP request). Afterwards, we enlist different applications where VANTAGE6 has been successfully utilized in the field of cancer informatics, including a few potential future use cases. We proceed to outline the media and outreach channels that we have established for the VANTAGE6 community. Lastly, we close the paper our overall conclusions.

CONCEPTUAL DESIGN

Before describing the architecture of VANTAGE6, we need to outline a few concepts. We define a *party* as an entity that takes part in one (or more) collaborations. We define a *collaboration* as an agreement between two or more parties to participate in a study (i.e., to answer a research question). Moreover, there are a three fundamental functional aspects of FL infrastructures that are worth describing (and that are often overlooked[8]):

Autonomy. All involved parties should remain independent and autonomous. In practice, this translates to each party being in charge of the control and management of their own data, without the need of the infrastructure itself to do so. Furthermore, each party should be able decide with how much of its data will contribute to the solution of the collaboration's global model (e.g., number of patients) and which algorithms will be allowed to be executed.

Heterogeneity. Parties should be allowed to have differences in hardware and operating systems. FL systems should also enable collaborations among parties of different nature (e.g., between a registry and a biobank or between hospitals of different countries). Not only does this diversity have the potential to enrich the data to answer the question at hand, but also allows posing and answering more distinct, interesting research questions.

Flexibility. Related to the latter, a FL infrastructure should not limit the use of relevant data. The research question might need either horizontally- or vertically-partitioned data (Fig. 1) to be answered. The supporting FL system should be able to handle these two (very different) scenarios.

We incorporated these characteristics into VANTAGE6's design from the very beginning, as described in the following section.

ARCHITECTURE

VANTAGE6 uses a client-server model, which is shown in Fig. 2. In this scenario, the researcher can pose a question and using his/her preferred programming language, send it as a task (also known as computation request) to the (central) server through function calls. The server is in charge of processing the task as well as of handling administrative functions such as authentication and authorization. The requested algorithm is delivered as a Docker image to the nodes, which have access to their own (local) data. When the algorithm has reached a solution, it is transmitted via the server to the researcher. A more detailed explanation of these components is given as follows.

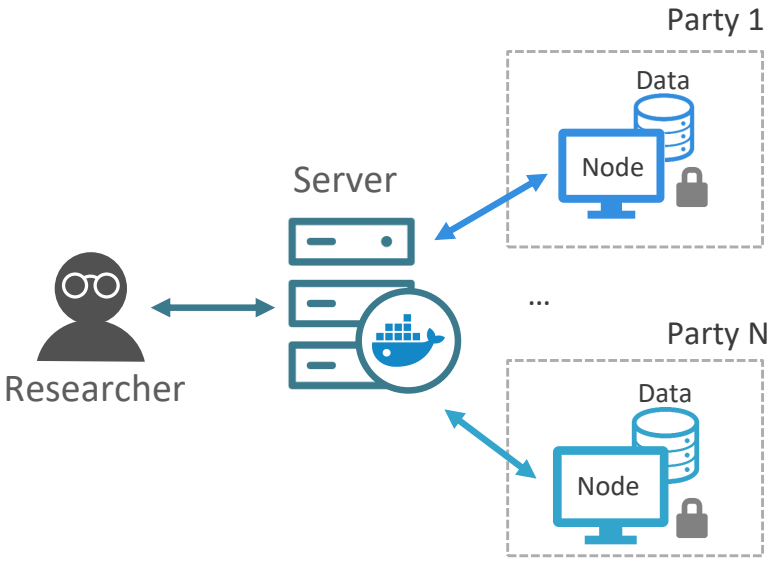


Figure 2: General diagram of the basic components of VANTAGE6. More detailed schematics of the server and nodes are shown in Fig. 3 and 5, respectively

Researcher

First, the researcher defines a question. In order to answer it, (s)he identifies which parties possess the required data and establishes a collaboration with them. Then, the parties specify which variables are needed and, more importantly, they agree on their definition. Preferably, this is done following previously established data standards suitable for the field and question at hand. Moreover, it is strongly encouraged that the parties adhere to practices and principles that make their data FAIR (findable, accessible, interoperable, and reusable)[14].

Once this is done, the researcher can pose his/her question as a task to the server in an HTTP request. VANTAGE6 allows the researcher to do so using any platform of his/her preference (e.g., Python, R, Postman, etc.). The request contains a JSON body which includes information about the collaboration and the party for which the request is intended, a reference to a Docker image (corresponding to the selected algorithm), and optional inputs (usually algorithm parameters). By default, the task is sent to all parties.

VANTAGE6's processing of the task (i.e., server and nodes functionality) occurs behind the scenes. The researcher only needs to deal with his/her working environment (e.g., Jupyter notebook, RStudio).

Once the results are ready, the researcher can obtain them in two ways: on demand (i.e., polling), or through a continuous connection with the server where messages can be sent/received instantly (i.e., WebSocket channel). Due to its speed and efficiency, the latter is preferred.

Server

Figure 3 shows a more detailed diagram of VANTAGE6's server. First, the server is configured by an administrator through a command line interface. The server's parameters (e.g., IP, port, log settings, etc.) are stored into a configuration file. The latter is loaded when the server starts. Once the server is running, entities (e.g., tasks, users, nodes) can be managed through a RESTful API. Furthermore, a WebSocket channel allows communication of simple messages (e.g., status updates) between the different components. This reduces the number of server requests (i.e., neither the researcher nor the nodes need to poll for tasks or results), improving the speed and efficiency of message transmission.

The server is also a good place for hosting a private registry of Docker images (although any Docker registry can be used) together with its corresponding RESTful API. The Docker images correspond to the algorithms' implementations, which are delivered to the nodes, where they are executed. VANTAGE6 also allows the researcher to upload its own Docker images (i.e., algorithms) to the registry. However, in order to be executed, all Docker images must be approved by the involved nodes (i.e., parties). This way, parties can *autonomously* decide which algorithms are allowed to have access to their data. Additionally, in order to verify that the pulled container corresponds to an approved image, VANTAGE6 uses Docker Notary (a digital seal for publishing and managing trusted collections of content).

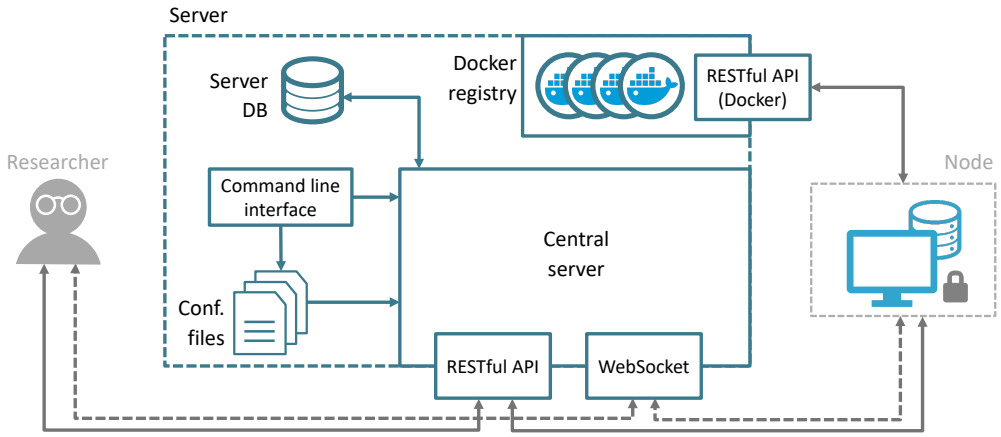


Figure 3: VANTAGE6's server. An administrator uses the command line interface to configure and start the server. After the server loads its configuration parameters (which are stored in a YAML file), it exposes its RESTful API. It is worth noting that the central server's RESTful API is different from that of the Docker registry.

The central server also stores metadata and information of the researcher (user), parties, collaborations, tasks, nodes, and results. Figure 4 shows its corresponding database model.

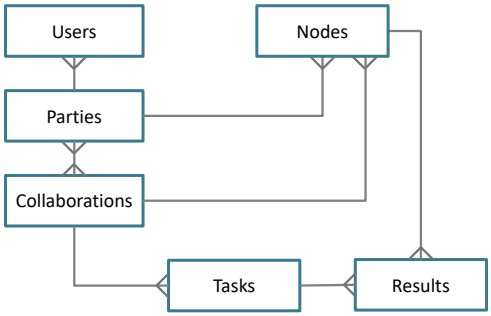


Figure 4: Database model of the central server (Fig. 3). The users are always members of a party, which can participate in multiple collaborations. Within a party, users can have different roles (e.g., an administrator is allowed to accept collaborations). For each collaboration a party takes place in, it should create a (running) node. Tasks are always part of a single collaboration and have one or multiple results. In turn, results are always part of a single task and node.

Node(s)

In order to host a node, the parties need to comply with a few minimal system requirements: Python 3.6+, Docker Community Edition (CE), a stable internet connection, and access to the data. Figure 5 shows a more detailed diagram of a single VANTAGE6 node.

In this case, an administrator uses a command line interface to configure the node's core and to start the Docker daemon. We can think of the latter as a service which manages Docker images, containers, volumes, etc. The daemon starts the node's core, which in turn instructs the daemon to create the data volume. The latter contains a copy of the host's data of interest. It is in this moment when the party can exert its *autonomy* by deciding how much of its data will it allow to contribute to the global solution at hand. After this step, all the pieces are in place for the task execution.

The node receives a task from the server (which could involve a master or an algorithm container) and executes it by downloading the requested (and previously approved) Docker image. The corresponding container accesses the local data through the node and executes the algorithm with the given parameters. Then, the algorithm outputs a set of (*intermediate*) results, which is sent to the server through the RESTful API. The user or the master container collects these results of all nodes. If needed, it computes a first version of the *global* solution and sends it back to the nodes, which use it to compute a new set of results. This process could be iteratively until the model's global solution converges or after a fixed number of iterations. This iterative approach is quite generic and allows *flexibility* by supporting numerous algorithms that deal with horizontally- or vertically-partitioned data[5].

It is worth emphasizing that the data always stay at their original location. It is only intermediate results (i.e., aggregated values, coefficients) that are transmitted, which immensely reduce the risk of leaking private patient information. Furthermore, all messages (node to node, node to user) are end-to-end-encrypted, adding an extra layer of security. It is also worth mentioning that the parties hosting the nodes are allowed to be *heterogeneous*: as long as they comply with the minimal system requirements, they can have their own hardware and operating system.

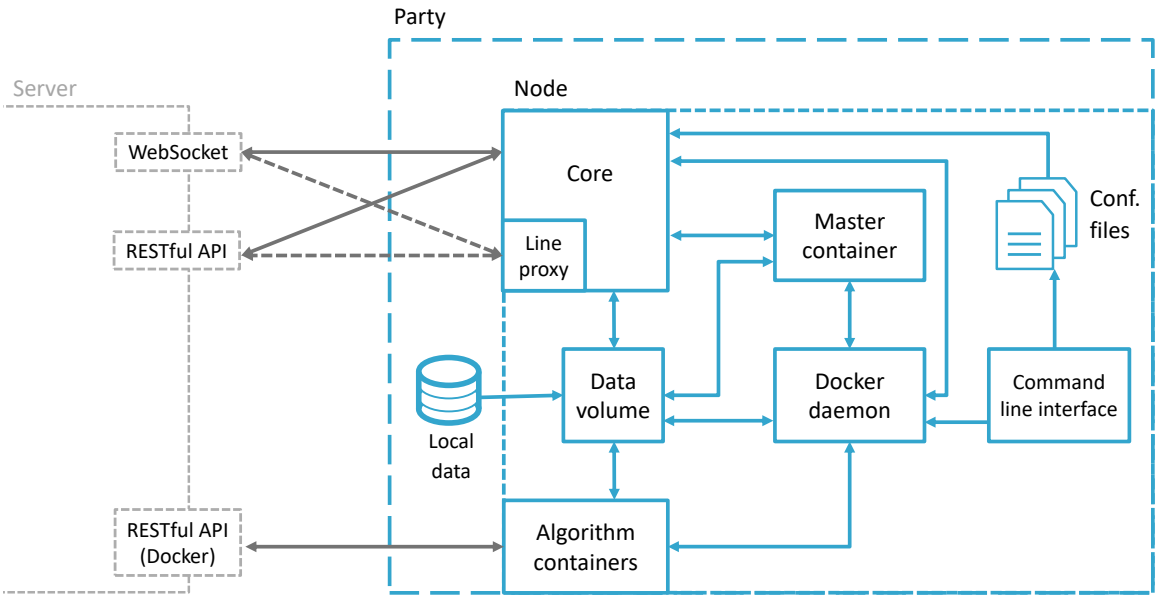


Figure 5: A party hosting a VANTAGE6 node. The party needs to have Python 3.6+, Docker Community Edition (CE), a stable internet connection, and access to its data. Similarly to the server's case, the node's configuration parameters are stored in a YAML file. It is worth emphasizing that the node only transmits to the server a set of aggregated values (i.e., coefficients) – the party's data never leave their original location.

APPLICATIONS & FUTURE WORK

We have used VANTAGE6 as a backbone of several projects that have been implemented and deployed successfully. For example, there is a huge difference in incidence of oral cavity cancer between the Netherlands and Taiwan. We hypothesized that different risk factors and treatment expertise could result in survival differences between these two countries. Unfortunately, due to legal and regulatory reasons, performing a centralized analysis using combined data from both parties was not possible. In collaboration with the Taiwan Cancer Registry, we investigated prognostic factors for survival of oral cavity cancer patients in the Dutch and Taiwanese populations. We did so by implementing a federated Cox Proportional Hazards algorithm for horizontally-partitioned data[15] on VANTAGE6. Without exchanging information at a patient-level, we found that the outcomes of patients treated in both countries were slightly but significantly different[16].

In another project, we were interested in investigating the impact of pathological synoptic (i.e., structured) reporting on Dutch prostate cancer patients. We believed that patients that received pathological synoptic reporting would present more favorable outcomes that patients that did not. In collaboration with the Nationwide Network and Registry of Histo- and Cytopathology (PALGA), we implemented a federated logistic regression for vertically-partitioned data[17] on VANTAGE6. Using the developed (federated) model, we found that pathological synoptic reporting had a consistent positive relation to patient survival[18].

Furthermore, we are actively using VANTAGE6 for FL analyses in other collaborations. For instance, we are comparing the survival of Dutch and Italian rectal cancer patients using a federated implementation of the χ^2 test. Moreover, we are also using VANTAGE6 as the core for developing secure-multiparty-computation-based solutions for survival analysis of vertically-partitioned data. Our future work will be focused on expanding the set of tools (i.e., algorithms) available for FL analyses. In order to help the parties evaluate the quality of their data, we are also interested in implementing a stage of data verification at the nodes.

These examples show how VANTAGE6's main focus and motivation is the support and development of FL projects. However, its architecture and modular components allow different types of applications, namely using it as a FAIR data station and as a model repository (Fig. 6). In the first case, VANTAGE6 would require setting up the server and a single node. Provided that the hosting party has taken care that its data are FAIR[14], researchers could easily query the node (within the limits set by the data provider) through an API interface. In the second case VANTAGE6 could function as a repository of a pre-computed model. In this case, access to the data is no longer required. The node could host the model and the researcher could query it (also through an API) to obtain predictions of interest.

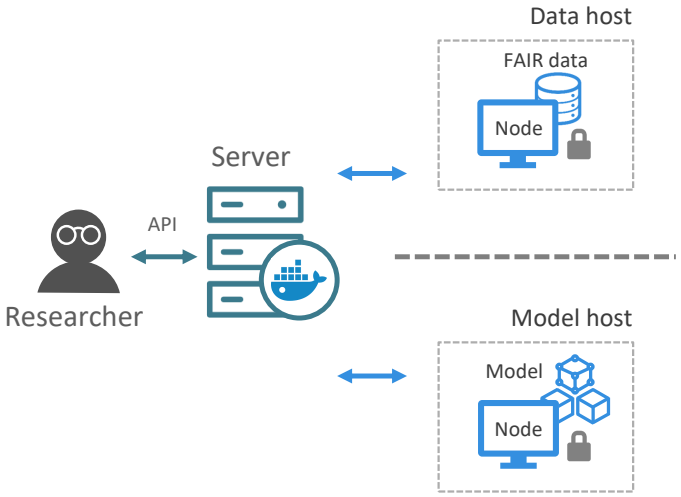


Figure 6: Besides serving as the backbone for FL projects, VANTAGE6's architecture and modular components allow for different potential use cases: FAIR (findable, accessible, interoperable, and reusable) data station and host of a pre-computed model. In both cases, the researcher can access the object of interest through an API interface.

In the Netherlands, FL has taken the shape of the so-called Personal Health Train, a network of health care and research institutes with health data from various sources. In this analogy, different parties (i.e., the stations) can answer their research questions (i.e., the trains) collaboratively by exchanging aggregated data and/or statistics. In this framework, VANTAGE6 serves as the backbone (i.e., the railways) upon which different projects will be developed on.

Lastly, it is worth noting that although VANTAGE6 was originally conceived and developed to be utilized in oncology, it can be easily adapted and extended to be used in other fields of health care or even outside of it, such as finance, education, and urbanism[6].

SUPPORT & COMMUNITY

We want to make VANTAGE6 an accessible and easy-to-use infrastructure for the FL field. To this end, we have built several services and communication channels to support users in setting up their own FL projects as quickly and efficiently as possible.

We have created and maintain a website for VANTAGE6, `https://vantage6.ai/`. Here, we provide a general overview of the infrastructure. More importantly, it serves as a hub for other useful resources. For instance, we maintain a blog, where we publish release notes, updates descriptions, and posts that are relevant for the community. It also hosts the corresponding documentation, which describes in detail how to install VANTAGE6, how to configure the server and nodes, how to use the RESTful API and WebSockets, and how to create new tasks and algorithms which can be tailored to the specific needs of the users. The website also links to VANTAGE6's Github repository[19], where the users can access the source code, submit contributions, and report issues.

Lastly, we have also created a Discord server. Its purpose is to encourage communication, interaction, and support between VANTAGE6's users.

We hope that these platforms will serve as a cornerstone upon which a VANTAGE6 community will be built around. Furthermore, we also hope that the community will help us grow and improve VANTAGE6 in the near and long-term future.

CONCLUSIONS

In this paper, we presented our priVAcy preserviNg federaTed leArninG infrastructurE for Secure Insight eXchange, VANTAGE6. It is a flexible, versatile platform capable of dealing with horizontally- *and* vertically-partitioned data that allows for parties' autonomy and heterogeneity. Its architecture uses Docker containers at its core, making it extremely flexible for researchers to implement a function or algorithm of their choice. Currently, it supports model aggregation and cryptographic privacy mechanisms.

We provided a few examples of FL projects where VANTAGE6 has been successfully utilized in the field of cancer informatics. However, VANTAGE6's use can extend beyond oncology and even outside health care.

Lastly, we enlisted the services and communication channels that we have established to help build and grow the community around VANTAGE6. Developing technology to support FL (such as VANTAGE6) will pave the way for the adoption and mainstream practice of this new approach for analyzing decentralized data.

References

1. J Ferlay, M Colombet, I Soerjomataram, et al. Cancer incidence and mortality patterns in europe: estimates for 40 countries and 25 major cancers in 2018. *European Journal of Cancer*, 2018.
2. Evert-Ben van Veen. Observational health research in Europe: understanding the General Data Protection Regulation and underlying debate. *European Journal of Cancer*, 104:70–80, 2018.
3. DLA Piper. Data protection laws of the world. Technical report, DLA Piper, 2020.
4. H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.
5. Wenrui Dai, Shuang Wang, Hongkai Xiong, and Xiaoqian Jiang. Privacy preserving federated big data analysis. In *Guide to Big Data Applications*, pages 49–82. Springer, 2018.
6. Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu. *Federated Learning – Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool Publishers, 2019.
7. Jie Xu and Fei Wang. Federated learning for healthcare informatics. *arXiv preprint arXiv:1911.06270*, 2019.
8. Qinbin Li, Zeyi Wen, and Bingsheng He. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *arXiv preprint arXiv:1907.09693*, 2019.
9. Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):12, 2019.
10. Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*, 2019.
11. Amadou Gaye, Yannick Marcon, Julia Isaeva, et al. DataSHIELD: taking the analysis to the data, not the data to the analysis. *International journal of epidemiology*, 43(6):1929–1944, 2014.
12. Theo Ryffel, Andrew Trask, Morten Dahl, et al. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.
13. Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, et al. Towards federated learning at scale: System design. In *Proceedings of the 2nd Conference on Systems and Machine Learning (SysML)*, 2019.
14. Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, et al. The FAIR guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.
15. Chia-Lun Lu, Shuang Wang, Zhanglong Ji, et al. WebDISCO: a web service for distributed cox model learning without patient-level data sharing. *Journal of the American Medical Informatics Association*, 22(6):1212–1219, 2015.
16. Gijs Geleijnse, RuRu Chun-Ju, Melle Sieswerda, et al. Prognostic factors for survival in patients with oral cavity cancer: a comparison of the Netherlands and Taiwan using privacy-preserving federated analyses. *Under review*, 2020.
17. Yong Li, Xiaoqian Jiang, Shuang Wang, Hongkai Xiong, and Lucila Ohno-Machado. VERTIcal Grid lOgistic regression (VERTIGO). *Journal of the American Medical Informatics Association*, 23(3):570–579, 2015.
18. Arturo Moncada-Torres, Frank Martin, Katja Aben, et al. The effect of synoptic reporting on prostate cancer survivability: Centralized and federated analysis. *Under review*, 2020.
19. Frank Martin, Melle Sieswerda, Johan van Soest, and Arturo Moncada-Torres. VANTAGE6. Available at https://doi.org/10.5281/zenodo.3686944, February 2020.