PyVertical: A Vertical Federated Learning Framework for Multi-headed SplitNN

Daniele Romanini *
OpenMined
daler.romanini@gmail.com

Adam James Hall *
Edinburgh Napier University / OpenMined
adam@openmined.org

Pavlos Papadopoulos *
Edinburgh Napier University / Apheris
pavlos.papadopoulos@napier.ac.uk

Tom Titcombe *
Tessella / OpenMined
tom.titcombe@tessella.com

Abbas Ismail
Birla Institute of Technology, Mesra
be10285.17@bitmesra.ac.in

Tudor Cebere
OpenMined
tudor@openmined.org

Robert Sandmann
Apheris
r.sandmann@apheris.com

Robin Roehm
Apheris
r.roehm@apheris.com

Michael A. Hoeh
Apheris
m.hoeh@apheris.com

Abstract

We introduce PyVertical, a framework supporting vertical federated learning using split neural networks. The proposed framework allows a data scientist to train neural networks on data features vertically partitioned across multiple owners while keeping raw data on an owner's device. To link entities shared across different datasets' partitions, we use Private Set Intersection on IDs associated with data points. To demonstrate the validity of the proposed framework, we present the training of a simple dual-headed split neural network for a MNIST classification task, with data samples vertically distributed across two data owners and a data scientist.

1 Introduction

With ubiquitous data collection, individuals are constantly generating diverse swathes of data, including location, health, financial information. These data streams are often collected by separate entities and are sufficient for high utility use-cases. A common challenge faced by data scientists is utilising data isolated in silos to train machine learning (ML) algorithms. When this data is commercially sensitive, personal or otherwise under strict legal protection, it cannot be simply merged with data controlled by another party. To ensure data privacy is not compromised during the training or inference process, several privacy-preserving ML techniques, such as Federated Learning (FL) (McMahan et al., 2016; Konečnỳ et al., 2016; McMahan & Ramage, 2017; Bonawitz et al., 2019; Ryffel et al., 2018), focus on training ML models on distributed datasets by keeping data in the custody of its corresponding holder. FL typically splits data horizontally. This is where datasets are distributed across multiple owners that have the same features and represent different data subjects (Kantarcioglu & Clifton, 2004). However, it is common in real-world scenarios to find datasets which are vertically distributed (McConnell & Skillicorn, 2004), i.e. different features of the same

---

* Authors have equal contribution in this work

data subject are distributed across multiple data owners. For example, specialists or general hospitals may hold different parts of a patient's medical data.

To address the issue of learning from vertically distributed data, we use Split Neural Networks (SplitNN) to first map data into an abstract, shareable representation. This allows information from multiple sources to be combined for learning without exposing raw data. We combine this with Private Set Intersection (PSI) to identify and link data points belonging to the same data samples shared among parties. This process facilitates Vertical Federated Learning (VFL) for non-linear functions.

1.1 CONTRIBUTIONS

In this work, we extend the proposal of (Angelou et al., 2020), regarding the use of (SplitNNs) and PSI in Vertical Federated Learning. We use the PySyft library for privacy-preserving machine learning (Ryffel et al., 2018) to train a Vertically Federated ML algorithm on data distributed across the premises of one or multiple data owners. This work is released as an open-source framework, PyVertical. To the best of our knowledge, this is the first open-source framework to perform machine learning on vertically distributed datasets using Split Neural Networks[1].

We verify our method on a two-party, vertically-partitioned MNIST dataset. Our work presents a dual-headed scenario, where data from two separate data owners (who holds different parts of the data samples) and a data scientist (who, in our case, holds data labels) are securely aligned and combined for model training. However, this work could be extended to multiple data owners using the same principle we describe here.

2 BACKGROUND KNOWLEDGE AND RELATED WORK

2.1 PRIVATE SET INTERSECTION

Private Set Intersection (PSI) (Freedman et al., 2004; Huang et al., 2012; De Cristofaro & Tsudik, 2010; Dachman-Soled et al., 2009) is a multi-party computation cryptographic technique which allows two parties, where each hold a set of elements, to compute the intersection of these elements, without revealing anything to the other party except for the elements in the intersection. Different PSI protocols have been proposed (Buddhavarapu et al., 2020; Ion et al., 2020; Chase & Miao, 2020; Pinkas et al., 2018) and employed for scenarios such as private contact discovery (Demmler et al., 2018) and also privacy-preserving contact tracing (Angelou et al., 2020).

In this work, we use a PSI implementation based on a Diffie-Hellman key exchange that uses Bloom filters compression to reduce the communication complexity (Angelou et al., 2020). This protocol works with two parties computing the intersection between their sets. However, the chosen PSI framework can be replaced with an alternative implementation, for instance, to compute directly the intersection of datasets coming from more than two parties (Hazay & Venkitasubramaniam, 2017).

2.2 SPLIT NEURAL NETWORKS

Split learning is a concept of training a model that is split into segments held by different parties or on different devices. A neural network model trained this way is called a Split Neural Network, or SplitNN. In SplitNN, each model segment transforms its input data into an intermediate data representation (as the output of a hidden layer of a classic neural network). This intermediate data is transmitted to the next segment until the training or the inference process is completed. During backpropagation, the gradient is also propagated across different segments. Compared to data-centric FL, split learning can also be useful to reduce the computational burden on data owners, who in many real-world scenarios may have limited computational resources (Gupta & Raskar, 2018; Vepakomma et al., 2018).

---

[1]Code is available at PyVertical: https://github.com/OpenMined/PyVertical

2.3 Vertical Federated Learning

Vertical federated learning (VFL) is the concept of collaboratively training a model on a dataset where data features are split amongst multiple parties (Yang et al., 2019). For example, different healthcare organizations may have different data for the same patient. Considering the sensitivity of the data, these two organizations cannot simply merge their information without violating that person's privacy. For this reason, a machine learning model should be trained collaboratively, and data should be kept on the corresponding premises. Machine learning algorithms for vertical partitioned data is not a new concept, and many studies for new models and algorithms have been proposed in this area (Feng & Yu, 2020; Liu et al., 2020; Du & Atallah, 2001; Du et al., 2004; Vaidya & Clifton, 2002; Karr et al., 2009; Sanil et al., 2004; Wan et al., 2007; Gascón et al., 2017; Thapa et al., 2020; Hardy et al., 2017; Nock et al., 2018). Existing open-source VFL frameworks include FedML (He et al., 2020), which implements multi-party linear models (Hardy et al., 2017).

Similarly to our work, the use of split networks for vertical federated learning has been proposed (Ceballos et al., 2020). However, differently from our work, the authors investigate multiple methods for combining information sent to the data scientist from disjoint datasets. Moreover, they do not consider the entity resolution problem for aligning data across parties, whereas we illustrate how PSI can be successfully exploited prior to the training process to account for this.

3 Framework Description

We introduce PyVertical, a framework written in Python for vertical federated learning using SplitNNs and PSI. PyVertical is built upon the privacy-preserving deep learning library PySyft (Ryffel et al., 2018) to provide security features and mechanisms for model training, such as pointers to data, without exposing private information.

A set of data features are distributed across one or more data owners. We refer to a full dataset split vertically across the features as a *vertical dataset*. Each of the data owners takes part in the model training, alongside a data scientist who orchestrates the process. The data scientist could also be a data owner itself, holding features or data labels. The data features in the vertical datasets may intersect. Each data point is associated with a unique ID based on the data point's subject, the format of which is agreed by the data owners (e.g. legal names, email addresses, ID card numbers). The data owners use PSI to agree on a shared set of data IDs (process described in Section 3.1); each data owner discards non-shared data from their datasets and sorts their datasets by ID, such that element n of each vertical dataset corresponds to the same data subject.

In our framework, the data scientist is able to define a split neural network model and send the corresponding model segments to the data owners. Each data owner's model segment maps their data samples to an abstract representation with k_i neurons. The output from each model segment (which would correspond to a hidden layer of a complete classic neural network) is sent to the data scientist and concatenated to form a $\sum_i k_i$ length vector. The data scientist also defines a model segment corresponding to the final part of the split neural network. This segment remains on the data scientist's premises and maps the concatenated, intermediate data (i.e. the output from data owners' model segments) into a shape relevant to the task. During model training, the data scientist calculates batch loss and updates their model segment's weights. The data scientist then sends the final gradient to the data owners, each of whom updates their own model segment by completing backpropagation independently. We assume all parties are honest-but-curious actors. Figure 1 demonstrates model inference under this framework for the experiment outlined in Section 4.

3.1 Data Resolution Protocol

We use a PSI Python library (Angelou et al., 2020) to identify intersections between data points in two datasets based on unique IDs. In this work, we consider a setting where the data scientist has access to ground truth labels. For all three parties (two data owners + one data scientist) to agree on data points shared among all datasets, the protocol works as follow: firstly, the data scientist runs the PSI protocol independently with each data owner. The intersection of IDs between the data scientist and each data owner is revealed to the data scientist. The data scientist computes the global intersection from the two previous independently computed intersections and communicates the global intersection to the data owners. In this setting, the data owners do not communicate
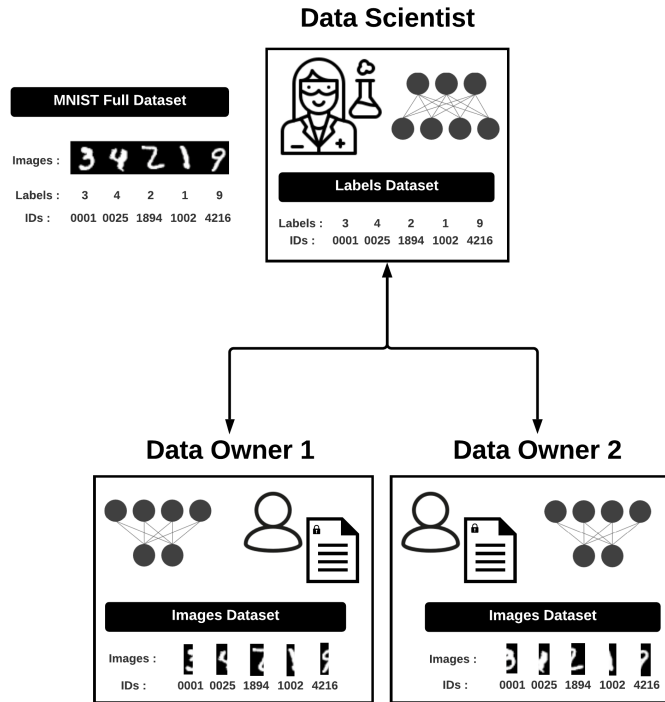
Figure 1: Parties and datasets in the conducted experiment. Data Scientist holds a part of the SplitNN and the labels dataset. Data Owners hold their images datasets and parts of the SplitNN

and are not aware of each other's identity in any regard. In practice, this is an ideal feature of the protocol as having knowledge of a group's or individual's participation in a training process can reveal sensitive commercial and personal information in and of itself. Moreover, as the single IDs' intersection lists are only revealed to the data scientist, there is no risk for the data owners to learn which information the other data owners owns. Each of the data owners learns only the information necessary for training or inference.

4 EXPERIMENT

To verify the validity of our framework, we train a dual-headed SplitNN on a vertically-partitioned version of the MNIST dataset. We generate the data by splitting the images in MNIST into left and right halves, providing a dataset of each half to different data owners. The data scientist defines and sends an identical, multi-layered neural network to each of the data owners that takes 392-length vectors as input (flattened representation of 28x14 pixel images). The data scientist also defines and keeps on its premises the second part of the neural network, which outputs a softmax layer for classification. The data scientist can access the ground truth labels and calculate the loss for each data batch. The data scientist controls the training process and hyperparameters. Appendix B provides more details on the specific values used in model training. The objective of this experiment is to demonstrate that the proposed framework allows vertically-partitioned learning. This specific experiment should be considered a proof-of-concept, thus not highly optimised for the specific task. Nevertheless, we report the results of the experiment in Figure 4 (in Appendix B).

5 EVALUATION AND CONCLUSION

We have developed and distributed our work as an open-source project. We hope that PyVertical can serve as a useful tool for researching neural-networks-based VFL. We find PSI an appropriate and useful method for resolving data subjects across datasets; many datasets and domains already collect unique IDs, such as usernames or national identifiers for medical data, making our method widely

applicable. Finally, we successfully train a dual-headed model on a vertically-partitioned MNIST dataset, demonstrating that the proposed framework and method work in principle.

5.1 LIMITATIONS AND FUTURE WORK

The experiment performed in this work assumes that all the parties involved (data owners and data scientist) act honestly. To develop a truly scalable, robust VFL system, additional precautions should be taken into account: identity management, validation of adherence to PSI protocol, and a method agreeing on data ID schema, to name a few.

This work investigates a symmetric SplitNN model: we assume that each data owner holds an identical model segment and that data points are split equally between data owners. Future work should investigate the impact of imbalanced vertical datasets (Liu et al., 2020) and the resulting difficulties from the asymmetric model segment convergence due to the use of different sized models and learning rates.

Finally, we illustrate an example of a training process with two data owners and a data scientist holding labels. While the proposed framework can support more parties in principle, we aim to investigate how to apply the process to massively multi-headed Vertical Federated Learning tasks. Additionally, we plan to research and integrate other privacy-preserving ML techniques into our workflow, such as decentralised identities (Papadopoulos et al., 2021; Abramson et al., 2020) and differential privacy (Dwork et al., 2006; Dwork, 2008; Titcombe et al., 2021), to further enhance privacy guarantees.

REFERENCES

Will Abramson, Adam James Hall, Pavlos Papadopoulos, Nikolaos Pitropakis, and William J. Buchanan. A distributed trust framework for privacy-preserving machine learning. *Lecture Notes in Computer Science*, pp. 205–220, 2020. ISSN 1611-3349. doi: 10.1007/978-3-030-58986-8_14. URL http://dx.doi.org/10.1007/978-3-030-58986-8_14.

Nick Angelou, Ayoud Benaissa, Bogdan Cebere, Will Clark, Phillipp Schoppmann, Rutuja Surve, Daniel Liu, and Ben Szymbow. PSI Source Code, 2020. URL https://github.com/OpenMined/PSI.

Nick Angelou, Ayoub Benaissa, Bogdan Cebere, William Clark, Adam James Hall, Michael A Hoeh, Daniel Liu, Pavlos Papadopoulos, Robin Roehm, Robert Sandmann, Phillipp Schoppmann, and Tom Titcombe. Asymmetric private set intersection with applications to contact tracing and private vertical federated machine learning. *arXiv preprint arXiv:2011.09350*, 2020.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.

Prasad Buddhavarapu, Andrew Knox, Payman Mohassel, Shubho Sengupta, Erik Taubeneck, and Vlad Vlaskin. Private matching for compute. *IACR Cryptol. ePrint Arch.*, 2020:599, 2020.

Iker Ceballos, Vivek Sharma, Eduardo Mugica, Abhishek Singh, Alberto Roman, Praneeth Vepakomma, and Ramesh Raskar. Splitnn-driven vertical partitioning, 2020.

Melissa Chase and Peihan Miao. Private set intersection in the internet setting from lightweight oblivious prf. In *Annual International Cryptology Conference*, pp. 34–63. Springer, 2020.

Dana Dachman-Soled, Tal Malkin, Mariana Raykova, and Moti Yung. Efficient robust private set intersection. In *International Conference on Applied Cryptography and Network Security*, pp. 125–142. Springer, 2009.

Emiliano De Cristofaro and Gene Tsudik. Practical private set intersection protocols with linear complexity. In *International Conference on Financial Cryptography and Data Security*, pp. 143–159. Springer, 2010.

Daniel Demmler, Peter Rindal, Mike Rosulek, and Ni Trieu. Pir-psi: Scaling private contact discovery. *IACR Cryptol. ePrint Arch.*, 2018.

Wenliang Du and Mikhail J Atallah. Privacy-preserving cooperative statistical analysis. In *Seventeenth Annual Computer Security Applications Conference*, pp. 102–110. IEEE, 2001.

Wenliang Du, Yunghsiang S Han, and Shigang Chen. Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *Proceedings of the 2004 SIAM international conference on data mining*, pp. 222–233. SIAM, 2004.

Cynthia Dwork. Differential privacy: A survey of results. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li (eds.), *Theory and Applications of Models of Computation*, pp. 1–19, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. ISBN 978-3-540-79228-4.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin (eds.), *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Siwei Feng and Han Yu. Multi-participant multi-class vertical federated learning. *arXiv preprint arXiv:2001.11154*, 2020.

Michael J Freedman, Kobbi Nissim, and Benny Pinkas. Efficient private matching and set intersection. In *International conference on the theory and applications of cryptographic techniques*, pp. 1–19. Springer, 2004.

Adrià Gascón, Phillipp Schoppmann, Borja Balle, Mariana Raykova, Jack Doerner, Samee Zahur, and David Evans. Privacy-preserving distributed linear regression on high-dimensional data. *Proceedings on Privacy Enhancing Technologies*, 2017(4):345–364, 2017.

Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*, 2017.

Carmit Hazay and Muthuramakrishnan Venkitasubramaniam. Scalable multi-party private set-intersection. In *IACR International Workshop on Public Key Cryptography*, pp. 175–203. Springer, 2017.

Chaoyang He, Songze Li, Jinhyun So, Mi Zhang, Hongyi Wang, Xiaoyang Wang, Praneeth Vepakomma, Abhishek Singh, Hang Qiu, Li Shen, et al. Fedml: A research library and benchmark for federated machine learning. *arXiv preprint arXiv:2007.13518*, 2020.

Yan Huang, David Evans, and Jonathan Katz. Private set intersection: Are garbled circuits better than custom protocols? In *NDSS*, 2012.

Mihaela Ion, Ben Kreuter, Ahmet Erhan Nergiz, Sarvar Patel, Shobhit Saxena, Karn Seth, Mariana Raykova, David Shanahan, and Moti Yung. On deploying secure computing: Private intersection-sum-with-cardinality. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 370–389. IEEE, 2020.

Murat Kantarcioglu and Chris Clifton. Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE transactions on knowledge and data engineering*, 16(9): 1026–1037, 2004.

Alan F Karr, Xiaodong Lin, Ashish P Sanil, and Jerome P Reiter. Privacy-preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics*, 25(1):125, 2009.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

Yang Liu, Xiong Zhang, and Libin Wang. Asymmetrically vertical federated learning. *arXiv preprint arXiv:2004.07427*, 2020.

Sabine McConnell and David B Skillicorn. Building predictors from vertically distributed data. In *Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research*, pp. 150–162, 2004.

Brendan McMahan and Daniel Ramage. Federated learning: Collaborative machine learning without centralized training data. *Google Research Blog*, 3, 2017. URL `https://ai.googleblog.com/2017/04/federated-learning-collaborative.html`.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Richard Nock, Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Entity resolution and federated learning get a federated resolution. *arXiv preprint arXiv:1803.04035*, 2018.

Pavlos Papadopoulos, Will Abramson, Adam J. Hall, Nikolaos Pitropakis, and William J. Buchanan. Privacy and trust redefined in federated machine learning. *Machine Learning and Knowledge Extraction*, 3(2):333–356, 2021. ISSN 2504-4990. doi: 10.3390/make3020017. URL `https://www.mdpi.com/2504-4990/3/2/17`.

Benny Pinkas, Thomas Schneider, and Michael Zohner. Scalable private set intersection based on ot extension. *ACM Transactions on Privacy and Security (TOPS)*, 21(2):1–35, 2018.

Theo Ryffel, Andrew Trask, Morten Dahl, Bobby Wagner, Jason Mancuso, Daniel Rueckert, and Jonathan Passerat-Palmbach. A generic framework for privacy preserving deep learning. *arXiv preprint arXiv:1811.04017*, 2018.

Ashish P Sanil, Alan F Karr, Xiaodong Lin, and Jerome P Reiter. Privacy preserving regression modelling via distributed computation. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 677–682, 2004.

Chandra Thapa, M. A. P. Chamikara, and Seyit Camtepe. Splitfed: When federated learning meets split learning, 2020.

Tom Titcombe, Adam J. Hall, Pavlos Papadopoulos, and Daniele Romanini. Practical defences against model inversion attacks for split neural networks. *In press*, 2021.

Jaideep Vaidya and Chris Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 639–644, 2002.

Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

Li Wan, Wee Keong Ng, Shuguo Han, and Vincent CS Lee. Privacy-preservation for gradient descent methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 775–783, 2007.

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):1–19, 2019.

A PYVERTICAL PROTOCOL

Figure 2 describes the PyVertical protocol applied to the MNIST dataset for a single data owner. The dual-headed PSI data linkage process is presented in Figure 3. Note that, in this illustration, there is only one data scientist; the duplicated icon is just to illustrate in more details how the data scientist runs a single PSI with each data owner separately, and that this could be done in parallel.

(a) Full dataset

(b) Split images and labels datasets

(c) PSI linkage and ordering
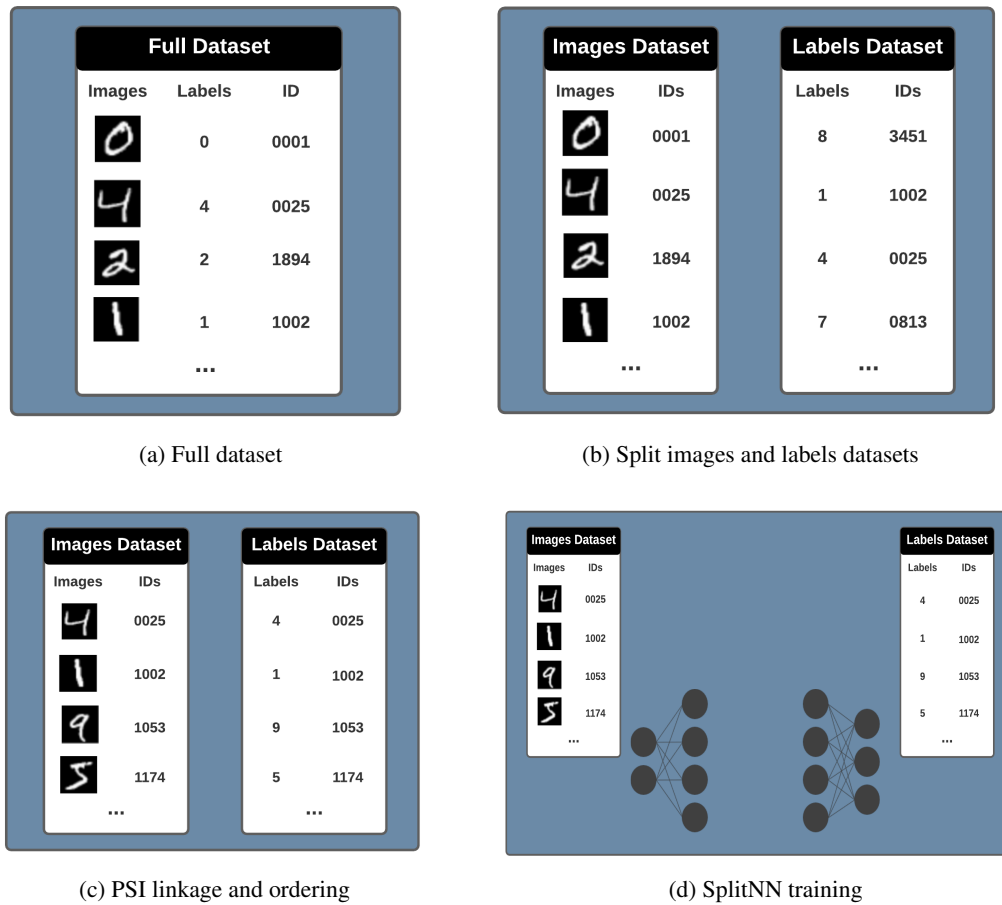
(d) SplitNN training

Figure 2: Vertical federated learning proof-of-concept implementation of Angelou et al. (2020)
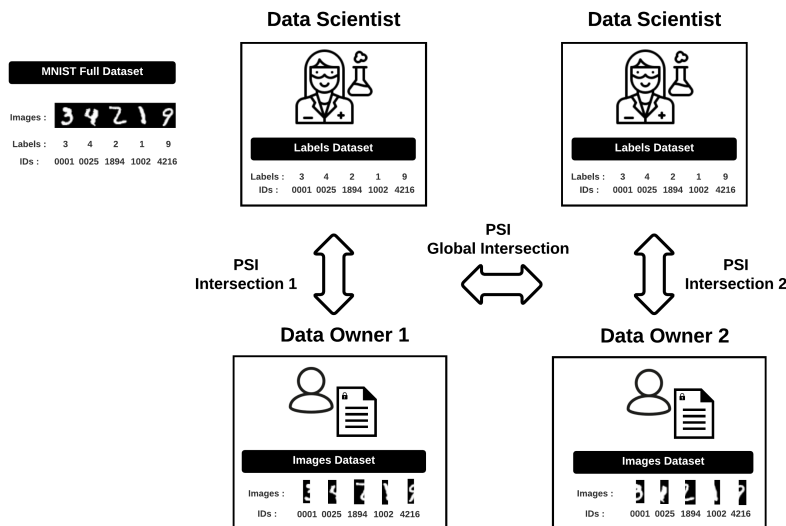

Figure 3: i) Data Scientist computes the intersection with Data Owner 1. ii) Data Scientist computes the intersection with Data Owner 2. iii) Data Scientist computes the global intersection.

B EXPERIMENTAL SETUP

The data owner model segment maps 392-length input into a 64-length intermediate vector with a ReLU activation, which is an abstract encoding of the data. The data scientist controls a separate neural network that takes as input a 128-length vector (concatenated data owner outputs) and transforms it into a softmax-activated, 10-class vector representing the possible digits in the dataset. The data scientist's model has a 500-length hidden layer with a ReLU activation. All layers are fully-connected. A learning rate of 0.01 is used for the data owner models and 0.1 for the data scientist model. Data is grouped into batches of size 128. Only the first 20,000 training images of MNIST are used, and the model is trained for 30 epochs.
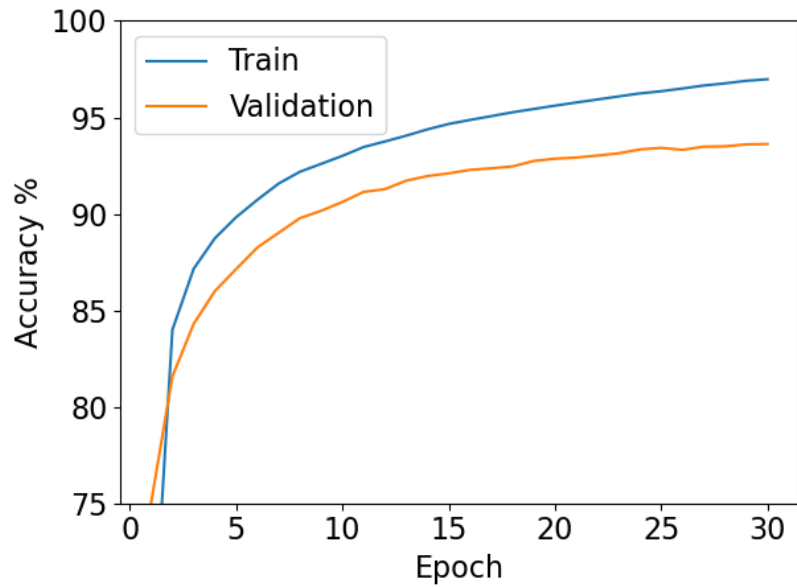

Figure 4: Train and validation accuracy for an unoptimised dual-headed SplitNN on vertically-partitioned MNIST.