Vertical Split Learning - an exploration of predictive performance in medical and other use cases

Corinne G. Allaart
Computer Science Dept
Vrije Universiteit
Amsterdam, Netherlands
0000-0002-5262-3723

Björn Keyser
Computer Science Dept
Vrije Universiteit
Amsterdam, Netherlands

Henri Bal
Computer Science Dept
Vrije Universiteit
Amsterdam, Netherlands
0000-0001-9827-4461

Aart van Halteren
Computer Science Dept.
Vrije Universiteit
Amsterdam, Netherlands
0000-0002-9631-0657

***Abstract*—In healthcare and other fields, data of an individual is often vertically partitioned across multiple organizations. Creating a centralized data store for AI algorithm development is cumbersome in such cases because of concerns like privacy and data ownership. Methods of distributed learning over vertically partitioned data could offer a solution here. While several studies have evaluated the feasibility, privacy and efficiency of such methods, an extensive evaluation of their impact on predictive performance compared to a centralized approach is missing. Vertical Split Learning (VSL) aims to provide vertical distributed learning through distributed neural network architectures. Our study adapts and applies VSL to 8 datasets, both in medicine and beyond, evaluating the impact of different network and (vertical) feature distributions on predictive performance. In most configurations VSL yields comparable predictive performance to its centralized counterparts. However, certain data and network distributions give an unexpected and severe loss of performance. Based on our findings we give some initial recommendations under which conditions VSL can be applied as a suitable alternative for data centralization.**

***Index Terms*—Vertically partitioned data, deep learning, split learning, medicine**

I. Introduction

Already for decades there is a great promise and prospect in the use of artificial intelligence (AI) in healthcare. AI is often promoted as a means to improve health outcomes, reduce costs and improve the healthcare experience for both patients and clinicians. But there are still technology, data and regulatory barriers that inhibit the widespread implementation of AI across the healthcare industry. One of these barriers is the use of centralised storage and computation for AI algorithm development [1]. Limited availability of data sets hinders training and validation [2]. As patients are usually managed across the care continuum, health data of one patient is typically stored at multiple providers. As a result, data is distributed and often needs to be brought together in a centralized repository for algorithm development. This can be time consuming or even impossible due to privacy concerns of patients, patient consent or data sharing policies of individual care institutions. Recent developments in distributed learning have opened up the possibility to develop AI algorithms without bringing data into a central repository. Application of distributed learning depends on the type of data distribution. When a data set is distributed across organisations, yet all the data of a single patient is in a single location, this is referred to as horizontally partitioned data. When vertically partitioned, the data of every single patient is distributed among different care institutions. While horizontally partitioned data implementations and solutions are widespread, vertically partitioned distributed learning remains an understudied topic [3]. Nevertheless, it is highly prevalent in medical use cases as patients often have multiple care providers, even for a single episode of care. For example, consider the development of an algorithm that predicts the outcome of Cerebrovascular Accident (CVA): after the CVA incident, the patient might be admitted to a hospital at first, but be transferred to a different clinic for their rehabilitation. Moreover, this patient might also have data at their general practitioner. To create prediction models using all essential information, it may be necessary to draw information from all involved institutions.

The straightforward approach for dealing with vertically partitioned data is to create a central database or registry. Unfortunately, this comes with concern of privacy as true anonymization of data is difficult [4]. Also, it needs approval from (medical) ethical boards of the different institutions, and there are questions of data ownership responsibility of maintaining and updating the central database. While distributed learning might not solve all these issues [2], it will make shared learning more feasible. Recently, the field of vertical federated learning (VFL) has started to offer such distributed solutions for vertically partitioned data. Several machine learning algorithms have been developed, such as for logistic regression [5], random forests [6], support vector machines [7], as well as deep learning (DL) [8] [9]. Unfortunately, these DL solutions still have issues in terms of predictive performance, privacy and efficiency. While several papers have addressed these last two issues [14] [12], predictive performance, especially for diverse sets of use cases, has not been properly investigated. A promising VFL approach is Vertical Split Learning (VSL) [11]. With VSL,

This publication is part of the project Enabling Personalized Interventions (EPI) with project number 628.011.028 of the research programme Commit2Data-Data2Person, which is (partly) financed by the Dutch Research Council (NWO).

not only the data but also the neural network is distributed over the locations. This creates the opportunity to bring partial networks to the vertically partitioned data, thereby creating a vertically split neural network, where data remains locally, and only intermediate outputs and gradients are shared between the different locations. However, this distribution creates the possibility of loss of predictive performance compared to centralized learning [11]. While VSL has been tested on a few use cases, it is important to investigate this in a larger set of use cases and distribution scenarios. VSL negates the need for a central database, and offers opportunity to perform deep learning on distributed, more complete datasets. Therefore, a small loss compared to centralized learning (CL) could be acceptable, as it could increase the availability of usable data. Nevertheless, it is essential to investigate in which cases the VSL is a good alternative to CL, and to show how the pay-off between effort and potential gain of predictive performance of CL skews.

The goal of this paper is to study in which situations CL can lead to a gain in predictive performance in medical use cases, by comparing the predictive performance of VSL for several different scenarios in comparison with CL. It will look at different datasets, including several medical and non-medical datasets to cover a range of different use cases, as well as different feature distributions in these datasets and the impact of the different set ups of the Vertical Split Neural Network. The contributions of this paper are as follows:

1) We develop a method to extensively evaluate the predictive performance of vertical split learning, where we allow for different network and feature distributions
2) We apply our developed evaluation method for vertical split learning on 8 datasets that provide a wide range of use cases where vertical partitioning is appropriate
3) Based on our evaluations, we offer some initial recommendations in which manner and in which use cases vertical split learning can be used instead of centralized learning without major loss in performance

II. Related work

A dataset can be distributed in many ways and each distribution raises different challenges. The partitioning is defined by the distribution of the samples and the features of the datasets over the different locations. Data can be horizontally partitioned (figure 1a), vertically partitioned (figure 1b), or a combination of the two, also referred to as arbitrarily partitioned (figure 1c). Of these situations, the most commonly occurring is horizontal data partitioning. This is for example the use case of standard or horizontal federated learning (HFL). For HFL, as every sample has all features in the same location, every location creates their local model on their local samples, whereafter these local models are used to iteratively combine into one global model [3]. HFL solutions can not be directly applied to vertically paritioned data (VPD), as they require the same features across all different partitions, but the field of vertical federated learning offers new solutions. For machine learning purposes, most algorithms have either through experimentation or mathematical proof been shown to be comparably accurate to their centralized counterparts. These include algorithms for logistic regression [5], random forests [6], and support vector machines [7]. With regards to deep learning on vertically partitioned data, several studies have been published with architectures that allow for vertical distribution [8] [9] [13].

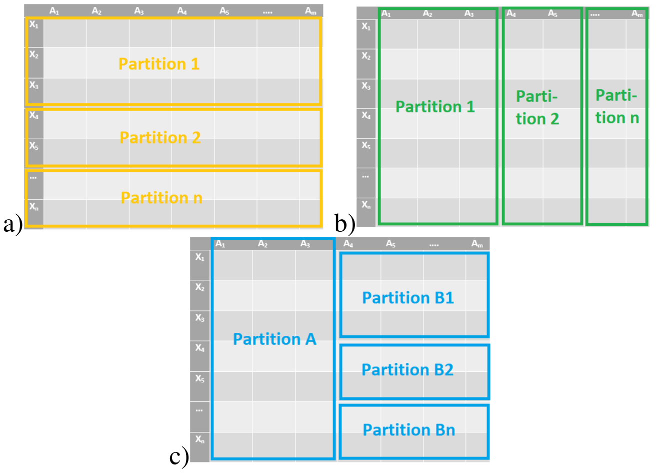

Fig. 1. shows the different distributions of data, with the rows representing samples and the columns representing features. Each partition illustrates the data at a local node. Data can be (a) Horizontally partitioned, (b) Vertically partitioned, (c) Arbitrarily partitioned

Due to the more complicated nature of VPD, as the data of one sample is spread out over different institutions, VFL cannot simply send complete neural network updates but has to rely on some kind of partial model. This can raise issues in terms of privacy, efficiency and predictive performance, with the first two being the subject of several recent VFL studies. Some studies have pointed out the risk of privacy leakage in VFL architectures, due to the sending of intermediary network information [15] [14]. To resolve this, TransNet adds a noise layer in the neural network to limit data exposure, at the cost of small loss in predictive performance [9], and [12] and [10] have argued to add extra privacy protecting measures. Nevertheless, we see that there is limited attention to the influence of vertical federated learning on predictive performance, even though many record a significant drop compared to centralized learning. Due to the distribution of VPD, it can be expected that the predictive performance will not only be affected by network architecture, but also by the type of data set and the feature distribution. While retention of predictive performance is essential for acceptation, especially in health care, we see very few papers that conduct a comparative analysis on the predictive performance. Often they test with an imaging dataset such as MNIST [8] [10] [13], which would not give an accurate representation of a vertically distributed dataset, or only selecting a single feature distribution with no information of the networks performance on other distributions [11] [9] [12]. Moreover, some algorithms assume that all parties or the server have access to the datasets labels [10] [9], a case not necessarily representative for a real life scenario. Vertical Split Learning [11], where a neural network is split up amongst several local nodes and a central server is promising. Its predictive performance seems comparable to the centralized learning in their experiments, and the architecture allows for the labels to be held by a single party. We believe an accurate and thorough evaluation of predictive performance is necessary for acceptation. This is especially important medical use cases, because of the possibility of an increase in bias with a drop in performance. Our goal is to test vertical split learning on a set of different use cases in different feature distributions, to show under which conditions VSL can be a suitable alternative to centralized learning.

III. Background

Vertical split learning originated from (horizontal) split learning (SL) [16], where the layers of a neural network are spread out over multiple clients. This can be beneficial for privacy as clients do not need to share their data, and efficient as calculating the network can happen in a distributed manner. VSL is an extension of SL, where the layers themselves can also be split among different locations. The distribution of the neural network is illustrated in figure 2a, where the neural network is split up among three parties, 2 local nodes and a central server. The k layer represents the layer in the neural network where the division between the local nodes and the server falls. The f count represents where the layers themselves are split up among the local nodes. In a real-world use case, the f count would depend on the local feature distribution, while k would be tunable. However, we will also simulate different feature distributions. In figure 2, we assume that the central server owns the labels of the dataset. Alternatively, the node owning the labels could act as central server, or a different architecture, with a central server acting as intermediary, can be implemented. In the set up of [11], the neural network is only recombined at the final activation function with a high k value. We designed our VSL set up with a tunable k as it could influence the predictive performance of the split network, due to the interconnectedness of the network nodes. A centralized neural network contains several layers of nodes, and between two consecutive layers, all nodes are connected. When a network is split vertically, this interconnectedness between the nodes is limited within the different locations. As the predictive strength of the neural networks depends on this interconnectedness and its ability to find feature interactions, a lack of interconnectedness could cause a model to perform less well. Therefore, earlier centralization of the model, with a lower k, might compensate a possible performance drop in certain instances. As previously mentioned, the vertical splitting of the networks will not always lead to a loss of performance, which implies the model architectures and data sets and their distribution over the clients influences response to the split architecture. To be able to estimate in advance in which situations split learning is a proper alternative to centralized learning, the relation between the distribution of the features and predictive performance should be investigated.

Figure 2b shows the architecture of vertical split learning, and the steps of the algorithm. The steps are as follows, and repeat until conversion:

1) The local nodes forward their samples through their partial models, up until the k layer
2) The outputs of the k layer are combined from the local node, and sent to the central server
3) The central server trains their part of the model using the samples available, and backpropagates until layer k
4) The backpropagated gradients are split and sent back to the local nodes, who backpropagate their partial model.

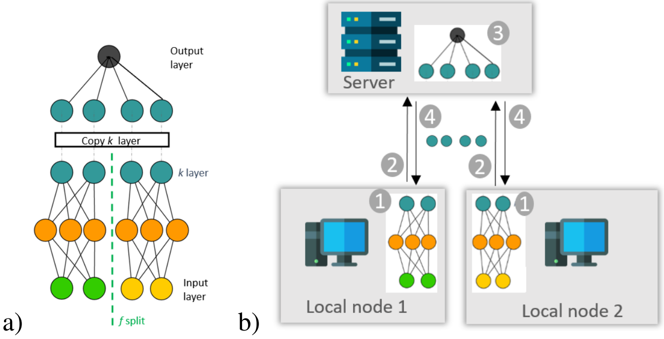

Fig. 2. a) This figure show the distribution of the neural network, distributed between the central server (top) and the local nodes (bottom). The values of k and f determine how the network is split b) VSL architecture

IV. Experiments

To investigate the performance of vertical split learning in a wide range of use cases and distribution set ups, several experiments were orchestrated in the following manner: 8 public datasets appropriate for vertical data partitioning were selected and 3 experiments were performed on these datasets, to illustrate both the influence of feature distribution and neural network distribution on predictive performance.

A. Datasets

For the experiments, 5 public medical datasets were selected based on the following criteria. The dataset needed to be publicly available and contain at least 500 samples to be suitable for a Multi-Layer Perceptron (MLP). Moreover, only datasets where a vertical partitioning would be a reasonable possibility were selected. As such, medical imaging datasets, or datasets covering a single event, like ICU datasets, were excluded. The datasets were furthermore selected to include a diverse set of medical use cases. Due to the limited accessibility of public medical datasets, we also include several non-medical datasets to provide a wider range of dataset diversity. To have relevance for vertical split learning, datasets with the purpose of fraud detection were chosen. Vertically partitioned data can be applicable here, as financial institutions might aim to join

Ref.	Dataset	# Samples	# Features	Frac. Pos. lab.
[17]	Cervical cancer	858	36	0.06
[18]	Early stage diabetes	520	17	0.62
[19]	Heart Disease	3749	15	0.15
[20]	Stroke	5110	12	0.05
[21]	Stroke Rehabilitation	1219	200	0.26
[22]	Provider Fraud	5410	28	0.09
[23]	PaySim	307511	11	<0.01
[24]	Insurance Claims	1000	43	0.25

TABLE I

DETAILS OF DATASETS. RIGHT COLUMN CONTAINS THE FRACTION OF POSITIVE LABELS PER DATASET.

forces in finding fraudulent behaviors of certain individuals. As such, the 8 datasets, summarized in table I are as follows:

- *Risk Factors of Cervical Cancer* Has been collected at Hospital Universitario de Caracas in Venezuela. The dataset, to predict the likelihood of cervical cancer, comprises of demographics, habits, medical history, and records of 858 female patients.
- *Early stage diabetes risk prediction* Contains information from direct questionnaires to patients of Sylhet Diabetes Hospital in Sylhet, Bangladesh. This questionnaire included demographic information and relevant medical history, is used to predict early stage diabetes.
- *Stroke rehabilitation* Originated from a year-long observational study by the University of Texas, following patients rehabilitating after a stroke incident. Includes demographics, medical history, data from questionnaires and medical examinations at multiple time points.
- *Framingham Heart disease:* Contains data from an ongoing observational study on the cardiovascular health of residents of Framingham, Massachusetts. The goal is to classify whether the patient has 10-year risk of coronary heart disease. The dataset provides the patients' demographics, and behavioral and medical risk factors.
- *Stroke* This dataset is used to predict whether a patient is likely to get a stroke based on features including demographics and medical history and risk factor behaviors.
- *Healthcare Provider Fraud Detection Analysis* is a dataset created for the task of predicting provider fraud; a form of organized crime which involves providers, physicians and beneficiaries making fraudulent claims. This particular dataset contained inpatient data, outpatient data and beneficiary demographics, preprocessed by [22].
- *Synthetic Financial Datasets For Fraud Detection (PaySim)* Synthetically dataset: generated using their proposed PaySim generator, which uses one month of real life financial logs from a mobile money service provider to generate realistic financial data. The data consists of financial logs, with the aim to predict fraudulent transactions.
- *Insurance Claim:* Likely synthetic dataset of insurance claims; there is no information available on Kaggle on the data's source. The data includes personal information such as hobbies and occupation.

B. Dataset preparation

Missing features were imputed by averages for continuous variables, and by the most common value for discrete variables. Features with an extremely high missing value count (more than 90%), were removed from the dataset. Because most datasets have a big class imbalance, a low sample count or both, we decided to aim to balance the datasets. The datasets were oversampled, undersampled using SMOTE or neither, depending on what was most suitable for the dataset. This was decided by which option gave the highest predictive performance per dataset after the parameter search.

C. Centralized model

Before the distributed experiments, a centralized MLP was designed and trained for each dataset. A parameter and hyperparameter search was performed to optimize the model. This search was performed using Optuna [25]. The proper configuration of the centralized model is essential, as the centralized models function both as the benchmark for the distributed models, as well as the foundation for the distributed model configurations. Therefore wide range of NN depth, width and types of nodes as well as hyperparameters were examined. In addition to the experiments described in section IV-E, several evaluations of the dataset and centralized model were performed to further illustrate their characteristics. These evaluations included a feature importance determination as well as feature correlation matrices. Feature importance was determined using [27], based on LIME [26]. The LIME method measures how much the features of the data sets cooperate to the final prediction for all the samples.

D. Experiment Set-up

In all experiments, only situations of data distribution among 2 local nodes were considered. For vertical distributions, data will in most situations not be spread out over a large set of locations. Moreover, the choice of two locations was also preferred, as it might provide a more straightforward image of the influence of feature distribution on the performance. The dataset samples were divided 3:1:1, for the training, validation and test set respectively. All tuning steps, including the complete training of the centralized model, were performed on the training set. With regards to the splitting on the MLP among the local nodes, the division of the MLP, in terms of nodes per layer, was proportional to the feature division among the nodes. The k layer, where the local partial networks were brought together depended on the depth of the network. All experiments were tested with all available layers as k, with the exception of the inputlayer. For the combination of outputs in the k layers, we chose concatenation as our strategy. This method was chosen above the other methods examined in [11], for its simplicity and as drop out of local nodes was not considered in these experiments. The focus of this paper is on predictive performance, and drop out of one of the two clients is not considered in these use cases. All experiments were performed in Pytorch version 1.4.0.

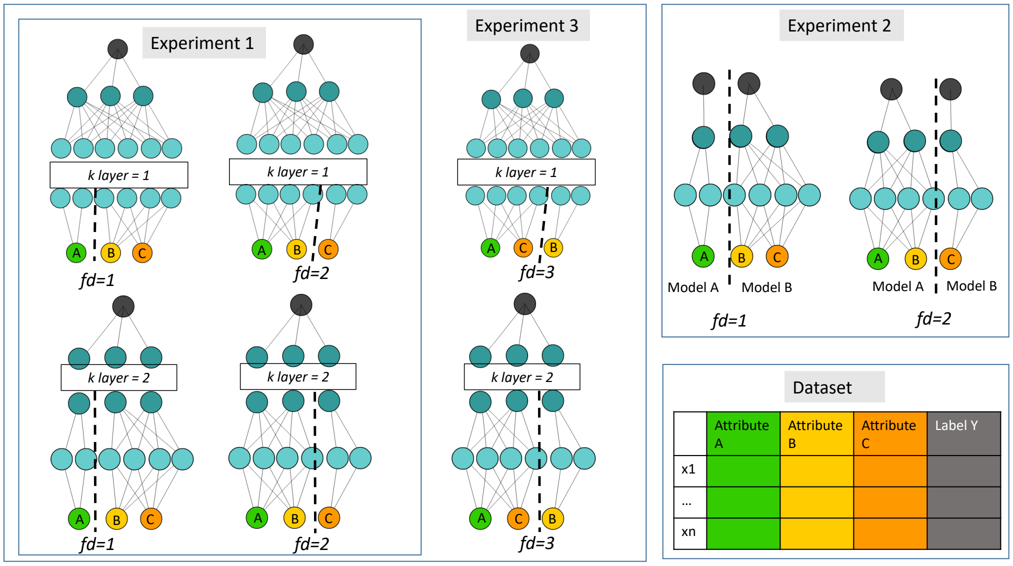

Fig. 3. Representation of the experiments as if they would be performed on a dataset with 3 features (orange, yellow and green), for a neural network with a depth of 2. The figures show all the subexperiments for each feature count *f*, feature distribution *fd* and connecting layer. The black dotted lines indicate the splits of the network among the different locations *k*

E. Performed Experiments

Three experiments were performed, illustrated in figure 3.

1) Split learning for all features, for one specific ordering of features: For the first experiment, all features are ordered in a specific ordering. This ordering is based on a logical distributions of the features. For example, in the stroke rehabilitation data set, the features are ordered chronologically. Then, after each feature in this ordering, a split distribution is established.

2) Centralized learning on the partial datasets: Using the feature distributions created in experiment 1, 2 smaller centralized models are developed with the 2 partial datasets of each feature distribution. The size of the model is proportionate with the size of the partial dataset. Then, the highest performance is picked. This experiment is performed in order to determine the necessity for any form of collaborative learning, and whether the partial datasets can explain the difference in performance.

3) Split learning for a set of features, with all possible combinations: Lastly, this experiment was performed for a more thorough overview of the influence of the interaction between the different features in the data set. Therefore, in this experiment we create a split NN for every possible feature split, not limiting ourselves to one specific ordering of features, as in the earlier two experiments. While not all of these combinations would be expected in a real life scenario, it limits possible bias in our results that could be caused by the chosen feature ordering. Due to an exponentially increasing set of combinations for each added feature, attempting this experiment with all features would be too computationally expensive. Therefore, this experiment was only performed with a set of the most important features: the top 10 was by LIME on the centralized model.

V. RESULTS

We chose to report predictive performance of the experiments in Area Under the Curve (AUC). Tt gives the most complete overview of accuracy, precision and recall. The results of the three experiments will be highlighted in their respective subsections.

A. Experiment 1

Figure 4 shows the results of experiment 1. The y-axis shows AUC, and the x-as the split in the feature distribution. The red horizontal line is the performance of the centralized model. As expected, the results of the VSL approach, compared to CL, is highly depended on the use case. In many of these evaluated use cases (a, b, c and h), we notice only small changes of predictive performance, both above and below the centralized model, as can be seen by the small differences in the AUC values on the y-axes. This could be by chance, caused by the relatively small datasets. This argues that in these use cases and feature distributions, VSL does not underperform compared to CL. However, this is not the case for all datasets. For both the cervical cancer (figure 4e) and the insurance claim (figure 4f) datasets, there is a noticeable drop in AUC: 0.85 to 0.62 (4e) and 0.96 to 0.78(4e). These drops show two of the same characteristics: they seem both depended on the feature distribution split, as well as the chosen *k* layer. Here, the higher the selected *k* layers (i.e. the more layers are trained locally among clients), the lower the AUC is on average. This indicates that the difference in predictive performance could be due to the lack of interconnectedness in these networks. However, this is not the case for all datasets. In the PaySim datasetfigure 4g, we also notice a few drops in predictive performance, but these occur when k=0. The drops disappear when the same feature distribution is trained with k=1.

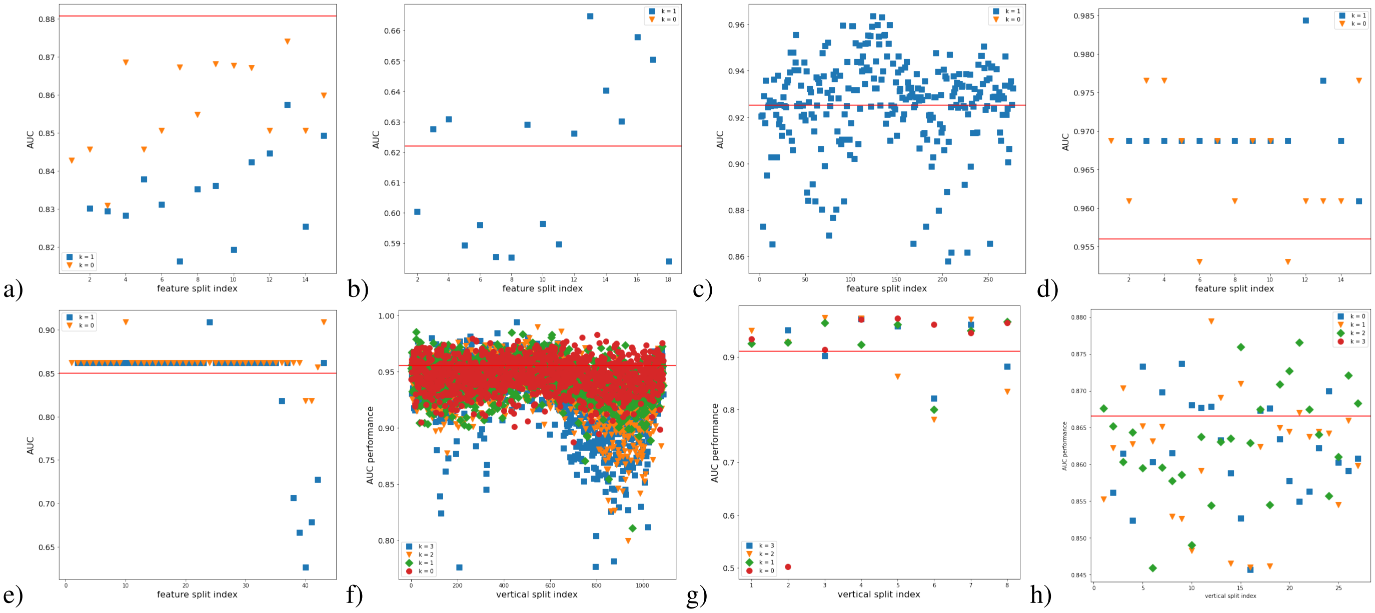

Fig. 4. Results experiment 1, where the x-axis shows the feature split f and the y-axis the AUC on the test set. The y-axes contain different scales. The horizontal red lines denote CL performance. a) Stroke, b) Heart Disease c) Rehabilitation d) Diabetes e) Cervical Cancer f) Insurance g) PaySim h) health care provider

B. Experiment 2

In experiment 2, we build a central NN on the partial datasets of each feature distribution. The same ordered feature distributions as in experiment 1 were selected. The results are shown in figure 5. In most of these figures (a,c,d,g,h), we can see the expected result. Here, the partial datasets with only very few features have low predictive performance, and as more features are added, the performance rises. In these cases one of the two partial datasets is not performing well, indicating VSL would lead to increase in predictive performance. In the cases where VSL did not function properly (e and f), the change in best performing network happens around the same value for f where performance dropped in experiment 1. Moreover, what we can see in all datasets, is that the addition of a few extra features does often not lead to an increase of performance. This can be due to low feature importance, or a high correlation of this feature to the other features in the dataset. This demonstrates that adding extra data does not necessarily lead to an increase in performance. Therefore it should be carefully considered whether split learning or centralizing data is even necessary in the first place.

C. Experiment 3

Figure 6 shows experiment 3, where all possible feature distributions of the top 10 features were evaluated for each dataset and each available k layer. As this leads to a multitude of distributions, we chose to quantify each distribution for its feature correlations in the partial dataset as we assume that the correlation between features is relevant to the performance of the split model. In this work we employ an evaluation metric based on Pearson correlation coefficients, correlation feature selection (CFS) measure [28]. When compared with the central performances in figure 1, most datasets show small drops of performance, but this could be explained by the selection of only the top 10 most important features. Again we see that most configurations perform well, with a few exceptions. However the k at which this occurs differs per dataset. In figure 6a, we see that a higher k generally leads to a lower AUC. In 6f, we see the opposite to experiment one, the lowest performance is obtained when k=0. In 6g, it happens around the middle k layers. All these drops occur when the CFS is near 0. This indicates that some subsets with low correlation could benefit from tuning to a different k. For the cervical cancer dataset (figure 6e), we do not see as clear of a drop in predictive performance as we saw in experiment 1.

VI. Discussion

The goal of this paper is to establish whether VSL can be an alternative to centralizing vertically partitioned data for deep learning. We tested a diverse set of vertically distributed datasets in several settings. In most of these situations, there was little difference in predictive performance between the centralized learning (CL) and VSL. This implies that, in most cases, VSL can serve as good alternative that eliminates the need for centralizing data while keeping similar predictive performance. Nevertheless, there are several situations in our results that did not meet these conclusions and require extra consideration. Firstly, when forming a decision on the necessity of VSL or CL in a certain use case, it is important to consider the results of experiment 2. Moreover, these experiments showed that for all datasets, split configurations can be found where one of the partial data sets does not under perform the CL on the complete data set. This implies

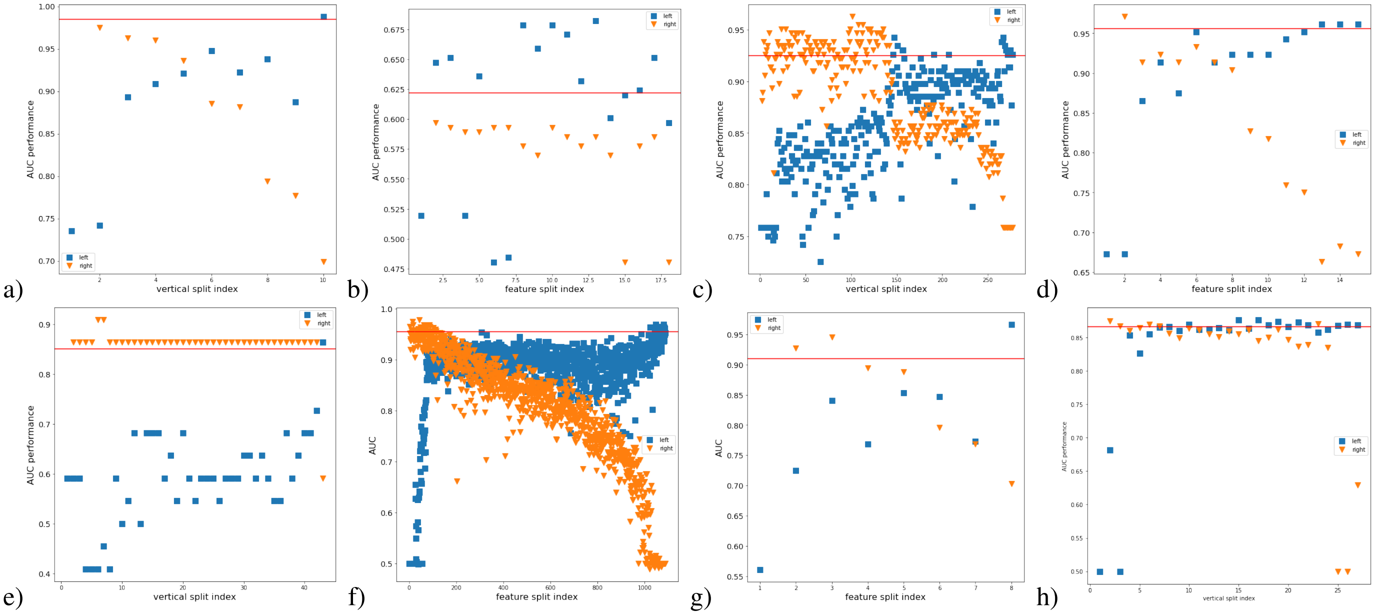

Fig. 5. Results experiment 2, where the x-axis shows the feature split *f* and the y-axis the AUC on the test set. Orange triangles show the results of the centralized models with features after *f*, blue with the features before *f*. The y-axes contain different scales. The horizontal red lines denote CL performance. Use cases are a) Stroke, b) Heart Disease c) Rehabilitation d) Diabetes e) Cervical Cancer f) Insurance g) PaySim h) health care provider

that the addition of the features of the matching partial data would not lead to an increase in predictive performance. Therefore, it remains essential to establish the necessity of any kind of distributed learning, as addition of extra features does not guarantee a better performance. Moreover, while in most cases VSL and CL performed comparably, there are several situations where VSL predictive performance did drop compared to CL. This is displayed in experiment 1, where both the cervical cancer as well as the simulated fraud data set have lower AUC in a specific set of feature configurations. In these cases, we see that a lower k limited the loss of performance, implying that the earlier the split models combined, the better the performance of VSL. This would fit with the expectations raised in section 2 as these earlier combined models would have a higher level of interconnectedness between the nodes. However, not all drops in VSL performance follow this pattern, as we see in the PaySim dataset, and further evidenced in experiment 3. We see a lower performance in several datasets that occur a lower k. We notice that this is dependent on the dataset and the feature distribution.

Whether VSL will perform comparably to CL is likely dependent on the dataset and the model architecture. While it performs well in most cases, and we were able to identify some patterns in the case of under performance, no generalized explanation was inferable for these experiments. To reliably use VSL to avoid centralization of data, a small suitability test could be conducted, where only a subset of data is centralized for a trial run of VSL. With a representative sample, possible failures could be detected. Moreover, more research is necessary to determine what characteristics of a dataset lie at the base of these occasional performance drops. To create a more comprehensive conclusion, a larger and more diverse set of data sets should be evaluated with these experiments. Currently, our evaluation method is applicable to other VFL methods, a larger set of uses cases could form a base to effectively compare their predictive performances. There are several limitations that should be noted, apart from the aforementioned limited set of datasets we could consider. In our experiments, we limited the configurations of splits to distributions of data of 2 clients. In some use cases, one could imagine situations in which data is distributed among more clients. However, due to the nature of vertically partitioned data, we do not expect situations where this number is greatly exceeded. Moreover, we did not apply a new hyperparameter search for each split configuration. It is plausible that this could negate the drop in predictive performance somewhat.

VII. Conclusion

In conclusion, VSL can be a viable alternative to centralized learning, as its performance is comparable in most situations. It occasionally underperforms depending on the use case, and more research is needed to clarify these cases, to create a better estimation on the possible gain of implementing VSL for a given use case. Another important factor for estimation of a payoff is efficiency, future work should aim to measure and improve the efficiency of VSL. Moreover, for cases of arbitrarily partitioning data or data where the samples on the different vertical datasets do not perfectly match, this technique does not suffice. Incorporating different techniques of distributed learning or entity matching into VSL could create a more robust system for a more diverse set of use cases. Lastly, the viability of techniques like VSL should be

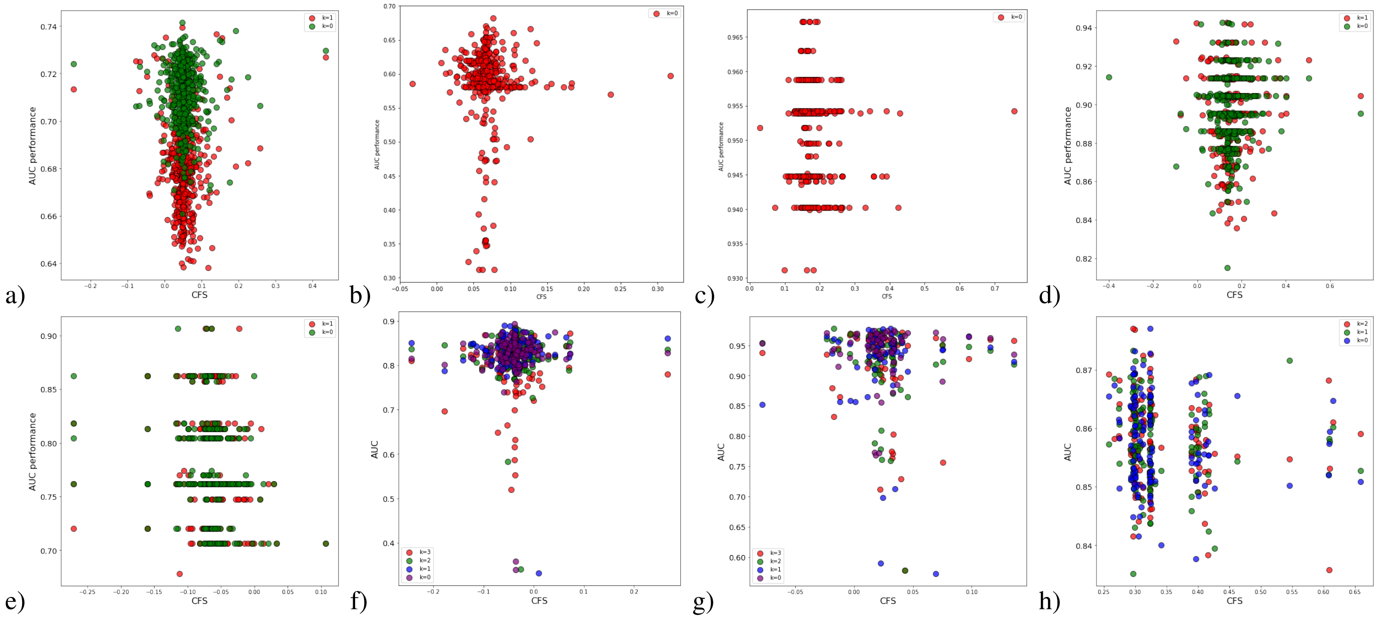

Fig. 6. Results experiment 3, where the x-axis shows the CFS of the reported feature distribution and the y-axis the AUC on the test set. a) Stroke, b) Heart Disease c) Rehabilitation d) Diabetes e) Cervical Cancer f) Insurance g) Paysim h) health care provider

investigated both in terms of risk of introducing bias as well as its legality under laws such as GDPR.

References

[1] He, J., Baxter, S.L., Xu, J. et al. 2019. The practical implementation of artificial intelligence technologies in medicine. Nat Med 25, 30–36. https://doi.org/10.1038/s41591-018-0307-0

[2] Dash S., Shakyawar S, Sharm, M. et al. 2019. Big data in healthcare: management, analysis and future prospects. J Big Data. 6:54 https://doi.org/10.1186/s40537-019-0217-0

[3] Kairouz P., McMahan B., Avent B., Bellet A., Bennis M. Bhagoji A, et al. 2021. Advances and Open Problems in Federated Learning. arXiv. 1912.04977

[4] General Data Protection Regulation (GDPR). 2018. General Data Protection Regulation (GDPR). https://gdpr-info.eu/

[5] Yang, K., Fan, T., Chen, T., Shi, Y., Yang, Q. (2019). A quasi-newton method based vertical federated learning framework for logistic regression. arXiv. 1912.00513

[6] Wu, Y., Cai, S., Xiao, X., Chen, G., Ooi, B. (2020). Privacy preserving vertical federated learning for tree-based models. arXiv preprint. 2008.06170 Xiv preprint. 1912.00513.

[7] Shen M., Zhang J., Zhu L., Xu K., Tang, X. 2019. Secure SVM training over vertically-partitioned datasets using consortium blockchain for vehicular social networks. IEEE Transactions on Vehicular Technology.

[8] Romanini D., Hall A., Papadopoulos P.,Titcombe T., Ismail A. , Cebere T. et al. 2021. PyVertical: a Vertical Federated Learning Framework for Multi-headed SplitNN. arXiv. 2104.00489.

[9] He Q., Yang W., Chen B., Geng Y., Huang L. 2020. TransNet: training privacy-preserving neural network over transformed layer. Proc. VLDB Endow. 13, 12, 1849–1862. https://doi.org/10.14778/3407790.3407794

[10] Chen, T., Jin, X., Sun, Y., Yin, W. 2020. Vafl: a method of vertical asynchronous federated learning. arXiv preprint http://arxiv.org/abs/2007.06081.

[11] I. Ceballos, V. Sharma, E. Mugica, A. Singh, A. Roman, P. Vepakomma, and R. Raskar. 2020. Splitnn-driven vertical partitioning. arXiv. 2008.04137

[12] Zhang Q., Gu B., Deng C., Huang H. 2021. Secure Bilevel Asynchronous Vertical Federated Learning with Backward Updating. arXiv. 2103.00958.

[13] Feng S., Yu H. 2020. Multi-Participant Multi-Class Vertical Federated Learning. arXiv. 2001.11154

[14] Sun J., Yao Y., Gao W., Xie J., Wang C. 2021. Defending against Reconstruction Attack in Vertical Federated Learning. arXiv. 2107.09898

[15] Luo X., Wu Y., Xiao X., Ooi B. 2021. Feature Inference Attack on Model Predictions in Vertical Federated Learning. International Conference on Data Engineering (ICDE):181-192, doi: 10.1109/ICDE51399.2021.00023.

[16] P. Vepakomma, O. Gupta, T. Swedish, and R. Raskar, Split learning for health: Dis-tributed deep learning without sharing. 1812.00564. http://arxiv.org/abs/1812.00564.

[17] Fernandes K., Cardoso J., Fernandes J. 2017. 'Transfer Learning with Partial Observability Applied to Cervical Cancer Screening.' Iberian Conference on Pattern Recognition and Image Analysis. Springer International Publishing.

[18] Islam M., et al. 'Likelihood prediction of diabetes at early stage using data mining techniques.' Computer Vision and Machine Intelligence in Medical Image Analysis. Springer, Singapore, 2020. 113-125.

[19] Ramachandran V., Benjamin E., Cupples L., Ellison R., Massaro J. 2020. Framingham Heart Study. Kaggle. https://www.kaggle.com/christofel04/cardiovascular-study-dataset-predict-heart-disea

[20] Kaggle. Stroke Prediction set. Kaggle. https://www.kaggle.com/fedesoriano/stroke-prediction-dataset

[21] Ostir G., Ottenbacher K., and Kuo Y. 2016. Stroke Recovery in Underserved Populations 2005-2006. Inter-university Consortium for Political and Social Research. https://doi.org/10.3886/ICPSR36422.v1

[22] Gupta R. 2019. Healthcare provider fraud detection analysis. Kaggle. https://www.kaggle.com/rohitrox/healthcare-provider-fraud-detection-analysis.

[23] Lopez-Rojas A., Axelssons. 2016. Paysim: A financial mobile money simulator for fraud detection. The 28th European Modeling and Simulation Symposium-EMSS.

[24] Sharma R. Insurance Claims (2019). Kaggle. https://www.kaggle.com/roshansharma/insurance-claim

[25] Akiba T., Sano S, Yanase T., Ohta T., Koyama M. 2019. Optuna: A next-generation hyperparameter optimization framework. arXiv. 1907.10902.

[26] Ribeiro M., Singh S., Guestrin C. 2016. Why should I trust you?: Explaining the predictions of any classifier. KDD '16, 1135–1144. https://doi.org/10.1145/2939672.2939778

[27] Lundberg, S., Lee, S. 2017. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems. 30.

[28] Hall M. 2000. Correlation-based feature selection for machine learning. University of Waikato. https://www.cs.waikato.ac.nz/ mhall/thesis.pdf