Federated data processing and learning for collaboration in the physical sciences

To cite this article: W Huang and A S Barnard 2022 *Mach. Learn.: Sci. Technol.* **3** 045023

IOP Publishing Mach. Learn.: Sci. Technol. 3 (2022) 045023 https://doi.org/10.1088/2632-2153/aca87c

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
9 August 2022

REVISED
13 November 2022

ACCEPTED FOR PUBLICATION
1 December 2022

PUBLISHED
16 December 2022

PAPER

Federated data processing and learning for collaboration in the physical sciences

W Huang and A S Barnard*

School of Computing, Australian National University, Acton ACT 2601, Australia
* Author to whom any correspondence should be addressed.

E-mail: amanda.s.barnard@anu.edu.au

Keywords: machine learning, federated learning, physical science, nanoparticles

Abstract

Property analysis and prediction is a challenging topic in fields such as chemistry, nanotechnology and materials science, and often suffers from lack of data. Federated learning (FL) is a machine learning (ML) framework that encourages privacy-preserving collaborations between data owners, and potentially overcomes the need to combine data that may contain proprietary information. Combining information from different data sets within the same domain can also produce ML models with more general insight and reduce the impact of the selection bias inherent in small, individual studies. In this paper we propose using horizontal FL to mitigate these data limitation issues and explore the opportunity for data-driven collaboration under these constraints. We also propose FedRed, a new dimensionality reduction method for FL, that allows faster convergence and accounts for differences between individual data sets. The FL pipeline has been tested on a collection of eight different data sets of metallic nanoparticles, and while there are expected losses compared to a combined data set that does not preserve the privacy of the collaborators, we obtained extremely good result compared to local training on individual data sets. We conclude that FL is an effective and efficient method for the physical science domain that could hugely reduce the negative effect of insufficient data.

1. Introduction

In recent years, scientists have been exploring possibilities of applying machine learning (ML) algorithms to a variety of applications in the physical sciences, including material science and nanotechnology, to acquire data-driven insights for the field [1–4]. The majority of methods to date have focused on supervised learning methods (such as regression) to predict structure/property relationships.

However, data sets in these domains tend to be small, particularly in nanotechnology, and this can limit the types of ML methods that are applicable, and the types of insight that can be extracted [5]. Reasons include the high cost of raw materials as well as the high cost of generating experimental results, due to the time involved and the instruments required [6, 7]. Creating synthetic computational data for specific materials can mitigate this issue, but this still requires high-performance supercomputers to simulate the results with extensive domain knowledge, which are time and energy-consuming to run [8–11]. The performance of many machine learning methods relies on the availability of large and reasonably comprehensive data sets, particularly neural networks. This mismatch in the types of data sets that are typical in materials science and nanotechnology, and the types of data sets required to train popular machine learning models, presents both a limitation and an opportunity.

Failing the ability to generate more data, one possible solution to this mismatch is to combine several different data sets from a group of different collaborators. Combining data can help scientists discover patterns beyond their specific experiments, overcome the selection bias inherent in individual studies, and collectively benefit from the increased amount of data and information. However, data from different sources tend to be inconsistent, since individual studies are focused on specific goals, with different numbers of sample instances, different numbers and types of feature variables, and different distributions. Additionally,

even when collaborating, sharing data may not be feasible due to privacy or confidentiality concerns, particularly in cases where potentially valuable intellectual property is involved.

Federated learning (FL) is a learning technique that has the potential to allow scientists to train models in a collaborative, decentralised, and privacy-preserving way without sharing their local data assets [12]. FL relies on collaborator hardware for the training process and uses a central coordinator (referred to as here as a *federator*) to aggregate sub-models into a global model, even when there are differences in the sample space (i.e. data identity) and feature space among the data sets. Previous applications for FL include Google virtual keyboard [13], and many more in medical science and financial technology (fintech) [14–17].

There are three different types of FL: horizontal, vertical, or federated transfer learning. A horizontal FL model takes advantage of the shared feature space among collaborators [18], making it similar to traditional distributed machine learning techniques where large data sets are deliberately distributed to different computing cores or devices by the coordinator for better parallelisation. This is consistent with the way ML is currently used in the physical sciences. However, since the federator does not have access to any of the local data, calculations including data pre-processing and model training are done locally on collaborator devices and the aggregation process can be different depending on the underlying FL model. This makes sharing knowledge of data cleaning or feature engineering such as data dimension reduction a challenge that is particularly relevant to this domain.

Another challenge when applying FL to science and technology is the overhead caused by model parameter transmission between the central federator and decentralised collaborators. This issue is similar to distributed ML where large volumes of data are transferred between machines on a frequent basis. Researchers have taken the approach of reducing the dimension of those data sets, either through feature selection or feature extraction, to cope with this overhead. This is also a standard method for reducing calculation time in ML tasks when confronted with data sets with large feature space typically used throughout (nano)materials informatics [19]. However, there has been little attention given to how a FL pipeline can incorporate dimension reduction undertaken by a collaborator. In distributed ML tasks dimension reduction can be performed centrally before distributing data to different calculation cores, but this is impossible in a FL workflow since no data is shared.

Therefore, a standard procedure for data processing and dimension reduction in FL context is essential to the application of FL in this domain, as an effective dimension reduction process could significantly improve the communication efficiency of a FL pipeline, making it more widely applicable [20]. In this paper we address this issue and propose a new method, FedRed, to allow collaboration-based dimension reduction in a federated data pre-processing phase (FDPP) for horizontal FL frameworks using a modified principle components analysis (PCA). The purpose of the FDPP is to allow different collaborators to acknowledge differences or similarities in their data distribution and act on them before training, so that collaborator data sets can be better prepared to suit the specific learning task. This method is developed and integrated into a FL workflow demonstrated on a group of different nanoparticle data sets. The results show that FL is an effective tool for collaboration in physical science and by adopting FedRed the characteristics of individual data sets (such as size and dimensionality) can be accounted for or emphasised.

2. Methodology

The underlying horizontal FL pipeline developed in this study was designed to execute a learning task with four steps, including 'connection establishing', 'meta-data exchanging', 'federated data pre-processing', and 'federated model training', where the third step corresponds to our proposed method, FedRed. When the pipeline terminates, a shared global model is generated. Pseudo-code for the federator and collaborator is provided below in algorithms 1 and 2. As mentioned above, this step draws on PCA to reduce the data sets in a consistent way that preserves as much local information as possible. This pipeline was implemented in Python, and is available online [21].

2.1. Principle component analysis

PCA aims to minimise the MSE caused by projecting an original data set $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\} \in \mathbb{M} \times \mathbb{N}$ onto a hyper-plane in dimension K, where $K << N$, by maximising the variance captured by the principle components (PCs), denoted by $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k$. The first PC, $\boldsymbol{u}_1$, is calculated by maximising the Lagrangian given by:

$$\boldsymbol{u}_1^T \boldsymbol{S} \boldsymbol{u}_1 + \lambda_1 (1 - \boldsymbol{u}_1^T \boldsymbol{u}_1) \tag{1}$$

where λ_1 is a factor and S is the covariance matrix defined as:

Algorithm 1. Federator Pipeline

```
InitialiseFederator(model, hyperparameters);
ListenOnPort();
while WaitForAllClients do
    if NewClientConnections then
        SaveClientInfo();
        AskClientToWait();
    end if
end while
forall ConnectedClients do
    Send(model, hyperparameters);
    AskClientToProceed();
end forall
FederatedDataPreprocess();
for i := 0 ... CommunicationRounds do
    while WaitForClientToReport do
        if ClientReports then
            SaveClientModelInfo();
            AskClientToWait();
        end if
    end while
    newGlobalModel ← AggregateClientsModelInfo();
    forall ConnectedClients do
        Send(newGlobalModel);
    end forall
end for
End();
```

Algorithm 2. Collaborator Pipeline

```
InitialiseClient();
if ConnectedToHostPort then
    SendSelfInfo();
end if
localModel ← GetFederatorMessage();
WaitForFederatorToProceed();
FederatedDataPreprocess();
for i := 0 ... CommunicationRounds do
    modelInfo ← Train(localModel, localData);
    SendToFederator(modelInfo);
    WaitForFederatorToProceed();
    localModel ← GetFederatorMessage();
end for
End();
```

$$\boldsymbol{S} = \frac{1}{K}\sum_{k=1}^{K}(\boldsymbol{x}_k - \bar{\boldsymbol{x}})(\boldsymbol{x}_k - \bar{\boldsymbol{x}})^T \tag{2}$$

$\boldsymbol{u}_1$ is calculated to be an eigenvector of $\boldsymbol{S}$ while λ_1 is the largest eigenvalue. Similarly, other PCs are calculated iteratively, and the final projection matrix we use to reduce the dimensionality of $\boldsymbol{X}$ can be denoted as $\boldsymbol{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k\} \in \mathbb{K} \times \mathbb{N}$. The ratio of variance captured (explained) can be calculated by divide the sum of eigenvalues of the PCs by the sum of all eigenvalues of the covariance matrix, which can be represented as:

$$R = \frac{\sum_{i=1}^{K}\lambda_i}{\sum_{j=1}^{N}\lambda_j}. \tag{3}$$

2.1.1. Federated PCA

The federated adaptation of PCA assumes each data set has a unique distribution the final projection must reflect this uniqueness to some degree. To have weighted contribution from each collaborator, we sum all the projection matrices $\{\boldsymbol{U}_1, \boldsymbol{U}_2, \ldots, \boldsymbol{U}_n\}$ and take the average as:

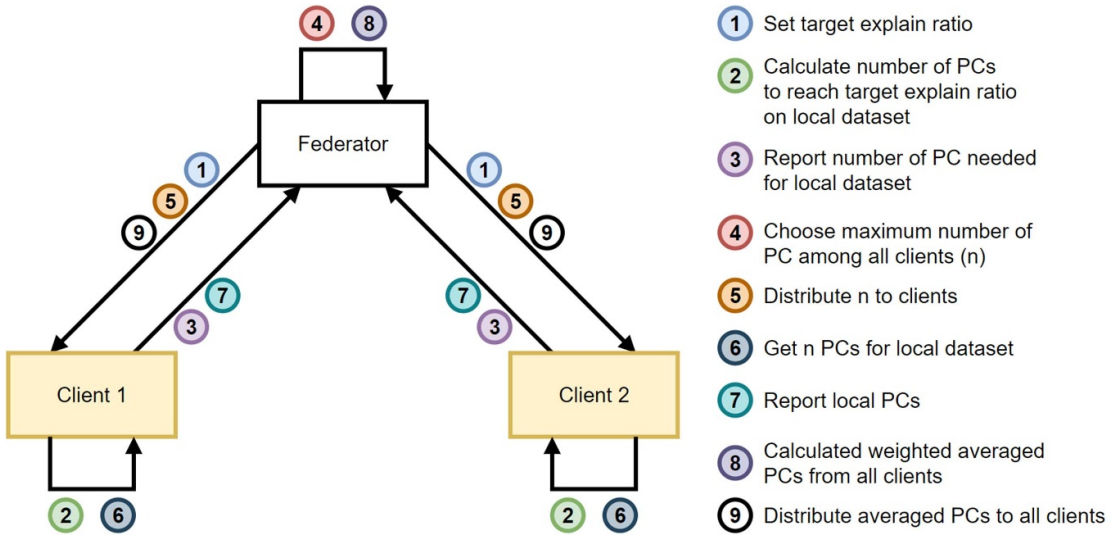

Figure 1. Multi-round communication for the federated PCA process with steps explained.

$$\overline{\boldsymbol{U}} = \frac{1}{n}\sum_{i=1}^{n} w_i \boldsymbol{U}_i \tag{4}$$

where $\overline{\boldsymbol{U}}$ is the final projection matrix for all collaborators.

Weighting can be done in a number of ways, including intuitive weighting metrics such as averaging, the size of a data set, or the explained variance ratio of a local PCA process. When using data sets sizes as the metric, $w_i = \frac{|\boldsymbol{X}_i|}{\sum_{j=1}^{n}|\boldsymbol{X}_j|}$; and when using explained variance ratio, $w_i = \frac{R_i}{\sum_{j=1}^{n} R_j}$, where R_i is the explain ratio defined in equation (3). Alternatives are possible, depending on the goals of the combined study.

In the present work we implemented federated PCA as a multi-step pipeline within the horizontal FL framework as shown in figure 1.

After receiving the 'averaged' PCs, individual collaborators use them to perform dimension reduction locally to reduce their local data set to a more general state. In our experiments, we compared the effectiveness of all three of the intuitive metrics mentioned above (*average*, *size*, and *explained variance ratio*). *Averaging* refers to cases where the final reduction matrix is the average of all locally generated reduction matrices. *Size* refers to cases where a weight that is equal to the percentage of data instance number of a data set among all data sets is applied to each of the reduction matrices before averaging. Similarly, the *explained variance ratio* refers to cases where the weight is the ratio of the explained of variance of a local PCA process. Since we set a target explained variance ratio for finding the number of principle components needed, a smaller data set typically needs a small number of PCs while a larger data set will need more. We take the largest number among all clients such that all clients reach the target explained variance ratio (allowing some to exceed it). By having more PCs than needed, smaller data sets have a higher explained variance ratio than larger data sets. By applying this value as a weight for a reduction matrix, smaller data sets can contribute more to the final reduction matrix, mitigating the possibility of collaborators with larger data sets biasing the final model.

2.1.2. Federated feature selection

Aside from feature extraction methods such as PCA, feature selection methods can be adjusted to a similar pipeline. Instead of setting a target explained variance ratio, the number of features can be agreed by collaborators at the outset. The mutually selected set of features could be chosen by voting and a threshold of vote numbers with potential weights on client votes. This approach is not presented here, but the overall workflow is the same.

2.2. Base models

We chose two models, linear regression and artificial neural network (ANN), as base models for the FL pipeline. Linear regression is well suited to finding linear relationships, while ANN adds possibilities of discovering non-linear patterns. However, when given insufficient training data, it is hard for both models to learn non-trivial patterns.

Table 1. Data sets and descriptions.

Name	Method	Condition	Temperature	Particle size	Instances
Pt-large (collab0)	Molecular Dynamics (MD)	Both Kinetically and Thermodynamically Limited	273 K 973 K	>6000 atoms	341
Pt-small (collab1)	Molecular Dynamics (MD)	Both Kinetically and Thermodynamically Limited	273 K 973 K	<3000 atoms	754
Ag-classical (collab2)	Molecular Dynamics (MD)	Kinetically Limited	273 K to 973 K	157 atoms to 14115 atoms	4000
Pd-highT (collab3)	Molecular Dynamics (MD)	Both Kinetically and Thermodynamically Limited	>773 K	158 atoms to 16262 atoms	1100
Ag-quantum (collab4)	Density Funcitonal Tight Binding (DFTB)	Thermodynamically Limited	0 K	13 atoms to 2947 atoms	425
Au-thermo (collab5)	Molecular Dynamics (MD)	Thermodynamically Limited	323 K to 923 K	236 atoms to 13804 atoms	2400
Au-kinetic (collab6)	Molecular Dynamics (MD)	Thermodynamically Limited	273 K to 973 K	236 atoms to 14277 atoms	1600
Pd-lowT (collab7)	Molecular Dynamics (MD)	Both Kinetically and Thermodynamically Limited	<373 K	236 atoms to 16083 atoms	1000

2.3. Evaluation metrics

We claim FL is an effective framework in this study by comparing the predictive behaviour of a global model's performance on data sets compared to a client's locally trained model's performance on another data set. The effect of the FedRed pipeline can be measured by comparing the reduction matrix generated on a combined data set with that generated via collaboration. We expect the collaboration-generated matrix to have a closer distance to the matrix generated when we have all collaborator data in one data set than when they remain separated. Upon this observation, we can claim that the FedRed pipeline is better in capturing the overall trend of a particular field.

In addition to this, each collaborator could be assigned a specific weight to make the final reduction matrix closer to, or farther away from, its locally generated one, such that the final reduction matrix reflects trend captured in this data set to a lesser or greater degree. After providing a weighting metric, we can compare whether the final model has the expected predictive performance on each of the data sets to validate the effectiveness.

2.4. Data sets

Our work uses eight computational nanoparticle data sets as described in table 1, each produced using either quantum mechanical electronic structure or classical molecular dynamics (MD) high-performance computer simulations. The data sets were not generated as part of this study, and are all publicly available (see Data Availability Statement).

Each data set has been been characterised using the same set of 105 features and one continuous numerical target label (the formation energy, in electron volts). There are no categorical features, and no instances are shared between the datsets. Each of these data sets is measured either on different materials (silver, gold, palladium, and platinum), was simulated using different method (i.e. MD and DFTB), was simulated under different conditions (thermodynamically-limited or kinetically-limited, high temperature or low temperature), or with different nanoparticle sizes (i.e. number of atoms) to reflect data diversity between collaborators. As we will show, FL is a suitable framework for tasks in this area, and by using FedRed the performance of the FL pipeline can be significantly improved.

3. Results and discussion

3.1. Baseline

To establish a baseline we joined all eight data sets together into one large comprehensive data set and used it to trains a linear regressor and an ANN. The result is shown in figure 2. Due to large training sample size, the loss in the baseline is very small (approximately 1.32×10^{-6} per instance on average), meaning the model has extremely good prediction ability with the presence of all available data. It is expected that result generated using FL techniques would be inferior to this baseline by a small amount due to statistical heterogeneity

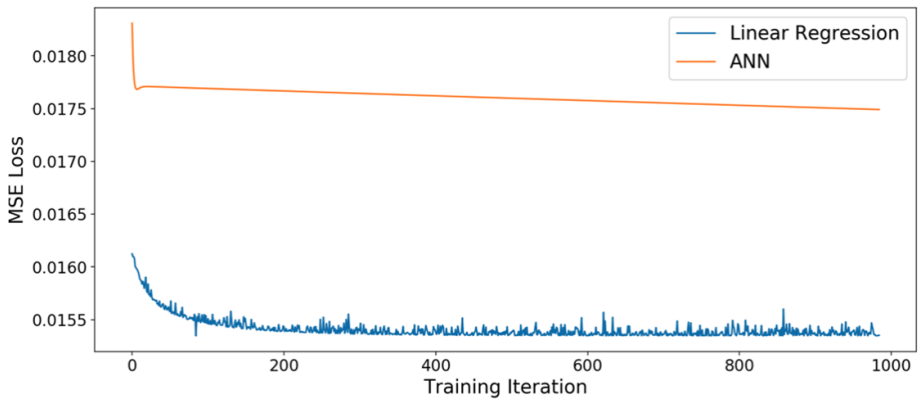

Figure 2. Loss for training on the joint data set over 1000 training iterations.

[12, 20, 22], but it is extremely advantageous to have a high performing baseline from which to assess the final loss.

It may appear counter-intuitive to see an ANN performing worse than a linear model in figure 2, however, the losses generated by both models are extremely small. Both models perform very well on this data set, which clearly has an underlying linear pattern. In such cases a more complicated model, such as ANN, will have over-fitting issues, which can lead to higher testing losses.

3.2. FL effectiveness

A series of experiments were carried out to validate the effectiveness of the FL framework in our study, comparing models trained only on the local data set of each collaborator with models trained collaboratively using FL. During the FL training process, Federated PCA was applied with even contributions from all collaborators.

Firstly, a separate, local model was trained on each of the eight data sets individually (representing a non-collaborative situation). Each of the eight locally generated models was then tested on all test sets with per-instance losses of each model for each data set plotted in figure 3(a) for the linear regressor and figure 3(b) for the ANN, where the legend indicates which data set the model was trained on. This could be considered an unsophisticated type of collaboration, sharing a model for others to use on their local data without retraining.

For the FL generated model, each of the eight data sets was treated as a different collaborator and a model was trained collaboratively with the same hyper-parameters as the local training approach. We plotted the per-instance losses on testing data sets held aside prior to the training in figure 4.

We observe a significant difference between the two training approaches. In the case of the individual locally trained models, the predictive ability was extremely unstable, even on the same data set they trained on (see figures 3(a) and (b)). The MSE loss varies significantly between data sets, however, the pattern of loss change with different models on each data set is similar. For example, Pt-large (see table 1) has the largest loss for all eight data sets while Pt-small has the lowest loss regardless of the model. These two data sets were generated using the same computational method and simulation approach, with the same temperature range but different nanoparticle size ranges. This indicates there may be some generic pattern that has been captured by different models. In addition to this, all losses have a relatively high value. The worst prediction, with models trained on Pd-highT data and tested on Pt-large data, yielded more than 10 000 MSE loss. This can be due to the two data sets being significantly different but also because there is insufficient data in Pd-highT to train on to achieve reasonable performance on Pt-large. Similar trends can be found when comparing results using ANN as the base model (see figure 3(b)). Both observations on this result suggest that the unsophisticated collaboration approach of training on one data set and using the model on another is unwise, even if it is compelling to researchers.

For the shared model generated from the FL framework in figure 4, the losses are extremely small and only vary in a small range. This indicates that in this use case, FL is capable of generating a much better global model that can be confidently used on all data sets compared to training using only local data sets.

Comparing with the baseline result in figure 2, however, we observe an increase in losses generated from models trained using the FL pipeline. This is an expected behaviour due to statistical heterogeneity between data sets from different collaborators, and the small size of some of the collaborator data set with respect to the comprehensive combination. Although it is generally desirable to have lower loss, it is not always possible

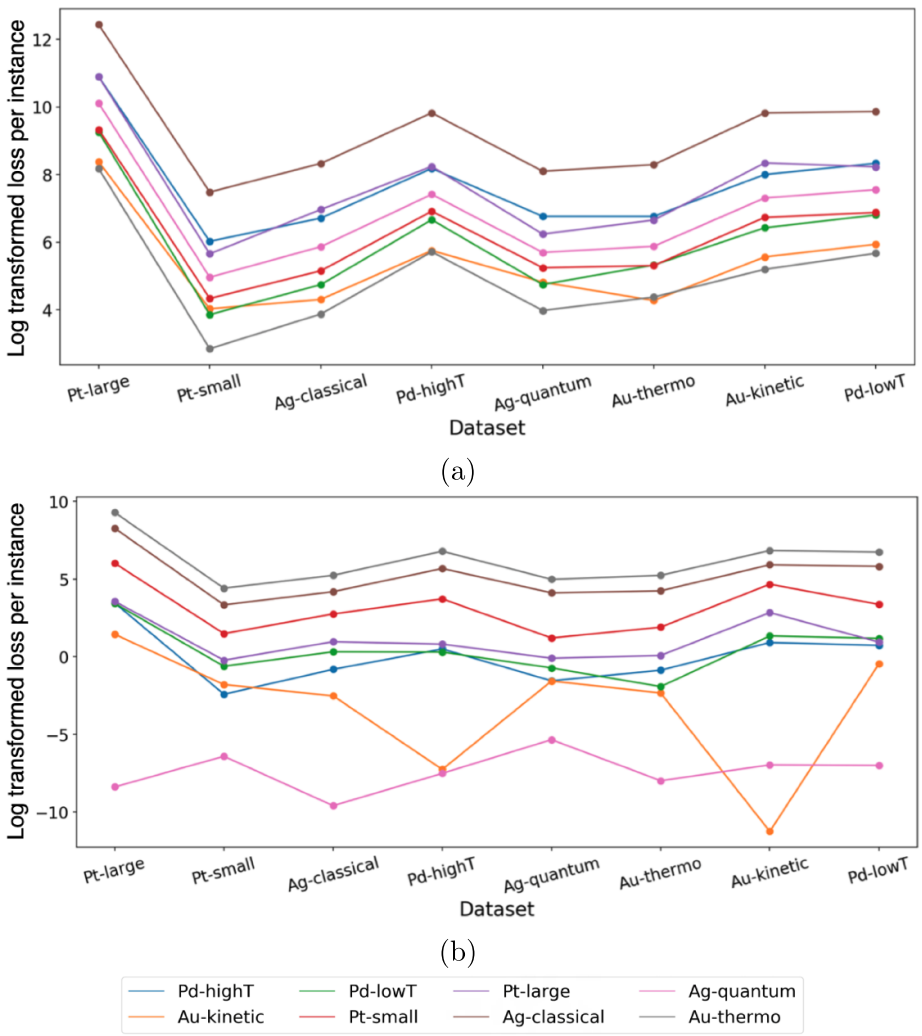

Figure 3. Locally generated (a) linear regression and (b) ANN model effects on each data set. Note that the *y*-axis differs to manage low losses using the ANN.

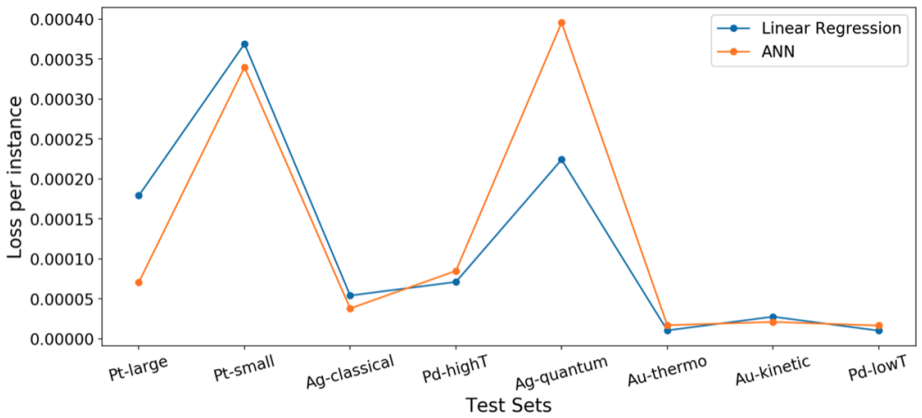

Figure 4. Effects of the federated learning generated models on testing sets.

to combine data sets (perhaps due to IP and privacy reasons). Fortunately losses from the FL pipeline is already extremely low compared to individual local training, meaning the pipeline is effective in this scenario. While the performance with a combined data set is higher, if this is not practical the losses due to federation are small when compared to no collaboration at all.

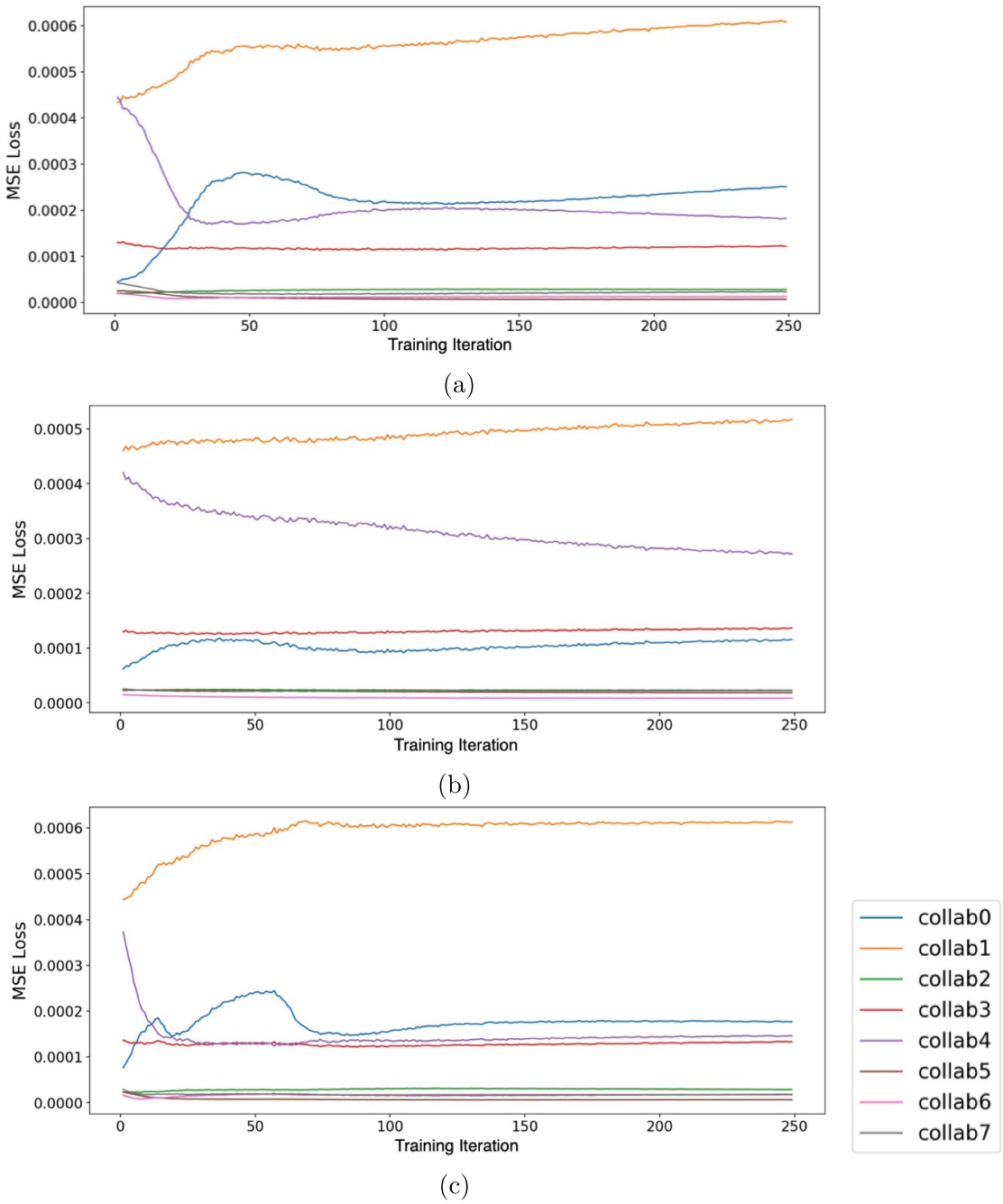

Figure 5. Per-collaborator, per-instance losses using different PC aggregation metrics, including (a) using weighting by the average, (b) weighting by the data set size, and (c) weighting by the ratio of the explained variance of each collaborator dimension reduction. The base model is linear regression for these plots.

3.3. FedRed effectiveness

To explore the effectiveness of our federated dimension reduction method, FedRed, we measured each collaborator testing loss during the training phase. From figure 5(a), we can see the collaborator testing error changes when we use the average of the reduction matrices generated from all collaborator data sets. The scores are similar to those seen in figure 4 when models converge. Figures 5(b) and (c) shows the same measurements weighed by the data set size and ratio of the explained variance as PC aggregation metrics, respectively. We can observe for Collaborator4, owner of the Ag-quantum data set, its per-instance loss is approximately 2×10^{-4} and 3×10^{-4} respectively when using average and size weightings. This reduces to 1.5×10^{-4} when we use the explained variance ratio. We believe this is a result of the change in reduction matrix because the Ag-quantum data set has the second smallest number of sample instances (only 3.6% of the whole population) and the final reduction matrix is closer to that of Ag-quantum, resulting in an model capable of better performance on this data set.

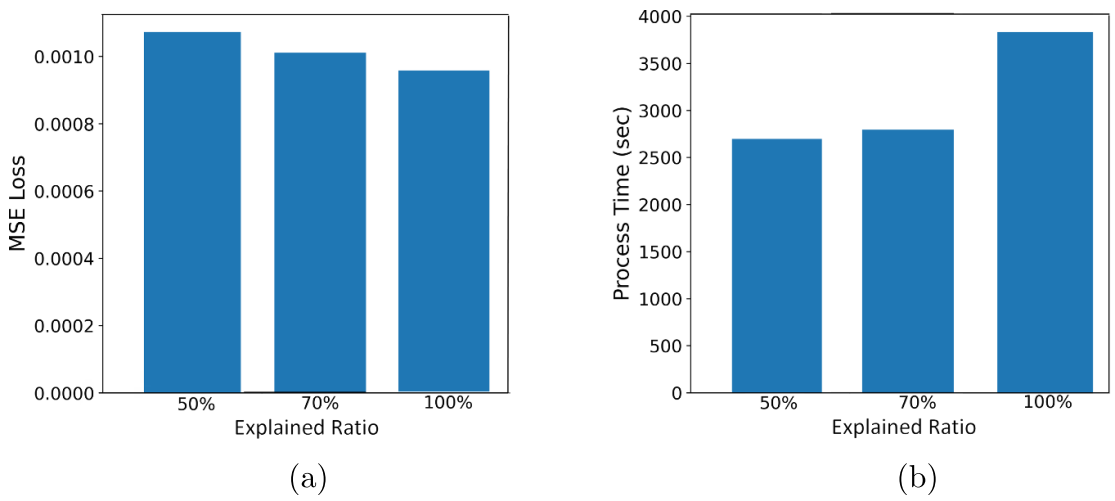

Figure 6. Impact of more aggressive dimension reduction, including (a) MSE loss generated by models with different explain ratios, and (b) processing time of 200 training iterations by models with different explain ratios. 100% explain ratio means no PCA applied and all data information is preserved.

However, we also observed behaviour that is unexpected. Collaborator0, owner of the Pt-large data set, has a much higher loss when we use the explained variance ratio aggregation metric. Collaborator0 has the smallest data set, but its behaviour is opposite to that of Collaborator4, with the Ag-quantum data set, who has the second smallest. This indicates a much more complicated relationship where the size of the set is insufficient to anticipate performance. Comparing these two data sets (see table 1), we can see they consist of two different chemical elements, but were also generated using two different computational methods, under different conditions and temperatures, and having totally different nanoparticle sizes. While the data sets are both relatively small, they differ in every other way possible. From a domain expert perspective, we believe the difference in training behaviour are a function of the physical differences of the nanoparticles and the computational difference of the simulations, highlighting that the FL pipeline, and the addition of FedRed, preserves the scientific information and does not homogenise the output. This capability may provide domain experts with clues when studying the physical properties of these nanoparticles, by discussing model outcomes with the other collaborators, but without ever sharing any data.

Despite possible explanations and implications to the domain, in this example all three models in figure 5 have similar cumulative losses among all testing instances. Therefore, it is hard to tell which model performs best on all data sets. However, this is not necessarily the case for every use case and data sets. It is recommended that researchers experiment with different settings before choosing their desired method. Future work to develop new evaluation methods specific to federated learning (as we have done here for federated dimension reduction) would be beneficial to the community.

Another aspect to consider is the affect of using different explained ratios. Figure 6(a) shows the cumulative loss of all clients when tested on models generated with different explained ratios (different numbers of components). In this experiment, higher explained ratio has lower loss. Due to the pattern of federation (figure 5) being largely unaffected by the number of components, the cumulative loss for these three experiments are similar. This may not necessarily be the case for all applications, but it encourages using fewer components and exploiting the other benefits. In contrast, the processing time shows a significant advantage on cases when FedRed is applied. Figure 6(b) shows the processing time of 200 iterations on eight collaborators and it is much faster when dimension reduction is used, though this advantage saturates due to the federation overhead. Both of these results indicate a general effectiveness of FedRed in addition to its advantages on specifying the aggregation metric.

In previous literature [23] distributed PCA has been well discussed in terms of convergence guarantees and communication efficiency optimisation. We acknowledge that the FedRed method has the same essence of obtaining an ‘average consensus’ among collaborators, meaning FedRed is in alignment with the previously reported results. However, there are two advances over the existing work. Firstly, FedRed could be used in a feature selection scenario in addition to PCA as described in section 2.1.2. Secondly, we emphasise there are possible domain implications that comes with different weighting metrics, so our approach gives researchers the opportunity to apply there domain knowledge or technical preferences, depending on the circumstances of the collaborations.

There are possible ways of improving the FedRed scheme that have not been discussed here. Iteratively improving the PCs on a local scale before each round of aggregation could be desirable propagandising local features, but introduces risks such as over optimisation (potentially resulting in increased over-fitting) and a reduction in global generality. Alternative feature extraction methods other than PCA can also be explored and analysed in this setting. These are interesting research questions that are worth exploring in future works.

At this point it is prudent to point out that in this demonstration we have only considered situations where all collaborators have data sets with the same features, i.e. using horizontal FL, as there is some consistency in the way physical scientists characterise chemicals, materials and nanoparticles. However, in some cases features in collaborators data sets may be different, making vertical FL and federated transfer learning [18, 24] more suitable to help learning new features from others' data sets. This is also a recommended direction for future research.

4. Conclusion

In this paper we have confirmed the applicability of horizontal federated learning to data sets typical of materials informatics and nanoinformatics, and designed and demonstrated a new federated dimension reduction scheme within the federated learning pipeline to help prepare heterogeneous data sets for learning. Upon applying federated learning to the nanoparticle case study in this paper, we observed significant predictive performance gains during testing. FedRed, helps to reduce the statistical heterogeneity of different data sets by providing a shared reduction matrix to all collaborators contributing to the federated model, while preserving the scientific differences of each local data set. The effectiveness of the method was validated, and was shown to outperform the crude but compelling reuse of individually trained models on alternative data sets. In general the use of the explained variance ratio for weighing the contribution from collaborating data sets brings the predictions closer to the benchmark of a fully shared, non-private, integrated data set.

Given the privacy-preserving nature of federated learning, we believe the incorporation of the new federated dimension reduction scheme is an effective framework towards solving the problem of data insufficiency in this field by providing researchers a means of collaboration using methods they can trust.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: http://doi.org/10.25919/5d395ef9a4291; http://doi.org/10.25919/5e30b5231c669; http://doi.org/10.25919/5d3958d9bf5f7; http://doi.org/10.25919/5d3958ee6f239; http://doi.org/10.25919/5d22d20bc543e.

Acknowledgment

Computational resources for this project were supplied by the National Computing Infrastructure (NCI) national facility under Grant p00.

Conflict of Interest

There are no conflicts of interest to declare.

ORCID iD

A S Barnard https://orcid.org/0000-0002-4784-2382

References

[1] Schmidt J, Marques M R G, Botti S and Marques M A L 2019 Recent advances and applications of machine learning in solid-state materials science *npj Comput. Mater.* **5** 1–36
[2] Ramprasad R, Batra R, Pilania G, Mannodi-Kanakkithodi A and Kim C 2017 Machine learning in materials informatics: recent applications and prospects *npj Comput. Mater.* **3** 1–13
[3] Rajan K 2005 Materials informatics *Mater. Today* **8** 38–45
[4] Rajan K 2015 Materials informatics: the materials "gene" and big data *Annu. Rev. Mater. Res.* **45** 153–69
[5] Barnard A S, Motevalli B, Parker A J, Fischer J M, Feigl C A and Opletal G 2019 Nanoinformatics and the big challenges for the science of small things *Nanoscale* **11** 19190–201
[6] Miracle D B, Li M, Zhang Z, Mishra R and Flores K M 2021 Emerging capabilities for the high-throughput characterization of structural materials *Annu. Rev. Mater. Res.* **51** 131–64
[7] Green M L *et al* 2017 Fulfilling the promise of the materials genome initiative with high-throughput experimental methodologies *Appl. Phys. Rev.* **4** 011105

[8] Luo S, Li T, Wang X, Faizan M and Zhang L 2021 High-throughput computational materials screening and discovery of optoelectronic semiconductors *Wiley Interdiscip. Rev.-Comput. Mol. Sci.* **11** e1489
[9] Sun B, Fernandez M and Barnard A S 2016 Statistics, damned statistics and nanoscience–using data science to meet the challenge of nanomaterial complexity *Nanoscale Horiz.* **1** 89–95
[10] Colón Y J and Snurr R Q 2014 High-throughput computational screening of metal–organic frameworks *Chem. Soc. Rev.* **43** 5735–49
[11] Afzal M A F, Browning A R, Goldberg A, Halls M D, Gavartin J L, Morisato T, Hughes T F, Giesen D J and Goose J E 2020 High-throughput molecular dynamics simulations and validation of thermophysical properties of polymers for various applications *ACS Appl. Polym. Mater.* **3** 620–30
[12] McMahan B, Moore E, Ramage D, Hampson S and Aguera y Arcas B 2017 Communication-efficient learning of deep networks from decentralized data *Artificial Intelligence and Statistics* (PMLR) pp 1273–82
[13] Hard A, Rao K, Mathews R, Ramaswamy S, Beaufays F, Augenstein S, Eichner H, Kiddon C and Ramage D 2018 Federated learning for mobile keyboard prediction (arXiv:1811.03604)
[14] Lee J, Sun J, Wang F, Wang S, Jun C-H and Jiang X 2018 Privacy-preserving patient similarity learning in a federated environment: development and analysis *JMIR Med. Inform.* **6** e7744
[15] Rieke N *et al* 2020 The future of digital health with federated learning *npj Digit. Med.* **3** 1–7
[16] Long G, Tan Y, Jiang J and Zhang C 2020 Federated learning for open banking *Federated Learning* (Berlin: Springer) pp 240–54
[17] Liu Y, Huang A, Luo Y, Huang H, Liu Y, Chen Y, Feng L, Chen T, Yu H and Yang Q 2020 Fedvision: an online visual object detection platform powered by federated learning *Proc. of the AAAI Conf. on Artificial Intelligence* vol 34 pp 13172–9
[18] Yang Q, Liu Y, Chen T and Tong Y 2019 Federated machine learning: concept and applications *ACM Trans. Intell. Syst. Technol.* **10** 1–19
[19] Motevalli B, Parker A J, Sun B and Barnard A S 2019 The representative structure of graphene oxide nanoflakes from machine learning *Nano Futures* **3** 045001
[20] Li T, Sahu A K, Talwalkar A and Smith V 2020 Federated learning: challenges, methods and future directions *IEEE Signal Process. Mag.* **37** 50–60
[21] Huang W FederatedLearning version 1.0.0 (available at: https://github.com/jacobvons/FederatedLearning)
[22] Yuan B, Song G and Xing W 2020 A federated learning framework for healthcare IoT devices (arXiv:2005.05083)
[23] Liang Y, Balcan M-F F, Kanchanapally V and Woodruff D 2014 Improved distributed principal component analysis *Advances in Neural Information Processing Systems* vol 27
[24] Zhang C, Xie Y, Bai H, Yu B, Li W and Gao Y 2021 A survey on federated learning *Knowl.-Based Syst.* **216** 106775