DiagNet: Machine Fault Diagnosis Using Federated Transfer Learning in Low Data Regimes

Zexi Yao[1], **Chao Jin**[1], **Mohamed Ragab**[1,2], **Khin Mi Mi Aung**[1], **Xiaoli Li**[1]

[1] Institute for Infocomm Research, A*STAR, Singapore
{yao_zexi,jin_chao,mi_mi_aung,xlli}@i2r.a-star.edu.sg
[2] School of Computer Science and Engineering, Nanyang Technological University
mohamedr002@e.ntu.edu.sg

Abstract

Data-driven fault diagnosis plays a key role in reducing maintaining costs and reducing down time for industrial machines. Deep learning has shown promising performance in identifying the different fault types. Yet, large amount of data is required to achieve satisfactory performance. In the real world, fault data is often rare, thus there is incentive for different corporations to work together to train a fault detection model. However, sharing data between different factories may not be applicable due to the data privacy concerns. Besides, distribution of data collected from different entities can be non i.i.d. As a results, a model trained on one machine can fail to generalise to different machines due to the distribution shift problem. In this work, we propose DiagNet, a federated transfer learning framework for machine fault diagnosis tasks. Specifically, to address the data privacy concerns, we employ the federated learning approach by jointly training a global model across multiple clients without sharing their raw data. However, the global model does not perform the best for each of the clients due to data distribution variances. To further tackle this problem, we employ the transfer learning approach to adapt the global model separately on each client with his own private machine data. Experimental results under low data regimes show that our DiagNet framework can significantly improve the fault-diagnosis model training accuracy by up to 28%.

1 Introduction

Data-driven fault diagnosis has witnessed a remarkable success in monitoring the health of industrial machines. One category of conventional machine learning based methods aim to learn the mapping between manually designed features (i.e., handcrafted features) and the corresponding faulty types. However, these methods rely on the domain knowledge when designing these handcrafted features. Deep learning with hierarchical multi-layer representation learning has achieved a great performance on fault diagnostic tasks. It directly learns features from raw input data without manual feature extraction (Sun et al. 2018; Zhao et al. 2020; Iqbal et al. 2019). However, the success of deep learning is mainly attributed to the availability of large amount of data, with existing research focusing primarily on comprehensive datasets with i.i.d. data that covers all fault classes in all training conditions(Sun et al. 2018; Zhao et al. 2020). Yet, in real world industry, obtaining large amount of labeled faulty data can be challenging due to the catastrophic consequences of failure of complex machines. Thus, our paper will focus on the specific problem of sparse non i.i.d. data conditions found in real world machine fault diagnosis.

One way to address this issue is to leverage data from different companies which use similar type of machines. This can have strong potential to show promising fault diagnosis performance. However, sharing data between different companies can face the following obstacles. First, sharing data between different factories and entities can be unattainable due to the data privacy and confidentiality issues. Second, even if the data can be shared, data from different companies usually have different distributions (i.e., non i.i.d). As such, a model trained on data collected from one factory can fail to generalize well when tested on data from different factories (Ragab et al. 2020).

To address all these issues, we propose our DiagNet approach, a federated transfer learning approach for fault diagnosis tasks. In particular, DiagNet employs federated learning to securely aggregate information from several clients to train a deep learning global model while preserving data privacy. Then, DiagNet further employs transfer learning to adapt the global model to the specific requirements and operating conditions of the machinery for each client. However, an important observation from us is that, under low data situations, the complex model can have sub optimal transfer learning performance due to the over fitting phenomena on the global model. Therefore, we introduce an adaptive early stopping approach to regularize the model and improve the overall performance.

The overall workflow of DiagNet is thus as follows: 1) The central server shares the initial model with each client. 2) Each client trains the model on their local data and sends the trained weights back to the central server. 3) The central server updates its global model using the weights from each client. 4) The process repeats until the global model is trained through Federated Learning, utilizing adaptive early stopping to halt training before over fitting. 5) The partially trained global model is then sent to each client. 6) Each client then conducts Transfer Learning by using the global model as the initial starting model and training it on their local dataset only. 7) Finally, each client will have a trained model tailored specifically to their domain that can be used for fault diagnosis for incoming data.

The major contributions of our paper are summarized as follows.

- We design and implement DiagNet, a Federated Transfer learning framework for machine fault diagnosis model training, and is compatible with most existing neural network models such as CNN, DNN and LSTM.
- We propose the method of adaptive early stopping, which terminates Federated training of the global model before reaching optimal accuracy in order to prevent overfitting and improve the effectiveness of the Transfer Learning stage.
- We demonstrate the efficiency of DiagNet on fault diagnosis sensory data from different operating conditions. We were able to improve the average accuracy of models trained by clients under very low local data conditions (70 to 200 samples per client) by 28%. Even clients with exceptionally low data samples of only 70, we were able to achieve local model accuracy in excess of 90%.

The rest of the paper is organized as follows. The next section introduces related work in the field of machine fault diagnosis, Federated and Transfer Learning. In the third section we describe DiagNet, our framework for Federated Transfer Learning for machine fault diagnosis. Experimental setup and methodology are discussed in the fourth section. In the fifth section, we analyze the experimental results. Lastly, in section six, we discuss findings of this paper, potential applications of DiagNet, and future research opportunities in this field.

2 Related Work

2.1 Machine Fault Diagnosis Models

Deep learning, with its automatic extraction of salient features, has widely acclaimed performance in various applications including computer vision, natural language processing, and speech recognition (Otter, Medina, and Kalita 2021). Recently, a growing body of literature has leveraged deep learning to automatically extract features from raw vibration signals for fault diagnosis. Chen *et al.* employed the convolutional neural network (CNN) with 1-dimensional kernels to extract transferable features for fault diagnosis (Chen, Gryllias, and Li 2019). Chuang *et al.* proposed a deep sparse autoencoder to tackle overfitting risk of deep models and improve the performance of fault diagnosis (Sun et al. 2018). Zhao *et al.* developed a novel deep residual shrinkage network to learn better feature representation from noisy vibration signals, resulting a better performance for fault diagnosis tasks (Zhao et al. 2020).

Nonetheless, most existing deep learning approaches focus on datasets with large amounts of i.i.d. data. For instance, Zhao used a comprehensive training dataset of 3,600 data samples from a single operating machine distributed evenly among 9 classes(Zhao et al. 2020). In the case of the CWRU bearing dataset, Chuang *et al.* used an i.i.d. dataset of 196 datasamples split evenly among 4 classes from a single operating condition(Sun et al. 2018). This may not reflect a realistic data environment in the real world, as machine fault data is difficult to collect, and varies considerably between operating conditions. This results in a sparse non i.i.d. dataset, where certain operating conditions may not have data entries in some fault classes at all.

To address this issue, data across different factories can be leveraged for larger amounts of data. However, this can raise data privacy concerns and cannot be applied in real world scenarios. Thus, we aim to use Federated Transfer Learning to allow for cooperative model training without requiring data sharing.

2.2 Federated Learning

Federated Learning (FL) is a machine learning setting that allows clients to exchange information for model training while keeping data decentralized, thus preserving data privacy. Instead, each client computes on their data locally, and creates an update package which is sent to the central server for aggregation. (Kairouz et al. 2021)

In 2017, McMahan *et al.* proposed FedAvg, one of the most popular federated training frameworks today, and demonstrated that training via FL can produce effective models that are superior to models trained only on locally hosted datasets (McMahan et al. 2017). Considerable research has shown that FL has a growing number of applications in various fields that have data privacy concerns, such as retail, banking and healthcare. (Yang et al. 2019). In the field of machine fault diagnosis, Zhang *et al.* proposed a Federated Learning method that aggregrates local client models to tackle the data island problem caused by a lack of data sharing (Zhang et al. 2021). Geng *et al.* developed FA-FedAvg, which improves the accuracy and speed of fault diagnosis through an optimized weighting strategy (Geng et al. 2021).

However, real world machine fault data is often non-i.i.d. in nature, as machinery from different clients usually operate under different conditions. This presents a challenge for Federated Learning models, as non i.i.d. causes global model divergence and thus reducing model performance, especially among deep networks (Zhao et al. 2018). Thus, traditional Federated Learning frameworks may perform sub optimally when faced with real world data, and an approach that is better suited to tackling non i.i.d. data is required.

2.3 Transfer Learning

Transfer Learning (TL) aims to transfer knowledge from one or more source domains to a target domain, in order to improve the performance of a model in the target domain. By transferring model features between domains, TL has been shown to be capable of reducing the amount of labeled data required for deep learning in the target domain, as opposed to training a model from scratch (Yosinski et al. 2014). TL has been shown to have a wide range of deep learning applications, such as Natural Language Processing, Computer Vision and Robotics. (Yu and Jiang 2016; Zhang et al. 2016; Rusu et al. 2017) Recent research has also been conducted for TL in the field of machine fault diagnosis. Wang *et al.* demonstrated the feasibility of transferring models from non-manufacturing settings to manufacturing machine fault detection, as well as between different machines and different operating conditions. (Wang and Gao 2020)

Transfer Learning however requires pre-trained models as a foundation for adaptation to the target domain. Such suitable models many not be available in the real world for many applications, and alternatives need to be found.

In our approach, we aim to combine Transfer Learning as a second stage to Federated Learning in DiagNet, by transferring knowledge from the global model to each individual's client domain. This ensures that Transfer Learning has a suitable source domain to pull from, and reduces divergence caused by non-i.i.d. data for Federated Learning.

3 Diagnet: Federated Learning Network for Machine Fault Diagnostics

In this section, we introduce our design of the DiagNet framework, which incorporates Federated Learning to train a global model before employing Transfer Learning in client specific transfer learning to adapt the global model's domain to suit each individual client.

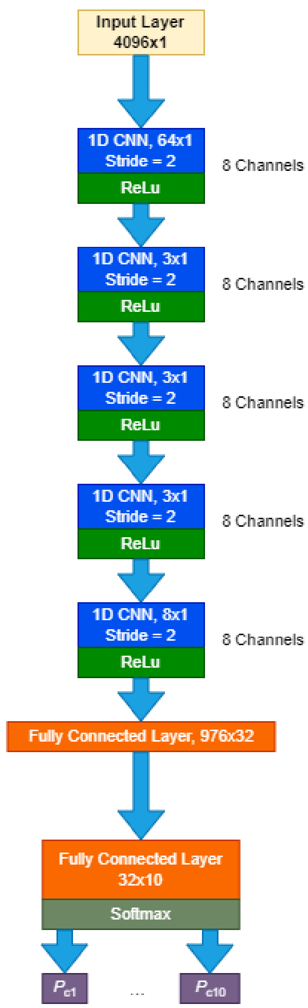


Figure 1: Fault Diagnosis Model

3.1 Machine Fault Diagnosis Model

The model we use in DiagNet for fault diagnosis is derived from our previous model in the PrivGD network(Jin, Ragab, and Aung 2020). It is a CNN based model which is composed of two components, a feature extractor and a classifier. In particular, the feature extractor is a 5-layer convolutional neural network with 1-dimensional kernels (1D-CNN). It aims to find a latent representation of the time-series data that could be class discriminative. The classifier is composed of a fully connected layer followed by a Softmax activation layer. It takes the extracted features from the1D-CNN network as inputs, and outputs the probabilities for the input sample belonging to each class.

This model was designed to operate with an initial training dataset of 2000 i.i.d. data entries per client. However this may not be a realistic reflection of real world conditions, as data may be less readily available and non-i.i.d. in nature. Thus in our paper, we will adapt this model to a more realistic sparse data environment, with only 70-200 data entries per client. In addition, we will use random sampling of the training data. Through this, we will demonstrate that existing models with ideal data assumptions can be adapted to work using DiagNet.

Compared to Chuang's approach, where he used 196 samples for a single operating condition, split evenly among 4 fault classes (Sun et al. 2018), our dataset presents a significantly more challenging environment. Instead of focusing on a single fault size, we opted to include all fault sizes, thus resulting in a total of 10 classes. As such, we have an average of 7-20 data samples per class per client, as opposed to a fixed 49 per class in Chuang's study. In addition, due to the random sampling and non i.i.d. nature of our data, there may not be data entries for all classes in all operating conditions. This presents a more realistic environment, where there is uneven distribution of data between clients and classes.

3.2 Federated Learning Framework

DiagNet's Federated Learning framework is comprised of a centralized server and several clients. In particular, the framework adopts a federated mini-batch stochastic gradient descent approach. A single batch size, b, is selected globally, and each client partitions their local dataset into batches of size b. Due to the non-i.i.d. nature of the data, it is likely that clients will have different numbers of batches. This will be accounted for during the the training process.

Figure 2 illustrates how the DiagNet processes. First, the centralized server sends the global model to each client. Each client then selects the relevant batch from their dataset and computes the resulting gradients of the model's weights for each sample. If a client has no data for this batch, such as when they have already exhausted all local data for this epoch, then the client sets the output gradients to 0. Each client then sums up all the gradients into a single result, and sends it back to the central server. The central server then uses these gradients to update the global model via SGD. The process repeats until all clients have conducted one full on their local dataset, which concludes one epoch of training.

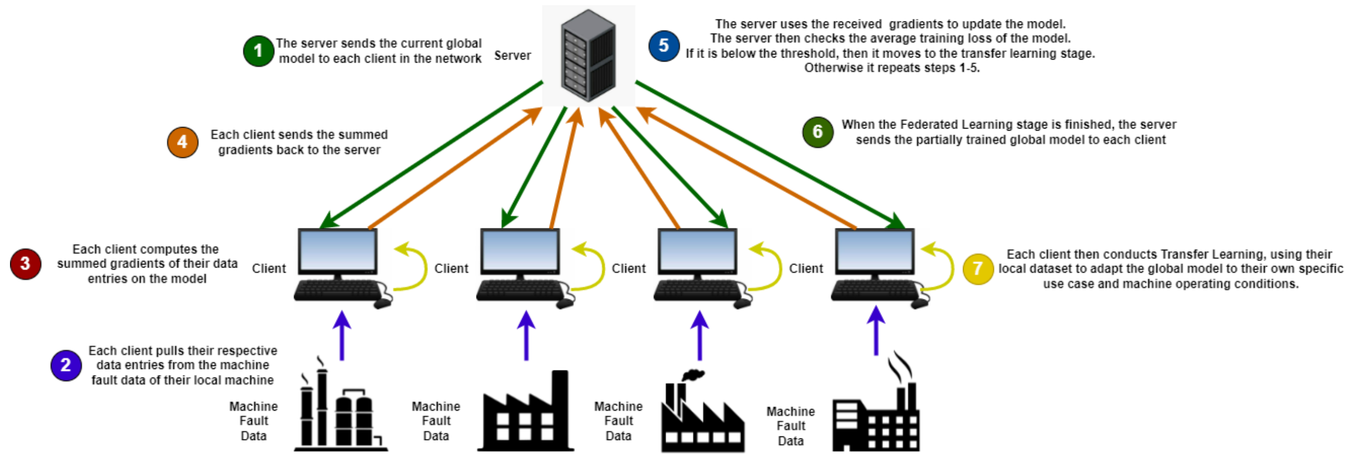


Figure 2: DiagNet Framework

Training of the global model continues until the server halts the process. Specifically, we used adaptive early stopping while training the global model before it begins over fitting. Each client then finishes the training process by transfer learning the model exclusively on their own local dataset.

Algorithm 1: DiagNet Server Algorithm

$EpochsUnderThreshold \leftarrow 0$
while $EpochsUnderThreshold < n$ **do**
  for $P \in S$ **do**
    for $a \in Clients$ **do**
      $sendToClient(a, (model, P))$
      $o \leftarrow receiveFromClient(a)$
      $(loss[P], acc[P], grad[P]) \leftarrow o$
    end for
    $SGD(model, avg(grad))$
  end for
  if $avg(loss) < T$ **then**
    $EpochsUnderThreshold{+}{+}$
  else
    $EpochsUnderThreshold \leftarrow 0$
  end if
end while
for $a \in Clients$ **do**
  $sendToClient(a, model)$
end for

3.3 Adaptive Early Stopping

Adaptive early stopping is necessary for our scenario since the number of epochs needed to train the model varies significantly. This is due to a combination of the non-i.i.d nature of our data and the low number of data samples. During initial experimentation , we found that the Client Specific Transfer Learning stage was ineffective due to overfitting of the global model during the Federated Learning stage. Due to the random sampling of the training data, the rate of training convergence varied considerably between runs, which made a traditional epoch-based early stopping unsuitable.

Algorithm 2: DiagNet Client Federated Learning Algorithm

$(model, P) \leftarrow receiveFromServer()$
for $i \in P$ **do**
  $entry \leftarrow localdata[i]$
  $loss[i], acc[i], o \leftarrow forward(model, entry)$
  $grad[i] \leftarrow backPropagation(model, entry, o)$
end for
$sendToServer(avg(loss), avg(acc), avg(grad))$

To remedy this issue, we implemented Adaptive Early stopping to prevent global model overfitting. Adaptive early stopping is conducted by setting a threshold T for training loss and threshold n for epoch count. When training the global model, when the training loss falls under the threshold for n consecutive epochs, the global training process is halted and client specific transfer learning begins. The loss threshold ensures that we reach a suitable point in the training process before transitioning to transfer learning, while the epoch threshold is necessary as the training loss would often dip below the loss threshold for 1-2 epochs early on in the training process. Algorithm 1 demonstrates how the DiagNet Server conducts Federated Learning with adaptive early stopping, and Algorithm 2 demonstrates the client side of the training process.

3.4 Client Specific Transfer Learning

In our scenario, each client is running their machine at a different operating condition. As such, directly applying the global model to each client will likely result in reduced accuracy. To mitigate this problem, we utilized transfer learning to adapt the global model to each client's domain.

Once training of the global model has concluded, the central server sends the global model to each client. Each client then uses the global model as the initial state to conduct training on their local dataset. The number of epochs for the transfer learning process needs to be carefully selected to maximize the accuracy of the result. Due to the low amount of data available in the local dataset, overfitting due to excessive training is highly likely. In the Experiment Evaluation section, we will show that the fine-tuning process reaches optimal validation loss after only 15-20 epochs of training, and begins overfitting afterwards. As such, we limited the number of epochs of client specific transfer learning to 20 epochs in our model.

Algorithm 3 illustrates the client specific transfer learning process of DiagNet.

Algorithm 3: DiagNet Client Finetuning Algorithm

$model \leftarrow receiveFromServer()$
for $(i = 0; i < FTepochs; i \leftarrow i + 1)$ **do**
  for $entry \in localdata$ **do**
    $loss[i], acc[i], o \leftarrow forward(model, entry)$
    $grad[i] \leftarrow backPropagation(model, entry, o)$
    $model \leftarrow SGD(model, grad[i])$
  end for
end for

Table 1: CWRU Bearing Dataset Labels

Class Label	Fault Type	Fault Size (in)
1	None	-
2	IF	0.007
3	IF	0.014
4	IF	0.021
5	OF	0.007
6	OF	0.014
7	OF	0.021
8	BF	0.007
9	BF	0.014
10	BF	0.021

4 Experiments

In this section we will first describe the datasets, implementation details and our experimental setup.

4.1 The Machine Vibration Sensor Datasets

The datasets used for our scenario are acquired from the Case Western Reserve University Bearing Data Center's website. The CWRU bearing dataset is time-series data that collected at 12k sampling rate. The dataset has 4 subsets with different loading torques, where the torque values ranges from 0 to 3. Each subset is assigned to a different client, to simulate different operating conditions of machines under each client. As such, the data is non-i.i.d., with a co-variate shift and concept drift between clients.

In each subset, the data instances fall into 4 different categories, with one non-faulty and three faulty categories. The three faulty categories are inner-race faults (IF), outer-race faults (OF), and bearing-race faults (BF). Each faulty category could have 3 fault sizes, 0.007 inches, 0.014 inches, and 0.021 inches, for a total of 10 classes (1 non-faulty class, and 9 faulty classes). Table 1 illustrates how the various classes are labelled.

Table 2: Client Data Sample Size

Client	Sample Size
1	70
2	100
3	150
4	200
Total	520

In addition, to simulate uneven distribution of data in the real world, each client has a different number of samples available for training. Table 2 shows the distribution of data samples sizes available for each client for our experiment.

4.2 DiagNet Experimental Procedure

The federated setting for DiagNet is simulated using four client datasets, each of a different operating condition. As our scenario calls for a low dataset environment, we reduced the samples available for training to each client by taking a random subset of the original data. This subset is not necessarily a representative sample of the original dataset, as machine fault data in the real world may not be representative of real world conditions.

To mitigate experimental error due to our low number of data samples, we conducted three runs of our experiment and averaged the results of the runs. The process of a single run is as follows:

- A random subset from each dataset is selected to represent the training data of the clients. The size of the training dataset is shown in Table 2
- A control group is trained without FL, with each client training their model using only their dataset.
- Using DiagNet, a global model is trained through Federated Learning utilizing adaptive early stopping with T = 0.05 and n = 3. These values were chosen due to experimental results in section 5.3 showing that it was the optimal threshold.
- A copy of the partially trained model is made for each client, and Transfer Learning is done by training each model on the respective client dataset for a further 20 epochs. This value is chosen as experimental results in section 5.2 showed that any further training resulted in overfitting.
- We also train a copy of the global model till convergence to obtain a baseline without TL.
- Lastly, we conducted TL on the converged global model for a further 20 epochs to obtain another set of results for FT+TL without adaptive early stopping.

To test the accuracy of the trained models, we assembled a validation dataset for each client. Each validation data set had 2000 entries. We then ran each of the models through the validation dataset of each client, and collated the results.

4.3 Implementation Details

We carried out the experiments on a server with an Intel Xeon Platinum 8170CPU @ 3.700GHz with 26 cores, and 188 GB RAM. The operating system is Arch Linux. The fault diagnosis model training and fine-tuning on data is done using Pytorch at version 1.8.1.

5 Results and Discussion

5.1 Ablation Study

As seen in Table 3, DiagNet is able to achieve a substantial improvement in accuracy over local learning for all clients, regardless of their local sample size, with all clients achieving an average accuracy of greater than 90%.

Notably, improvement becomes more significant the smaller the size of the local dataset available to the client. For instance, the client with largest sample size (i.e., Client 3) experienced an increase in accuracy of 8.3, while the client with smallest sample size (i.e., Client 1) has witnessed significant improvement of 40.1% with our DiagNet. In addition, due to the random sampling of training data for each client for each run, the variance of model accuracy is very high for local only training, with 2 out of 4 clients having standard deviations in excess of 10%. However, DiagNet was able to reduce the variance to under 2% for each client when training on the same data. This demonstrates that DiagNet can successfully tackle the challenge of low data sample environments in machine fault diagnosis.

In addition, DiagNet still showed an improvement in accuracy compared to the Federated Learning only approach for all clients of 0.4 to 1.4%. In addition, as seen in Figure 3(a), we see that DiagNet offers a substantial improvement in validation loss, from 0.287 to 0.193, a reduction of 32.8%. Thus, we see that the Transfer Learning component of DiagNet offers a significant benefit over traditional pure Federated Learning approaches.

Comparatively, without Adaptive Early Stopping, the FL+TL model achieved an improve in accuracy of only 0.3 to 0.9% over the pure FL model, which is notably smaller than DiagNet. In addition, we see from Figure 3(a), that the FL+TL approach without adaptive early stopping achieved almost no improvement in the validation loss over a purely FL approach. Thus, it is evident that adaptive early stopping has a significant positive influence on the effective of DiagNet. Further analysis of adaptive early stopping is conducted in section 5.3.

5.2 Impact of Overfitting during Transfer Learning

As mentioned previously, overfitting occurs during the transfer learning process due to the low sample sizes available to each client. To determine the degree of overfitting during the TL stage, and the ideal stopping point, we set up the following experimental procedure.

Table 3: DiagNet Accuracy (%)

Client	Local Only	FL Only	FL+TL No Adaptive Earlystop	FL+TL w/ Adaptive Earlystop
1	53.0 ±4.87	92.5 ±2.10	92.8 ±2.04	93.1 ±1.80
2	57.3 ±11.2	93.6 ±2.47	94.5 ±2.17	95.0 ±1.65
3	69.4 ±11.6	96.7 ±2.09	97.1 ±2.30	97.4 ±1.94
4	89.8 ±9.76	97.3 ±2.19	97.7 ±2.12	98.1 ±1.88
Avg	67.4 ±17.3	95.0 ±3.00	95.5 ±2.94	95.9 ±2.69

- Similar to the overall DiagNet experiment, a random subset from each dataset is selected to represent the clients.
- Using DiagNet, a global model is trained through Federated Learning. Adaptive early stopping is not used in order to ensure fair comparisons between runs. The stopping point for the global model is instead set at epochs = 30. We arrived at this value as most adaptive early stopping happens between epochs 20 to 40.
- A copy of the partially trained model is made for each client, and transfer learning is done by training each model on the respective client dataset for a further 60 epochs to reach convergence.

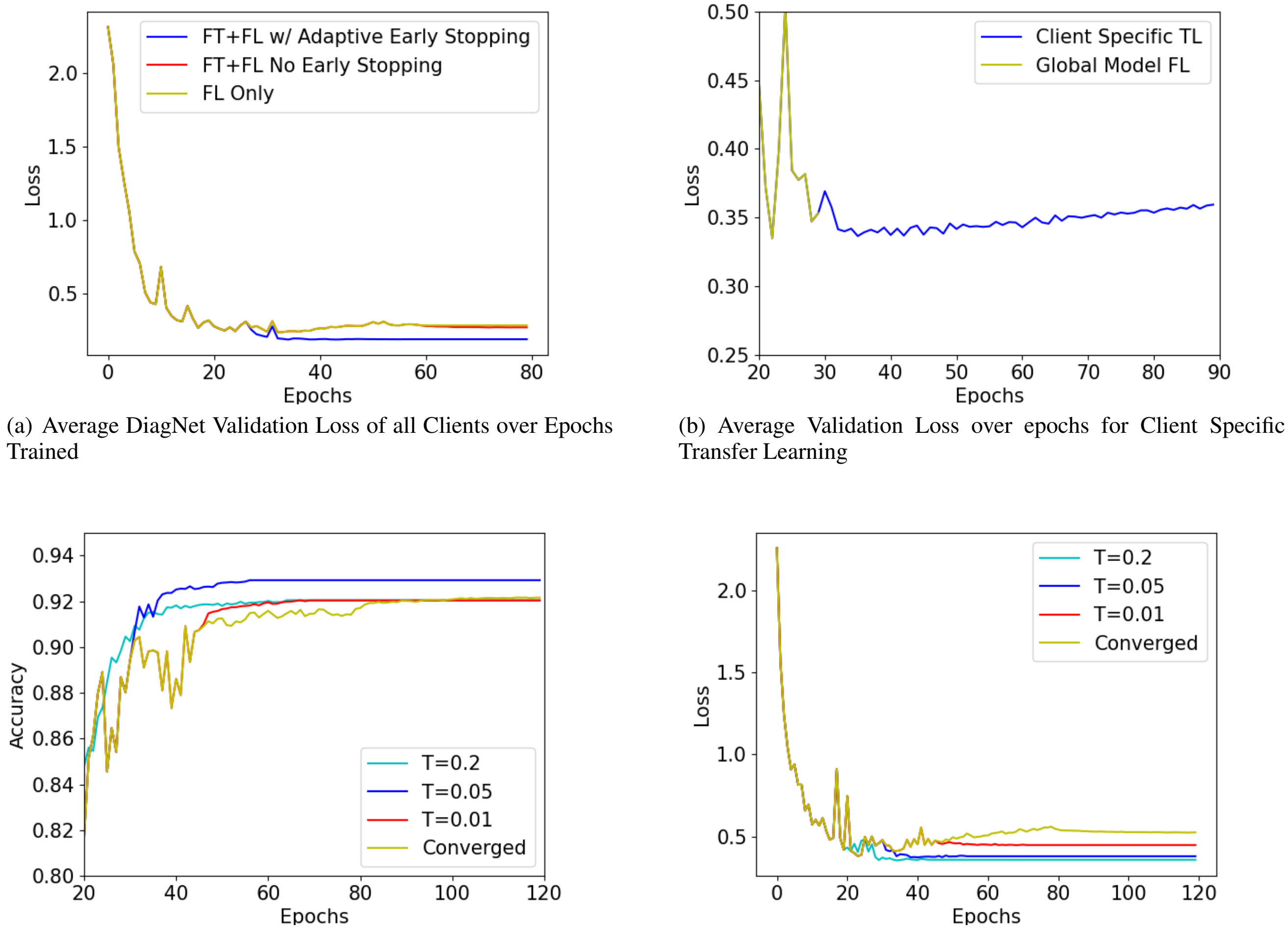
We conducted 3 runs of the experiment and averaged the results. The results can be seen in Figure 3(b).

The training of the global model stops at the 30 epoch mark, before transitioning to client specific transfer learning. The client models converge rapidly once TL begins, reaching their minimum between the 35 and 50 epoch mark. As training continues beyond the 50 epoch mark, we can see the validation loss steadily increasing once more due to overfitting. As such, we determined that 20 epochs of transfer learning is sufficient for transfer learning for our client models.

5.3 Effect of Adaptive Early Stopping

DiagNet's Adaptive Early Stopping algorithm relies on a training loss threshold to determine the point of early stopping during the training process. In order to determine the ideal threshold for the algorithm for this dataset, we ran the following experiment.

- Similar to the DiagNet experiment, a random subset from each dataset is selected to represent the clients.
- Using DiagNet, a global model is trained through Federated Learning. The thresholds for the different early stopping points are T = 0.2, 0.05, 0.01
- When the global model reaches a threshold, the current global model is branched, and each client does 20 epochs of transfer learning on that iteration of the global model.



(a) Average DiagNet Validation Loss of all Clients over Epochs Trained

(b) Average Validation Loss over epochs for Client Specific Transfer Learning

(c) Average Validation Accuracy with different Adaptive Thresholds

(d) Average Validation Loss with different Adaptive Thresholds

Figure 3: Experimental Results

Once the transfer learning is done, the results are stored, and the network continues training the global model.

- Lastly, a global model is trained for 80 epochs (i.e. convergence), and 20 epochs of client specific transfer learning is conducted to create a control group.

We conducted 3 runs of the experiment and averaged the results. Our findings are as follows.

As seen from Table 4 and Figure 3(d), all instances of Adaptive Early Stopping thresholds performed better no Adaptive Early Stopping in terms of validation loss. Notably, the global model suffered from overfitting from epoch 37 onwards, as seen by the steady increase in validation loss. Both T=0.2 and T=0.05 stopped global model training before epoch 37, and achieved a comparatively low validation loss rate. T=0.2 performed the best in terms of validation loss with 0.358, with T=0.05 shortly after at 0.380. T=0.01 stopped the training after overfitting of the global model had already begun, and thus achieved a substantially worse validation loss of 0.448.

Table 4: Adaptive Threshold Loss

Client	T=0.2	T=0.05	T=0.01	Convergence
1	0.512	0.583	0.670	0.791
2	0.476	0.493	0.555	0.647
3	0.229	0.240	0.328	0.390
4	0.216	0.201	0.238	0.278
Avg	0.358	0.380	0.448	0.526

In terms of accuracy, Figure 3(c) shows that despite the low validation loss, T=0.2 achieved only an accuracy of 92.1%, nearly identical to the control group's 92.2%. T=0.01 similarly also achieved an accuracy of 92.1%. Only T=0.05 performed substantially better, at 92.9%. This experiment further demonstrates the effectiveness of adaptive early stopping, as well as the importance of selecting an appropriate threshold. We can see that a balance must be struck in order to achieve optimal results, with the threshold ideally as close to the global model's minimum as possible, but before overfitting begins.

Thus, we chose T=0.05 for our program's threshold for adaptive early stopping, as it stopped the global model training closest to the global loss minimum and led to the best model accuracy after transfer learning.

6 Conclusion

In this paper, we constructed the DiagNet framework, a Federated Learning combined with transfer learning framework for machine fault diagnosis, and demonstrated its effectiveness in a sparse non-i.i.d. dataset which simulates real world conditions. Our future investigations will focus on applying DiagNet to other machine fault datasets and real world situations, as well as exploring the possibility of utilizing DiagNet in other domains that have similar issues with data collection, such as for medical diagnosis.

7 Acknowledgements

This research project is supported by A*STAR under its RIE2020 Advanced Manufacturing and Engineering (AME) Programmatic Programme (Award A19E3b0099).

References

Chen, Z.; Gryllias, K.; and Li, W. 2019. Intelligent fault diagnosis for rotary machinery using transferable convolutional neural network. *IEEE Transactions on Industrial Informatics*, 16(1): 339–349.

Geng, D.; He, H.; Lan, X.; and Liu, C. 2021. Bearing fault diagnosis based on improved federated learning algorithm.

Iqbal, R.; Maniak, T.; Doctor, F.; and Karyotis, C. 2019. Fault Detection and Isolation in Industrial Processes Using Deep Learning Approaches. *IEEE Transactions on Industrial Informatics*, 15(5): 3077–3084.

Jin, C.; Ragab, M.; and Aung, K. M. M. 2020. Secure Transfer Learning for Machine Fault Diagnosis Under Different Operating Conditions. In *Provable and Practical Security*.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A. N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; D'Oliveira, R. G. L.; Eichner, H.; Rouayheb, S. E.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.; Konečný, J.; Korolova, A.; Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Raykova, M.; Qi, H.; Ramage, D.; Raskar, R.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2021. Advances and Open Problems in Federated Learning. arXiv:1912.04977.

McMahan, H. B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. arXiv:1602.05629.

Otter, D. W.; Medina, J. R.; and Kalita, J. K. 2021. A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2): 604–624.

Ragab, M.; Chen, Z.; Wu, M.; Li, H.; Kwoh, C.-K.; Yan, R.; and Li, X. 2020. Adversarial multiple-target domain adaptation for fault classification. *IEEE Transactions on Instrumentation and Measurement*, 70: 1–11.

Rusu, A. A.; Večerík, M.; Rothörl, T.; Heess, N.; Pascanu, R.; and Hadsell, R. 2017. Sim-to-real robot learning from pixels with progressive nets. In *Conference on Robot Learning*, 262–270.

Sun, C.; Ma, M.; Zhao, Z.; and Chen, X. 2018. Sparse Deep Stacking Network for Fault Diagnosis of Motor. *IEEE Transactions on Industrial Informatics*, 14(7): 3261–3270.

Wang, P.; and Gao, R. X. 2020. Transfer learning for enhanced machine fault diagnosis in manufacturing. *CIRP Annals*, 69(1): 413–416.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated Machine Learning: Concept and Applications. arXiv:1902.04885.

Yosinski, J.; Clune, J.; Bengio, Y.; and Lipson, H. 2014. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, 3320–3328.

Yu, J.; and Jiang, J. 2016. Learning Sentence Embeddings with Auxiliary Tasks for Cross-Domain Sentiment Classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 236–246.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zhang, W.; Li, X.; Ma, H.; Luo, Z.; and Li, X. 2021. Federated learning for machinery fault diagnosis with dynamic validation and self-supervision. *Knowledge-Based Systems*.

Zhao, M.; Zhong, S.; Fu, X.; Tang, B.; and Pecht, M. 2020. Deep Residual Shrinkage Networks for Fault Diagnosis. *IEEE Transactions on Industrial Informatics*, 16(7): 4681–4690.

Zhao, Y.; Li, M.; Lai, L.; Suda, N.; Civin, D.; and Chandra, V. 2018. Federated Learning with Non-IID Data. arXiv:1806.00582.