Federated Learning Applications

Diksha Srivastava
College of Science and Engineering
National University of Ireland
Galway, Ireland
D.Srivastava1@nuigalway.ie

Saumya Goel
College of Science and Engineering
National University of Ireland
Galway, Ireland
S.Goel1@nuigalway.ie

Sungamesh Jayveera Prasad
College of Science and Engineering
National University of Ireland
Galway, Ireland
s.jayaveeraprasad1@nuigalway.ie

***Abstract*—Federated learning, often known as collaborative learning, is a decentralized machine learning approach. Its applications serve as a stepping-stone for machine learning algorithms to learn from a wider range of data sets collected through different mediums. Because these data sets are stored in various locations, the number of hardware infrastructures required is reduced. Federated learning is adopted in a lot of fields like healthcare, Internet of Things, Industrial applications, and many more. In this paper, we discuss different application wise issues faced in the above-mentioned fields and measures to overcome these issues. The main aim of these improvements is to bring new level of security, reliability, efficiency, and enhance the performance.**

Keywords— Federated learning, machine learning, Internet of Things, healthcare, Industrial applications.

I. Introduction

The importance of data science has increased exponentially in various fields in recent years and there has been an explosive development in artificial intelligence, machine learning, smart production, and deep learning. However, deep learning has a few challenges. Firstly, deep learning networks are called data hungry because they have millions of parameters which need large amount of training samples and getting labelled data is a big challenge. Secondly, anonymizing the data to be GDPR compliant and to keep it secure for transferring such large amount of data over the internet continuously is another issue and it is very costly.

A solution to all these problems is Federated Learning where it is not needed to transfer the data or to minimize the data. Federated learning is a machine learning procedure where the goal is to train a high-quality model with the data distributed over a several independent providers instead of gathering the data on a single central server, the data remain locked on those servers and the algorithm/predictive models travel between them.

Transferring trained model from server to remote organization to fine-tuning toward the local data of that organization. Once the network is fine-tuned, it is resent to the server to update the baseline model which is further forwarded to other organizations or clients. This cycle continues and the model is regularly kept updated on the data from each organization without leaving the premises or compromising the privacy or security.

Federated learning has two main components: Central server and client component. The central server initiates and orchestrate the training process whereas the client device performs the actual model training.

Due to multiple advantages of Federated Learning, it is being used in various fields like Healthcare, IOT devices, Industrial Engineering, and many more. This paper reviews

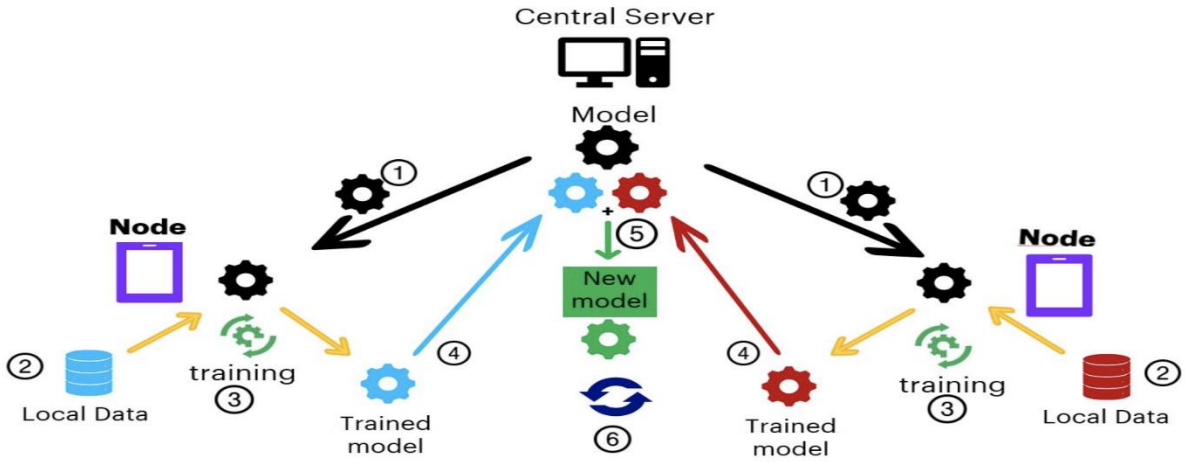


Fig 1: Federated Learning Architecture [1]

how Federated Learning is used in collaboration with Deep Learning models to harness the maximum benefits such as ensuring privacy, compliance to GDPR, quick deployment of base models, and cross-industry applications.

This paper is organized as follows. Besides the introduction, we cover the Literature Review in section 2. In section 3, we review and compare the models, techniques, and results of the Federated Learning applications. Furthermore, in section 4, we cover the advantages and disadvantages of using FL approach. In section 5, we suggest some modifications and ensures data security of clients, possible future work to the FL approach. Finally, we conclude our paper.

II. LITERATURE REVIEW

Most of the researchers in various fields used Machine Learning (such as deep learning) along with Big Data to train inference models. For Example, V. Anuja Kumari, and R. Chitra [2] proposed a supervised Machine Learning approach i.e., SVM (Support Vector Machine) to classify the diagnosis of diabetes. Similarly, for prediction of out-of-vocabulary words, Moses Charikar, Kevin Chen, and Martin [3] proposed using trivial data structures like hash maps, count sketches, to identify the frequently used words and returning it in the prediction. Y. Zheng, F. Liu, and H. Hsieh [4] proposed semi-supervised learning approach based on co-training to infer the air quality according to the data provided by the monitor stations. Similarly, for visual inspection tasks, Deep learning was being used extensively. Yu, Z., Wu, X., Gu, X. [5] used lower layers of pre trained CNN with VGG or inception and then retrain. A. Jindal, G.S. Aujla [6] used SDN-enabled deep learning architecture i.e., CNN for controlling traffic in vehicular systems.

There is also a need to deal with data heterogeneity which is inherent in IoT environments and geographical locations. In such scenarios, federated learning in its base form lags in capturing such granular details. Z. Zhang, H. Ma [7] proposed in their paper to use multiple kernels which fuse these heterogeneous and complementary features that have different notions. H. Brendan McMahan, Felix X. Yu [8] proposed federated learning in their paper for improving communication efficiency, but it allows single global model across the shared data which limits their ability to deal with non-IID data.

All the aforementioned approaches proved to be helpful in increasing accuracies when compared to traditional models such as pattern matching, hash maps, count sketches etc. but came with its own set of challenges. Deep Learning approaches need lot of training data, exposure of private information, omission of detailed characteristics, inability to capture diversity in data.

These issues can be dealt with recently researched Federated Learning which encompasses various applications like analytics of mobile data, visual inspection, healthcare, autonomous driving, environment sensing, etc. In addition to this, Federated Learning approach can be customized to suit to specific applications.

Federated learning is a machine learning technique in which an algorithm is trained across numerous decentralized edge devices or servers keeping local data samples without their being exchanged. FL can be categorized in three groups namely horizontal FL, vertical FL, and federated transfer learning. In horizontal FL, there is some overlap between the features of data dispersed across multiple nodes, despite the fact that the data are fairly diverse in sample space. Vertical FL is appropriate for scenarios in which data is partitioned vertically based on feature dimension. Transfer learning allows us to transfer knowledge from one domain (the source domain) to another domain (the target domain) to get higher learning outcomes, which is ideal in most cases. Moreover, FL can be personalized to form application specific models such as Federated Region Learning, Federated Averaging to address the intricacies of the data.

III. REVIEW OF FEDERATED LEARNING APPLICATIONS

FEDERATED LEARNING OF PREDICTIVE MODELS FROM ELECTRONIC HEALTH RECORDS

A. *Introduction*

In the era where big data plays a major role, having a solution which is computationally efficient and privacy-aware have become increasingly important. This is true for the healthcare domain as well, where large amounts of data are held in multiple locations and owned by multiple entities. Previously, central data repository was used to store and process data from all agents. However, it was an inefficient approach because it is not a feasible solution for large-scale datasets, and it can lead to single point of failure which can cause privacy issues. A decentralized approach could better solve this problem.

The aim of this paper is to predict the hospitalizations of people due to cardiac arrest. It is a binary supervised classification problem which is solved using a general decentralized optimization framework which enables the collaboration of multiple data holders, converge to a predictive model and does not exchange raw data. This decentralized framework is created using an iterative **cluster Primal Dual Splitting (cDPS)** algorithm which solves the large-scale **sparse Support Vector Machine (sSVM)** classifier problem.

B. *Motivation*

Due to the increase in volume, variety, velocity, veracity of clinical data, a computational efficient model is required to mine these data. The insights obtained from the result could help in creating policies, detect disease, provide medical solutions, etc. Related works suggest that sparse classifiers (that use less features), have better performance and generalize well on the dataset. Further, the results obtained from these models are interpretable and protected from data breaches. The following three challenges are addressed through the model proposed by them:

- Data is present in different locations (hospitals, patients' smartphones, doctors' offices).

- Large scale data, which makes scalable frameworks important.
- Storing data in a central server is unnecessary as it cannot store large amounts of data and leads to privacy issues as if the central server is attacked by hackers, then the complete data is at risk of exposure.

C. Aim

The aim is to create a distributed/federated model to predict the hospitalisations of people due to cardiac arrest during a targeted year. The medical history of the patients is used for making such predictions. This data is obtained from Electronic Health Records (EHRs) of different hospitals or from the patient's smartphones. Thus, the collaboration of different agents is required to create a predictive model. Since, it is a binary supervised classification problem, a distributed soft-margin l1-regularized sparse support vector machines algorithm is used. SVM is used because it is an effective classifier.

D. Algorithm

- Patient's demographic data such as age, gender, weight, height, BMI, previous diagnosis, procedures, drug prescriptions, etc. acts as feature vectors.
- Hospitalizations is represented by label $l_i = +1$ and non-hospitalizations by label $l_i = -1$.
- Using the SVM, we need to find the hyperplane that maximises the distance between these two classes. Furthermore, since few features needs to be used for classification, a sparse SVM problem arises.

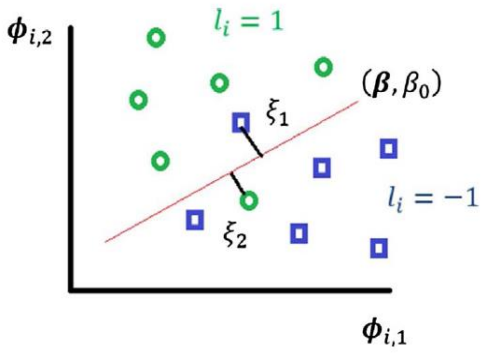


Fig 2: Support Vector Machines [9]

- This problem can be resolved using incremental sub gradient method. However, this only works over network with ring structure. Another approach is using linear time-average consensus optimization algorithm (LAC). It is an iterative algorithm that takes small steps to reach towards an optimal solution.
- However, for this paper, a new approach called **cluster primal dual splitting** is used. In this approach, it is assumed that there is a network of agents holding parts of data. These agents can be hospitals that process the data of their patients only or patients' personal data maintained on smartphones. These agents are connected through a communication network, which is an undirected graph but are connected and information exchange is only between neighbors.

Method	Decentralized?	Per iteration complexity	ε–accuracy iterations
Subgradient Descent	×	O(nd)	$O(1/\varepsilon^2)$
Incremental Subgradient	×	O(d)	$O(1/\varepsilon^2)$
Linear Average Consensus (LAC)	√	$O(n^2 + nd)$	$O(1/\varepsilon^2)$
Cluster Primal Dual Splitting (cPDS)	√	$O((n + m^2)d)$	$o(1/\varepsilon)$

Fig 3: Theoretical comparison of various techniques to solve sSVM problem [9]

- Each agent holds its own model parameters and using cPDS, an optimal value of parameters is found by updating them. This all happens using only the local information.

E. Results

The patient dataset is split into training set and test set. Then the model is trained on the training set and evaluated on test set. The performance is measured using AUROC (Area under the Receiver Operator Characteristic). This curve is plotted between true positive rate i.e., out of the hospitalized patients, how many were correctly hospitalized and the false positive rate which is the measure that out of the non-hospitalized patients, how many were hospitalized. The computation time is also measured which is the measure of time taken to train the model at all nodes. The communication cost is twice the product of number of edges.

The Fig 4 shows the comparison between cPDS and the centralized barrier method, the SubGD, the IncrSub and the LAC scheme. It shows that the AUC (Area under the ROC curve) for cPDS is high (i.e., 0.7806) as compared to other centralized methods. The computation cost is the effort made at all nodes. The more hospitals are, the more is the computation cost, however, if edges are more, the less time will be taken for convergence as it would be easy to reach other nodes. Hence, the computation cost is less for cPDS. However, the communication cost is high for cPDS as it is the measure of the number of messages exchanged between nodes.

Experimental comparison for various methods that solve the sSVM problem.

Method	Distributed?	AUC	Number of iterations	Computation cost (sec)	Communication cost
Subgradient Descent	×	0.7667	1500	2055	N/A
Barrier	×	0.7688	32	40,174	N/A
Incremental Subgradient	×	0.7734	554	6.3485	N/A
Linear Average Consensus (LAC)	√	0.7683	200	27,703	1.04e + 11
Cluster Primal Dual Splitting (cPDS) (m = 10, random graph)	√	0.7806	100	544	2600

Fig 4: Accuracy of various models [9]

FEDERATED LEARNING OF OUT-OF-VOCABULARY WORDS

A. *Introduction*

The Google Keyboard- Gboard provides features like prediction of next words, auto-correction, gesture typing, and many more. Identifying frequently typed words is important for mobile keyboards. However, words not in the vocabulary are not predicted. These are termed as "out-of-vocabulary (OOV)" words. Thus, this paper focuses on learning OOV words without transferring data on centralized servers. Thus, they make use of Federated Learning framework for neural networks where a federated character based recurrent neural network (RNN) is trained on device.

B. *Algorithm*

- Reddit comment dataset was used to train the LSTM model with a simulated FL environment. The user ID from the dataset can be used to learn from each client's local data.
- The LSTM model used is a variant with a Coupled Input and Forget gate (CIFG), peephole connection, and a projection layer.
- With the help of CIFG, the number of input parameters is decreased by 25%. Furthermore, the projection layer helps in reducing the dimension and speed-up training. The peephole connection let the layers of gate look at the cell state.
- A multilayer LSTM approach is used to increase the representation power of the model.
- The sampling process begins with the start of word token, it is executed in a multi-threaded way. At each step, each thread generates a random index. This is done iteratively till end of the word is reached. Thus, the use of thread avoids any dependence between each sampling thread.
- FL is useful for OOV learning. A Federated Averaging approach is used to combine client updates after being locally trained to produce a global weight. An adaptive L2-norm is performed on gradient of each client as it improves the convergence.
- The experiments were conducted on Reddit dataset and on-device FL where the data always stays locally. Some filters were applied to remove invalid "OOV" patterns. The filter excludes emoji, numbers, repetitive patterns like "hahaha" or "yesss".

C. *Evaluation Metric*

In OOV learning, the aim was to find either missing or daydreamed from the model sampling. In simulated FL, since the gold labels are available for the dataset. Thus, Precision and Recall can be used for evaluation. However, this is not possible for the on-device FL setting. Thus, it is evaluated by showing that the model can converge to good CE loss and top-k character level prediction accuracy.

	FL_S^{SGD}	FL_L^{SGD}	FL_L^M
m	0.0	0.0	0.9
B_s	64	64	64
S	0.0	0.0	6.0*
η	1.0	1.0	1.0
η_{client}	0.1	0.1	0.5
N_r	2	3	3
N	256	256	256
d	16	128	128
N_p	64	128	128

Fig 5: Hyper-parameters for three different FL settings. [10]

Fig 5 shows three different model hyper-parameters. N_r, η, m, and B_s refers to number of RNN layers, server-side learning rate, momentum, and batch size, respectively. FL_S^{SGD} and FL_L^{SGD} applies SGD without momentum. They have different LSTM model architectures. FL_S^{SGD} contains 216K parameters and FL_L^{SGD} contains 758K parameters. FL_L^{SGD} and FL_L^M have same architecture. FL_L^M converges when η_{client} = 0.5, while other two FL techniques diverges.

D. *Results*

For the simulated FL on Reddit data, FL_L^M converges with an accuracy of 66.3% and 1.887 CE loss. Momentum and adaptive clipping help in faster convergence.

yea	0.0050	yea	0.0057
upvote	0.0033	upvote	0.0040
downvoted	0.0030	downvoted	0.0033
alot	0.0026	alot	0.0029
downvote	0.0023	downvote	0.0026
downvotes	0.0018	downvotes	0.0022
upvotes	0.0016	upvotes	0.0021
wp-content	0.0016	op's	0.0019
op's	0.0015	wp-content	0.0017
restrict_sr	0.0014	redditors	0.0016

Fig 6: Top 10 OOV words and their probabilities from ground truth. [10]

Fig 6 shows the top 10 OOV words with their probability. The model learns the probability of occurrence of a word.

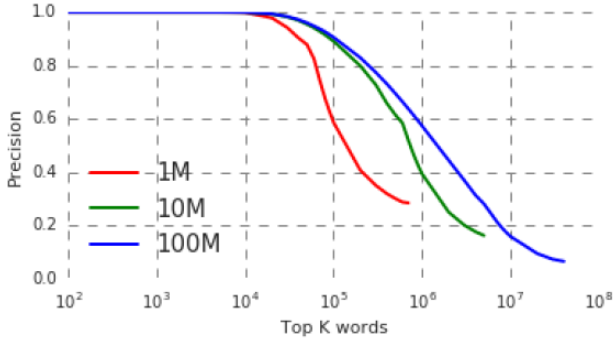


Fig 7: Precision in simulated FL [10]

FL_L^M shows a precision of about 90.56%. Curves in red, green, and blue represent a sampling of 10^6, 10^7, 10^8. Precision increases as sampling increases. Thus, this method can learn OOV setting in a real-world effectively.

FEDERATED LEARNING FOR VISUAL INSPECTION

A. Introduction

Federated Learning is widely for industrial engineering and visual inspection of products is one of the applications. Federated learning has replaced traditional methods of automated vision technologies such as pattern matching which needs extensive development and expertise to build algorithms for each structure. Deep Learning is an alternative which has shown promising results in inspection applications domain, but lack of data samples is one of the biggest challenges for classification tasks of defect detection. Therefore, Dataonomy is used to train the model with relatively small data samples. It is also extended to build cross-industry base models instead of focusing on single type of product or industry. Federated Learning along with Dataonomy and CNN provides a framework which allows quick deployment of base model for clients from new industry and perform automated inspection while servers work on improving the model using new datasets generated from clients. Also, it allows data to stay in the user devices with only the model information sent to the server. This ensures data security of clients. Dataonomy followed by transfer learning are perfectly compatible with FL framework.

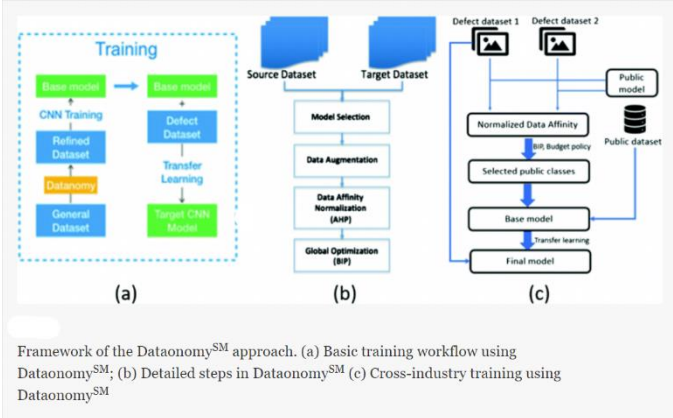
B. Dataonomy for Cross-Industry Applications

I. Dataonomy Approach workflow

Fig 1a and 1b shows the approach used by Dataonomy to prepare big datasets for the base model of the specific industry. This is later used for transfer learning to obtain final model.

It involves the following steps:

- To test the defect dataset, Dataonomy involves using a pretrained public available CNN model e.g., Inception V3
- Obtain mean probabilities of the defect dataset
- The probabilities are used for data augmentation. It quantifies how well the differences between defect classes in the defect dataset can be related to each class in the public dataset.
- The final step is to obtain global mapping scheme. This maximizes the performance and minimizes the supervision.
- This approach works for single type of surface mateial.
- When base model is shared by defect datasets from different industries, it can be used for cross-industry applications.
- There are highly relevant group of data classes for each industry in the total of selected classes for the base model training.



Framework of the Dataonomy^{SM} approach. (a) Basic training workflow using Dataonomy^{SM}; (b) Detailed steps in Dataonomy^{SM} (c) Cross-industry training using Dataonomy^{SM}

Fig 8: Framework of the Dataonomy approach [11]

- Data augmentation and AHP is applied separately.
- BIP method is applied to obtain selected classes for base model training.

II. Budget Policy for Cross-Industry Base Model

The aim of budget policy is to balance the number of selected classes for each industry. The selected classes for different

industries are balanced based on the differences in the training difficulty, available defect data size, and importance among different industries. The BIP design works under the premise that better predictions are yielded by base models trained with more classes. Using the new set of classes selected in the BIP process, the cross-industry base model can be created by retraining the model.

B. Federated Learning Framework

The framework has three stages:

a) Deployment Stage: A request is sent to the server for a new model for the inspection task by the client. The request is sent along with some basic information regarding the data. The server then decides on the approach to make. Typically, if the new client is from a new industry, a generalized cross-industry base model is given to be deployed directly for the production line initially. In case the new client is similar to the old client, a readily available customised model is given as the base model to train on its available data and run surface inspection tasks.

Deployment stage takes most of the time in the cycle. The expectation from the client is to collect new data at this stage to prepare for new model updates.

b) Server Model Update Stage: At regular intervals, server updates the model using the information from alive clients. An alogrithm is sent to the client along with the base model to extract low-level features. These features are extracted from the low-level convolutional layers of newly trained model. The CNN changes the output features from the raw image which ensures data privacy. These features are then fed into truncated CNN and the server uses Dataonomy to update the new base model in the server with more weightage for the new image features.

c) Client Model Update Stage: The server sends the updated base models to the clients. Each client train the new model locally with a small size of raw data. This results in low cost. Post this, the final models are applied in the inspection systems and new cycle starts from deployment stage.

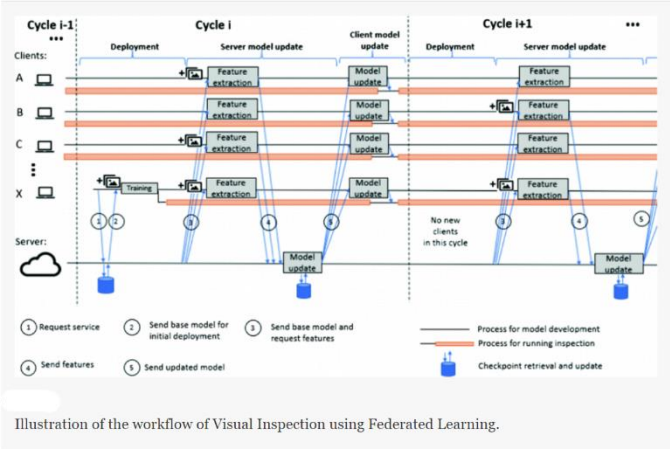


Fig 9: Illustration of the workflow of Visual Inspection [11]

C. Results

The base model trained on data sets from two different industries is used to obtain high quality models for the defect inspection tasks of each industry and the same is shown in the table below.

Model	Accuracy	Training time
Base model	85.5%	6.6 h
Wood model	100.0%	6 min
Texture model	99.7%	54 min

Fig 10: Models involved in the cross-industry approach (all trained on 8 GPU) [11]

It can also be seen from simulated FL process in one cycle on wood dataset that models are updated efficiently by the server.

Stage	Client computing cost	Client data size	Server training cost	Server data size
Deployment	1 min (few layers)	1.0 MB (42 images) and 92 MB (base model)	NA	NA
Server update	48 min (all layers)	23.3 MB (1573 images) and 92 MB (model)	19 h	5.2 GB (features) and 368 MB (models)
Client update	2 min (few layers)	1.0 MB (42 images) and 92 MB (base model)	NA	NA

Fig 11: Federated Learning process time cost and data size involved in one cycle. [11]

FEDERATED REGION LEARNING FOR URBAN ENVIRONMENT SENSING

A. Introduction

Fighting urban pollution requires environmental data collection and synthesis that relies on monitoring sites which are usually insufficient to obtain fine-grained environment status over the whole city. This leads to sparse sensory data which becomes the main challenge of fine-grained environment sensing. Researchers began to try big data technology along with some latest advances in ML such as deep learning to train inference models. To implement this, uniform model is built with centralized model. This gives rise to several problems such as computational efficiency and model performance as it omits the regional characteristics and needs massive training samples. To overcome this, Federated Region-Learning (FRL) is used which can learn a global model by aggregating locally computed updates.

B. Proposed Framework

The framework is divided into two phases [12]. In first phase, air quality monitoring sites are divided into regions. The sites tend to be redundant in some regions but lacking in others. To extract the regional characteristics of data and manage data

before training regional models, the sites are divided on the basis of geographical locations and other important features. A clustering algorithm is used for regionalization purpose which can be divided into two phases:

a) Based on sites location and relevance, construct a weighted network. The sites are considered vertexes (λ) and edge weight is equal to the ratio of correlation between changes in air quality to the distances between each site.
b) Using Girvan-Newman algorithm, the weighted network is divided to obtain the regional division of sites.

The second phase involves the use of Federated region-learning which focuses on the lower layer model such as a regional model instead of the central model on the top. The micro cloud collects the data within its own region and then download the shared model from the central server. Based on the global model, each micro cloud trains regional model using regional data. The lower layer models are tested and saved. The best regional model that infers the air quality data of the region with highest accuracy is uploaded to the central server to obtain a new global model. This new global model is distributed to each micro cloud in each iteration.

The federated region learning approach helps in reducing the communication cost significantly along with increased accuracy. It has proved to more effective and efficient with reduced requirement for massive training samples and without omitting the regional characteristics.

C. Results and Evaluation

For experiment, W&A dataset with $PM_{2.5}$ categories generated by weather/air- quality monitoring sites is used. This dataset contains 12 categories related to $PM_{2.5}$ such as temp., humidity, etc. Data from first 25 days of the month is used as the training set and the remaining days is used as the test set.

The centralized, standard and federated region-learning are then compared to evaluate their efficiency and effectiveness. To converge the W&A data as the dataset of each region. The data is tested separately for each region and the average value is picked up as the basis for evaluation. Post filtering invalid and default data, we are left with 4000 data for training and 1000 data for testing approximately per district. Table below shows the results for the highest accuracy models, and it is evident that the regional models achieved the higher accuracy in less communication rounds.

	Centralized training		Federated Learning		Federated Region-Learning	
B	ACC	CR	ACC	CR	ACC	CR
96	80.27	2560	75.64	492	**81.09**	820
196	79.64	1440	76.25	756	**81.05**	533.4

Fig 12: Comparison between the Centralized Training, Federated Learning and Federated Region-Learning [12]

FEDERATED LEARNING FOR IOT DEVICES

With rapid increase in smart devices and mobile networks, a new phase has emerged in the field of Internet of Things (IOT). As a result, we have access to large user data to produce insights, train task-specific machine learning models, and ultimately deliver high-quality smart services and products with the help of the Internet of Things. Lately, federated learning has been presented to train a globally shared model using many user-generated data samples on IoT devices while avoiding data loss. Traditional federated learning, on the other hand, has significant hurdles in complex IoT systems due to device, statistical, and model heterogeneity, making it unsuitable for straightforward deployment.

A. Main Challenges of Federated Learning in IoT Environments

- *Device Heterogeneity:* Currently, there are a significant amount of IoT devices with different hardware (CPU, memory), network conditions (3G, 4G, Wi-Fi) and power (battery level), resulting in a wide range of computational, storage, and communication capacity. This poses a challenge for federated learning as it results in high communication costs due to sluggish network or connection difficulties and devices with limited computation resources could result in slow model updates than other devices making it stragglers. Henceforth, it is important to address these challenges.
- *Statistical Heterogeneity*: The generated personal data from multiple devices may naturally display non-IID distributions due to users' varying usage environments and patterns. Feature distribution skew, label distribution skew, and concept shift are all examples of non-IID user data. Federated Averaging (FedAvg) was created to overcome these issues by working on non-IID data. But performance of this approach deteriorated when the data distribution is highly skewed compared to traditional centralized approach.
- *Model Heterogeneity:* Devices that participated in traditional federated learning framework have to agree on a specific training model architecture so that the global model could be efficiently produced by aggregating the model weights gathered from local models. But due to computational capacity, various devices try to craft their own models according to application environment and due to privacy concern the models generated might not be shared. Due to this, model architectures of various devices might exhibit diverse shapes making traditional federating learning impossible to perform aggregation.

B. *Cloud-edge Framework for Personalized Federated Learning*

As mentioned in section A, the challenges exist in device, statistical and model heterogeneity. To overcome these challenges personalization is required to achieve great flexibility by developing and utilizing more advanced federated

learning methods, allowing individual devices to craft their own personalized models to meet their computational resource and application requirements while still benefiting from federated learning for collective knowledge sharing. The PerFit framework is adopted, which is built on a cloud-edge architecture, which provides on-demand edge computing capacity near IoT devices. As a result, each IoT device may choose to outsource its demanding computing activities to the edge (i.e., edge gateway at home, edge server at work, or 5G MEC server outside) through wireless connections, allowing IoT applications to meet their criteria for high processing efficiency and low latency. Federated learning is used to train the shared global model by combining locally computed models across end devices, edge servers, and the cloud keeping sensitive data private.

C. *Experimental Results*

In this experiment, the results of personalized federated learning are compared with both traditional federated learning with FedAvg method and centralized learning. For personalized FL, two approaches were used: federated transfer learning (FTL) and federated distillation (FD). For centralized, SVM, kNN, RF and CNN are used, and large amount data are used to train these models. As we can see in figure 4, the accuracies of all the models. The FTL can reach an accuracy up to 95.37%, which is 11.12% more than the traditional federated learning approach.

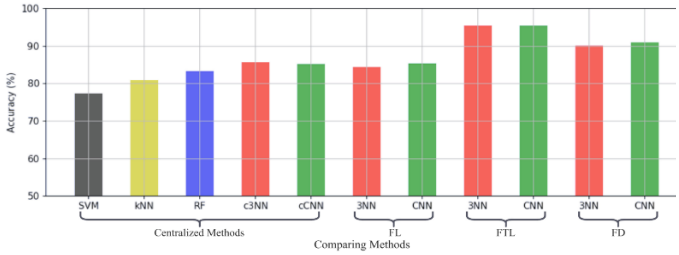


Fig 13: The accuracy of various learning approaches in human activity recognition [13]

FEDERATED LEARNING FOR VEHICULAR NETWORKS

Machine learning is being used in Vehicular networks for object detection, road safety, traffic management and autonomous driving. As the safety features are being enhanced in the autonomous driving, there has been a significant increase in the data being generated at the rate of 1 Giga Byte per sec from various sensors like RADAR, camera, etc. Moreover, training the ML model adds considerable cost to data communication between the parameter server and the vehicle's edge devices. The federated learning (FL) framework was recently proposed as an essential technique with the purpose of decreasing transmission overhead while simultaneously maintaining privacy by transmitting just model updates of the learnable parameters instead of the entire dataset.

A. *Federated Learning for Distributed Training in Vehicular Networks*

Traditional methods for training ML models in automotive applications use a central server that takes raw data from vehicular edge devices, calculates gradients based on the present state of the model and the new incoming data, and then changes the model parameters correspondingly. Because of the highly dynamic and harsh communication channel, many nodes in a given vehicular network cannot meet the requirements for high transmission operational costs and raw data privacy, impacting the model's local adaptation capability as some of the local data nodes cannot contribute to training the model.

FL's ability to adapt to local changes is dependent on the "mini-batch learning" method employed in conventional ML model training, in which the dataset is broken into smaller sub-blocks that are utilized for parameter updates instead of the entire dataset. The gradients are then determined for all these mini-batches, and then average of group of these gradients, is often used as gradient value to update parameter in each iteration, that is done repeatedly until convergence is reached. The central server then basically aggregates the received gradients and updates the model, after which the updated model parameters are transmitted to each automobile. This decreases communication overhead while maintaining privacy.

B. *Research Challenges and Future Directions*

The key problems that must be overcome in order to use FL in vehicle environments and take advantage of its promises of lower communication overhead, raw data confidentiality, and more flexible and efficient ML models, are shown in fig 14.

Although the communication overhead is lowered, effectively training ML models over vehicular networks with FL remains difficult due to the dynamic nature of the communication channel in the vehicular environment, which results in frequent dropouts and handovers owing to excessive vehicle mobility and diverse weather conditions. Furthermore, the decrease in communication overhead comes at the expense of higher computational overhead at vehicular edge devices.

The main Learning-Related challenges that are faced by FL approach in the paper [14] are:

- *Data diversity*, caused because of irregular distribution of dataset at edge devices. For example, the features of image data in various places enhance the dataset's heterogeneity, making NN incapable of performing feature extraction and feature representation. A viable choice would be to increase the model size, which includes increasing the width and depth of the NN model.
- *Labelling,* as supervised ML techniques are used, labelled dataset is necessary to train the model and data labelling requires a certain amount of work. Need an efficient approach in the future to overcome this. Possible solution would be reinforcement learning.
- *Efficient Model Training,* while performing transfer learning, the diversity of the datasets might cause complications because of dataset's non-uniform distribution, allowing for a shallow ML model without

the emphasis on TL As a result, new techniques are required and should be created to make FL model training more effective in Applications for vehicular networks.

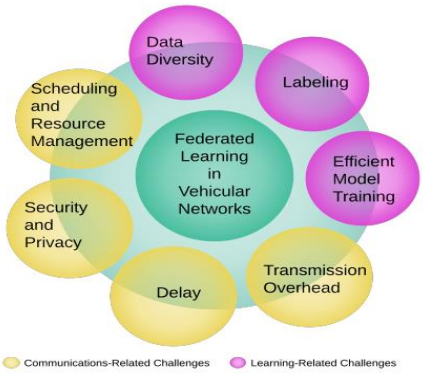


Fig 14: Design and research difficulties for federated learning in vehicular networks are summarized. [14]

COMPARISON

	Electronic Health Records	Visual Inspection	Intelligent IoT Applications	Out-Of-Vocabulary words	Urban Environment Sensing	Vehicular Networks
Application Domain	Predict future hospitalizations for patients.	Image Detection in Industrial Engineering.	To cope up heterogeneity issues in IoT environment using personalized federated learning methods.	Smart phone keyboard which provides suggestions, auto-correction, gesture typing, and many more features.	Urban environment sensing with air quality monitoring.	To cope up with sensor data in autonomous vehicles.
Model Used	Federated Learning + Cluster Primal Dual Splitting algorithm.	Federated Learning + Dataonomy + CNN	Federated Transfer Learning + Federated Distillation - 3NN, CNN.	Federated Averaging + RNN (Recurrent Neural Network) + Multilayer LSTM models.	Girvan-Newman clustering algorithm + Federated Region Learning.	Federated Learning+ Lyapunov optimization.
Performance	Area under ROC obtained is 78.06%.	Accuracy obtained is 97.7% on build-in Inception V3, 99.12% on Wood dataset.	Highest Accuracy obtained in personalized federated learning is 95.37% with FTL.	90.56% Precision and 81.22% Recall with an accuracy of 66.3% and 1.887 CE loss.	3.12 times and 2.7 times less communication rounds and obtains 5.45% and 4.8% higher accuracy than other models.	37 times lower transmission overhead as compared to ML.
Pros	Uses only a few features to build classifiers.	Able to detect defects with less number of training samples.	Need less samples to train the data and data heterogeneity problem is also taken care in this approach.	The vocabulary is increased without exporting any sensitive information from the user's device.	No omission of regional characteristics.	Decreases transmission overhead.
Constraints	Requires more iterations to achieve convergence.	Need quick model deployment to serve various industries.	Need more time to train the shared model compared to traditional approach.	Relies heavily on probabilistic mode.	It is a two-layer structure. To cover wider area, multi-layer structure is required.	Higher computational overhead at vehicular edge devices.

IV. FINDINGS AND LIMITATIONS OF FEDERATED LEARNING

FINDINGS

- In this approach, model is sent to client location for training where data transfer is not required. This anonymizes the data to be GDPR compliant and keep it secure. This ensures data privacy.
- Using local data at client location for training purpose prevents data transfers over the internet continuously which, otherwise, is very costly (communication cost).
- Since it is a decentralized system, data doesn't need to be stored at one location i.e., server. This resolves data storage issue and allows usage of large-scale data.
- Models using FL approach can be trained using less sample data.
- To overcome the limitations in computing and communication part, edge devices are used in FL for IoT devices.
- The data is generated continuously from the clients which requires model to be retrained at regular intervals. As both the data and model are residing at client location, it is easy and less time-consuming to train the model.
- An advantage of this approach is that it allows clients from a new industry to deploy a base model quickly and perform training with their own dataset and send it back to server which updates the global model. This enables cross-industry collaboration.

LIMITATIONS

- Since there are multiple clients involved in FL, thus, the interactions between these clients and server will be more, which further increases the communication cost.
- In the process of enhancing data privacy, the model performance or system efficiency gets compromised. This leads to trade-offs between them.
- The data generated on client-side may be mislabelled or non-labelled.
- Domain experts are expected to be involved in the FL process to provide professional guidance as users tend to trust the experts.
- Handling data diversity in the to train an efficient model is a challenge in FL.
- FL approaches are well suitable for large data. For less dataset, the performance will be average.

V. SUGGESTIONS FOR MODIFICATIONS

- To enable secure computation as well as low communication and computation, SAFER is used. It compresses the model and then sends it to the server.
- A semi-supervised learning-based technique can be used to label the mislabelled or non-labelled client data. However, the solution may require dealing with the data privacy, heterogeneity, and scalability issues.
- Unsupervised approach like Collaborative and Adversarial Network (CAN) can be used, which reduces the need of labelled data and shows effectiveness and high performance.
- A Muti-task learning approach can be used. A multi-task approach solves multiple learning task at the same time by using the common properties and differences of the tasks. This overall improves the efficiency and accuracy of the algorithm.
- Using multi-layer structures of FRL, distributed learning problem can be solved in many more fields.
- Transfer learning and distributed learning approach yields efficient results, if used with federated learning.

VI. CONCLUSION

This study explores the applications of Federated Learning in fields of Healthcare, Industrial Engineering, and IoT applications. Post reviewing a few papers from these domains, it is evident that FL has made significant improvements in terms of data privacy, reliability, efficiency, and performance. Also, one of the major bottlenecks that can be observed in all these domains is that they require large datasets, however, FL approach can achieve the same or even higher accuracy with smaller datasets as well. We further conclude that the cPDS approach developed for the healthcare domain is fully decentralized and yields classifiers using only limited features. It also showed improved convergence rate over other methods. Furthermore, experiments demonstrated the viability and efficiency of cross-industry modeling in the FL framework. Personalized FL handles device, statistical, and model heterogeneity in IoT environments by pooling local updates from remote IoT devices and utilizing the benefits of edge computing to develop a globally shared model. This has helped industrial engineering where detailed characteristics of data were getting omitted due to traditional approaches leading to data sparsity issues. FL in vehicular networks helps in lowering communication overhead, raw data confidentiality. Overall, federated learning holds the advantage over all the traditional approaches.

VII. CONTRIBUTIONS

Abstract	Sungamesh Jayveera Prasad
Introduction	Diksha Srivastava
Literature Review	Saumya Goel
Federated Learning of predictive models from electronic health records, Federated Learning of out-of-vocabulary words.	Diksha Srivastava
Federated Learning for visual inspection, Federated region learning for urban environment sensing.	Saumya Goel
Federated Learning for IoT Devices, Federated Learning for Vehicular Networks.	Sungamesh Jayveera Prasad
Comparison	Diksha, Saumya, and Sungamesh
Findings	Diksha Srivastava
Limitations	Saumya Goel
Suggestions for modifications	Everyone contributed as per their papers.
Conclusion	Sungamesh Jayveera Prasad

VIII. REFERENCES

[1] "Architecture," [Online]. Available: https://miro.medium.com/max/4888/1*2lWjCr7FRAu-MblI_8shtw.png.

[2] R. V. Anuja Kumari, "Classification of diabetes disease using Support Vector Machine," vol. 3, no. 2, 2013.

[3] K. C. M. F. C. Moses Charikar, "Finding frequent items in DataStreams," 2002.

[4] Y. Zheng, F. Liu and H. Hsieh, "U-Air: When Urban Air Quality Inference Meets Big Data," 2013.

[5] Z. Yu, X. Wu and X. Gu, "Fully Convolutional Networks for Surface Defect Inspection in Industrial Environment," 2017.

[6] A. Jindal, G. S. Aujla, N. Kumar, R. Chaudhary, M. S. Obaidat and I. You, "SeDaTiVe: SDN-Enabled Deep Learning Architecture for Network Traffic Control in Vehicular Cyber-Physical Systems," 2018.

[7] Z. Zhang, H. Ma, H. Fu, L. Liu and C. Zhang, "Outdoor Air Quality Level Inference via Surveillance Cameras," 2015.

[8] J. Konecny, H. B. McMahan, F. X. Yu, A. T. Suresh and D. Bacon, "Federated Learning: Strategies for improving communication efficiency," 2017.

[9] R. C. T. M. A. O. I. C. P. W. S. Theodora S. Brisimia, "Federated learning of predictive models from federated

Electronic Health," *International Journal of Medical Informatics,*, pp. 61-63, 2018.

[10] R. M. T. O. F. B. Mingqing Chen, "Federated Learning of Out-Of-Vocabulary words," 2019.

[11] H. Y. H. G. Xu Han, "Visual inspection with federated learning," in *Image analysis and recognition*, 2019.

[12] B. Hu, Y. Gao, L. Liu and a. H. Ma, "Federated Region-Learning: An Edge Computing Based Framework for Urban Environment Sensing," 2018.

[13] K. H. a. X. C. Q. Wu, "Personalized Federated Learning for Intelligent IoT Applications: A Cloud-Edge Based Framework," *IEEE Open Journal of the Computer Society*, vol. 1, pp. 35-44, 2020.

[14] A. &. C. S. Elbir, "Federated Learning for Vehicular Networks".