FEDERATED LEARNING IN SPAM DETECTION WITH LOGISTIC REGRESSION ON DIFFERENT FEATURES

Aalto University, Espoo, Finland

ABSTRACT

This paper studies federated learning methods for spam detection. The logistic regression is leveraged as local model. To optimize the local model, FedSGD is selected as the optimization algorithm. Two methods will be applied to study the data features and explore the influence of data feature to the FL model. The empirical graph is constructed to build communication with different local models. The experimental result shows our model is efficient.

***Index Terms*—** Federated Learning, Networks, Spam Detection

1. INTRODUCTION

Spam detection plays a crucial role in ensuring a clean and secure communication environment [1, 2]. However, traditional approaches often require centralizing data, raising privacy concerns and legal constraints. In recent years, federated learning has emerged as a promising solution, enabling collaborative model training while preserving data privacy [3]. Federated learning based spam detection, where models are trained on decentralized nodes without the need to transfer sensitive data to a central server. By leveraging the collective intelligence of multiple clients, federated learning offers an innovative approach to combat spam while respecting privacy constraints. In this paper, we construct a federated spam detection model and explore different types of local models and different edge weights of empirical graphs to prove federated model is more efficient and safe than centralized model.

In order to protect the privacy, SMPAI [4] was proposed to protect against the attacks based on secure multiparty computation and differential privacy. Two mechanisms are applied to generate the distributed noise of each client: one is to encrypt the noise distribution from other clients and add to the client's own distributions; the other one is to aggregate its neighbor's noise distribution and the final result will be calculated by server.

Federated learning is applied in many fields except spam detection in computer vision, natural language processing and other fields. MOON [5] combines generative model to utilize the similarity between model representations to correct the local training of individual parties and improve the deep learning models' performance on image datasets. CNFGNN [6] combines federated learning with graph neural network to process Spatio-Temporal data, which can be applied in traffic flow prediction. FedNLP [7] applies federated learning on NLP tasks such as question answering and text classification, which is highly relevant with spam detection.

The following parts of this paper are organized as follows: Section 2 will introduce the dataset, local models and the global objective function. Section 3 will explain the data preprocessing method, local model choice and the FL algorithm selected to train the models. Section 4 will introduce the results obtained from each local model and the final chosen method. The loss function of final model will also be introduced.

2. PROBLEM FORMULATION

In this section, the dataset and empirical graph will be introduced.

2.1. Datasets

The selected dataset is called 'Smsspamcollection' [1], which is collected from Kaggle[2]. The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged acording being ham (legitimate) or spam. Each data point contain 1 processed message and label. In the Smsspamcollection dataset, the quantity of interest (label) is whether each SMS message is classified as spam or ham. In this paper, The local dataset is defined as

$$\mathcal{D}^{\rangle} = \{(\mathbf{x}^{(i,1)}, y^{(i,1)}), \ldots, (\mathbf{x}^{(i,m_i)}, y^{(i,m_i)})\}, \qquad (1)$$

where x and y respectively denote the features and the label of the r-th data point in the local dataset $D^{(i)}$.

2.2. Empirical Graph

The empirical graph is defined following [3]:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \qquad (2)$$

[1]https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset
[2]https://www.kaggle.com/

where nodes $\mathcal{V}$ denote the local dataset $D^{(i)}$ and edge $(i, j) \in \mathcal{E}$ denote the similarity between local datasets i and j. In our model, the empirical graph contains five nodes, which indicates that we have five local models and datasets.

To explore the data intrinsic similarity of local datasets, the parameters $\mathbf{w}^{(i)}$ of each local model will be projected to euclidean space. The distance between the nodes in the empirical graph can be written as:

$$A_{i,j} = ||\mathbf{w}^{(i)} - \mathbf{w}^{(j)}||^2 \tag{3}$$

The distance can be calculated through different distance measurement methods such as KL-divergence, JS-divergence or Wasserstein distance [8, 9, 10]. To learn the best parameters $\mathbf{w}$, the objective function of GTVMin can be written as:

$$\hat{w} \in \arg\min_{w \in \mathcal{W}} \text{GTVMin} \sum_{i \in \mathcal{V}} L_i(\mathbf{w}^{(i)}) + \lambda ||\mathbf{w}||_{\text{GTV}}, \tag{4}$$

where $L(\cdot)$ denotes the loss function of local model and $\lambda > 0$ denotes the regularization parameter.

3. METHODS

This section explains our data split strategy and feature preprocessing method in detail. After the data is prepared, the local model choice and variation measurement will also be introduced. Then, the applied FL algorithm will be introduced.

3.1. Data split

Because our dataset has 5,574 data points, to ensure the fairness of local dataset, we used the first 5,500 data points and equally divide it into five local datasets: $\{D^{(0)}, \ldots, D^{(4)}\}$. We preserve 10% of the data points as our test set, and the other data points are used as train set which is fairly divided into 5 local datasets. In each local data set, we use 90/10 strategy to divide the local dataset into train set, validation set and test set, which means that 90% of the data points will be used in training the model and 10% will be used to validate the local model's performance and find the best parameters.

3.2. Feature Selection

In our dataset, the feature selection process involves transforming the raw text data (email content) into numerical features that can be used by machine learning algorithms for classification. "Bag-of-Words" representation is applied with the help of the CountVectorizer class from scikit-learn[3]. Before using CounterVectorizer, we first fill the missing values by '0' and randomly sample 5,500 data points. The CountVectorizer is initialized and fitted to the text data, which constructs a vocabulary of unique words in the training data. The text data is then transformed into a numerical feature matrix where each row represents a document (email) and each column represents the frequency of a specific word. It is worth noticing that the stopping words (e.g., "a," "the," "and") may not provide much discriminatory power in classification. In our processing, the stopping word will also be removed.

[3]https://scikit-learn.org/

3.3. FL Algorithm

In this part, FedSGD [3] will be introduced.FedSGD (Federated Stochastic Gradient Descent) is an optimization algorithm which involves multiple communication rounds between a central server and participating clients. In each round, the server distributes the global model to the clients, who train the model on their local datasets using mini-batch stochastic gradient descent. The clients send their updated local models back to the server, which aggregates the models to update the global model. This iterative process allows collaborative training of a global model while preserving data privacy and leveraging local data and computing resources on client devices. The objective function can be written as:

$$\nabla f(\mathbf{w}) = (\mathbf{g}^{(0)}, \ldots, \mathbf{g}^{(n)})^T \tag{5}$$

. Take L2 norm as an example, The gradient $\mathbf{g}^{(r)}$ is expressed as:

$$\mathbf{g}^{(r)} := \nabla L_i(w^{(i)}) + 2\lambda \sum_{i' \in \mathcal{N}^{(i)}} A_{i,j}(\mathbf{w}^{(i)} - \mathbf{w}^{(j)}) \tag{6}$$

.

3.4. Local Model Selection

In this part, the local model, loss function and discrepancy measurement will be introduced.

The logistic regression [11] is selected as the local model. Logistic regression is selected due to its simplicity, efficiency, and interpretability. As a binary classification algorithm, logistic regression is well-suited for tasks where the objective is to distinguish between two classes, such as spam and ham classification in the Smsspamcollection dataset. Logistic regression models can be trained efficiently on local datasets, requiring less computational resources compared to more complex models. Moreover, the resulting coefficients in logistic regression provide interpretable insights into the importance and direction of features, aiding in understanding the factors influencing the classification decision. Hence, logistic regression serves as a practical and effective choice for local models in federated learning scenarios.

To measure the edge weight between two nodes in the empirical graph, we use Wasserstein distance. This divergence measure quantifies the difference between two probability distributions and can be used to assess the variation in the models' predictions.

The loss function L is logistic loss, which is also called binary cross-entropy loss. The logistic loss is commonly used for binary classification tasks, such as spam detection. It is suitable when the goal is to estimate the probability of an instance belonging to a particular class (e.g., spam or ham). The logistic loss penalizes incorrect predictions by assigning higher loss values when the predicted probability deviates from the true label. It is a smooth and differentiable loss function, making it compatible with gradient-based optimization algorithms such as stochastic gradient descent (SGD) used in logistic regression.

4. RESULTS

In this part, we run our model on the 'Smsspamcollection' dataset in two methods: the first one is that each local model learns the first 50 features, and the second one is that each local model learns the first 80 features. We construct the edges in the empirical graph in this method: each node is connected to 2 nearest neighbours with distance being measured via wasserstein distance because wasserstein distance can measure the distribution of labels and prdicted values, which reflects the similarity of two local datasets. Compared with KL divergence, wasserstein distance is symmetric and smooth, which provides gradients when using gradient descent algorithms to optimize the model. In our experiment, we found that using KL-divergence as the edge weight will produce similar results with the situation that applies wasserstein distance. The constructed empirical graph based on the wasserstein distance is shown in Fig. 1 and Fig. 2.

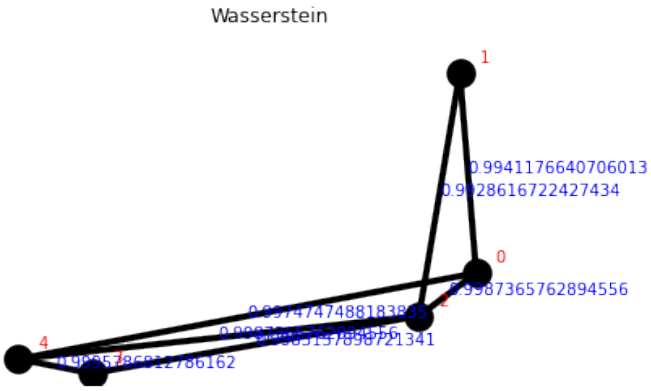

Fig. 1: The empirical graph with local dataset contains first 50 features.

Our results of the mentioned method are in Table 1 and Table 2. We get this result according to the parameters after cross validation through grid search method: maxmimum number of iterations = 1000 ,learning rate= 0.01, regulation parameter $\lambda = 0.1$. The results indicate that the loss is in a low level, which proves that our method is feasible and effective on spam detection task. It also shows that logistic regression can achieve better performance with more features. Therefore, our final data will contain 80 features in a data point.

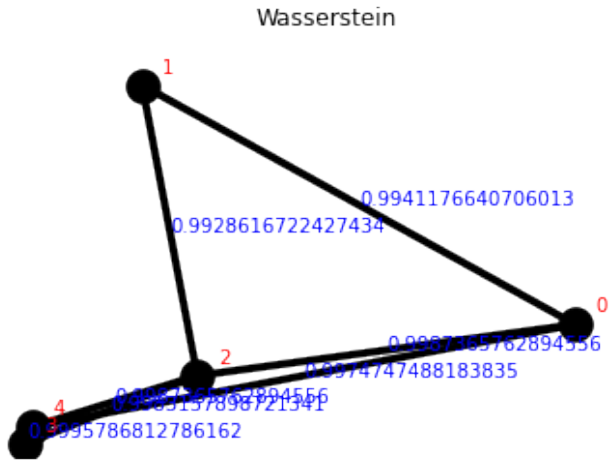

Fig. 2: The empirical graph with local dataset contains first 80 features.

	Node 1	Node2	Node 3	Node 4	Node 5
Train Loss	2.31	2.23	2.36	2.49	2.49
Val Loss	1.56	1.57	2.35	2.74	5.50
Test Loss	2.32	1.72	2.26	2.67	2.45

Table 1: The train and val loss on local models when the first 50 features are selected.

	Node 1	Node2	Node 3	Node 4	Node 5
Train Loss	1.83	1.66	1.66	1.70	1.83
Val Loss	1.17	1.96	1.96	1.96	3.53
Test Loss	1.44	1.44	1.70	1.70	1.56

Table 2: The train and val loss on local models when the first 80 features are selected.

The loss curve is in Fig. 3. The loss curve shows that the local model converges in few iterations.

In conclusion, With the experiment results, we finally choose logistic regression with 80 features as our local model. The empirical graph first connect the nodes with its two nearest neighbors and calculate the edge weight through wasserstein distance. FedSGD is applied to optimize the local models. The test error of each local model,which is shown in the two tables, is nearly equal to or slightly less than the train error.

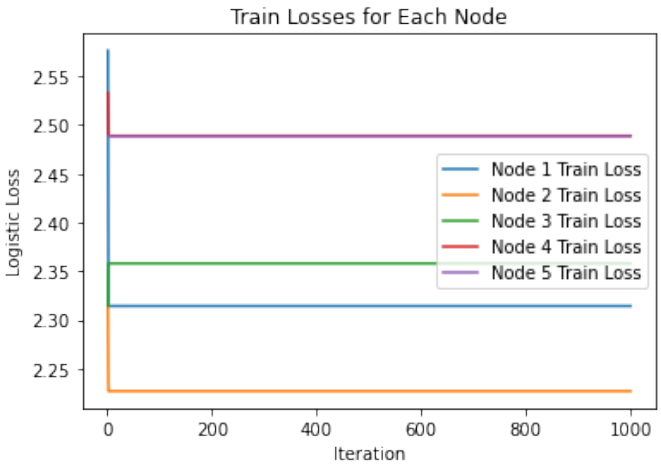

Fig. 3: The train loss curve of each local model.

5. CONCLUSION

In this paper, we explored the federated learning method on spam detection. The logistic regression is applied and FedSGD is leveraged to optimize the local model. We construct the empirical graph with five nodes and each node is connected to its two nearest neighbors in wasserstein distance instead of KL divergence due to wasserstein distance's characters. Through setting different numbers of feature, we explore the feature's influence on local model's performance. The results shows that our final chosen method is efficient and feasible on spam detection task. The results may be improved by more complex models such as MLP or decision trees.

However, there still some points can be optimized: in data preprocessing, due to the limited computational resource, we applied the CounterVectorizer but not modern embedding methods such as Word2Vec [12]. The local models can also be replaced by deep network if there is enough computational resource.

6. REFERENCES

[1] Nitin Jindal and Bing Liu, "Review spam detection," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 1189–1190.

[2] Arvind Mewada and Rupesh Kumar Dewang, "A comprehensive survey of various methods in opinion spam detection," *Multimedia Tools and Applications*, vol. 82, no. 9, pp. 13199–13239, 2023.

[3] Alex Jung, "Introductory lectures on federated learning," 2023.

[4] Vaikkunth Mugunthan, Antigoni Polychroniadou, David Byrd, and Tucker Hybinette Balch, "Smpai: Secure multi-party computation for federated learning," in *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, 2019.

[5] Qinbin Li, Bingsheng He, and Dawn Song, "Model-contrastive federated learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10713–10722.

[6] Chuizheng Meng, Sirisha Rambhatla, and Yan Liu, "Cross-node federated graph neural network for spatio-temporal data modeling," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1202–1211.

[7] Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr, "Fednlp: A research platform for federated learning in natural language processing," *arXiv preprint arXiv:2104.08815*, 2021.

[8] "Kullback–Leibler divergence - Wikipedia," .

[9] "Jensen–Shannon divergence - Wikipedia," .

[10] "Wasserstein metric - Wikipedia," .

[11] Michael P LaValley, "Logistic regression," *Circulation*, vol. 117, no. 18, pp. 2395–2399, 2008.

[12] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

Response to the Reviewer Comments on Manuscript "Federated learning in Spam Detection"

July 2, 2023

We express our sincere gratitude for the insightful and constructive comments and suggestions provided by the reviewers. We have tried to address all these comments to the extent possible. Major modifications we implemented in the revised manuscript include the following:

- We have modified the source code and imported the necessary packages to fix the problem in the implementation.
- We have modified the title to summarize the motivation of this paper.
- We now optimized the figure of empirical graph to show the edge weight.

In the following, we respond to the reviewer comments in a point-by-point manner. Section, page, equation, and reference numbers in the copied action editor and reviewer comments (typeset in italic print) refer to the original manuscript whereas those in our response (typeset in upright print) refer to the revised manuscript unless indicated otherwise.

Comments of Reviewer #1

1.1 *Title "FEDERATED LEARNING IN SPAM DETECTION" is describing the content, however could be improved to be more spesific with used methods*

We have revised the title to better clarify the used methods. The new title is "FEDERATED LEARNING IN SPAM DETECTION WITH LOGISTIC REGRESSION ON DIFFERENT FEATURES".

1.2 *The jupyter notebook page seemed not to exist.*

The uploaded version has some mistakes, but the jupyter notebook was uploaded as "FLProj.zip". The new version of the jupyter notebook has fixed all the problems.

Comments of Reviewer #2

2.1 *The description or motivation could have been more detailed. The model parameters of the emp.graph could have been given for each FL method.*

The parameters of the empirical graph is evaluated by Wasserstein distance when evaluating Logistic regression with 50 features and with 80 features. The motivation is expanded in the paper.

Comments of Reviewer #3

3.1 *Final graphs are showcased in figures 1 and 2, but the weights are not mentioned by number.*

We have added the edge weights to the figures.

Comments of Reviewer #4

4.1 *Does the report discuss at least two different FL methods that are based on GTV minimization? Only FedSgd and logistic regression for local models is discussed.*

We discussed and evaluated the performance of logsitic regression with different feature numbers, which can be considered as two different FL methods and fulfills the requirement of the project.

4.2 *The use of wasserstein distance is mentioned, but not really motivated.*

We have added the reason why wasserstein distance is applied, and the difference of wasserstein distance and KL divergence is also discussed.

4.3 *There doesnt seem to be much of a distinction between test and validation sets. It would be good to state what goes in validation and what in test instead of saying: "10% will be used to validate the local model's performance and find the best parameters."*

The validation set is used to select the parameters of local models, and the test set is used to evaluate the performance of the model. There is no special distinction between test and validation set.

4.4 *I guess it's implied that the model with 80 features is selected, but all that is said is that that model performs better, and "With the experiment results, we finally choose logistic regression as our local model."*
I think it would be food to explicitly state that you pick the 80 feature model instead of saying you chose logistic regression, since that was anyway the only model that was discussed.

The conclusion is modified according to this valuable comment.

4.5 *the empirical graphs are visualized, which is nice, but I don't see anything about edge weigths, maybe those could be added in.*

The edge weights now are shown in figure 1 and figure 2.

4.6 *The conclusions give a decent overview, but I feel like they're a bit short. Also I'm not sure why the train loss is plotted in the conclusions, I feel like test would be more appropriate.*

We have added some details into the conclusion: why wasserstein distance is selected and what we explored on local model. As for why there is no test loss plots, the reason is that the train loss shows that FedSGD optimize the local model properly, and the test loss is shown in Table 1 and 2 in a concise but precise way.

Comments of Reviewer #5

3.1 *kneighbors_graph call in def AddEdge is not defined.*

The jupyter notebook is modified and now it works well.