Cross-Silo Horizontal Federated Learning for Flow-based Time-related-Features Oriented Traffic Classification

Umer Majeed, Latif U. Khan, and Choong Seon Hong
Department of Computer Science & Engineering, Kyung Hee University
Yongin, South Korea
{umermajeed, latif, cshong}@khu.ac.kr

***Abstract*—Traffic classification (TC) has a principal function in autonomous network management. Recently, deep learning and machine learning-based TC have become popular than the traditional port-based and protocol-based TC due to practices such as port disguise and payload encryption. The flow-based TC is reliable as it relies on time-related statistical features. Federated learning is a distributed machine learning technique to train improvised deep/machine learning models with less privacy distress. The organizations or enterprises having similar business models may take participation in building a federated model for their network traffic characterization. In this study, we build a cross-silo horizontal federated model for TC using flow-based time-related features. The federated model shows comparable performance to the centralized model.**

***Index Terms*—Cross-silo, Horizontal federated learning, Tensorflow federated, Traffic classification**

I. Introduction

The introduction of bandwidth-intensive applications, has risen the need for network traffic engineering. Traffic classification (TC), as part of network traffic engineering, is essential for maintaining the quality of Experience (QoE) and Quality of Service (QoS). Enterprises may do TC to monitor their users' activity, to maintain efficient access control, and to allocate network resources in an optimized manner. ***Virtual private networks*** (VPN) are utilized by enterprises to securely connect employees to the enterprise network. VPN encrypts the traffic at the packet-level which makes TC a tough task. ***Traffic classification*** tasks may be further decoupled as encrypted/non-encrypted identification, VPN/non-VPN characterization, and application-level TC.

Multiple organizations may collaborate to build a better network traffic classifier. However, the traffic data may be sensitive, and may not be centralized due to privacy issues. The ***federated learning*** (FL) [1], [2], [3], [4] approach allows building machine learning models without the need to centralized the data of the central server. ***Cross-Silo FL*** [5] involves a small number of relevant companies which share incentive to train a shared model, but are unable to share their data due to privacy concerns. To mitigate privacy issues, these companies or organizations collaborate with reliable communications and substantial computing capabilities to build a FL model. ***Horizontal federated learning*** (HFL) [6], alternatively called as sample-based FL is FL paradigm where the samples in the clients' datasets have a lot of overlapping features. HFL is typically done between the clients who are operating on similar business models.

In this work, we did cross-silo HFL for TC using Tensorflow Federated (TFF). Whereas, TFF [7] is an open-source FL framework released by Google in March 2019. The framework provides a local single-machine run-time simulation environment for machine learning on decentralized datasets.

The contribution of this study is summarized as follows:

- We proposed and designed a cross-silo horizontal federated learning (HFL) scheme for TC based on supervised feature-based deep learning.
- We trained a federated learning (FL) model to classify the traffic as VPN or non-VPN network traffic in cross-silo HFL settings based on flow-based time-related features.
- We trained a FL model for application-level TC (e.g. P2P, VoIP, VPN-P2P, etc) in cross-silo HFL settings on basis of flow-based time-related features.
- The FL models outperforms the individual models of each silo but shows comparable accuracy to the centralized models without compromising the privacy of the traffic in two silos for both scenarios.

The rest of the paper is organized in the following way: Section II gives a brief literature review for TC. The system model is presented in Section III followed by Problem Formulation and Dataset details in Section IV and Section V respectively. Section VI gives evaluation results and Section VII will conclude the paper.

II. Related Work

The main techniques for TC are port-based, flow-based, deep packet inspection (DPI)-based, and behavior-based TC [8].

The port-based TC assumes that unique ports are assigned to the underlying application as per the policies of Internet Assigned Numbers Authority (IANA) [9]. The port-based TC methods have become lesser accurate because of random port assignment, shared IP address, and port obfuscation. Many

This work was supported by Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(2020-0-00364, Development of Neural Processing Unit and Application systems for enhancing AI based automobile communication technology) *Dr CS Hong is the corresponding author.

contemporary applications are now operating on port numbers originally associated with other network traffic types.

The flow-based TC methods generally use time-based statistical features. This technique relies on the presumption that these statistical features have unique patterns for different applications. Draper-Gil *et al.* in [10] introduced the state-of-art "ISCX VPN-nonVPN traffic dataset" in 2016. They classify traffic type as well as application types on this dataset using KNN and C4.5 decision tree algorithms on time-related statistical features for both non-VPN and VPN-tunneled network traffic. However, KNN has high memory requirement and is computationally expensiveness at the prediction stage. Moreover, non-parametric models such as KNN and C4.5 algorithms are not feasible in FL settings as they require the centralization of explicit features values.

Lotfollahi *et al.* in [11] studied deep packet inspection (DPI) based TC through deep learning. The convolutional neural network (CNN) and stacked autoencoder (SAE) were employed for traffic characterization, application-level identification as well as VPN/non-VPN characterization. The main advantage of DPI is it can be applied directly to TCP/UDP packets without explicit feature extraction from the traffic data. The TC through DPI techniques assumes that the payload of the packets is available for inspection by traffic classifier, which is subject to privacy concerns. However, in FL settings, each enterprise can build its local model. These local models are aggregated to build a global model for improved performance.

Host-behavior-based TC depends on observation of the social interaction among end-points (servers and hosts) to classify traffic types. X. Zeng *et al.* in [12] used host behavior and flow context for Shadowsocks's traffic identification with high accuracy than current state-of-art methods. Their approach relies on the inference that the connection patterns (such as number of other end-points contacted, number of ports, transport-layer protocol employed) between source and destination hosts, attributes related to flow correlation, as well as host-DNS behavior, are discriminative enough to classify underlying traffic type running on hosts.

III. System Model

We consider the federation O of two organization $I, J \in O = \{I, J\}$ having similar business models, which take part in Horizontal Federated Learning (HFL). The dataset of organization I and J are $D_I = \{(\mathcal{X}_i, \mathcal{Y}_i)\}$ and $D_J = \{(\mathcal{X}_j, \mathcal{Y}_j)\}$ respectively. The sample space of these datasets is different, but feature space and label space pair of the two datasets i.e., $(\mathcal{X}_i, \mathcal{Y}_i)$ and $(\mathcal{X}_j, \mathcal{Y}_j)$ are same. Formally [13],

$$\mathcal{X}_i = \mathcal{X}_j, \quad \mathcal{Y}_i = \mathcal{Y}_j, \quad \forall \mathcal{D}_i, \mathcal{D}_j, \quad i \neq j \tag{1}$$

$$\text{where} \quad \forall \mathcal{D}_i \in D_I \text{ and } \forall \mathcal{D}_j \in D_J, \quad I \neq J. \tag{2}$$

Whereas, I and J are organization identifiers and are assumed to be different.

Fig. 1 depicts the system model. M_I^F and M_J^F denotes the local models trained in FL setting on D_I and D_J respectively. Whereas, M_F denotes the FL model trained by the federated averaging process.

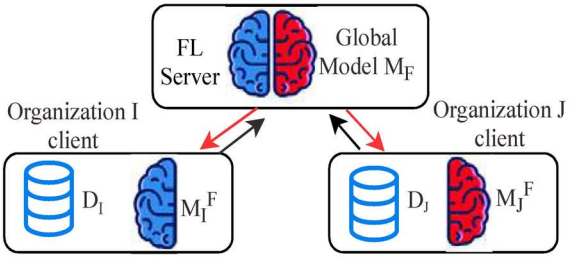

Fig. 1. Cross-Silo Federated Learning between two organizations

IV. Problem Formulation

We use the deep dense neural network ((DNN) to predict the target classes from the feature-based dataset using a supervised learning approach. We consider the C class classification task on a compact Euclidean space $\mathcal{X}$ and label space $\mathcal{Y} = [C]$, where $[C] = \{1, ..., C\}$. The objective function of the DNN model is to minimize overall loss function as

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) \quad \text{where} \quad f(\boldsymbol{w}) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{r=1}^{n} f_r(\boldsymbol{w}). \tag{3}$$

where n is total number of all the data points. The multiclass cross-entropy loss over one-hot-encoded labels on data point $\{\boldsymbol{x}, y\}$ is given as [14]

$$f_r(\boldsymbol{w}) = -\sum_{q=1}^{C} \mathbb{1}_{y=q} \log p_q(\boldsymbol{x}, \boldsymbol{w}). \tag{4}$$

Where, $p_q(\boldsymbol{x}, \boldsymbol{w})$ is the probability that $x \in X$ belongs to class q given $\boldsymbol{w}$ as the weight matrix of the neural network. In federated settings, the overall loss function is given as

$$f(\boldsymbol{w}) = \sum_{k \in O} \frac{n_k}{n} F_k(\boldsymbol{w}), \tag{5}$$

$$\text{where} \quad F_k(\boldsymbol{w}) = \frac{1}{n_k} \sum_{r \in \mathcal{D}_k} f_r(\boldsymbol{w}). \tag{6}$$

here, F_k denotes the local loss.

The Federated averaging (FedAvg) [1] to aggregate the local model weights at global iteration $t+1$ is given as

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \sum_{k \in O} \frac{n_k}{n} g_k, \text{ where} \quad g_k = \nabla F_k(\boldsymbol{w}_t) \tag{7}$$

$$\text{then,} \sum_{k \in O} \frac{n_k}{n} g_k = \nabla F(\boldsymbol{w}_t). \tag{8}$$

The local model weights at global iteration $t+1$ are updated as

$$\forall k, \boldsymbol{w}_{t+1}^k \leftarrow \boldsymbol{w}_t - \eta g_k. \tag{9}$$

The global model weights at global iteration $t+1$ are updated as

$$\boldsymbol{w}_{t+1} \leftarrow \sum_{k \in O} \frac{n_k}{n} \boldsymbol{w}_{t+1}^k. \tag{10}$$

V. Dataset

We used UNB ISCX VPN-NonVPN network traffic dataset [10] for TC. The dataset provides the time-related features and classes for the 4 different timeouts (120s, 60s, 30s, 15s). The list of time-related features can be seen in Table 2. of [10].

1) Scenario A: This scenario differentiates between the regular traffic and traffic tunneled through VPN. So we have two classes VPN and Non-VPN in this scenario.

2) Scenario B: The scenario characterize the traffic type in addition to VPN identification. So we have 14 types of different traffic classes.

A. Preprocessing

For preprocessing, we computed the z-score normalization individually for each timeout, as time-related features are strongly dependent on timeout. The z-score is computed for every feature except for the encoded labeled target classes as $z = \frac{x-\mu}{\sigma}$. where z is z-score, x is a raw datum, μ is the mean and σ denotes the standard deviation.

B. Splitting

We take 20% of the dataset as a validation dataset D_V. The rest of the 80% dataset is split equally between each of two organizations as D_I and D_J. We assume that the D_V is openly available, without any privacy concerns, so that all the models are validated on D_V.

VI. Evaluation

As the baseline, we trained models on individual datasets of two organizations. M_I denotes the model trained on D_I and M_J denotes the model trained on D_J without a FL approach. We also trained the baseline centralized model denoted by M_C which is trained on $D_C = \{D_I \cup D_J\}$. We use TensorFlow with Keras library to train M_C, M_I, and M_J. M_F denotes the FL model trained by the federated averaging (FedAvg) process. We use the FedAvg process of "TensorFlow Federated" with Keras library for training the vanilla federated model M_F.

A. Simulation Results - Scenario A

The distribution of data instances between D_I, D_J and D_V for Scenario A are shown in Table. I. The DNN model for VPN/ non-VPN TC is shown in Table. II. The model has one input layer, six hidden layers with 256 neurons and Relu activation, and an output layer with 2 neurons with softmax activation. The optimizer used is stochastic gradient descent (SGD).

TABLE I
DATASET DETAILS FOR VPN/ NON-VPN CLASSIFICATION

	D_I	D_J	D_V	Total
VPN	12010	12064	6019	30093
Non-VPN	11872	11818	5923	29613
Total	23882	23882	11942	59706

TABLE II
MODEL FOR VPN/ NON-VPN CLASSIFICATION

Sr	Layer	Value	Activation	Sr	Layer	Value	Activation
1	Input	(23,)	-	6	Dense	256	Relu
2	Dense	256	Relu	7	Dropout	0.2	-
3	Dense	256	Relu	8	Dense	256	Relu
4	Dropout	0.2	-	9	Dense	256	Relu
5	Dense	256	Relu	10	Dense	2	Softmax

The training accuracy for Scenario A is shown in Fig. 2. The validation accuracy (VA) for Scenario A is shown in Fig. 3. The VA of the federated model M_F slightly lower than the VA of centralized model M_C. However, VA of the federated model M_F outperforms the VA of both M_I and M_J.

The performance metrics (PM's) on the D_V for the four models are shown in Table. III. The precision, recall, F1-score as well as accuracy of M_F is better than the corresponding metrics of M_I and M_J. The PM's of M_F are slightly lower

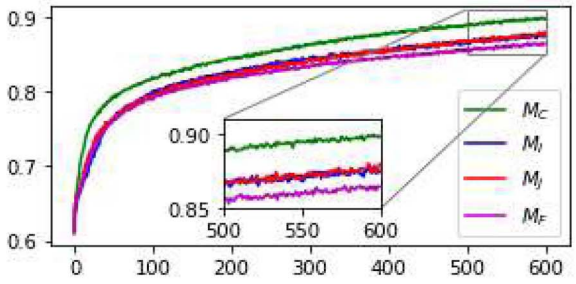

Fig. 2. Training Accuracy - Scenario A

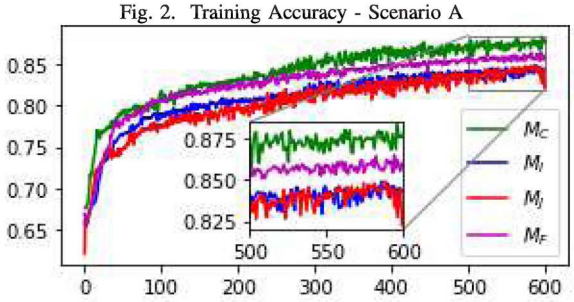

Fig. 3. Validation Accuracy- Scenario A

than the centralized model M_C as expected. However, M_F provides the benefit of training a learning model with privacy preservation.

TABLE III
PERFORMANCE METRICS ON VALIDATION DATASET D_V - SCENARIO A

	Precision	Recall	F-1	Accuracy
M_I	0.84	0.84	0.84	0.84
M_J	0.85	0.82	0.81	0.82
M_F	**0.87**	**0.86**	**0.86**	**0.86**
M_C	0.88	0.88	0.88	0.88

B. Simulation Results - Scenario B

The distribution of data instances between D_I, D_J and D_V for Scenario B are shown in Table. IV. The DNN for application-level traffic characterization is shown in Table. V. The model has one input layer, nine hidden layers with 512 neurons and Relu activation, and output layer with 14 neurons and softmax activation. The optimizer used is SGD.

TABLE IV
DATASET DETAILS FOR APPLICATION-LEVEL TRAFFIC CLASSIFICATION

	D_I	D_J	D_V	Total
BROWSING	4025	3975	2000	10000
VPN-BROWSING	3961	4039	2000	10000
CHAT	1016	988	501	2505
VPN-CHAT	1119	1152	568	2839
STREAMING	499	528	257	1284
VPN-STREAMING	462	430	223	1115
MAIL	528	563	273	1364
VPN-MAIL	928	1027	489	2444
VOIP	2535	2653	1297	6485
VPN-VOIP	2249	2212	1115	5576
P2P	1646	1554	800	4000
VPN-P2P	1394	1338	683	3415
FT	1637	1543	795	3975
VPN-FT	1883	1880	941	4704
Total	23882	23882	11942	59706

The training accuracy for Scenario B is shown in Fig. 4. The validation accuracy (VA) for Scenario B is shown in Fig. 5.

TABLE V
MODEL FOR APPLICATION-LEVEL TRAFFIC CLASSIFICATION

Sr	Layer	Value	Activation	Sr	Layer	Value	Activation
1	Input	(23,)	-	8	Dense	512	Relu
2	Dense	512	Relu	9	Dropout	0.2	-
3	Dense	512	Relu	10	Dense	512	Relu
4	Dense	512	Relu	11	Dense	512	Relu
5	Dropout	0.2	-	12	Dense	512	Relu
6	Dense	512	Relu	13	Dense	14	Softmax
7	Dense	512	Relu	-	-	-	-

The VA of the federated model M_F is slightly lower than the VA of centralized model M_C. However, VA of the federated model M_F outperforms the VA of both M_I and M_J.

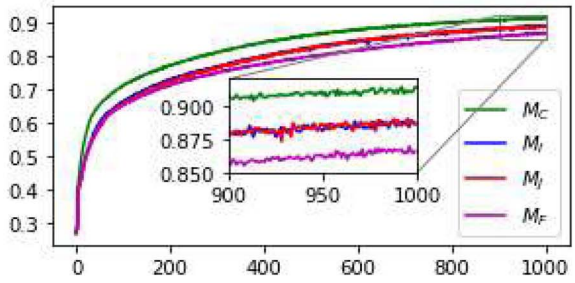

Fig. 4. Training Accuracy - Scenario B

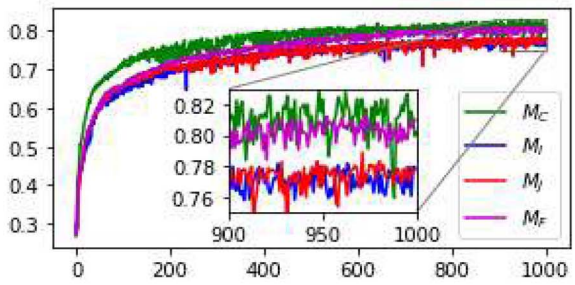

Fig. 5. Validation Accuracy - Scenario B

The performance metrics (PM's) on D_V for the four models are shown in Table. VI. The PM's of M_F is better than the corresponding metrics of M_I and M_J. The PM's of M_F are approximately equal to the PM's of the centralized model M_C. However, M_F provides the additional benefit of training a learning model with privacy.

TABLE VI
PERFORMANCE METRICS ON VALIDATION DATASET D_V - SCENARIO B

	Precision	Recall	F-1	Auccuracy
M_I	0.77	0.73	0.74	0.78
M_J	0.77	0.74	0.75	0.78
M_F	**0.80**	**0.77**	**0.78**	**0.81**
M_C	0.80	0.76	0.77	0.80

C. Discussion

The training accuracy (TA) of the federated model M_F is the mean of TA reported by local models M_I^F and M_J^F. These local federated model accuracy is computed on the individual training set D_I and D_J respectively, in FL settings by each silo. Since, the local model is replaced by a federated global model for the training of the next local model update at each global iteration, each of the the two local models M_I^F and M_J^F can not over-fit their individual data sets, therefore TA of federated model M_F is lower than the TA of centralized model M_C as well as M_I and M_J in both scenarios.

VII. CONCLUSION AND FUTURE WORK

Traffic classification is one of the important task in network operation and management systems. This paper proposed the cross-silo horizontal federated learning (HFL) for feature-based traffic classification. Simulation results show that the federated learning (FL) model shows similar results to the centralized learning approach, with the additional benefit of data-privacy. Moreover, the FL approach showed superior accuracy than the individual models trained on local datasets without a FL scheme. In our future work, we will consider a FL approach for deep packet inspection (DPI) based traffic classification, anomaly detection, and intrusion detection in network traffic.

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, and B. A. y Arcas, "Federated learning of deep networks using model averaging," *arXiv preprint arXiv:1602.05629*, 2016.

[2] U. Majeed and C. S. Hong, "FLchain: Federated Learning via MEC-enabled Blockchain Network," in *2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2019, pp. 1–4.

[3] L. U. Khan, N. H. Tran, S. R. Pandey, W. Saad, Z. Han, M. N. Nguyen, and C. S. Hong, "Federated learning for edge networks: Resource optimization and incentive mechanism," *arXiv preprint arXiv:1911.05642*, 2019.

[4] U. Majeed and C. S. Hong, "EFLChain: Ensemble Learning via Federated Learning over Blockchain network: a framework," *Proc. of the KIISE Korea Software Conference (KSC)*, pp. 845–847, 2019.

[5] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[6] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[7] "TensorFlow Federated: Machine Learning on Decentralized Data." [Online]. Available: https://www.tensorflow.org/federated

[8] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1988–2014, 2019.

[9] T. T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[10] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of Encrypted and VPN Traffic using Time-related Features," in *Proceedings of the 2nd international conference on information systems security and privacy (ICISSP)*, 2016, pp. 407–414.

[11] M. Lotfollahi, M. J. Siavoshani, R. S. H. Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, vol. 24, no. 3, pp. 1999–2012, 2020.

[12] X. Zeng, X. Chen, G. Shao, T. He, Z. Han, Y. Wen, and Q. Wang, "Flow context and host behavior based shadowsocks's traffic identification," *IEEE Access*, vol. 7, pp. 41 017–41 032, 2019.

[13] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[14] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," *arXiv preprint arXiv:1806.00582*, 2018.