Cross-Silo Model-Based Secure Federated Transfer Learning for Flow-Based Traffic Classification

Umer Majeed, Sheikh Salman Hassan, Choong Seon Hong
Department of Computer Engineering
Kyung Hee University
Yongin, South Korea
{umermajeed, salman0335, cshong}@khu.ac.kr

***Abstract*—Traffic classification is crucial for autonomous network management. Deep learning-based traffic classification methods are in demand because of their ability to accurately classify even encrypted traffic. Federated learning is a way to collaboratively train learning models with privacy-preservation. Transfer learning allows learning models to share knowledge between tasks from different but related domains. Federated Transfer Learning allows collaborative training of privacy-preserving models with knowledge sharing from source to target domains. In this paper, we did secure federated transfer learning for improvising the training-time and accuracy of the target-federated-model for traffic classification. The target-federated-model outperforms the baseline-federated-model trained from scratch. We implemented a simple cross-silo secure aggregation protocol for security.**

***Index Terms*—Cross-Silo, Federated Learning, Federated Transfer Learning, Horizontal Federated Learning, Tensorflow Federated, Transfer Learning, Secure Aggregation**

I. Introduction

Traffic classification is the process to categorize network traffic into relevant classes. With the emergence of bandwidth-intensive services [1], traffic classification has a significant role in network traffic engineering [2]. Traffic classification is a prerequisite for malware detection, intrusion prevention, price adjustment, resource management, and maintaining the quality of service (QoS) [3].

Traffic classification allows enterprises to ensure compliance with enterprise network usage policies. ***Virtual Private Network*** (VPN) technology allows secure encrypted data transmission between enterprises and individuals (employees). However, VPN encryption poses an obstacle to traditional traffic classification schemes. Deep learning enabled flow-based traffic classification schemes are admired for their ability to accurately classify normal traffic, but also VPN encrypted traffic without explicit feature search.

Few organizations may collaborate to build a traffic classification model, but the sharing of raw traffic data has privacy and security concerns. To mitigate these issues, ***Federated Learning*** (FL) [4], [5], [6], [7] provides a way to conjointly learn a common model without centralizing the raw data. Because of the changing network traffic dynamics and usage of new applications in the network [8], [9], existing traffic classification models may become obsolete. ***Transfer Learning*** allows the learning knowledge to transfer between models and then quickly adapt to a new domain without starting the learning from scratch. ***Federated Transfer Learning*** not only allows the knowledge sharing between source and target domains but allows collaborative training of learning models without transferring users' data to the cloud server.

Cross-Silo Horizontal Federated Learning allows enterprises/ organizations or silos to train collaborative models where the datasets of silos have a lot of overlapping features.

In this study, we did model-based federated transfer learning for traffic classification where the source and target models are trained in cross-silo horizontal federated learning settings using cross-silo secure aggregation protocol.

The contribution of this paper is highlighted below:

- We propose and devise a cross-silo model-based federated transfer learning scheme for traffic classification based on supervised feature-based deep learning.
- We train a source-federated-model for application-level traffic classification (e.g. P2P, VoIP, VPN-P2P, etc) based on flow-based time-related features in a cross-silo horizontal federated learning configuration.
- We transfer the weights from the source-federated-model to the target-federated-model. Where the target-federated-model is further train afterward for VPN\non-VPN identification as binary classification. The target model training is done based on flow-based time-related features in a cross-silo horizontal federated learning configuration.
- The target-federated-model outperforms the baseline model for both validation accuracy and training-time efficiency.
- We applied the cross-silo secure aggregation technique for security and privacy-preserving federated learning.

The rest of the paper is organized as follows: Section II gives a brief overview about federated learning and transfer learning. The system model is presented in Section III followed by the problem formulation in Section IV. Section V describes the employed secure aggregation protocol for cross-silo federated learning. Section VI describes the dataset used. Section VII gives simulation results. Section VIII concludes our work.

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2020R1A4A1018607). *Dr. CS Hong is the corresponding author.

II. PRELIMINARIES

A. Federated Learning

Federated learning [10], [11], [12], [13] is a privacy-preserving distributed machine learning process to train a shared model from distributed datasets with more computation at the edge. For each global iteration, the federated learning server aggregates local model updates from learners to update the global model. We will briefly formulate the federated learning process [14] below:

Consider a neural network that classifies the input data to C classes. The input data has compact Euclidean feature space $\mathcal{X}$ mapped on the label space $\mathcal{Y} = [C]$, where $[C] = \{1, ..., C\}$.

The cross-entropy loss for a datapoint $\{x, y\}$ with one-hot encoded label is given as [15]

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

Where, the probability of $x \in \mathcal{X}$ being mapped to class q is denoted as $p_q(\boldsymbol{x}, \boldsymbol{w})$. while $\boldsymbol{w}$ denotes the weight matrix for the artificial neural network (ANN). The local loss F_k can be written as

$$F_k(\boldsymbol{w}) = \frac{1}{n_k} \sum_{r \in D_k} f_r(\boldsymbol{w}). \tag{2}$$

where D_k is dataset of k^{th} client and $n_k = |D_k|$ are number of samples in D_k. Then the local gradient is determined as

$$g_k = \nabla F_k(\boldsymbol{w}_t) \quad \text{where} \quad \delta_k = |D_k| \, g_k. \tag{3}$$

At global iteration $t+1$, the local model weights are updated as

$$\boldsymbol{w}_{t+1}^k \leftarrow \boldsymbol{w}_t - \eta g_k, \quad \forall k. \tag{4}$$

The overall global loss in federated learning settings is calculated as

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

Where ψ is the set of federated learning clients. Then global gradient is determined as

$$\nabla F(\boldsymbol{w}_t) = \sum_{k \in \psi} \frac{n_k}{n} g_k = \frac{\sum_{k \in \psi} \delta_k}{\sum_{k \in \psi} |D_k|}. \tag{6}$$

At global iteration $t + 1$, the global model weights are updated using Federated averaging (FedAvg) [10] as

$$\boldsymbol{w}_{t+1} \leftarrow \sum_{k \in \psi} \frac{n_k}{n} \boldsymbol{w}_{t+1}^k \tag{7}$$

or

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \eta \nabla F(\boldsymbol{w}_t). \tag{8}$$

The overall purpose is to minimize the global loss during federated learning process as:

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}). \tag{9}$$

B. Transfer Learning

Transfer learning facilitates the reuse of the experience from the source domain to another related target domain to quickly adapt to the target domain or task. There are several transfer learning schemes such as instance-based, feature-based, model-based, related-based transfer learning [16]. Here, we will briefly describe model-based transfer learning only.

In Model-based Transfer Learning, parameters or hyper-parameters of learning models from the source domain, are assigned to the parameters or hyper-parameters of learning models in the target domain. So, the pre-trained models can be employed in whole or part as initial weights of target models [16]. Afterward, the target model is further trained as per the target domain.

III. System Model

A federation O has two organization $I, J \in O = \{I, J\}$. Another federation P has two organization $K, L \in P = \{K, L\}$.

Organization I has dataset D_I having sample space $Z_I = (\mathcal{X}_I, \mathcal{Y}_I)$ and organization J has dataset D_J having sample space $Z_J = (\mathcal{X}_J, \mathcal{Y}_J)$.

Organization K has dataset D_K having sample space $Z_K = (\mathcal{X}_K, \mathcal{Y}_K)$ and organization L has dataset D_L having sample space $Z_L = (\mathcal{X}_L, \mathcal{Y}_L)$.

Where $\mathcal{X}_\gamma$ is feature space, $\mathcal{Y}_\gamma$ is label space, and $\gamma \in \{I, J, K, L\}$.

Data-sets D_I and D_J have different sample space. However, the feature space and label space pair of these two datasets i.e., $(\mathcal{X}_I, \mathcal{Y}_I)$ and $(\mathcal{X}_J, \mathcal{Y}_j)$ is same. Formally [17]

$$\mathcal{X}_I = \mathcal{X}_J, \;\; \mathcal{Y}_I = \mathcal{Y}_J, \;\; Z_I \neq Z_J, \;\; D_I \neq D_J, \;\; I \neq J \tag{10}$$

Similarly, datasets D_K and D_L have different sample space. However, the feature space and label space pair of these two datasets i.e., $(\mathcal{X}_K, \mathcal{Y}_K)$ and $(\mathcal{X}_L, \mathcal{Y}_L)$ is same. Formally [17],

$$\mathcal{X}_K = \mathcal{X}_L, \;\; \mathcal{Y}_K = \mathcal{Y}_L, \;\; Z_K \neq Z_L, \;\; D_K \neq D_L \tag{11}$$

$$\text{where} \;\; K \neq L \tag{12}$$

Fig. 1 illustrates the system model.

Organization I and J collaboratively trains a source-federated-model M_S^F, while M_I^F and M_J^F symbolize the local models trained in federated learning setting on D_I and D_J respectively. These entities are indicated in the source module in the system model.

Similarly, organization K and L collaboratively trains a target-federated-model M_T^F, while M_K^F and M_L^F symbolize the local models trained in federated learning setting on D_K and D_L respectively. These entities are indicated in the target module in the system model.

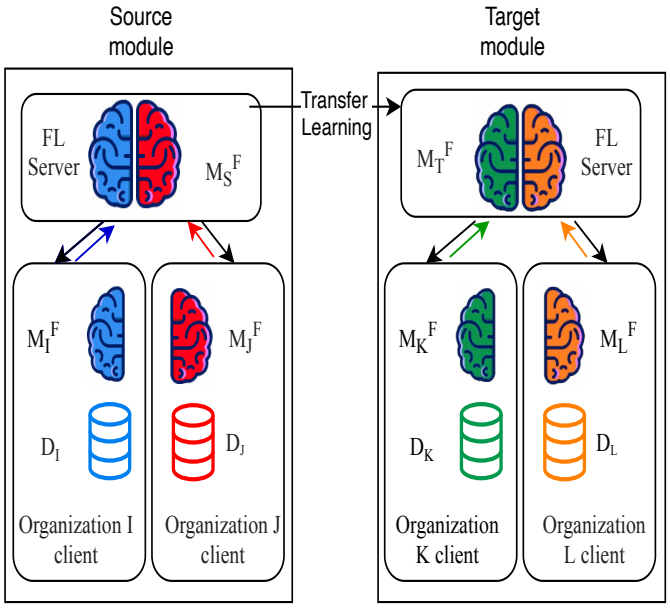

Fig. 1. Cross-Silo Model-based Federated Transfer Learning

Algorithm 1: Procedure for model-based secure Federated Transfer Learning - Source Module

Initiate M_I^F, M_J^F, M_S^F via $\mathcal{T}_S = \{\mathcal{Y}_S, \dot{f}_S^F(x;\theta^S)\}$
for $u \in \{1, 2, 3, ..., \hat{I}_S^G\}$ } **do**
  update M_I^F on D_I using Eq. 4
  update M_J^F on D_J using Eq. 4
  apply secure aggregation protocol - Section V-A
  aggregate M_I^F and M_J^F using Eq. 7 and secure aggregation protocol (Section V-A) to get M_S^F
Send M_S^F to Target Module (Algorithm. 2)

Algorithm 2: Procedure for model-based secure Federated Transfer Learning - Target Module

Initiate M_K^F, M_L^F, M_T^F via $\mathcal{T}_T = \{\mathcal{Y}_T, \dot{f}_T^F(x;\theta^T)\}$
for $\{g \in G\}$ **do**
  assign $\theta_g^T = \theta_g^S$ for M_I^F, M_J^F, M_S^F
for $v \in \{1, 2, 3, ..., \hat{I}_T^G\}$ **do**
  update M_K^F on D_K using Eq. 4
  update M_L^F on D_L using Eq. 4
  apply secure aggregation protocol - Section V-A
  aggregate M_K^F and M_L^F using Eq. 7 and secure aggregation protocol (Section V-A) to get M_T^F
deploy target federated transfer model M_T^F

IV. Problem Formulation

In this section, we formulate our federated transfer learning problem corresponding to the system model in Section. III. We have the same feature space of source and target domain dataset. However, the label space of target domain and source domain datasets is different. Formally,

$$\mathcal{X}_S = \mathcal{X}_T, \quad \mathcal{Y}_S \neq \mathcal{Y}_T, \quad D_S \neq D_T, \quad S \neq T \tag{13}$$

$$\text{where} \quad S \in \{O = \{I, J\}\} \text{ and } T \in \{P = \{K, L\}\} \tag{14}$$

For source domain $\mathcal{D}_S$, We have source task $\mathcal{T}_S = \{\mathcal{Y}_S, \dot{f}_S^F(x;\theta^S)\}$. Where $\dot{f}_S^F(x;\theta^S)$ is the source predictive function from feature space $\mathcal{X}_S$ to label space $\mathcal{Y}_S$ for source-federated-model M_S^F and θ^S denotes the weight matrix for M_S^F.

Similarly, for target domain $\mathcal{D}_T$, We have target task $\mathcal{T}_T = \{\mathcal{Y}_T, \dot{f}_T^F(x;\theta^T)\}$. Where $\dot{f}_T^F(x;\theta^T)$ is the target predictive function from feature space $\mathcal{X}_T$ to label space $\mathcal{Y}_T$ for target-federated-model M_T^F and θ^T denotes the weight matrix for M_T^F.

Let G be set of layers set as freezed in target-federated-model M_T^F. θ_g^S denotes the weights of g^{th} layer of source model and θ_g^T denotes the weights of g^{th} layer of target model. After the trained source model is available from the source domain, we can assign the weights to the target model M_T^F as

$$\theta_g^T = \theta_g^S \quad , \quad \forall g \in G \tag{15}$$

We formulate our federated transfer learning problem as: given source domain $\mathcal{D}_S$ with source task $\mathcal{T}_S$ and target domain $\mathcal{D}_T$ with target task $\mathcal{T}_T$, increase the learning accuracy of $\dot{f}_T^F(x;\theta^T)$ in $\mathcal{D}_T$ and decrease corresponding training time t_T using the knowledge from $\mathcal{D}_S$ and $\mathcal{T}_T$, where,

$$\mathcal{D}_T \neq \mathcal{D}_S, \quad \mathcal{T}_S \neq \mathcal{T}_T, \quad S \neq T, \tag{16}$$

and source-federated-model M_S^F and target-federated-model M_T^F are trained in cross-silo horizontal federated learning settings. Algorithm. 1 and Algorithm. 2 shows the procedure for model-based federated transfer learning for source and target module respectively.

V. Security for Cross-Silo Federated Learning

For secure federated learning, we have employed secure aggregation protocol which is described below:

A. Secure Aggregation

In a cross-silo federated learning setting of the source module with federation O, we consider a secure and reliable communication channel between organization I and J as well as between organizations and federated learning server. Considering Eq. 6 and Eq. 8, The organization I and J have to just share $\langle |D_k|, \delta_k \rangle$ for aggregation of local models to compute the global model update. Here we formulate our simple secure aggregation protocol [18].

Consider that organization $u \in O$ holds private vector e_u with dimension d. Where $e_u, \sum_{u \in O} e_u \in \mathbb{R}^d$. Where, $\mathbb{R}$ is set of real numbers.

Each organization $u \in O$ agree on matched pair of masked perturbation for every other organization $v \in O$, that is $b_{u,v}$ and $b_{v,u}$ are uniformly sampled from $(-R, R)^d$, where R is some threshold. The organization exchange $b_{u,v}$ and $b_{v,u}$ over secure channel. Afterwards, they compute $a_{u,v} = b_{u,v} - b_{v,u}$. where $a_{u,v} = -a_{v,u}$ and $a_{v,u} = 0$ when $v = u$.

Every organization sends masked vector $h_u = e_u + \sum_{v \in O} a_{u,v}$ to the server. The server aggregates masked vectors to compute unmasked aggregated vector as

$$\bar{e} = \sum_{u \in O} h_u = \sum_{u \in O} e_u + \sum_{u \in O} \sum_{v \in O} a_{u,v} = \sum_{u \in O} e_u + \sum_{u \in O} \sum_{v \in O} b_{u,v} - \sum_{u \in O} \sum_{v \in O} b_{v,u} = \sum_{u \in O} e_u \quad (17)$$

The secure aggregation scheme above is used for computing the global model at each global iteration. For the target module, the same scheme is applied.

VI. Dataset

A. Dataset Details

The dataset we engage for our federated transfer learning-enabled traffic classification is a publicly available UNB ISCX VPN-NonVPN network traffic dataset [19]. This dataset is designed by the Research Center of the University of New Brunswick in Canada. The dataset has time-related features for flow-based traffic data with labeled classes. The dataset has four timeout intervals (120s, 60s, 30s, 15s). Table. I briefly describes these features.

1) Scenario A: Scenario A dataset distinguishes the traffic secured using VPN and non-VPN network traffic. This scenario has two classes i.e. VPN and Non-VPN.

2) Scenario B: The primary objective of scenario B is to differentiate the traffic type besides VPN-non VPN recognition. This scenario has fourteen traffic classes namely BROWSING, CHAT, STREAMING, MAIL, VOIP, P2P, FT, VPN-VOIP, VPN-CHAT, VPN-STREAMING, VPN-FT, VPN-BROWSING, VPN-P2P, and VPN-MAIL.

TABLE I
LIST OF TIME-RELATED FLOW-BASED FEATURES [19]

Feature	Details
duration	The flow's duration.
fiat	Forward Inter Arrival Time indicates the time duration amid two packets transmitted in the forward direction (min, max, mean, std).
flowiat	Flow Inter Arrival Time indicates the time duration between two packets transmitted in either direction (min, max, mean, std).
biat	Backward Inter Arrival Time indicates the time duration amid two packets transmitted in the backward direction (min, max, mean, std).
idle	The time span a flow was idle prior to going into active state(min, max, mean, std).
active	The time duration a flow was active prior to going into idle state(min, max, mean, std).
fp-psec	Number of Flow Flow packets transmitted per second.
fb-psec	Number of Flow Bytes transmitted per second.

B. Data Preprocessing

For data preprocessing, we first separated the data based on timeout. Since, time-related features have a high positive correlation with timeout interval, we later performed the z-score normalization for each timeout separately. The z-score is performed on all the time-related flow-based features except for the encoded label.

$$z = \frac{x - \mu}{\sigma} \quad (18)$$

where z denotes the z-score, x denotes the raw datum, μ denotes the mean and σ is the standard deviation.

C. Splitting

The 20 percent of the dataset is assigned as a D_V (validation dataset). The validation dataset is publicly accessible without any secrecy issues. The rest of 80 percent of the dataset is evenly divided between the two organizations in each module.

Specifically, for the source task, 80% of scenario B dataset is equally divided between organization I and J, and 20% is used as D_V. Similarly, for the target task, 80% of scenario A dataset is equally divided between organization K and L, and 20% is used as D_V. Corresponding D_V is used for validation of all related models.

VII. Simulation Results

We use Tensorflow Federated (TFF) [20] for federated learning while the Tensorflow Keras library was used for learning transfer from source to target module. We used

TABLE II
SOURCE-FEDERATED-MODEL M_S^F AND TARGET FEDERATED MODEL M_T^F - LAYERED ARCHITECTURE

			source-federated-model M_S^F	**target-federated-model** M_T^F	
Sr	**Layer**	**Activation**	**Value**	**Value**	**Trainable/ non-Trainable**
1	Input	-	(23,)	(23,)	-
2	Dense	Relu	512	512	non-Trainable
3	Dense	Relu	512	512	non-Trainable
4	Dense	Relu	512	512	non-Trainable
5	Dropout	-	0.2	0.2	-
6	Dense	Relu	512	512	non-Trainable
7	Dense	Relu	512	512	non-Trainable
8	Dense	Relu	512	512	non-Trainable
9	Dropout	-	0.2	0.2	-
10	Dense	Relu	512	512	Trainable
11	Dense	Relu	512	512	Trainable
12	Dense	Relu	512	512	Trainable
13	Dense	Softmax	14	2	Trainable

callbacks in Tensorflow Federated to implement the secure aggregation protocol. Tensorflow Federated was released by Google in March 2019 as an open-source single-machine federated learning framework.

The layered based-architecture for source model M_S^F and target model M_T^F is shown in Table. II. The trainable/ non-trainable layers for target model M_T^F are also specified. Stochastic gradient descent (SGD) is exploited as an optimizer during the federated learning process.

Table. III shows the dissemination of data samples between D_I, D_J and D_V^S for source module.

TABLE III
DATASET DISTRIBUTION FOR APPLICATION-LEVEL TRAFFIC CLASSIFICATION - SOURCE-FEDERATED-MODEL

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

First, the source-federated-model M_S^F was trained for 1000 epochs in the source module on D_I and D_J in cross-silo horizontal federated learning settings. In 1000 epochs, the source-federated-model with maximum validation accuracy was picked out for further processing. The training and validation accuracy for source-federated-model M_S^F is illustrated in Fig. 2. The related performance metrics are given in Table. IV.

TABLE IV
PERFORMANCE METRICS OF M_S^F ON VALIDATION DATASET D_V

	Precision	Recall	F-1	Accuracy
M_S^F	0.85	0.79	0.81	0.83

Subsequently, the weights of layers from source-federated-model M_S^F were assigned to corresponding non-trainable layers of target-federated-model M_T^F. Afterward, the trainable layers of target-federated-model M_T^F were trained on D_K and D_L in cross-silo horizontal federated learning settings.

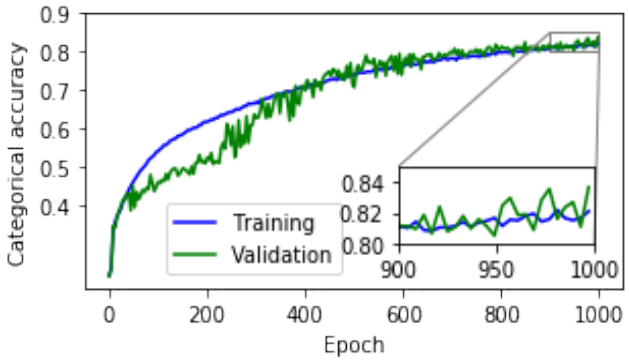

Fig. 2. Training and validation accuracy for source-federated-model M_S^F

Table. V shows the dissemination of data samples between D_K, D_L and D_V^T for target module.

TABLE V
DATASET DISTRIBUTION FOR VPN/ NON-VPN IDENTIFICATION - TARGET-FEDERATED-MODEL

	D_K	D_L	D_V^T	Total
Non-VPN	11872	11818	5923	29613
VPN	12010	12064	6019	30093
Total	23882	23882	11942	59706

As a baseline, the baseline-federated-model M_B^F was trained on D_K and D_L in cross-silo horizontal federated learning settings from scratch. The architecture of baseline model M_B^F is the same as of M_T^F except that all layers of M_B^F are trainable. The target-federated-model M_T^F and baseline model M_B^F were trained for 600 epochs. The models with maximum validation accuracy were picked out for further processing using call-backs.

Fig. 3 shows the training and validation accuracy for target-federated-model M_T^F and baseline-federated-model M_B^F on Validation dataset D_V. The target-federated-model M_T^F gained maximum validation accuracy of 0.8969 at epoch 595, while baseline-federated-model M_B^F gained maximum validation ac-

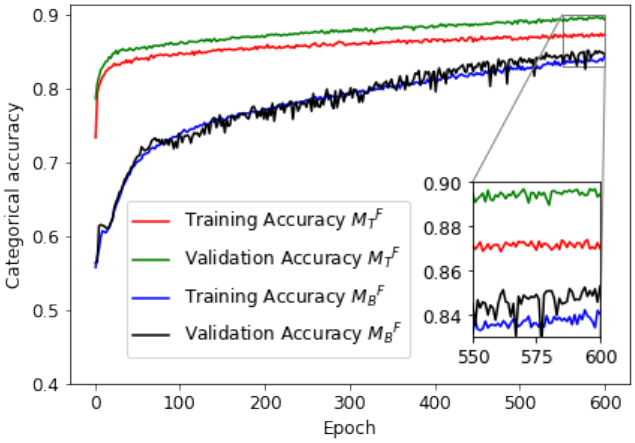

Fig. 3. Training and validation accuracy for target-federated-model M_T^F and baseline-federated-model M_B^F

curacy of 0.8530 at epoch 599. We measured the time taken by M_B^F and M_T^F for 600 epochs alongside the secure aggregation protocol. Fig. 4 shows that target-federated-model M_T^F takes less time compared to baseline-federated-model M_B^F for training as there are less number of training parameters in target-federated-model M_T^F. Consequently, time for secure aggregation of target-federated-model M_T^F per global iteration is also than baseline-federated-model M_B^F. Table. VI shows the corresponding performance metrics.

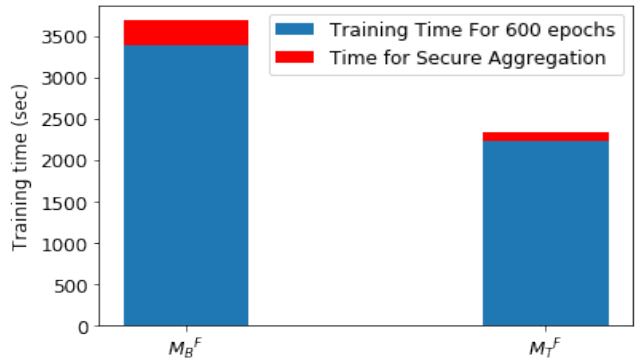

Fig. 4. Training time for target-federated-model M_T^F and baseline-federated-model M_B^F

TABLE VI
PERFORMANCE METRICS OF TARGET-FEDERATED-MODEL M_T^F AND BASELINE-FEDERATED-MODEL M_B^F ON VALIDATION DATASET D_V

	Precision	Recall	F-1	Accuracy
$\boldsymbol{M_T^F}$	0.91	0.90	0.89	0.90
$\boldsymbol{M_B^F}$	0.86	0.85	0.84	0.85

VIII. CONCLUSION

Network traffic classification is an indispensable component for intelligent autonomous network management. In this work, we designed a cross-silo model-based federated transfer learning scheme for traffic classification. The models are based on supervised deep learning on feature-based datasets. The source-federated-model was trained for application-level traffic classification on time-related flow-based features in cross-silo horizontal federated learning settings. We assigned the weights of the source-federated-model to the weights of the target-federated-model. The target-federated-model is then further trained in cross-silo horizontal federated learning settings on the time-related flow-based features for VPN/non-VPN recognition. The target-federated-model outperforms the baseline-federated-model both in terms of accuracy and training-time efficiency. Moreover, we applied the secure aggregation protocol for secure and privacy-preserving federated learning.

REFERENCES

[1] L. U. Khan, I. Yaqoob, M. Imran, Z. Han, and C. S. Hong, "6G Wireless Systems: A Vision, Architectural Elements, and Future Directions," *IEEE Access*, vol. 8, pp. 147 029–147 044, 2020.

[2] M. R. P. Santos, R. M. C. Andrade, D. G. Gomes, and A. C. Callado, "An efficient approach for device identification and traffic classification in iot ecosystems," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, 2018, pp. 00 304–00 309.

[3] S. Rezaei and X. Liu, "Deep learning for encrypted traffic classification: An overview," *IEEE Communications Magazine*, vol. 57, no. 5, pp. 76–81, 2019.

[4] L. U. Khan, S. R. Pandey, N. H. Tran, W. Saad, Z. Han, M. N. H. Nguyen, and C. S. Hong, "Federated Learning for Edge Networks: Resource Optimization and Incentive Mechanism," 2019.

[5] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," 2020.

[6] 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.

[7] L. U. Khan, U. Majeed, and C. S. Hong, "Blockchain-assisted Ensemble Federated Learning for Automatic Modulation Classification in Wireless Networks," *Proc. of the Korean Network Operations and Management (KNOM)*, pp. 111–113, 2020.

[8] ——, "A Hierarchical Caching Framework for 5G Cellular Networks," *Proc. of the KIISE Korea Computer Congress (KCC)*, pp. 1289–1291, 2019.

[9] I. Yaqoob, U. Majeed, and C. S. Hong, "Towards Real-Time Analytics for Mobile Big Data using the Edge Computing," *Proc. of the KIISE Korea Computer Congress (KCC)*, pp. 338–340, 2018.

[10] 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.

[11] 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.

[12] L. U. Khan, U. Majeed, and C. S. Hong, "Federated Learning for Cellular Networks: Joint User Association and Resource Allocation," in *2020 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 2020, pp. 405–408.

[13] U. Majeed and C. S. Hong, "Blockchain-assisted Ensemble Federated Learning for Automatic Modulation Classification in Wireless Networks," *Proc. of the KIISE Korea Computer Congress (KCC)*, pp. 756–758, 2019.

[14] U. Majeed, L. U. Khan, and C. S. Hong, "Cross-Silo Horizontal Federated Learning for Flow-based Time-related-Features Oriented Traffic Classification," in *The 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, Daegu, Korea (South), Sep. 2020.

[15] 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.

[16] Q. Yang, Y. Zhang, W. Dai, and S. J. Pan, *Transfer Learning*. Cambridge University Press, 2020.

[17] 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.

[18] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical Secure Aggregation for Federated Learning on User-Held Data," 2016.

[19] 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.

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