IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 26 June 2022, accepted 9 July 2022, date of publication 20 July 2022, date of current version 5 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3192454

RESEARCH ARTICLE

Privacy-Preserving Federated Transfer Learning for Driver Drowsiness Detection

LINLIN ZHANG[1,2], HIDEO SAITO[1], (Senior Member, IEEE), LIANG YANG[3], AND JIAJIE WU[2]

[1]Graduate School of Science and Technology, Keio University, Tokyo 223-8522, Japan
[2]China Automotive Technology and Research Center Co., Ltd., Tianjin 300300, China
[3]China Auto Information Technology Co., Ltd., Tianjin 300300, China

Corresponding author: Linlin Zhang (zhanglinlin@catarc.ac.cn)

ABSTRACT Drowsiness affects the drivers' sensory, cognitive, and psychomotor abilities, which are necessary for safe driving. Drowsiness detection is a critical technique to avoid traffic accidents. Federated learning (FL) can solve the problem of insufficient driver facial data by utilizing different industrial entities' data. However, in the FL system, the privacy information of the drivers might be leaked. In addition, reducing the communication costs and maintaining the model performance are also challenges in industrial scenarios. In this work, we propose a federated transfer learning method with the privacy-preserving protocol for driver drowsiness detection, named PFTL-DDD. We use fine-tuning transfer learning on the initial model of the drowsiness detection FL system. Furthermore, a CKKS-based privacy-preserving protocol is applied to preserve the drivers' privacy data by encrypting the exchanged parameters. The experimental results show that the PFTL-DDD method is superior in terms of accuracy and efficiency compared to the conventional federated learning on the NTHU-DDD and YAWDD datasets. The theoretical analysis demonstrates that the proposed transfer learning method can reduce the communication cost of the system, and the CKKS-based security protocol can protect personal privacy.

INDEX TERMS Driver drowsiness detection, transfer learning, federated learning, privacy-preserving.

I. INTRODUCTION

Driver drowsiness is a major cause of road crashes and serious injuries inflicted in traffic accidents [1]. Timely prediction of driver drowsiness and provision of driving assistance can reduce the economic lossses and casualties effectively [2]. European Union (EU) 2019/2144 regulation requires certain types of vehicles to be equipped with driver drowsiness and attention warning (DDAW) systems. The external non-invasive observation of a driver's status is a well-known technique for driver drowsiness detection. Specific facial expressions, such as frequency of nodding [3], yawning [4], and elongated eye closure time [5], indicate drivers' drowsiness.

The deep learning techniques excel in object detection, segmentation, classification, and behavior prediction tasks. With the popularity of Internet of Vehicles (IoV) functions, driver drowsiness detection has become a research hotspot in artificial intelligence (AI) and big data applications [6]. Drowsiness detection is a typical case of deep learning for behavior prediction related to automated driving.

Ghoddoosian *et al.* [5] proposed an early drowsiness detection method by using a hierarchical multiscale long short-term memory (HM-LSTM) network. This method predicts drowsiness by considering eye blinking as a time series. Ramos *et al.* [4] adopted an API-based histogram of oriented gradients (HOG) and linear support vector machine (SVM) to predict driver drowsiness by utilizing Euclidean distance between closed eyes and yawns. Weng *et al.* [7] proposed a new hierarchical temporal deep belief network by using two hidden Markov models to capture the interaction between eyes, mouth, and head movements for detecting drowsiness. This work also provides a dataset for driver drowsiness detection. Savas *et al.* [8] introduced a multi-task CNN for simultaneously classifying eyes and mouth features.

The centralized deep learning for drowsiness detection is premised on the sharing and collection of the personal data of drivers. The facial data samples of drivers are scattered on

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Yan.

different edge nodes, including vehicles, OEMs, and IoV service enterprises. Usually, these entities are reluctant to share their datasets because of business competition and transmission costs. Moreover, many countries [9] have strict regulations on the use of personal data due to privacy breaches caused by data sharing. Zafar *et al.* [10] implemented federated learning (FL) for driver drowsiness detection to preserve the drivers' privacy. The FL system achieves cooperative training of edge nodes without transmitting the raw data. Please note that transferring a complete model between a cloud server and edge nodes in a FL system consumes huge communication resources. Besides, the previous works [11], [12] show that the model exchanged in a FL system runs the risk of revealing original local information after extrapolation. In paper [13], the authors analyze several privacy-preserving schemes commonly used in federated learning, including differential privacy and homomorphic encryption. The encryption scheme of differential privacy consumes extremely high communication resources and costs extra model training time. The encryption scheme of Paillier is an extremely time-consuming and communication-consuming training strategy. Inspired by the studies of [14]–[16], we introduce the probabilistic encryption method CKKS into the driver drowsiness detection federated learning framework to protect the privacy of the edge nodes. In this work, we analyze the security of CKKS-based encryption schemes and prove that the CKKS is IND-CPA secure, and if there is no collusion between edge nodes and the cloud server/ external attacker, the privacy of the driver's facial data of the edge nodes can be protected.

In this study, a privacy-preserving federated transfer learning method is applied to driver drowsiness detection. Predicting driver drowsiness as accurately as possible is crucial to improving traffic safety. In addition, using driver facial data for federated learning remains the risk of individual information being eavesdropping on. To solve the issues, transfer learning is introduced to the initial model of federated learning to promote the model's accuracy and reduce the communication cost. Besides, the CKKS-based scheme is adopted to protect the drivers' individual information of the edge nodes. Compare to the FL-based methods, the main contributions of this work are presented below:

1) We adopt the pre-learning mechanism of transfer learning to the initial model in the federated learning system for driver drowsiness detection. The proposed method achieves better performance than the conventional FL method by transferring the knowledge of similar tasks. Furthermore, the PFTL-DDD method only exchanges the parameters of the classification layer between the edge nodes and the cloud server, which reduces the communication overhead compared with the existing FL-based method.
2) Existing encryption schemes for the FL-based method include differential privacy and homomorphic encryption, etc. Considering the trade-off between the effect and efficiency of encryption, the proposed method selects a CKKS scheme with excellent computational speed. The CKKS-based protocol meets CPA-secure [11] providing a security framework for the proposed method.
3) We evaluate the performance of the proposed privacy-preserving federated transfer learning method on the NTHU-DDD dataset and the YAWDD dataset. The experimental results show that the PFTL-DDD method has better accuracy and saves more communication resources than the FL-based method. Besides, the security analysis of the proposed method indicates that the CKKS-based protocol can resist the risk of privacy leakage even if there are eavesdroppers in the system.

II. PROBLEM STATEMENT AND THEORETICAL BACKGROUND

A. PROBLEM STATEMENT

Federated learning (FL) [17]–[19] is a method for collaborative training among entities that cannot share data. The participants train models on their respective datasets and draw on the knowledge of other participants for model updates. In this work, the federated learning system has three entities, namely trust party, cloud server, and edge nodes, as shown in Figure 1.

1) Trust party: The trust party provides a secure communication protocol by generating public and secret keys for edge nodes as the system initializes.
2) Cloud server: The cloud server provides an initial model for the edge nodes. After receiving local parameters from the edge nodes, the cloud server performs parameter aggregation and updates the global parameters $\mathbf{w}_{\text{global}}(t)$ to the edge nodes.
3) Edge nodes: The edge nodes train on the local datasets with the global parameters downloaded from the cloud server and send the new local parameters $\mathbf{g}_{e,i}(t+1)$ to the cloud server for model aggregation.

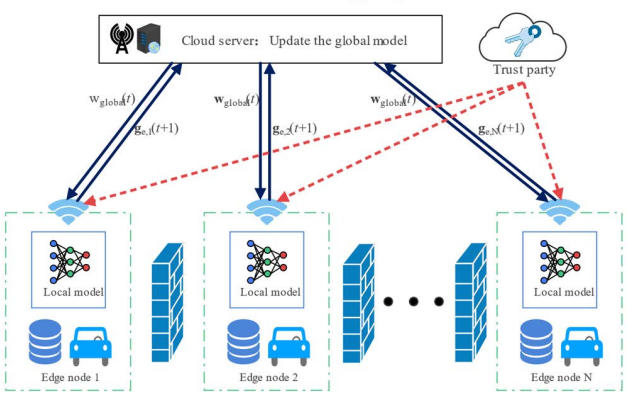

FIGURE 1. The proposed model of privacy-preserving federated learning.

Thread Model: In the proposed method, the trust party and the edge nodes are assumed to be honest and curious that will not divulge the keys and information to the cloud server or external eavesdroppers. However, the cloud server

and these participants might attempt to infer excess information from the exchange parameters. Several studies [11], [12] show that the parameters exchanged in FL system may leak the privacy information of the edge nodes. Based on the above assumptions, the proposed security framework aims to protect the personal data of the edge nodes in the driver drowsiness detection task.

B. TRANSFER LEARNING

Transfer learning is a method that uses the knowledge of a specific domain to perform tasks in a novel domain [20], [21]. Recent works [22]–[24] show that the transfer learning method is adequate for tasks with insufficient data. It improves the learning efficiency [25] by reusing and fixing the pre-trained model. The fine-tuning technique [26], [27] has been widely used in transfer learning. In this technique, the samples related to the novel task are selected for pre-training and only a few last layers are fine-tuned while freezing multiple initial layers of the model.

In CNNs, the convolutional layers (layers close to the input images) extract basic features, such as texture and shape, whereas the fully connected layers classify high-level and abstract features for specific tasks. The fine-tuning transfer learning method reuses (freezes) the pre-trained convolutional layers and only trains the fully connected layers for a novel task, as shown in Figure 2. The CNN model is trained on a source dataset beforehand. Then, the convolutional layers of this pre-trained CNN model are copied (reused) to the novel model. The parameters of the fully connected layers of the novel model are randomly initialized. The novel model is trained on the target dataset with a smaller learning rate.

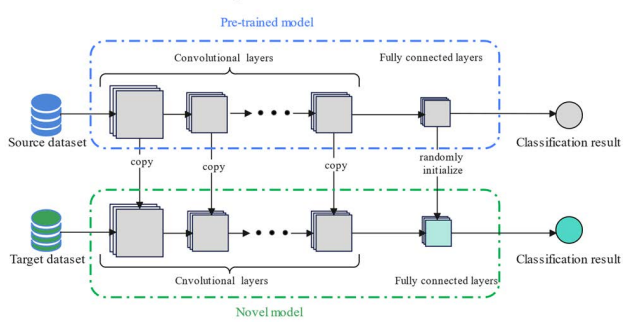

FIGURE 2. The conceptual figure of fine-tuning transfer learning.

C. MODEL FOR DROWSINESS DETECTION

In this work, we use VGG-16 as the baseline network to detect the behaviors related to drowsiness. This network comprises 13 convolutional layers, with kernels of size 3*3. Three fully connected layers are used to classify the features extracted by the convolutional layers, i.e., features belonging to the eyes, head, and mouth. The loss function is mathematically expressed as follows:

$$\mathrm{L} = \frac{1}{n_{batch}} \sum_{i} -\left[y_i \times \log\left(p_i\right) + \left(1 - y_i\right) \times \log\left(1 - p_i\right)\right] \tag{1}$$

where, n_{batch} denotes the batch size, and i represents the four labels of head, mouth, eyes, and drowsiness.

The model for drowsiness detection consists of a feature extraction layer and a classification layer. After training the VGG-16 on the drowsiness-related dataset, the convolution layers are reused for extracting the features, that is, the feature extraction layer. The fully connected layers of the VGG-16 are utilized for classification, that is, the classification layer. The structure of the model for drowsiness detection are presented in Figure 3.

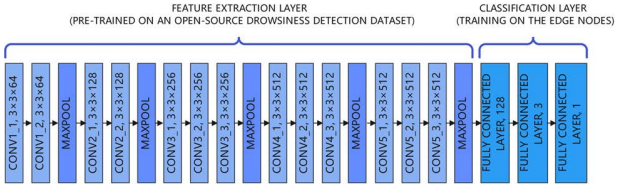

FIGURE 3. The structure of the model for drowsiness detection.

D. FULLY HOMOMORPHIC ENCRYPTION

Various research works [28]–[30] revealed that differential privacy and homomorphic encryption are usually used in the state-of-the-art privacy-preserving federal learning method. The differential privacy security protocol may result in significant degradation of model performance. In the driver drowsiness detection scenario, the accuracy reduction may lead to serious casualties. Consequently, the differential privacy protocol has not been implemented in the proposed method.

Fully homomorphic encryption is widely used in privacy protection that entrusts a third party to participate in information processing while protecting the local plaintext, i.e., computing the ciphertexts before decrypting is equivalent to executing the same operation on the plaintext after decrypting [31]. CKKS is a fully homomorphic encryption method that adds noise at the end of the truncated ciphertext. Comparing several homomorphic encryption algorithms, CKKS overcomes Paillier and RSA [32], [33] in encryption speed. To pursue superior performance and training speed, we select the CKKS to protect the data of the edge nodes in the proposed method. In this work, the local plaintexts of the edge nodes are encrypted via the CKKS-based method. The cloud server calculates the ciphertexts and sends back the results. During the whole process, the cloud server performs mathematical operations without obtaining the true value of local data.

III. THE PROPOSED FRAMEWORK

A. WORKFLOW

The workflow of the privacy-preserving federated transfer learning for driver drowsiness detection (PFTL-DDD) method comprises three major steps, namely system initialization, local training by edge nodes, and parameters aggregation by the cloud server, as shown in Figure 4.

1) SYSTEM INITIALIZATION

At initialization, the cloud server pre-trains the initial model with a drowsiness database and builds a communication channel between the cloud server and each edge node. Then, the cloud server sends the pre-trained initial model to the edge nodes. At the same time, the trust party sends the public and secret keys generated by the CKKS-based encryption protocol to all the edge nodes. The edge nodes download the pre-trained initial model and upload the encrypted local parameters $\mathrm{En}(\mathbf{g}_{\mathrm{e,i}}(1))$ after training the model on their local data, where $\mathbf{g}_{\mathrm{e,i}}$ represents the classification layer parameters of the local model in the proposed method.

2) PARAMETERS AGGREGATION BY CLOUD SERVER

When obtaining $\mathrm{En}(\mathbf{g}_{\mathrm{e,i}}(t))$ from all edge nodes, the cloud server aggregates $\mathrm{En}(\mathbf{g}_{\mathrm{e,i}}(t))$ to update $\mathrm{En}(\mathbf{w}_{\mathrm{global}}(t))$ based on the following equation, where $\mathbf{w}_{\mathrm{global}}$ represents the classification layer parameters of the global model and t represents the round of aggregation.

$$\mathrm{En}(\mathbf{w}_{\mathrm{global}}(t)) = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \mathrm{En}(\mathbf{g}_{e,i}(t)) \tag{2}$$

The updated $\mathrm{En}(\mathbf{w}_{\mathrm{global}}(t))$ are sent to the edge nodes for local training.

3) LOCAL TRAINING BY EDGE NODES

The edge nodes download $\mathrm{En}(\mathbf{w}_{\mathrm{global}}(t))$ from the cloud server and use the secret key for decryption. After loading $\mathbf{w}_{\mathrm{global}}(t)$ in their local model, the edge nodes retrain the classification layer by using the local resources. Afterward, the updated local parameters of the classification layer $\mathbf{g}_{\mathrm{e,i}}(t+1)$ are encrypted with the public key. Then, $\mathrm{En}(\mathbf{g}_{\mathrm{e,i}}(t+1))$ will be sent to the cloud server.

B. DROWSINESS DETECTION ON EDGE NODES

The model for drowsiness detection is presented in Section C of Chapter II. In this section, we design an input layer for image normalization. The architecture of the proposed model comprises three layers. The workflow of the edge nodes is demonstrated in Figure 5. The input layer reshapes the images to 224*224 pixels and loads them in the model. The extraction layer of the initial model is transferred from the pre-trained model and not updated to the cloud server after the local training. The classification layer is trained with a small learning rate to recognize eyes, head, and mouth. It is noteworthy that in the proposed work, the classification layers of the edge nodes are updated by downloading the global weights from the cloud server.

Algorithm 1 Privacy-Preserving Federated Transfer Learning for Driver Drowsiness Detection

Input: the baseline model
Initialization:
Cloud Server:
a). Pre-train the baseline model with a drowsiness database.
b). Establish channels for the edge nodes and send the pre-trained
initial model to the edge nodes.
Trust Party:
c). Generate public key and secret key according to CKKS-based
protocol and send them to the edge nodes.
Edge Nodes:
d). Train the pre-trained initial model on the local dataset of the edge
nodes and extract the classification layer parameters $\mathbf{g}_{e,i}$.
e). Encrypt $\mathbf{g}_{e,i}(1)$ with the public key and upload to the cloud server.
Procedure:
for t = 1, 2, …, T **do**
    (I). Cloud Server:
    Collect the $\mathrm{En}(\mathbf{g}_{\mathrm{e,i}}(t))$, where i = 1,2, …, N;
    Update the global parameters $\mathrm{En}(\mathbf{w}_{\mathrm{global}}(t))$ according to Eq.2;
    Send the $\mathrm{En}(\mathbf{w}_{\mathrm{global}}(t))$ to all edge nodes;
    (II). Edge Nodes: // Parallel computing
    for edge node i = 1, 2, …, N **do**
        Download the encrypted global parameters $\mathrm{En}(\mathbf{w}_{\mathrm{global}}(t))$
        from the cloud server;
        Decrypt the $\mathrm{En}(\mathbf{w}_{\mathrm{global}}(t))$ with the private key;
        Load the $\mathbf{w}_{\mathrm{global}}(t)$ to the classification layer of the local
        model;
        Train the local model and extract the classification layer
        parameters $\mathbf{g}_{\mathrm{e,i}}\ (t+1)$;
        Encrypt the $\mathbf{g}_{\mathrm{e,i}}(t+1)$ with public key and send
        $\mathrm{En}(\mathbf{g}_{\mathrm{e,i}}(t+1))$ to the cloud server;
    end
end

C. CKKS-BASED PRIVACY-PRESERVING COMMUNICATION PROTOCOL

In this work, the CKKS-based privacy-preserving communication protocol includes five functions.

The plaintext space $\mathfrak{R}_{\mathrm{Q}}$ is represented as a polynomial ring $\mathbb{Z}_{\mathrm{Q}}[\mathfrak{X}]/(\mathfrak{X}^{\mathrm{n}}+1)$, where n denotes the power of 2. Assuming that the scale of the plaintext is not large, the coefficients do

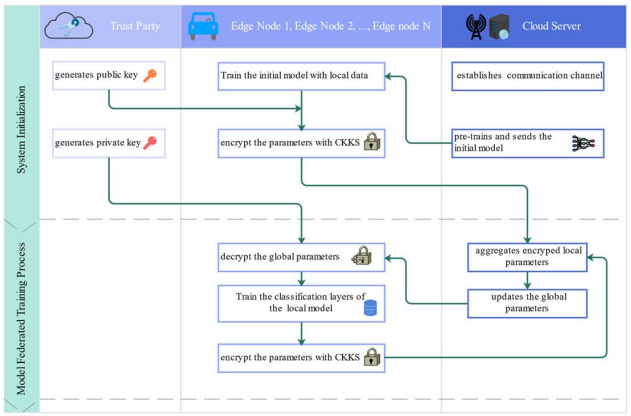

FIGURE 4. The flowchart of the proposed PFTL-DDD.

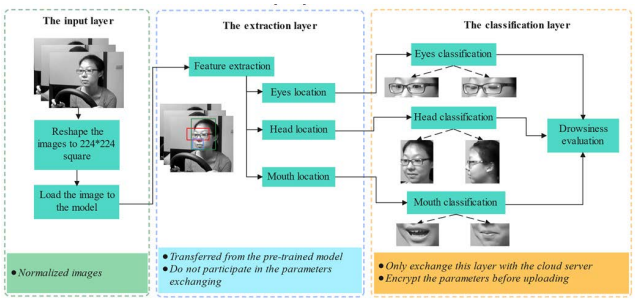

FIGURE 5. The workflow of edge nodes.

not exceed the modulus Q after encoding. We use $< \mathbf{j}, \mathbf{k} >$ to denote the inner product between two vectors $\mathbf{j}$ and $\mathbf{k}$.

1) Setup(λ): Given a security parameter λ and the max-depth L, we define $\mathrm{Q} = q_0{}^{\cdot L}$ and set a special modulus p, which is coprime of q, where q is a divisor of Q. We use χ_{key} to denote a distribution of secret key and χ_{err} to denote an error distribution over $\mathcal{R}_{\mathrm{Q}}$. Similarly, χ_{enc} is used to denote a random distribution for encryption.
2) KeyGenerate(): Sample $s \leftarrow \chi_{key}$, $a \leftarrow \mathcal{R}_{\mathrm{Q}}$, $e \leftarrow \chi_{err}$. The secret key $\mathcal{SK}$ and the public key $\mathcal{PK}$ are generated by the trust party. $\mathcal{SK} \leftarrow (1, s)$ and $\mathcal{PK} \leftarrow (b, a) \in \mathcal{R}_{\mathrm{Q}}^2$, where $b = (-a \cdots + e) \bmod \mathrm{Q}$.
3) ParaEnc($str, \mathcal{PK}$): Sample $v \leftarrow \chi_{enc}$ and $e_0, e_1 \leftarrow \chi_{err}$. $\mathcal{PK}$ is used to encrypt the local parameters in edge nodes based on the following expression:

$$\mathrm{En}(str) \leftarrow [v \cdot \mathcal{PK} + (str + e_0, e_1)] \bmod \mathrm{Q} \tag{3}$$

4) ParaAggregate($\mathrm{En}(str_1), \ldots, \mathrm{En}(str_N)$): The encrypted parameters $\mathrm{En}(str_k) \in \mathcal{R}_{\mathrm{Q}}^2$ are aggregated by the cloud server based on the following relation:

$$\mathrm{En}(str_{agg}) \leftarrow \mathrm{sum}[\mathrm{En}(str_1), \ldots, \mathrm{En}(str_N)] \bmod q \tag{4}$$

1) ParaDec ($\mathrm{En}(str_{\mathrm{agg}}), \mathcal{SK}$): The encrypted global parameter $\mathrm{En}(str_{agg})$ is decrypted by using the secret key $\mathcal{SK}$ at the edge nodes based on the following expression:

$$m \leftarrow < En(str_{agg}), \quad \mathcal{SK} > \bmod q \tag{5}$$

IV. THEORETICAL ANALYSIS

A. SECURITY ANALYSIS

A public-key encryption scheme $\sigma = (\mathrm{Gen}, \mathrm{En}, \mathrm{Dec})$ is CPA-secure if any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ cannot distinguish between the two equal length messages $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ which are intercepted from the message space $\mathcal{M}$. For any PPT adversary $\mathcal{A}$ there exists a negligible function negFun( ) such that:

$$\Pr\left[\mathrm{EXP}_{\mathcal{A},\sigma}^{cpa}(\lambda) = 1\right] \leq \frac{1}{2} + \mathrm{negFun}(\lambda) \tag{6}$$

where the probability is derived from the random coins of the adversary $\mathcal{A}$ and its indiscernibility experiment (generate a public key with λ, random bit $b \in \{0, 1\}$). According to this definition, we can draw the following conclusion: A CPA-secure encryption scheme guarantees security, even if there are eavesdroppers. The encryption scheme that meets the definition of CPA security must be probabilistic, and any deterministic encryption scheme does not fulfill CPA security.

Theorem 1: In the PFTL-DDD, if CKKS is CPA-secure, and there is no collusion between edge nodes and the cloud server/external eavesdroppers, the privacy of the driver's facial data of the edge nodes can be protected.

Proof: Assuming that the communication channels are adequately secure, the malicious activities performed by the eavesdroppers will be detected by the cloud server. In the proposed PFTL-DDD method, the parameters exchanged in the communication channel include the global parameters $\mathrm{En}(\mathbf{w}_{\mathrm{global}})$ and local parameters $\mathrm{En}(\mathbf{g}_{\mathrm{e,i}})$, which are encrypted by the secret key $\mathcal{SK}$. As described in Section C of Chapter III, the $\mathcal{SK}$ is generated by the security parameter λ. Therefore, even if the encrypted parameters are stolen by an eavesdropper, the eavesdropper cannot generate the secret key $\mathcal{SK}$ to decrypt the parameters without λ. If the edge nodes do not collude with the external eavesdropper, the eavesdropper is unable to obtain the original data without $\mathcal{SK}$. In addition, the parameters of the edge nodes uploaded to the cloud server are encrypted. The channels between the edge nodes and the cloud server are independent and secure. Therefore, without colluding with the cloud server, the edge nodes cannot infer the real information of other edge nodes by obtaining the parameters of them. Therefore, the proposed PFTL-DDD method can preserve the privacy of driver's facial data of the edge nodes.

B. COMPUTATION COMPLEXITY

In this section, the computation complexity of the CKKS in the proposed method is analyzed. The cryptography operation of the proposed method includes encryption, decryption, and

ciphertext addition. Table 1 shows the cryptography operation execution time of CKKS and Paillier. It indicates that the CKKS has a smaller computation cost compared to Paillier. When the parameter scale increases, the computation performance of Paillier decreases significantly. The computation time of CKKS is almost linearly positive correlation with the parameter scale. In the PFTL-DDD method, the classification layer parameters of the model need to take part in cryptography computing. Set En, Dec, and CipherAdd to represent the computational complexity function of encryption, decryption, and ciphertext addition operation of the proposed CKKS protocol, respectively. The computational complexity of each edge node is $\mathrm{En}\times|\mathbf{g}_{e,i}|+\mathrm{Dec}\times|\mathbf{w}_{global}|$ and the computational complexity of the cloud server is $\mathrm{CipherAdd}\times|\mathbf{g}_{e,i}|\times \mathrm{N}$.

TABLE 1. Execution time of CKKS and paillier.

Security protocol	Cryptography operation	Parameter scale		
		10000	20000	40000
CKKS	Encryption	38.18ms	80.01ms	0.16s
	Decryption	5.15ms	11.02ms	21.89ms
	Ciphertext addition	0.41ms	0.81ms	1.64ms
Paillier	Encryption	3.88s	14.64s	403.11s
	Decryption	1.05s	4.14s	115.03s
	Ciphertext addition	7.77ms	30.03ms	0.87s

C. COMMUNICATION COST

In the industrial scenario of driver drowsiness detection, reducing FL communication costs is significant. The parameters number of the initial model is 17.93 million, where the extraction layer contains 14.7 million parameters and the classification layer contains 3.23 million parameters. The proposed method reduces the scale of communication parameters from 17.93 million to 3.23 million by freezing the extraction layer in the edge nodes and only exchanges the classification layer between the cloud server and the edge nodes. On the contrary, the conventional FL method exchanges all the parameters, which will consume huge communication resources.

V. EXPERIMENT EVALUATION

A. DATASET

The proposed PFTL-DDD method is evaluated on the NTHU-DDD and YAWDD benchmark video datasets which are widely used in driver drowsiness detection researches [34]–[43]. The NTHU-DDD is an open-source driver drowsiness video dataset collected by the Computer Vision Lab of National Tsing Hua University [7]. This dataset contains recordings of about 9 and a half hours of both male and female drivers' behaviors, including normal driving, yawning, blinking, falling asleep, laughing, etc. It also contains videos of drivers with and without glasses/sunglasses during the daytime and nighttime. The resolution of the videos is 640 x 480 at 15/30 frames per second (FPS). The videos are segmented into 1,653,952 images, then labeled with eyes, head, mouth, and drowsiness. The YAWDD is also an open-source driver drowsiness video dataset collected by the DISCOVER Lab of the University of Ottawa [39]. The videos in this dataset are taken in a real driving scenario in the daytime. The resolution of the videos is 640 x 480 at 24/30 FPS. This dataset contains 320 recordings of both male and female drivers' behaviors, including 105 normal videos, 100 talking videos, 102 yawning videos, and 13 talking and yawning videos.

B. SETUP

We use Intel i7-8700, 32G RAM, and NVIDIA GeForce GTX3090ti for performing experiments. The proposed PFTL-DDD method is implemented using Python 3.8, TensorFlow 2.7.0, and MATLAB.

In this work, the PFTL-DDD system consists of four edge nodes. The NTHU-DDD and YAWDD were successively used as the experimental dataset to evaluate the proposed method. The NTHU-DDD experimental dataset is randomly split into three subsets (40%, 20%, 40%). The pre-training subset (40% of the dataset) is used for the initial model pre-training. The test subset (20% of the dataset) is used for testing the model. The last subset (the rest 40% of the dataset) is distributed to the edge nodes. The YAWDD experimental dataset is randomly divided into two subsets, 63 normal videos and 129 not normal videos for training and the rest 42normal videos and 86 not normal videos for testing. The video is segmented into images and then marked as normal and not normal. In the experiment, the classification threshold is set to 10%, that is, if more than 10% of the images in the video have not normal classification results, the video will be classified as abnormal.

C. PARAMETERS AND RUNNING TIME

The workflow of the proposed PFTL-DDD method is discussed in Section A of Chapter III. The cloud server pre-trains the baseline model on the pre-training subset of the experimental dataset and sends the pre-trained initial model to all edge nodes. The trust party generates the public and secret keys based on the CKKS protocol and sends them to the edge nodes. Each edge node preprocesses the local dataset and uses it for training the model. By fixing the extraction layer, the edge nodes upload the encrypted parameters of the classification layer to the cloud server, as discussed in Section B of Chapter III. The encryption process is presented in Section C of Chapter III. During the training process, we use Adam optimizer, 10 training epochs, a batch size of 64, learning rate of 0.001, and a dropout of 0.3.

D. COMPARISON OF DIFFERENT METHODS

In this section, we present the functional comparison of various driver drowsiness detection methods, such as HTDBN-DDD [7], ETL-DDD [44], FLDSM [10] and FedSup [45]. The functional comparisons of the methods are presented in Table 2.

The HTDBN-DDD and ETL-DDD do not support multi-nodes collaborative training. Driver drowsiness data are usually distributed stored in different edge nodes which rarely

share the facial data with the central server for centralized learning due to privacy concerns. However, the transfer learning method used by ETL-DDD provides the inspiration to improve the performance of federated learning. The FLDSM and FedSup adopt federated learning for driver drowsiness detection. However, they do not provide security frameworks to protect the privacy of edge nodes, nor use transfer learning to improve the performance. The PFTL-DDD adopts a federated learning framework for driver drowsiness detection which protect the privacy of the edge nodes by using a CKKS-based encryption scheme while improving the performance of the system model by using the transfer learning mechanism.

TABLE 2. Functional difference of comparative methods.

	HTDBN-DDD	ETL-DDD	FLDSM	FEDSUP	PFTL-DDD
COLLABORATIVE TRAINING	×	×	√	√	√
TRANSFER LEARNING	×	√	×	×	√
PRIVACY-PRESERVING	×	×	×	×	√
COMMUNICATION RESOURCE SAVING	—	—	×	√	√

E. EXPERIMENTAL RESULTS

1) CASE 1 (COMPARISON OF CL, TCL, FL, AND PFTL-DDD)

In this case, driver drowsiness detection is realized by centralized learning (CL), centralized transfer learning (TCL), federated learning (FL), and PFTL-DDD respectively. The classification results of the four methods on the NTHU-DDD and YAWDD are presented in Table 3 and Table 4, respectively. Taking the experimental results of the NTHU-DDD for analysis, the accuracy, precision, and recall of TCL are 85.58%, 85.52%, and 86.76%, respectively. The classification performance of TCL is better than CL, whose metrics are 81.87%, 82.43%, and 82.45%, respectively. The performance gap between CL and TCL reflects the improvement effect of transfer learning. The evaluation metrics of PFTL-DDD are 83.48%, 83.56%, and 84.68%, which are superior to the FL and the CL. There are various works [46]–[48] show that the performance gap between decentralized learning and centralized learning is pervasive. The adoption of transfer learning enables PFTL-DDD to outperform centralized learning. The analysis shows the experimental results of the YAWDD were similar to those of the NTHU-DDD. The classification results of the PFTL-DDD on the YAWDD are better than those of the FL and the CL (as shown in Table 4). The accuracy curves of the four methods on the NTHU-DDD and YAWDD are presented in Figure 6 and Figure 7, respectively. It can be seen that the convergence speed of the PFTL-DDD is faster than FL and CL.

2) CASE 2 (IMPLEMENT PFTL-DDD WITH DIFFERENT NETWORKS)

In this case, VGG-16, ResNet, and DenseNet are adopted as the baseline network model to realize the proposed PFTL-DDD method, respectively. The classification results of the NTHU-DDD and the YAWDD are presented in Table 5 and Table 6, respectively. Taking the experimental results of the NTHU-DDD for analysis, the PFTL-DDD method implemented by VGG-16, ResNet, and DenseNet has stable classification performance that the metrics of the three models for the PFTL-DDD method are in the range of 83%∼85%, 83%∼85%, and 84%∼86%, respectively. The experimental results of the different networks on the YAWDD dataset were similar to those of the different networks on NTHU-DDD. The accuracy curves of the three networks on the NTHU-DDD and YAWDD are presented in Figure 8 and Figure 9, respectively, which indicate that the convergence rates of different networks are similar. This case illustrates that the proposed method has good scalability and can be implemented on models with different structures.

TABLE 3. The classification results of CL, TCL, FL, and PFTL-DDD on the NTHU-DDD.

	Accuracy	Precision	Recall
CL	81.87%	82.43%	82.45%
TCL	85.58%	85.52%	86.76%
FL	76.31%	76.96%	77.23%
PFTL-DDD	83.48%	83.56%	84.68%

TABLE 4. The classification results of CL, TCL, FL, and PFTL-DDD on the YAWDD.

	Accuracy	Precision	Recall
CL	75.00%	58.93%	78.57%
TCL	89.06%	79.16%	90.47%
FL	71.09%	58.49%	73.80%
PFTL-DDD	85.93%	74.00%	88.10%

TABLE 5. The classification results of the NTHU-DDD of PFTL-DDD realized by VGG-16, ResNet, and DenseNet.

	Accuracy	Precision	Recall
VGG-16	83.48%	83.56%	84.68%
ResNet	84.15%	84.25%	85.21%
DenseNet	84.19%	84.24%	85.32%

TABLE 6. The classification results of the YAWDD of PFTL-DDD realized by VGG-16, ResNet, and DenseNet.

	Accuracy	Precision	Recall
VGG-16	85.93%	74.00%	88.10%
ResNet	85.93%	75.00%	85.71%
DenseNet	86.71%	75.51%	88.10%

VI. DISCUSSIONS

A. THE ADVANTAGES OF THE PFTL-DDD METHOD

The PFTL-DDD surpasses the drowsiness detection FL method in many respects. The advantages of the PFTL-DDD can be summarized as follow:

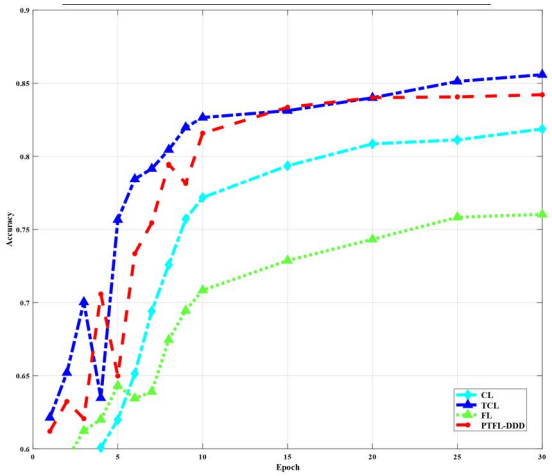

FIGURE 6. The accuracy curves of the four methods on the NTHU-DDD.

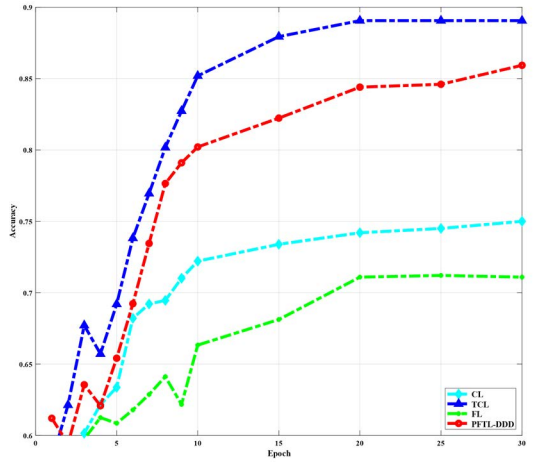

FIGURE 7. The accuracy curves of the four methods on the YAWDD.

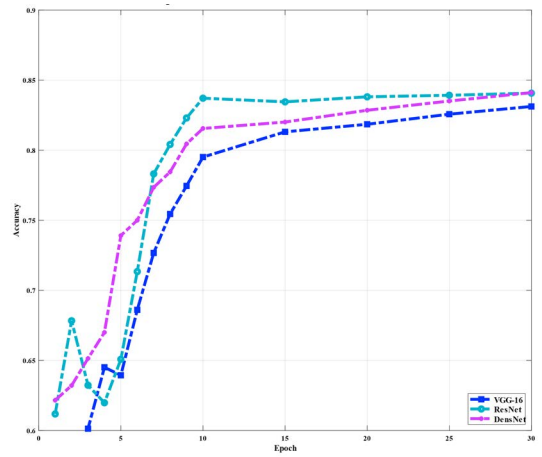

FIGURE 8. The accuracy curves of PFTL-DDD realized by VGG-16, ResNet, and DenseNet on the NTHU-DDD.

1) The PFTL-DDD method achieves better classification performance than centralized learning method even if the driver information is kept in the edge nodes.

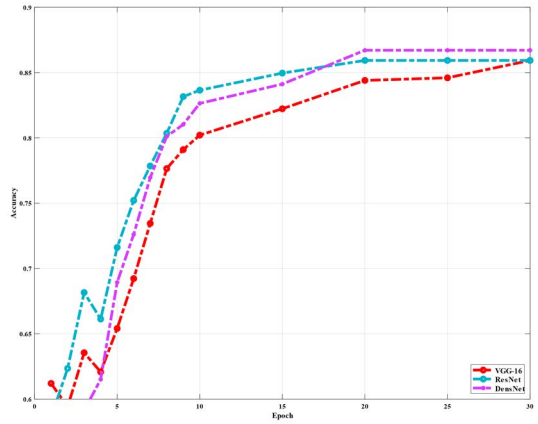

FIGURE 9. The accuracy curves of PFTL-DDD realized by VGG-16, ResNet, and DenseNet on the YAWDD.

2) The PFTL-DDD method reduces the communication cost by freezing the extraction layer of the collaborative training model.
3) The PFTL-DDD method provides a security framework based on the CKKS protocol that protects the drivers' privacy data in the federated learning process.
4) The PFTL-DDD method has good model structure scalability and can be adopted in other driver abnormal actions detection such as smoking, calling, and drinking.

B. THE SHORTCOMINGS OF THE PFTL-DDD METHOD

The shortcomings of the PFTL-DDD can be summarized in two aspects:

1) The proposed method minimizes the loss function with gradient descent (GD). GD is a step-by-step method that the parameters update of the current iteration depends on the parameters of the previous iteration. Only referring to a single gradient descent direction will reduce the model convergence speed.
2) Dataset in industrial scenarios is usually Non-IID (Non-independent and identically distributed) which leads to the loss of accuracy in federated learning method with low global model update frequency. However, increasing the frequency of global aggregation has the side effect of increasing communication overhead. A much more efficient global aggregation algorithm is essential for driver drowsiness detection federated learning method.

VII. CONCLUSION

Federated learning is frequently adopted in those industrial scenarios where the datasets belong to different ownerships. Collaborative training solves the problem of data isolation, but decentralized learning creates a new problem of performance degradation. Considering that the training data is personal biometric information, the risk of privacy leakage in the FL system becomes even more prominent in

the scenario of driver drowsiness detection. In this work, we propose a privacy-preserving federated transfer learning method (PFTL-DDD) that introduces transfer learning and CKKS-based privacy-preserving protocol in the drowsiness detection FL system. The experiment results on the driver drowsiness dataset show that the PFTL-DDD method exhibits outstanding performance and saves considerable communication resources as compared to the conventional drowsiness detection FL systems. Moreover, the analysis shows that the security framework of the proposed method protects the privacy of edge nodes.

In future studies, we will further explore the usage of momentum for GD to accelerate the convergence, as well as the optimization of the global parameter aggregation algorithm in the future.

REFERENCES

[1] V. Saini and R. Saini, "Driver drowsiness detection system and techniques: A review," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, no. 3, pp. 4245–4249, 2014.

[2] M. Ramzan, H. U. Khan, S. M. Awan, A. Ismail, M. Ilyas, and A. Mahmood, "A survey on state-of-the-art drowsiness detection techniques," *IEEE Access*, vol. 7, pp. 61904–61919, 2019.

[3] S. Kaplan, M. A. Guvensan, A. G. Yavuz, and Y. Karalurt, "Driver behavior analysis for safe driving: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3017–3032, Dec. 2015.

[4] A. L. A. Ramos, J. C. Erandio, D. H. T. Mangilaya, N. D. Carmen, E. M. Enteria, and L. J. Enriquez, "Driver drowsiness detection based on eye movement and yawning using facial landmark analysis," *Int. J. Simul.-Syst., Sci. Technol.*, vol. 20, pp. 37.1–37.8, 2019.

[5] R. Ghoddoosian, M. Galib, and V. Athitsos, "A realistic dataset and baseline temporal model for early drowsiness detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 178–187.

[6] D. J. Fremont, E. Kim, Y. V. Pant, S. A. Seshia, A. Acharya, X. Bruso, P. Wells, S. Lemke, Q. Lu, and S. Mehta, "Formal scenario-based testing of autonomous vehicles: From simulation to the real world," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–8.

[7] C. H. Weng, Y. H. Lai, and S. H. Lai, "Driver drowsiness detection via a hierarchical temporal deep belief network," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 117–133.

[8] B. K. Savas, "Real time driver fatigue detection system based on multi-task ConNN," *IEEE Access*, vol. 8, pp. 12491–12498, 2020.

[9] P. Voigt and A. Von dem Bussche, "The EU general data protection regulation (GDPR)," in *A Practical Guide*, vol. 10, 1st ed. Cham, Switzerland: Springer, 2017, Art. no. 3152676.

[10] A. Zafar, C. Prehofer, and C.-H. Cheng, "Federated learning for driver status monitoring," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 1463–1469.

[11] Z. Zhang, L. Zhang, Q. Li, K. Wang, N. He, and T. Gao, "Privacy-enhanced momentum federated learning via differential privacy and chaotic system in industrial cyber-physical systems," *ISA Trans.*, Sep. 2021, doi: 10.1109/TII.2022.3140806.

[12] Z. Zhang, N. He, Q. Li, K. Wang, H. Gao, and T. Gao, "DetectPMFL: Privacy-preserving momentum federated learning considering unreliable industrial agents," *IEEE Trans. Ind. Informat.*, early access, Jan. 6, 2022, doi: 10.1109/TII.2022.3140806.

[13] E. Novikova, D. Fomichov, I. Kholod, and E. Filippov, "Analysis of privacy-enhancing technologies in open-source federated learning frameworks for driver activity recognition," *Sensors*, vol. 22, no. 8, p. 2983, Apr. 2022.

[14] J. Cabrero-Holgueras and S. Pastrana, "SoK: Privacy-preserving computation techniques for deep learning," *Proc. Privacy Enhancing Technol.*, vol. 2021, no. 4, pp. 139–162, Oct. 2021.

[15] Y. Rahulamathavan, "Privacy-preserving similarity calculation of speaker features using fully homomorphic encryption," 2022, *arXiv:2202.07994*.

[16] E. M. Shiriaev, A. S. Nazarov, N. N. Kycherov, and N. A. Sotikova, "Efficient implementation of the CKKS scheme using a quadratic residue number system," in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng. (ElConRus)*, Jan. 2021, pp. 665–669.

[17] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowl.-Based Syst.*, vol. 216, Mar. 2021, Art. no. 106775.

[18] H. Zhu, H. Zhang, and Y. Jin, "From federated learning to federated neural architecture search: A survey," *Complex Intell. Syst.*, vol. 7, no. 2, pp. 639–657, Apr. 2021.

[19] 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," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 3, pp. 1759–1799, Jun. 2021, doi: 10.1109/COMST.2021.3090430.

[20] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jul. 2020.

[21] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3320–3328.

[22] G. Vrbancic and V. Podgorelec, "Transfer learning with adaptive fine-tuning," *IEEE Access*, vol. 8, pp. 196197–196211, 2020.

[23] B. Q. Huynh, H. Li, and M. Giger, "Digital mammographic tumor classification using transfer learning from deep convolutional neural networks," *J. Med. Imag.*, vol. 3, no. 3, Aug. 2016, Art. no. 034501.

[24] D. S. Kermany, M. Goldbaum, W. Cai, C. C. Valentim, H. Liang, S. L. Baxter, A. McKeown, G. Yang, X. Wu, and F. Yan, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell*, vol. 172, no. 5, pp. 1122–1131, 2018.

[25] C. Wang, D. Chen, J. Chen, X. Lai, and T. He, "Deep regression adaptation networks with model-based transfer learning for dynamic load identification in the frequency domain," *Eng. Appl. Artif. Intell.*, vol. 102, Jun. 2021, Art. no. 104244.

[26] L. Zhu, S. Ark, Y. Yang, and T. Pfister, "Learning to transfer learn: Reinforcement learning-based selection for adaptive transfer learning," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 342–358.

[27] Y. Cui, Y. Song, C. Sun, A. Howard, and S. Belongie, "Large scale fine-grained categorization and domain-specific transfer learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4109–4118.

[28] M. Hao, H. Li, G. Xu, S. Liu, and H. Yang, "Towards efficient and privacy-preserving federated deep learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6, doi: 10.1109/ICC.2019.8761267.

[29] C. Zhou, A. Fu, S. Yu, W. Yang, H. Wang, and Y. Zhang, "Privacy-preserving federated learning in fog computing," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10782–10793, Apr. 2020, doi: 10.1109/JIOT.2020.2987958.

[30] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, Q. S. T. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020, doi: 10.1109/TIFS.2020.2988575.

[31] M. Sumathi and S. Sangeetha, "A group-key-based sensitive attribute protection in cloud storage using modified random Fibonacci cryptography," *Complex Intell. Syst.*, vol. 7, no. 4, pp. 1733–1747, Jun. 2020.

[32] W. Ou, J. Zeng, Z. Guo, W. Yan, D. Liu, and S. Fuentes, "A homomorphic-encryption-based vertical federated learning scheme for rick management," *Comput. Sci. Inf. Syst.*, vol. 17, no. 3, pp. 819–834, 2020.

[33] H. Chen, W. Dai, M. Kim, and Y. Song, "Efficient multi-key homomorphic encryption with packed ciphertexts with application to oblivious neural network inference," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 395–412.

[34] J. Bai, W. Yu, Z. Xiao, V. Havyarimana, A. C. Regan, H. Jiang, and L. Jiao, "Two-stream spatial-temporal graph convolutional networks for driver drowsiness detection," *IEEE Trans. Cybern.*, early access, Oct. 4, 2021, doi: 10.1109/TCYB.2021.3110813.

[35] Q. Zhuang, Z. Kehua, J. Wang, and Q. Chen, "Driver fatigue detection method based on eye states with pupil and iris segmentation," *IEEE Access*, vol. 8, pp. 173440–173449, 2020.

[36] J. Lyu, Z. Yuan, and D. Chen, "Long-term multi-granularity deep framework for driver drowsiness detection," 2018, *arXiv:1801.02325*.

[37] T. H. Vu, A. Dang, and J.-C. Wang, "A deep neural network for real-time driver drowsiness detection," *IEICE Trans. Inf. Syst.*, vol. E102.D, no. 12, pp. 2637–2641, 2019.

[38] S. Park, F. Pan, S. Kang, and C. D. Yoo, "Driver drowsiness detection system based on feature representation learning using various deep networks," in *Proc. Asian Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 154–164.

[39] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: A yawning detection dataset," in *Proc. 5th ACM Multimedia Syst. Conf.*, 2014, pp. 24–28.

[40] R. M. Salman, M. Rashid, R. Roy, M. M. Ahsan, and Z. Siddique, "Driver drowsiness detection using ensemble convolutional neural networks on YawDD," 2021, *arXiv:2112.10298*.
[41] S. Junaedi and H. Akbar, "Driver drowsiness detection based on face feature and PERCLOS," *J. Phys., Conf.*, vol. 1090, Sep. 2018, Art. no. 012037.
[42] S. Saurav, S. Mathur, I. Sang, S. S. Prasad, and S. Singh, "Yawn detection for driver's drowsiness prediction using bi-directional LSTM with CNN features," in *Proc. Int. Conf. Intell. Hum. Comput. Interact.* Cham, Switzerland: Springer, 2019, pp. 189–200.
[43] N. N. Pandey and N. B. Muppalaneni, "Real-time drowsiness identification based on eye state analysis," in *Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS)*, Mar. 2021, pp. 1182–1187.
[44] M. V. S. Laxshmi and L. Chandana, "An enhanced driver drowsiness detection system using transfer learning," in *Proc. 5th Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Dec. 2021, pp. 1671–1678.
[45] C. Zhao, Z. Gao, Q. Wang, K. Xiao, Z. Mo, and M. J. Deen, "FedSup: A communication-efficient federated learning fatigue driving behaviors supervision framework," 2021, *arXiv:2104.12086*.
[46] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 1205–1221, Jun. 2019, doi: 10.1109/JSAC.2019.2904348.
[47] L. Zhang, Z. Zhang, and C. Guan, "Accelerating privacy-preserving momentum federated learning for industrial cyber-physical systems," *Complex Intell. Syst.*, vol. 7, no. 6, pp. 3289–3301, Dec. 2021, doi: 10.1007/s40747-021-00519-2.
[48] W. Liu, L. Chen, Y. Chen, and W. Zhang, "Accelerating federated learning via momentum gradient descent," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 8, pp. 1754–1766, Aug. 2020, doi: 10.1109/TPDS.2020.2975189.

LINLIN ZHANG received the B.S. degree from the College of Software, Nankai University, China, in 2011, and the M.S. degree in computer science and technology from Nankai University, in 2015. She is currently pursuing the Ph.D. degree with the Department of Information and Computer Science, Keio University, under the supervision of Prof. Hideo Saito. Since 2015, she has been with the China Automotive Technology and Research Center (CATARC). She is also a Registered Expert of the International Organization for Standardization (ISO). Her research interests include the application of computer vision in intelligent and connected vehicles.

HIDEO SAITO (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from Keio University, Japan, in 1992. Since 1992, he has been with the Faculty of Science and Technology, Keio University. From 1997 to 1999, he joined the Virtualized Reality Project at the Robotics Institute, Carnegie Mellon University, as a Visiting Researcher. Since 2006, he has been a Full Professor with the Department of Information and Computer Science, Keio University. His research interests include computer vision and pattern recognition and their applications to augmented reality, virtual reality, and human-robotic interaction. His recent activities in academic conferences include being the Program Chair of ACCV 2014, the General Chair of ISMAR 2015, the Program Chair of ISMAR 2016, and the Scientific Program Chair of EuroVR 2020.

LIANG YANG received the B.S. degree from the College of Software, Nankai University, China, in 2014, and the M.D. degree in computer science and technology from Nankai University, in 2017. He is currently a Senior Scientist at China Auto Information Technology Company Ltd. His research interests include artificial intelligence, deep learning, and image processing.

JIAJIE WU was born in Guangyuan, Sichuan, China, in 1996. He received the B.S. degree in optical and electronic information from the Huazhong University of Science and Technology, and the M.S. degree in computer engineering from Syracuse University, NY, in 2021. From 2017 to 2018, he followed a Prof. Mengfan Cheng researching in chaotic cryptography. During his Graduate career, from 2019 to 2021, his research interest is to solve vehicle routing problems (VRP) by utilizing DNN. Since 2021, he has been a standardization fellow of the China Automotive Technology and Research Center (CATARC) and commits to the standardization of intelligent connected vehicles (ICV). He is currently a member of Chinese delegation in World Forum for Harmonization of Vehicle Regulations (WP.29) at UNECE. His researches standards and regulations of ICV of China and others countries, and commits to the standardization of Chinese GB standards and UN regulations.