Privacy-Preserving Driver Drowsiness Detection with Spatial Self-Attention and Federated Learning

Tran Viet Khoa, Do Hai Son, Mohammad Abu Alsheikh, Yibeltal F Alem, and Dinh Thai Hoang

***Abstract*—Driver drowsiness is one of the main causes of road accidents and is recognized as a leading contributor to traffic-related fatalities. However, detecting drowsiness accurately remains a challenging task, especially in real-world settings where facial data from different individuals is decentralized and highly diverse. In this paper, we propose a novel framework for drowsiness detection that is designed to work effectively with heterogeneous and decentralized data. Our approach develops a new Spatial Self-Attention (SSA) mechanism integrated with a Long Short-Term Memory (LSTM) network to better extract key facial features and improve detection performance. To support federated learning, we employ a Gradient Similarity Comparison (GSC) that selects the most relevant trained models from different operators before aggregation. This improves the accuracy and robustness of the global model while preserving user privacy. We also develop a customized tool that automatically processes video data by extracting frames, detecting and cropping faces, and applying data augmentation techniques such as rotation, flipping, brightness adjustment, and zooming. Experimental results show that our framework achieves a detection accuracy of 89.9% in the federated learning settings, outperforming existing methods under various deployment scenarios. The results demonstrate the effectiveness of our approach in handling real-world data variability and highlight its potential for deployment in intelligent transportation systems to enhance road safety through early and reliable drowsiness detection.**

***Index Terms*—Intelligent transportation, federated learning, and drowsiness detection.**

I. Introduction

Technological advancements in transportation have significantly improved road safety and driving efficiency, particularly in autonomous and semi-autonomous vehicles. Various driver assistance systems, including fatigue detection technologies, have been developed to mitigate risks associated with human error. Despite these improvements, driver fatigue remains a critical factor in road accidents, contributing to 10%–20% of serious crashes worldwide [1]. In Australia, fatigue is recognized as one of the "fatal five" causes of road accidents, alongside speeding, drug and alcohol impairment, failure to wear seatbelts, and driver distraction [1].

There are two major approaches to detecting drowsiness to reduce these risks: physiological signals and vision-based methods [2]. Physiological signals, such as Electroencephalogram (EEG) [3] and Electrocardiogram (ECG) [4], rely on biological signals, including brain waves and heart rate variations, to identify specific biological markers of drowsiness. However, this approach requires complex systems to record an individual's biosignals, making it suitable for lab environments but challenging to deploy in practical settings [2]. In contrast, vision-based drowsiness detection [5] uses visual cues, such as head position, eye movements, and mouth activities, to assess driver fatigue. Effective drowsiness detection must identify subtle signs of drowsiness, such as small changes in eye behavior or facial expressions, while adapting to individual differences and varying in-vehicle conditions. This approach is more user-friendly and widely applied in various domains, including in-vehicle driver assistance [6] and driver fatigue monitoring [7]. Additionally, recent advances in machine learning have enabled the development of robust, real-time fatigue detection systems aimed at enhancing driver safety and preventing accidents [8]. However, the accuracy of vision-based methods can be influenced by factors such as background complexity, video quality, and dataset heterogeneity across different individuals [2]. Moreover, since drowsiness-related data is inherently distributed across various locations, federated learning offers a promising solution. It enables accurate detection in a decentralized setting while preserving data privacy and minimizing network overhead by avoiding the transfer of large datasets [9].

There are several challenges in data distribution in vision-based drowsiness detection systems. First, due to the nature of drivers' faces dataset, each individual's face is unique, leading to heterogeneous data [9], [10]. This data can cause deep learning models to fall into local optima, preventing convergence and reducing accuracy [11]. This challenge becomes more significant in decentralized environments, where analyzing data requires combining knowledge from multiple deep learning models. Second, in federated learning, the overall system accuracy can be compromised if individual clients contribute low-quality models that have learned patterns significantly different from those of other clients [9]. Such inconsistencies may lead to disruptive updates during model aggregation at the central server, ultimately degrading the performance of the global model. This challenge is particularly relevant in drowsiness detection, where variations in driver behavior, environment, and recording conditions across clients can result in highly divergent local models [9]. Third, existing datasets are often limited in size and diversity, failing to capture the full range of real-world driving conditions. This lack of representativeness can hinder the generalizability of drowsiness detection models. Moreover, each video frame includes complex background environments, and variations in

T. V. Khoa, M. Abu Alsheikh and Y. F. Alem are with the University of Canberra, Australia (e-mail: {khoa.tran, mohammad.abualsheikh, yibe.alem}@canberra.edu.au).

D. H. Son is with the School of Electrical Engineering, Computing and Mathematical Sciences, Curtin University, Australia and the VNU Information Technology Institute, Hanoi, Vietnam (e-mail: dohaison1998@vnu.edu.vn).

D. T. Hoang is with the School of Electrical and Data Engineering, University of Technology Sydney, Australia (e-mail: hoang.dinh@uts.edu.au).

lighting, particularly between day and night, can significantly degrade detection accuracy [12].

To deal with the first challenge of handling heterogeneous facial data, our proposed framework employs SSA to emphasize the most important features of each extracted face. By applying an SSA mechanism, the model can focus on regions such as the eyes and mouth, which are critical for drowsiness detection. This approach reduces variations between images, ensuring more consistent data. By emphasizing key facial features, SSA improves the model's ability to adapt to different conditions, resulting in more reliable and accurate drowsiness detection. To address the second challenge, we implement GSC in the federated learning model to filter out dissimilar learned knowledge from operators. This ensures the quality of local models while maintaining the accuracy and robustness of the aggregated global model. The GSC identifies which updates are most aligned with the global learning objective, ensuring that only meaningful updates contribute to model improvement. By selecting gradients across operators, this approach enhances model convergence and reduces the impact of noisy or biased updates, leading to more stable and reliable learning outcomes. Finally, to address the third challenge, we build a customized frame extraction and augmentation tool that automatically extracts frames from videos. It can also perform face detection and extraction to remove the background elements of the images. After that, we deploy augmentation techniques to generate variations of the original images, enhancing the dataset. This process not only increases the diversity of the data but also improves the model's ability to recognize faces under different conditions, ensuring more robust performance in real-world applications. Our contributions can be summarized as follows:

- We propose a novel framework for detecting driver drowsiness that can effectively handle heterogeneous data. This framework is capable of functioning in decentralized environments, ensuring its applicability in real-world scenarios where data is often distributed and privacy concerns are paramount.
- We develop a preprocessing tool that extracts frames from raw video streams, performs face detection and cropping, and applies frame augmentation techniques to increase dataset variability and model generalization.
- We develop a highly effective federated learning-based model integrating with GSC at the server side to select the appropriate models from operators for aggregation. In addition, our model employs SSA and LSTM at local training operators to improve the accuracy of detection. This approach ensures that the global model remains accurate and robust, even when working with decentralized and diverse data sources, while also preserving data privacy.
- We perform extensive simulations in a real-world dataset to evaluate our system. The results show that our model outperforms existing methods in both centralized and federated learning. Our proposed model can achieve up to 89.9% accuracy with federated learning. Additionally, our experiments demonstrate that the model can easily adapt to new participants without prior training, making it practical for real-world applications.

II. Related works

There are several works trying to deal with detecting drowsiness using computer vision. In [13], the authors use various machine learning and deep learning models (i.e., K-Nearest Neighbour, Naïve Bayes, Logistic Regression, Decision Trees, Random Forest, XGBoost, MLP, and CNN) to compare their performance in drowsiness detection in the University of Texas at Arlington Real-Life Drowsiness Dataset (UTA-RLDD). The simulation results show that the Logistic Regression achieves the highest accuracy of up to 75.67%. In [14], the authors introduce an isotropic self-supervised learning with momentum contrast (IsoSSL-MoCo) model to learn the representations of participants' images and exploit the complementarity of multimodal data. They propose a fusion model that is pretrained by the IsoSSL-MoCo to improve the performance of driver drowsiness detection. The simulation results with the NTHU-DDD dataset show that their proposed solution can achieve an accuracy of up to 93.71%. In [15], the authors propose two models for detecting drowsiness from a dataset. The first model (Model-A) combines YOLOv3 [16] and LSTM, while the second model (Model-B) integrates CNN and LSTM. The simulation results show that although Model-A is more complex than Model-B, it achieves a lower accuracy of 86% compared to Model-B's 97.5%. However, Model-A offers advantages in training efficiency over Model-B. In [17], the authors propose using Vision Transformers (ViT) for driver drowsiness detection. The simulation results show that their approach achieves a test accuracy of up to 98.10% and an average prediction time of approximately 17 ms per frame. In [18], the authors propose an approach that pretrains the dataset using YOLOv5 to detect and extract participants' faces. After that, Vision Transformers are employed to detect drowsiness. The simulation results show that their approach can achieve an accuracy of up to 95.5%.

All the above methods focus on centralized learning where all data is collected into a central server for analysis. However, in practice, due to the nature of decentralisation of car-driving environment, it is difficult to gather all data into a centralized server without the risk of compromising data privacy. In [19], the authors propose a federated learning approach for detecting fatigue driving behaviors. Their method uses edge servers to manage client data, while federated learning on cloud servers aggregates the learned knowledge from these edge servers. The proposed model, FedSup, enhances model sharing efficiency and reduces communication overhead. The simulation results show that their approach achieves an accuracy of approximately 90% in detecting fatigue driving behaviors.

We observe that many existing methods formally divide data into training and testing datasets. However, one of the biggest limitations in many drowsiness detection studies is that the same participants often appear in both training and testing datasets [20]. Specifically, all of the aforementioned methods create training and testing datasets using different features from the same participants. This thus allows the

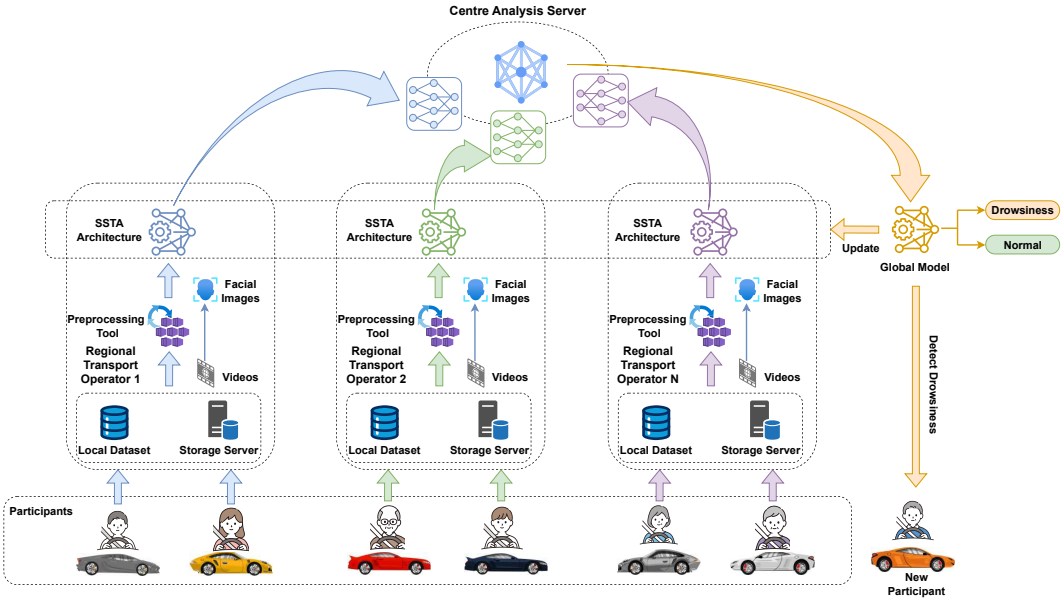


Fig. 1: The proposed system model.

learning model to learn individual-specific patterns, such as unique facial features or eye movement behaviors. As a result, the reported performance may not reflect how well the model works on completely new users. Therefore, in this paper, we consider a more practical scenario where a trained model can be used to detect drowsiness for new participants who have not been previously included in the training process. Recently, the authors in [21] propose to use separating participants in the training and testing datasets. They also employ federated learning for fatigue detection in a decentralized driving environment. However, their primary focus is on evaluating the impact of noise in federated learning to enhance the privacy of drivers' data. Although their approach uses different participants for training and testing, it achieves limited performance, with a maximum accuracy of approximately 70% for drowsiness detection. In this paper, we also consider a decentralized driving environment while evaluating the performance of participant separation in training and testing datasets, but with a focus on improving detection accuracy while maintaining data privacy. To achieve this, our model is designed to easily adapt to new participants for drowsiness detection. This adaptability enhances its practicality for real-world applications, where new users may continuously join the system. Extensive simulation results show that our proposed model achieves an accuracy of 89.9% on the testing dataset which includes both trained and untrained participants.

III. Proposed Drowsiness Detection Framework

In this paper, we propose a framework for detecting drowsiness in a decentralized car-driving environment. Fig. 1 describes our proposed system model. As shown in Fig. 1, there are N regional transport operators (operators), each collecting data from local participants who may vary in age, gender, and background environment.

In practice, video recordings of participants will be collected during the experiment and subsequently reviewed by trained operators for labeling. Each operator will carefully analyze the videos to identify and classify the participants' states of alertness, marking specific time segments where signs of drowsiness are observed. In particular, transport operator experts can review video recordings of drivers and tag time segments as drowsy or alert based on clear observable cues. For example, they can mark a period as drowsy whenever the driver shows obvious fatigue indicators, e.g., prolonged eye closures (extended blinks beyond the normal 0.1 -0.4 second range) [22], frequent yawning, or episodes of head nodding where the driver's head briefly droops [23]. To keep the labeling consistent and objective, the operators can follow standardized guidelines, e.g., defining any blink longer than a certain threshold (e.g., 0.4 seconds) as a drowsiness event [22], so that each video is judged by measurable behaviors rather than personal guesswork. This labeling process is designed to be feasible in real operational settings, i.e., it relies only on regular camera footage and human observation, without any specialized medical instruments or clinical tests.

Since facial videos are sensitive data, they are not shared across operators or networks. Furthermore, because each operator only has access to a limited amount of participant data, it becomes challenging to train accurate models locally. To address these issues, our framework enables collaborative model training across the operators while preserving user privacy. This allows operators to improve the performance of drowsiness detection models without sharing raw data. Each operator has a storage server for managing its local dataset, ensuring that sensitive data remains private and is not transmitted across the network. Within each operator, to prepare this data for training and detection, a preprocessing tool is used to convert the recorded videos into sequences of facial images. This includes steps such as face detection,

face extraction, and data augmentation, which are explained in detail in Section III-A. After preprocessing, a detection module based on SSA and LSTM is used to identify signs of drowsiness in facial image sequences. More details about this processing architecture can be found in Section III-B.

Due to the limited training data available in each operator, it is essential for operators to exchange learned knowledge to enhance detection accuracy. To address this, each operator sends its trained model to a central analysis server (e.g., the National Transport Authority). The central analysis server uses the trained models from operators to compute gradients, selects the most valuable information through gradient selection, and aggregates it into a new global model. This global model is then used to update the deep learning models within each regional transport operator, enhancing their ability to detect drowsiness. More importantly, the updated model can also be used by new participants to detect drowsiness while driving, improving the general adaptability and effectiveness of the system. This decentralized learning strategy is also used in real-world, large-scale systems. For instance, Tesla's Autopilot system adopts a fleet learning approach, where each vehicle processes driving data locally and sends only summarized model updates to a central server. This enables the global model to improve continuously while ensuring that raw video and sensor data remain on the vehicle [24], [25].

Overall, in this paper, we propose a novel framework that can learn from different groups of people's videos to detect drowsiness with high accuracy while preserving privacy. Our proposed framework includes three main processes as follows: the Preprocessing Process, the SSA and Temporal Aggregation network (SSTA) architecture, and the Federated Learning Drowsiness Detection.

A. Preprocessing Process

We develop a tool that preprocesses video data by extracting multiple frames from videos, detecting and extracting faces, and augmenting frames. The processes of this tool are described in Fig. 2. In the first step, the tool first extracts the video into multiple frames within a predefined time window and then performs the next processes as follows.

1) Face Detection and Extraction: First, we detect and extract the face from a frame to support the processing model in the next step, which focuses on detecting user drowsiness. To do that, we integrate the face recognition framework [26] into our tool. This framework uses the Histogram of Oriented Gradients (HoG) [27] to capture the gradient structure of a frame for object detection, e.g., faces. HoG operates by computing gradient orientations and magnitudes over a frame. Denoting a frame as $F(x, y)$, the gradients in the horizontal direction $G_x(x, y)$ and vertical direction $G_y(x, y)$ can be calculated as follows [28]:

$$G_x(x, y) = \frac{\partial F(x, y)}{\partial x}, \quad G_y(x, y) = \frac{\partial F(x, y)}{\partial y}, \tag{1}$$

where x and y are pixel positions of a frame. The gradient magnitude $M(x, y)$ and orientation $\epsilon(x, y)$ are then calculated as follows [28]:

$$\begin{aligned} M(x, y) &= \sqrt{G_x(x, y)^2 + G_y(x, y)^2}, \\ \epsilon(x, y) &= \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right). \end{aligned} \tag{2}$$

The frame is then divided into small spatial regions called cells, where the gradient orientations are quantized into bins, and a histogram of these orientations is built. The bin E_j (the value of the j-th orientation range in a histogram) is updated as follows [27]:

$$E_j = \sum_{(x,y)\in \text{cell}} M(x, y)\, \psi(\epsilon(x, y), j), \tag{3}$$

where $\psi(\epsilon, j)$ is a weighting function that distributes the gradient magnitude into adjacent bins based on linear interpolation. Each cell thus produces a histogram $\mathbf{E}^{(\text{cell})} = [E_1, E_2, \ldots, E_c]$, where c is the number of orientation bins per cell histogram. The histograms from cells are grouped into spatial regions called blocks. Suppose each block contains a cells. Then, for block b, the concatenated histogram vector is constructed by stacking the cell histograms [27]:

$$\mathbf{V}^{(b)} = [E_1^{(b)}, E_2^{(b)}, \ldots, E_{ac}^{(b)}] \in \mathbb{R}^{ac}, \tag{4}$$

where $E_k^{(b)}$ represents the k-th bin from the collection of cell histograms within block b. To improve robustness against illumination changes, block normalization is applied using the L2-norm [27]:

$$\mathbf{V}'^{(b)} = \frac{\mathbf{V}^{(b)}}{\sqrt{\|\mathbf{V}^{(b)}\|^2 + \eta^2}}, \tag{5}$$

where η is a small constant to avoid division by zero. Finally, the normalized vectors from all blocks are concatenated to form the global HoG feature descriptor [27]:

$$\mathbf{F}' = [\mathbf{V}'^{(1)}, \mathbf{V}'^{(2)}, \ldots, \mathbf{V}'^{(L)}] \in \mathbb{R}^{Lac}, \tag{6}$$

where L is the total number of blocks in the image. Finally, a Support Vector Machine (SVM) classifier is used to distinguish between face and non-face regions based on the extracted HoG features [27]. Based on the HoG feature descriptor $\mathbf{F}'$ and the SVM classifier, the facial region is located and extracted from the original input frame. Let us denote the original image frame as F_{img}. The cropped face image used for further processing is then defined as:

$$I' = \text{Crop}(F_{\text{img}}), \tag{7}$$

where $\text{Crop}(\cdot)$ is a function that extracts the detected face region from the input frame F_{img}. This image I' serves as the input to the facial augmentation process described in the next subsection. Fig. 2 presents an illustration of the face detection implemented in this tool.

2) Facial Augmentation: In vision-based object detection, increasing the amount of training data is crucial for improving model accuracy. Facial augmentation is a widely used technique that artificially expands the dataset by applying a series of transformations to the original facial images. Such transformations include flipping, rotation, scaling, cropping,

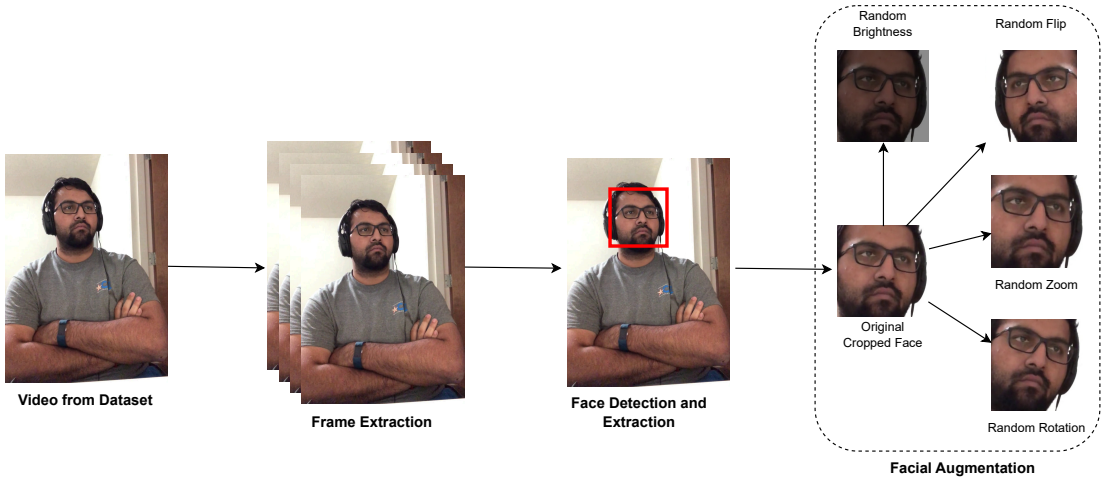


Fig. 2: Overview of the face recognition tool, which processes video through three stages: frame extraction, face detection (to isolate user faces and remove backgrounds), and facial augmentation to enhance dataset diversity.

and adjustments to brightness, contrast, or saturation.

Formally, let $I' \in \mathbb{R}^{H\times W\times C}$ denote an input image, where H, W, and C represent the height, width, and number of color channels, respectively. With Z representing the total number of different augmentation transformations applied to the original image I', the facial augmentation applies a set of transformation functions $\mathcal{T} = \{T_0, T_1, T_2, \ldots, T_Z\}$ to generate augmented images:

$$I_z = T_z(I'), \quad z = 0, 1, 2, \ldots, Z, \tag{8}$$

where each T_z represents a specific augmentation operation (e.g., rotation or flipping), and $z = 0$ means no transformation applied to the original facial image. The augmented dataset is thus composed of the original facial image along with its transformed variants:

$$\mathcal{D}_{\text{aug}} = \{I_0, I_1, I_2, \ldots, I_z\}. \tag{9}$$

By creating diverse variations of the original data, the facial augmentation helps the model generalize better across different scenarios, thereby enhancing robustness and accuracy during training. Fig. 2 illustrates the augmentation process applied to a cropped face image.

B. Proposed SSTA Architecture

The proposed SSTA architecture is described in Fig. 3. The images are processed by an **SSA** block [29]. They are then converted to vectors by a Fully Connected (FC) Layer with a linear function. After that, continuous images are aggregated into a Facial Features Table. Finally, a Temporal Component with an **LSTM** [30] is used to analyze a series of continuous images to create the output.

1) The SSA Block: Fig. 4 describes the SSA block [29], which enhances focus on key facial regions in each image, such as the eyes, mouth, and nose. First, a two-dimensional convolutional layer (Conv2D) is applied using a filter bank to extract visual features from the images. Let $\boldsymbol{I}$ denote the set of training images, where $t \in \{1, \ldots, T\}$ is the operator index, i is the image index within operator t, r is the training iteration, and $n \in \{1, \ldots, N\}$ denotes the convolutional layer index. The input image at layer n, operator t, image i, and iteration r is represented as $\boldsymbol{I}^{i,r}_{n,t}$, and the corresponding output feature map is denoted by $\boldsymbol{S}^{i,r}_{n,t}$. The output of convolutional layer n for image i at iteration r is calculated as follows [31]:

$$\boldsymbol{S}^{i,r}_{n+1,t} = \theta_{n,t}\Big(\boldsymbol{S}^{i,r}_{n,t} * \boldsymbol{B}_{n,t}\Big), \tag{10}$$

where $\theta_{n,t}$ is the activation function, $(*)$ is the convolutional operation, and $\boldsymbol{B}_{n,t}$ is the filter bank of layer n in the operator t. We then use two functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$ to transform $\boldsymbol{S}^{i,r}_{n+1,t}$ into two different feature spaces for attention computation with $\boldsymbol{f}(\boldsymbol{S}^{i,r}_{n+1,t}) = \boldsymbol{\Omega}_f \boldsymbol{S}^{i,r}_{n+1,t}$ and $\boldsymbol{g}(\boldsymbol{S}^{i,r}_{n+1,t}) = \boldsymbol{\Omega}_g \boldsymbol{S}^{i,r}_{n+1,t}$. Let P denote the total number of spatial positions in the feature map. The attention score between the q-th and p-th spatial locations is computed by taking the dot product of the corresponding feature vectors from the transformed feature spaces [32]:

$$s_{q,p,t} = \boldsymbol{f}(\boldsymbol{S}^{i,r}_{n+1,t})^T_q \, \boldsymbol{g}(\boldsymbol{S}^{i,r}_{n+1,t})_p. \tag{11}$$

The attention weight $\gamma_{q,p,t}$ is then obtained using the softmax function [29], [32]:

$$\gamma_{q,p,t} = \frac{\exp(s_{q,p,t})}{\sum_{p'=1}^{P} \exp(s_{q,p',t})}, \tag{12}$$

where $\gamma_{q,p,t}$ represents how much the model in the operator t attends to the p-th location when synthesizing the representation for the q-th location. The output of the SSA block with image i of operator t can be calculated as follows [32]:

$$\boldsymbol{S}^{i,r}_{n+2,t} = \boldsymbol{v}\left(\sum_{p=1}^{P} \gamma_{q,p,t}\, \boldsymbol{h}(\boldsymbol{S}^{i,r}_{n+1,t})_p\right), \tag{13}$$

where $\boldsymbol{h}(\cdot)$ and $\boldsymbol{v}(\cdot)$ are learnable transformations defined as $\boldsymbol{h}(\boldsymbol{S}) = \boldsymbol{\Omega}_h \boldsymbol{S}$ and $\boldsymbol{v}(\cdot) = \boldsymbol{\Omega}_v(\cdot)$. After this layer, $\boldsymbol{S}^{i,r}_{n+2,t}$ is flattened into a vector by a fully connected layer (FC), denoted as $\boldsymbol{S}^{i,r}_{n+3,t} = \text{FC}(\boldsymbol{S}^{i,r}_{n+2,t})$, where $\text{FC}(\cdot)$ is the function of the fully connected layer. The SSA and FC processes are

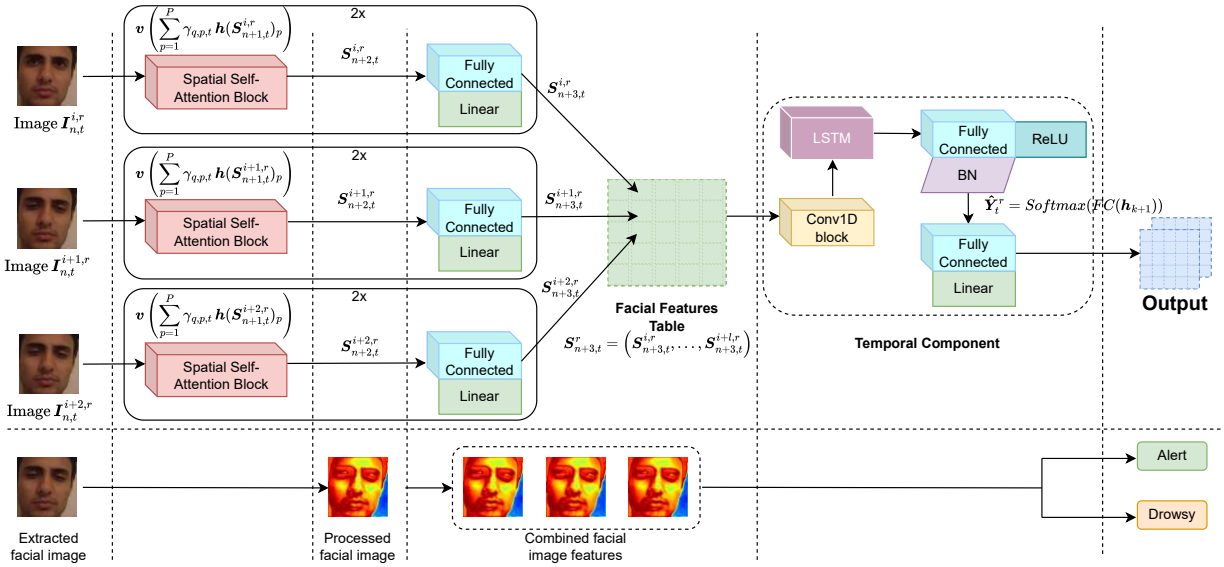


Fig. 3: The combination of CNN, SSA, and LSTM to analyze time-series frames.

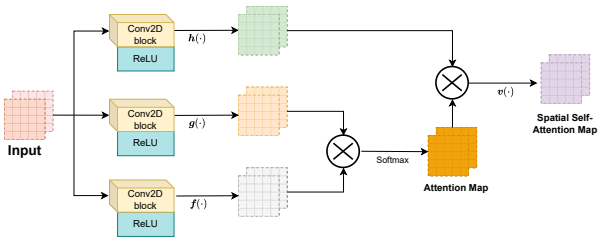


Fig. 4: An SSA Block.

applied a second time to each image and repeated across l consecutive images. Afterward, a Facial Features Table is constructed as a sequence of training samples: $\boldsymbol{S}_{n+3,t}^{r} = \left(\boldsymbol{S}_{n+3,t}^{i,r}, \boldsymbol{S}_{n+3,t}^{i+1,r}, \ldots, \boldsymbol{S}_{n+3,t}^{i+l,r}\right)$.

2) The Temporal Component with an LSTM: After the Facial Features Table is generated, a Temporal Component based on LSTM with multiple memory cells is applied to analyze the features across sequential images more effectively [33]. The LSTM used three weight functions $\boldsymbol{\Phi}_A$, $\boldsymbol{\Phi}_B$, and $\boldsymbol{\Phi}_C$. We denote $\boldsymbol{S}_{n+4,t}^{r} = \text{Conv1D}(\boldsymbol{S}_{n+3,t}^{r})$ as the output of the 1-dimensional convolution layer (Conv1D), $\sigma(\cdot)$ as the sigmoid function, $\phi(\cdot)$ as the tanh function, $\boldsymbol{h}_k$ as the previous output of the LSTM, $\boldsymbol{d}_k$ as the cell at state k of LSTM. The output of LSTM can be calculated as in the following equations [34], [35]:

$$\begin{aligned}
\boldsymbol{a}_k &= \sigma(\boldsymbol{\Phi}_{AS}\boldsymbol{S}_{n+4,t}^{r} + \boldsymbol{\Phi}_{Ah}\boldsymbol{h}_k),\\
\boldsymbol{b}_k &= \sigma(\boldsymbol{\Phi}_{BS}\boldsymbol{S}_{n+4,t}^{r} + \boldsymbol{\Phi}_{Bh}\boldsymbol{h}_k),\\
\boldsymbol{c}_k &= \phi(\boldsymbol{\Phi}_{CS}\boldsymbol{S}_{n+4,t}^{r} + \boldsymbol{\Phi}_{Ch}\boldsymbol{h}_k),\\
\boldsymbol{d}_k &= \boldsymbol{d}_{k-1} + \boldsymbol{a}_k \otimes \boldsymbol{c}_k,\\
\boldsymbol{h}_{k+1} &= \boldsymbol{b}_k \otimes \phi(\boldsymbol{d}_k),
\end{aligned} \tag{14}$$

where $\otimes$ is the element-wise multiplication of two vectors. At operator t, iteration r, we denote $\hat{\boldsymbol{Y}}_t^r$ as the final predicted output of the neural network. $\hat{\boldsymbol{Y}}_t^r$ can be calculated as follows:

$$\hat{\boldsymbol{Y}}_t^r = \text{Softmax}(\text{FC}(\boldsymbol{h}_{k+1})). \tag{15}$$

We denote $\boldsymbol{Y}_t^r$ as the label. We can calculate the loss using the categorical cross-entropy loss function as follows:

$$\mathcal{L}_t^r = -\sum_{p=1}^{M}\sum_{q=1}^{C} y_{p,q,t}^r \log(\hat{y}_{p,q,t}^r), \tag{16}$$

where M is the number of samples, C is the number of classification classes, $y_{p,q,t}^r \in \boldsymbol{Y}_t^r$, and $\hat{y}_{p,q,t}^r \in \hat{\boldsymbol{Y}}_t^r$. Using (16), we can calculate the gradient of the framework of operator t as follows:

$$\nabla\boldsymbol{\beta}_t^r = \frac{\partial\mathcal{L}_t^r}{\partial\boldsymbol{w}_t^r} = \frac{\boldsymbol{w}_t^r - \boldsymbol{w}_t^{r-1}}{\mu}, \tag{17}$$

with μ as the learning rate, $\boldsymbol{w}_t^r$ as the new weight matrix of the neural network of operator t at iteration r, $\boldsymbol{w}_t^{r-1}$ as the previous weight matrix of the neural network of operator t. Algorithm 1 summarizes this process.

Algorithm 1 Our Proposed SSTA Framework

1: **for** $f \in (i, i+l)$ **do**
2: Operator t uses SSA with functions $\boldsymbol{f}(\cdot), \boldsymbol{g}(\cdot), \boldsymbol{v}(\cdot)$ to calculate $\boldsymbol{S}_{n+2,t}^{f,r}$,
3: Operator t then uses $\text{FC}(\cdot)$ to calculate $\boldsymbol{S}_{n+3,t}^{f,r}$,
4: **end for**
5: Operator t uses the results of l image to create $\boldsymbol{S}_{n+3,t}^{r}$,
6: Operator t then uses LSTM to calculate $\boldsymbol{h}_{k+1}$ and $\hat{\boldsymbol{Y}}_t^r$,
7: Operator t then uses labels and loss function to calculate $\mathcal{L}_t^r$ and $\nabla\boldsymbol{\beta}_t^r$.

C. Federated Learning Drowsiness Detection

1) Increasing Number of Local Epochs: In federated learning, increasing the number of local training epochs in the

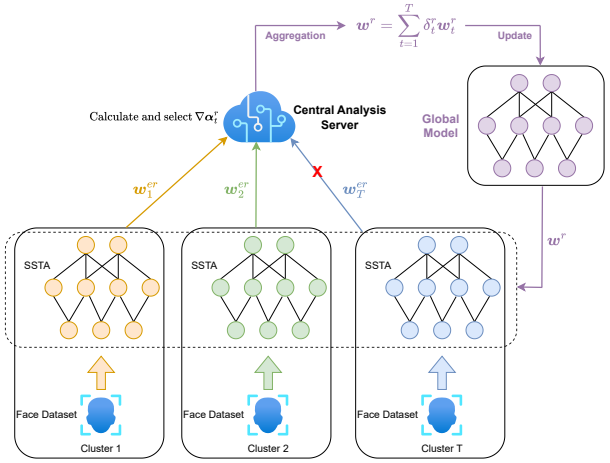


Fig. 5: FL with GSC: Each operator sends its trained model to the central analysis server, which performs model selection based on GSC to identify suitable models for aggregation. The selected models are then aggregated to form a new global model.

Algorithm 2 Our Proposed Drowsiness Detection Framework

while $r \leq$ maximum number of iterations **do**
  for $\forall t \in T$ **do**
    Operator t uses SSA and LSTM calculate $\hat{\boldsymbol{Y}}_t^r$ as in Equation (15),
    Operator t uses $\hat{\boldsymbol{Y}}_t^r$, its labels $\boldsymbol{Y}_t^r$, and categorical cross-entropy loss function to update $\boldsymbol{w}_t^{er}$,
    Operator t sends $\boldsymbol{w}_t^{er}$ to the central analysis server.
  end for
  The central analysis server uses $\boldsymbol{w}_t^{er}$ from operators to calculate $\nabla\boldsymbol{\alpha}^r$ as in Equation (18) and $\bar{\boldsymbol{p}}^r$ as in Equation (20) for each operator,
  The central analysis server calculates $\Re^r$ as in Equation (21) and δ_t^r as in Equation (22).
  The central analysis server calculates the final aggregated weight $\boldsymbol{w}^r$ as in Equation (23).
  The central analysis server sends $\boldsymbol{w}^r$ to all operators to update their neural network.
  $r = r + 1$.
end while
Operators use the optimized model to detect drowsiness.

FedAvg algorithm is essential. Running more local epochs reduces the frequency of communication between operators and the central analysis server, which helps lower the communication overhead [36] and keeps a robust convergence for the global model [37]. In this paper, we use multiple local epochs within each global round to make our proposed model more robust to data heterogeneity. We denote e as the number of local epochs, the trained model (weight) $\boldsymbol{w}_t^{er}$ is sent from operator t to the central analysis server at global iteration r, corresponding to local epochs er. In the central analysis server, the gradient $\nabla\boldsymbol{\alpha}_t^r$ can be calculated as [36]:

$$\nabla\boldsymbol{\alpha}_t^r = \nabla\boldsymbol{\beta}_t^{er} = \frac{\boldsymbol{w}_t^{er} - \boldsymbol{w}_t^{er-1}}{\mu}. \tag{18}$$

2) Gradient Similarity Comparison: After receiving trained models from operators, the central analysis server calculates a gradient $\nabla\boldsymbol{\alpha}^r$ for each operator as described in Fig. 5. It then performs GSC to select suitable gradients to aggregate. To do that, the central analysis server first calculates the pairwise GSC matrix between operators as follows:

$$\boldsymbol{P}_{t,u}^r = \frac{\nabla\boldsymbol{\alpha}_t^r \nabla\boldsymbol{\alpha}_u^r}{\|\nabla\boldsymbol{\alpha}_t^r\| \|\nabla\boldsymbol{\alpha}_u^r\|}, \quad t, u \in \{1, ..., T\}. \tag{19}$$

It then calculates the average similarity of operator t with other operators as follows:

$$\bar{\boldsymbol{p}}_t^r = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{P}_{t,u}^r. \tag{20}$$

The central analysis server then creates a threshold to define a group of operators. We denote θ as the similarity threshold to define a set of valid operators:

$$\Re^r = \{t | \bar{\boldsymbol{p}}_t^r \geq \theta\}. \tag{21}$$

We then can calculate the weights using the softmax function and the temperature parameter τ:

$$\delta_t^r = \begin{cases} \frac{\exp(\bar{\boldsymbol{p}}_t^r/\tau)}{\sum_{u \in \Re^r} \exp(\bar{\boldsymbol{p}}_u^r/\tau)} & \text{if} \quad t \in \Re^r \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

Fig. 5 describes this process. In this figure, the central analysis server does not calculate the gradient $\nabla\boldsymbol{\alpha}_t$ of operator t when the $\bar{\boldsymbol{p}}_t$ of operator t is smaller than θ ($\bar{\boldsymbol{p}}_t < \theta$). The gradient $\nabla\boldsymbol{\alpha}_t$ is ignored when $\bar{\boldsymbol{p}}_t < \theta$ because a low similarity score suggests that the operator's update is different from the others, possibly due to noise or data drift. This helps ensure that only consistent and reliable gradients are used in the global model aggregation. We then use Equation (22) to calculate the final global aggregated weight as follows:

$$\boldsymbol{w}^r = \sum_{t=1}^{T} \delta_t^r \boldsymbol{w}_t^r, \tag{23}$$

The central analysis server then sends the final global aggregated weights to all operators to update the operators' neural networks for the next iteration. This process is continuously repeated until reaching a predefined number of iterations. Once this threshold is met, the system produces a final optimized model, which can be deployed across all operators to detect drowsiness in both existing participants and new users who were not part of the initial training. This process is summarized in Algorithm 2.

IV. Experiment Setup

A. Dataset

In this paper, we use the University of Texas at Arlington Real-Life Drowsiness Dataset (UTA-RLDD) [38], [39] to evaluate the performance of our proposed framework in comparison with other state-of-the-art models. This dataset was developed by the University of Texas for multi-stage

drowsiness detection. The UTA-RDD dataset contains approximately 30 hours of RGB video recordings, totaling 180 videos from 48 healthy participants. Each participant contributed three videos, one for each of the three classes: alertness, low vigilance, and drowsiness. The videos were recorded from various angles in diverse real-world settings and backgrounds. All videos were recorded at an angle that ensured both eyes were visible, with the camera positioned within arm's length of the participant. These instructions were designed to make the recordings resemble videos captured in a car, where a phone is placed in a dashboard-mounted holder while driving. Each participant self-recorded their videos using either a smartphone or a webcam. The frame rate remained below 30 fps, aligning with the typical frame rates of consumer-grade cameras.

B. Evaluation Method

The confusion matrix [40], [41] is widely used to evaluate machine learning models and is particularly effective in assessing the performance of drowsiness detection [2]. In this context, TP, TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative, respectively. In this paper, we evaluate model performance using key metrics derived from the confusion matrix, namely accuracy, precision, and recall. The accuracy of a model is calculated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{24}$$

In addition, we use precision and recall to evaluate the performance of the models. Given B as the number of classification groups (i.e., alert and drowsy), the precision is calculated as follows:

$$\text{Precision} = \sum_{b=1}^{B} \frac{\text{TP}_b}{\text{TP}_b + \text{FP}_b}. \tag{25}$$

Similarly, the recall of the system can be calculated as follows:

$$\text{Recall} = \sum_{b=1}^{B} \frac{\text{TP}_b}{\text{TP}_b + \text{FN}_b}. \tag{26}$$

C. Simulation Setup

As described above, the UTA-RLDD dataset includes 48 participants, each participant is categorized into 3 classes corresponding to drowsiness levels: alertness, low vigilance, and drowsiness. To demonstrate that our proposed model can efficiently work with different individuals without any prior training, we divide the participants into training and testing datasets. The training dataset includes 42 participants. The testing dataset consists of 12 participants, including 6 from the training dataset and 6 who are not included in the training data. These 6 unseen participants are used to evaluate the model's performance on individuals it has not encountered before. The training dataset is split with 80% of the data for training and 20% for validation. The validation data is used to evaluate the accuracy and the convergence of the training process. In our experiment, our preprocessing tool first extracts frames from the training videos. After that, it performs face detection and extraction, generating a dataset containing 292,220 frames. Due to the high computational cost and time required to process the entire dataset, we randomly use 50% of the generated frames to perform experiments.

We consider two different scenarios, including centralized and federated learning. The centralized learning scenario is considered as the benmarks for our proposed FL model. In this scenario, all participants are used to train the machine learning models to evaluate performance. In contrast, in federated learning scenarios, the participants are grouped into different operators. Each operator includes a distinct set of participants, i.e., in the case of five operators, each operator consists of eight participants, whereas in the case of 42 operators, each operator contains a single participant. We perform experiments using five workstations running the Ubuntu operating system, each equipped with a GPU. The setup includes two NVIDIA GeForce RTX 3090, two NVIDIA RTX 6000 Ada Generation, three NVIDIA A100, and one NVIDIA GeForce RTX 4090 graphics cards, using the PyTorch framework for computational tasks.

V. Performance Evaluation

We compare our proposed framework with other state-of-the-art models to demonstrate the outperformance of our proposed framework. We consider two different scenarios, including the centralized and federated learning models.

A. Centralized Learning Evaluation

In the centralized learning scenario, we consider a scenario in which a centralized server can be used to collect all training datasets from the operator to perform a centralized training process. This serves as a baseline to compare our proposed approach with other state-of-the-art machine learning techniques. In particular, we evaluate the performance of our proposed SSTA model in comparison with other state-of-the-art framework such as Vision Transformer (ViT) [42], LSTM [43], Convolutional Neural Network (CNN) [44], Multilayer Perceptron (MLP) [45], Decision Tree (DT) [13], K-Nearest Neighbor (KNN) [13], Linear Regression (LR) [13], Random Forest (RF) [13], Support Vector Machine (SVM) [13], and Extreme Gradient Boosting (XGBoost) [13].

1) Visualization of SSA Patterns: Fig. 6 shows the output of the CNN baseline and our proposed model after the SSA blocks at epoch 5 (early stage of training) and epoch 200 (late stage of training), across 12 participants from the testing dataset. To ensure diverse testing conditions, we select participants with varying characteristics, such as participant 13 wears glasses, participant 25 whose videos are captured at a large rotation angle, participant 12 covers his mouth when feeling drowsy, participants 5 and 11 are in low-light environments while in a drowsy state, and participants 12 and 22 use headphones. Compared to the CNN, our model produces stronger activations, represented by orange to red colors, around key facial areas such as the eyes and mouth at both training stages. These visualizations indicate that our model can be more effective in focusing on important facial regions (eyes, nose, and mouth), enhancing its ability to detect signs of drowsiness throughout the training process.

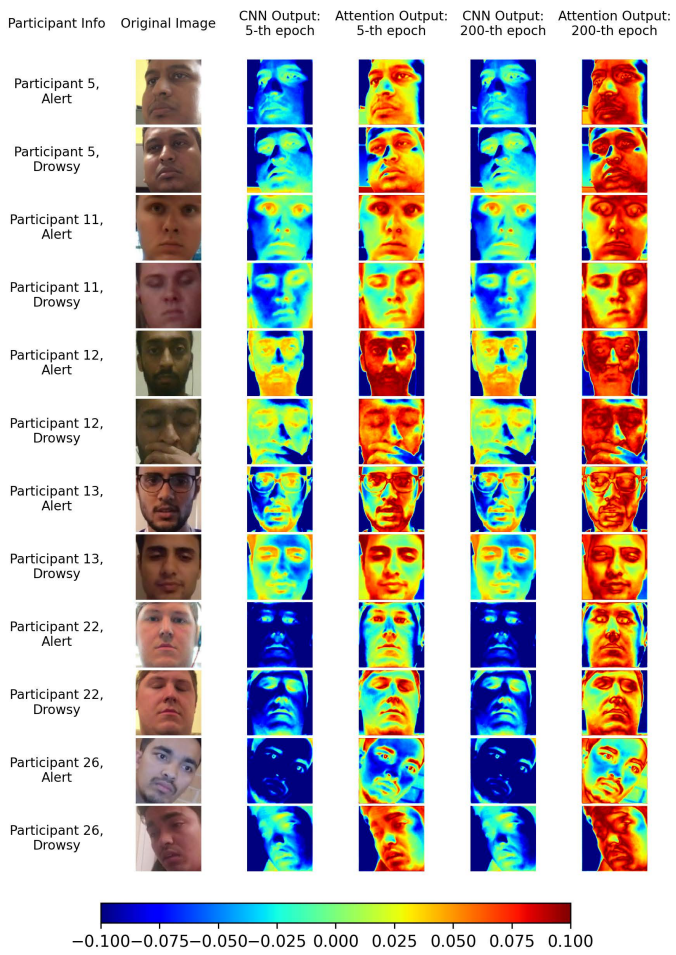


Fig. 6: Visualization of the SSA mechanism's output across selected participants in the testing dataset.

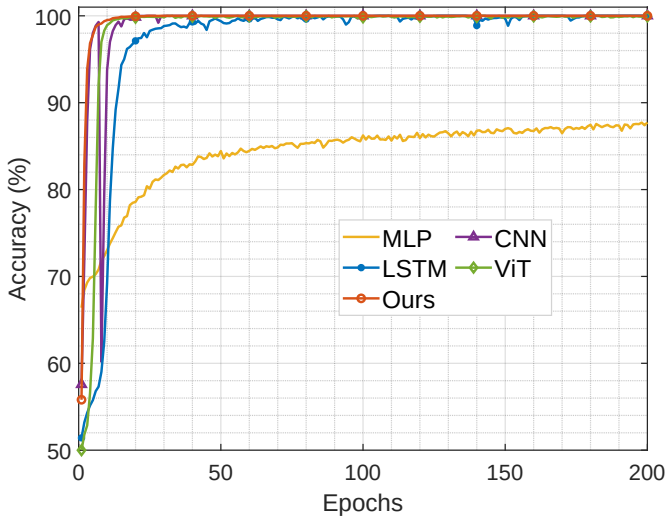


Fig. 7: Training accuracies of different centralized learning approaches evaluated on participants from the training dataset.

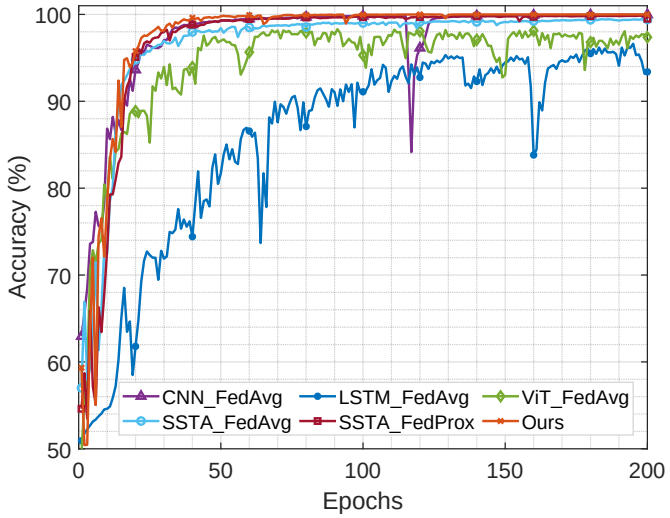


Fig. 8: The accuracies of different FL models in the case of 5 operators during the training process.

2) Model Accuracy and Convergence Comparison: Fig. 7 describes the convergence and accuracy of the training process for our proposed SSTA model compared to other DL models. As observed, both our proposed model and the ViT achieve convergence within the first 20 epochs. Notably, during the training process, our proposed SSTA model converges more rapidly and attains the highest accuracy (100%) compared to the others. Although both CNN and ViT models achieve near-perfect accuracy (100%) after 20 epochs, the CNN experiences a significant drop in accuracy around epoch 10, while the MLP stabilizes at approximately 80% accuracy after 100 epochs of the training process. Table I shows the performance results of our proposed SSTA model in terms of accuracy, precision, and recall, compared to other models on the testing dataset, which includes 6 previously untrained and 6 other trained participants. As shown in the table, traditional machine learning models, i.e., DT, KNN, LR, RF, SVM, XGBoost, exhibit relatively poor performance, achieving accuracies ranging from approximately 60% to 77% on the testing dataset. MLP also demonstrates suboptimal performance, with an accuracy of 71.3%. In contrast, deep learning models such as CNN, LSTM, and ViT perform significantly better, achieving accuracies of 83.92%, 90.38%, and 89.32%, respectively. Notably, with the testing dataset, our proposed SSTA model (Ours) outperforms all other models by getting the highest accuracy, precision, and recall of 91.83%, 91.96%, and 91.83%, respectively.

B. Federated Learning Evaluation

In this section, we consider a decentralized federated learning setting in which participants are organized into operators for the training process. Each operator may contain one or more participants. To evaluate the impact of clustering, we perform experiments with 5, 10, and 42 operators. A total of 42 participants in the training dataset are randomly and equally assigned to the operators. This results in 8 participants per operator for 5 operators, 4 participants per operator for 10 operators, and 1 participant per operator when 42 operators are used.

1) Comparative Analysis of Federated Strategies: Fig. 8 shows the accuracy curves during the training process of

TABLE I: Centralized mode: Performance comparison on testing dataset, including 6 trained and 6 untrained participants.

	DT [13]	**KNN** [13]	**LR** [13]	**RF** [13]	**SVM** [13]	**XGBoost** [13]	**MLP** [45]	**CNN** [44]	**LSTM** [43]	**ViT** [42]	**Ours**
Accuracy	71.1254	77.8734	61.2936	77.9235	64.5475	72.6672	71.3056	83.9277	90.3805	89.3219	**91.8341**
Precision	71.3131	77.8845	61.4571	77.9756	64.9342	72.6910	71.3168	83.9819	90.3826	89.3913	**91.9607**
Recall	71.1254	77.8734	61.2936	77.9235	64.5475	72.6672	71.3056	83.9277	90.3805	89.3219	**91.8341**

TABLE II: FL mode with 5 operators: Performance comparison on testing dataset, including 6 trained and 6 untrained participants.

	CNN_FedAvg	**LSTM_FedAvg**	**ViT_FedAvg**	**SSTA_FedAvg**	**SSTA_FedProx** [37]	**Ours**
Accuracy	76.5521	80.8601	81.2739	74.9849	85.1201	**89.9253**
Precision	77.5237	82.5897	81.2739	76.4987	85.1376	**90.6377**
Recall	76.5521	80.8601	81.2739	74.9849	85.1201	**89.9253**

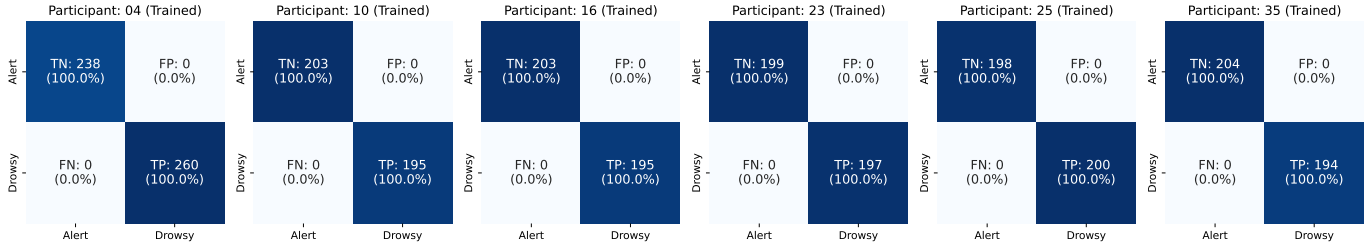
different combinations of deep learning and federated learning approaches with 5 operators. The approaches include CNN-FedAvg, LSTM-FedAvg, ViT-FedAvg, as well as SSTA-FedAvg, SSTA-FedProx, and our proposed approach: the complete SSTA with GSC (SSTA-GS). All approaches eventually reach convergence; however, LSTM-FedAvg and ViT-FedAvg show noticeable fluctuations, indicating their instability during the training process. Although CNN-FedAvg appears to converge, it experiences a significant drop in accuracy around epoch 118, raising concerns about its reliability. In contrast, our proposed SSTA-based approaches, when combined with FedAvg, FedProx, or GSC, achieve more stable and consistent convergence, suggesting improved training performance in the 5 operators setting.

Table II presents the performance of the different approaches on the testing dataset. Our proposed approach achieves the highest results, with an accuracy of 89.92%, precision of 90.63%, and recall of 89.92%. These results are approximately 4-5% higher than those of SSTA_FedProx across all three metrics and outperform the remaining approaches. These results show that our model is more effective at learning relevant patterns from the heterogeneous data, suggesting its strong potential for real-world applications in federated learning settings. Fig. 9 shows the classification results of our proposed model using federated learning with 5 operators on the testing dataset. Our model is 100% accurate in detecting drowsiness among participants who are part of the training dataset. The model also performs well in classifying previously unseen participants, detecting drowsiness with 100% accuracy in three participants and more than 90% accuracy in two participants. The model, however, achieves a lower accuracy of 73% in detecting drowsiness in the remaining participant, which may be due to differences in data patterns that are not well represented during training. This highlights the need for more diverse training data or personalized adjustments to improve performance for all users.

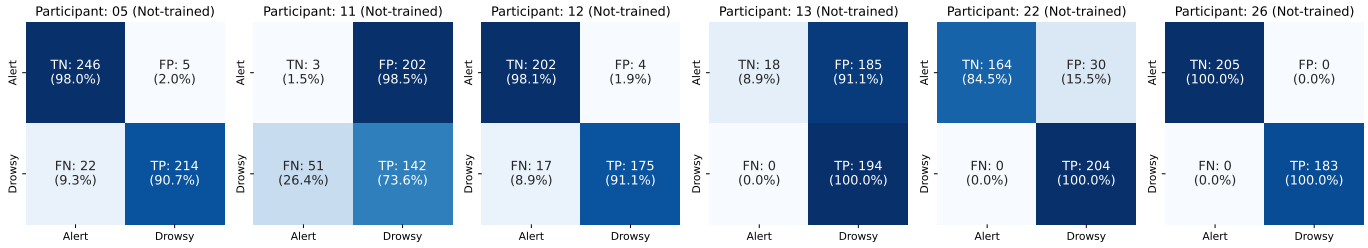
Fig. 10 shows the accuracy of different federated learning algorithms using the SSTA model during the training process. We compare three approaches: FedAvg, FedProx, and our proposed approach under a scenario with 10 operators. When the number of operators increases from 5 to 10, data heterogeneity also increases, posing significant challenges for federated learning. This increased heterogeneity directly affects the stability and convergence of the training process, as reflected in the accuracy trends over time. In particular, the FedAvg method shows considerable fluctuations in accuracy throughout training, indicating its instability when handling non-IID data across multiple operators. Besides, FedProx, designed to address some of these issues, achieves a more stable performance than FedAvg but still experiences a sharp drop in accuracy around epoch 85, highlighting its limitations under high heterogeneity. In contrast, our proposed approach consistently outperforms both baselines in terms of accuracy and stability. The training curve of our proposed model is smooth and stable, demonstrating strong convergence behavior even under challenging conditions. This result highlights the robustness of our method in handling heterogeneous data distributions and maintaining reliable performance across a larger number of operators. These findings confirm that our approach offers a more effective and resilient solution for federated learning in realistic, non-IID environments.

2) Robustness to Varying Data Heterogeneity: As mentioned in the previous section, increasing the number of operators also increases data heterogeneity. Fig. 11 illustrates the accuracy curves when using different numbers of operators with our proposed model during the training process. We observe that FL with 5 operators converges the fastest, followed by 10 operators. FL with 5 operators reaches 100% accuracy after approximately 50 epochs. In comparison, FL with 10 operators takes around 100 epochs to converge, while FL with 42 operators (i.e., one participant per operator) requires about 160 epochs. This demonstrates the increased complexity of training under highly heterogeneous data conditions. While increasing the number of operators leads to greater data heterogeneity, our model remains effective in handling these variations, achieving over 80% training accuracy after 160 epochs, despite a slower convergence rate in comparison with other cases.

Table III presents the performance in terms of accuracy, precision, and recall for different operator settings with our proposed model on the testing dataset. FL with 5 operators achieves the highest performance, with an accuracy of 89.92%, precision of 90.63%, and recall of 89.92%. Even in the most heterogeneous case with 42 operators, the model still performs well, achieving 83% accuracy, 84.47% precision, and 83.02%



(a) Trained participants



(b) Not-trained participants

Fig. 9: Classification report of FL with 5 operators on testing dataset (6 trained and 6 untrained participants).

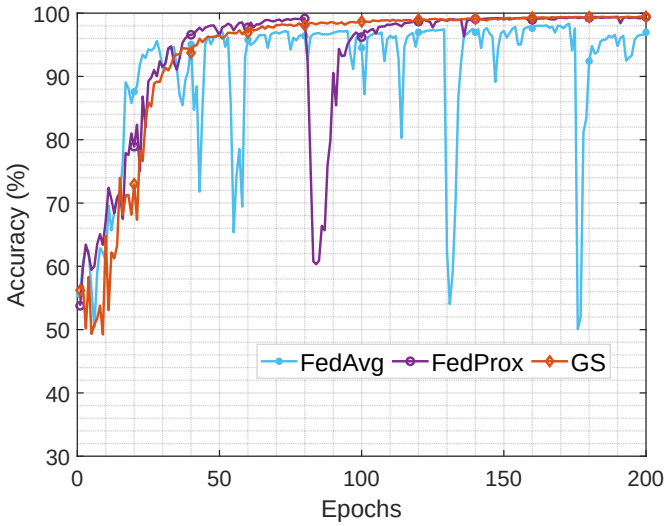


Fig. 10: The stability during the training process of different aggregation schemes on our SSTA network with 10 operators.

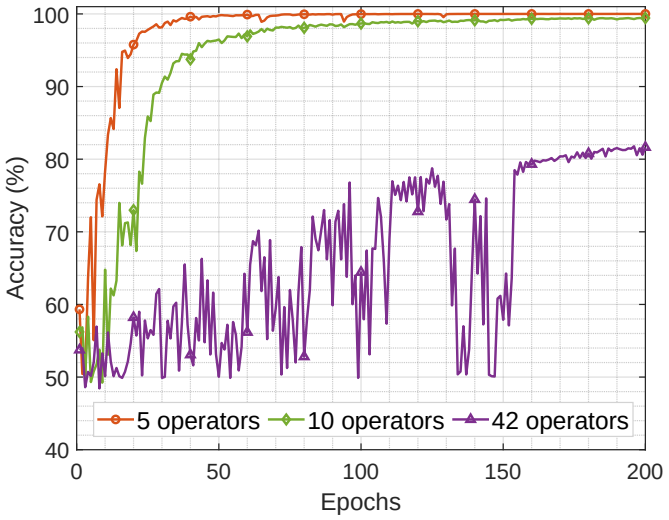


Fig. 11: The accuracies of our proposed model in the training process when the number of operators varies.

TABLE III: The stabilities of accuracies in the testing dataset when the number of operators increases.

	5 operators	**10 operators**	**42 operators**
Accuracy	**89.9253**	88.6332	83.0204
Precision	**90.6377**	88.6692	84.4740
Recall	**89.9253**	88.6332	83.0204

recall. These results highlight the robustness of our proposed model across varying levels of data heterogeneity. The model's strong and consistent performance demonstrates its suitability for real-world federated learning scenarios. Moreover, the relatively small performance drop in testing accuracy in highly heterogeneous conditions suggests that the model effectively mitigates the adverse effects of non-IID data distributions, which is a common challenge in practical FL applications.

3) Impact of Local Epochs on FL Performance: The number of local epochs plays a critical role in determining the accuracy and convergence behavior of federated learning (FL), particularly in decentralized environments with heterogeneous data distributions [37]. Increasing the number of local epochs allows operators to perform more local updates before communicating with the central analysis server, which can be beneficial in reducing communication overhead and improving model personalization [36]. However, when local datasets are highly heterogeneous, excessively increasing the number of local epochs can lead to overfitting on local data and cause operator models to converge to local optima. This divergence in local updates may negatively impact the global model's

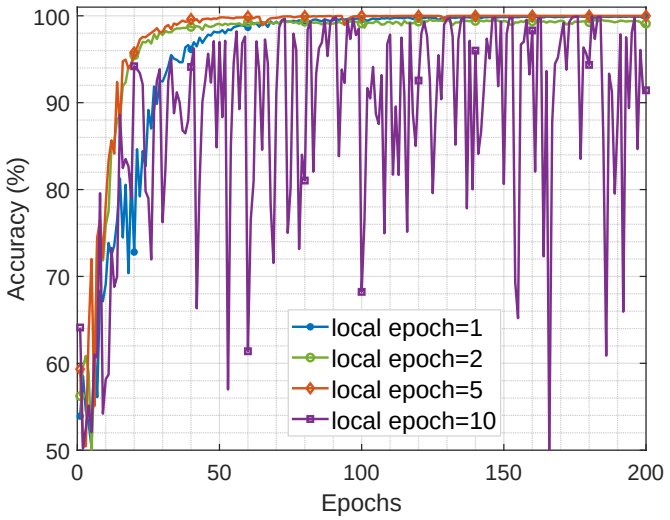


Fig. 12: The accuracies of the proposed SSTA model in the training process while varying the number of local epochs.

ability to converge effectively. Nevertheless, prior studies have shown that selecting an appropriate number of local epochs can improve overall model performance while maintaining stable convergence [37].

In this section, we evaluate the impact of varying the number of local epochs on the stability of the training process. Fig. 12 illustrates the effects of different local epoch settings (1, 2, 5, and 10) on training performance in a federated learning scenario. The curve corresponding to 1 local epoch demonstrates slower convergence, likely due to insufficient local training that leads to high communication frequency with limited improvement per round. In contrast, the model with 5 local epochs converges significantly faster and achieves superior performance, indicating an effective balance between local update depth and global model alignment. On the other hand, using 10 local epochs introduces instability and noticeable fluctuations in the learning curve, suggesting overfitting to local data and difficulty in achieving coherent global aggregation.

Based on these empirical results, we adopt 5 local epochs as the optimal setting for all experiments in this study. This choice offers a practical trade-off between communication efficiency, model convergence speed, and global accuracy in the presence of non-IID data.

VI. Conclusion

In this paper, we developed a novel framework for driver drowsiness detection in decentralized environments with heterogeneous facial data. To improve detection accuracy, we combined an SSA mechanism with an LSTM network, helping the model focus on important facial features across different individuals. Moreover, we integrated the GSC into our model to support federated learning, allowing the system to select and combine models from similar client clusters. This improves both the accuracy and robustness of the final global model. Additionally, we built a preprocessing tool that can perform frame extraction from videos, face detection and extraction, and frame augmentation to enhance the data quality of the dataset. Extensive simulations demonstrate that our approach outperforms existing methods in both accuracy and computational efficiency. Furthermore, by enabling decentralized learning without compromising performance, our framework enhances data privacy, making it well-suited for real-world applications in intelligent transportation systems.

References

[1] M. Thomas, C. Gupta, M. Sprajcer, D. Demasi, E. Sach, G. Roach, C. Sargent, D. Dawson, and S. Ferguson, "Fatigue and driving: An international review," Available at: https://www.aaa.asn.au/wp-content/uploads/2021/10/Fatigue-Driving-Literature-Review-FINAL.pdf, 2021, [Online; accessed 31 October 2024].

[2] B. Fu, F. Boutros, C.-T. Lin, and N. Damer, "A survey on drowsiness detection: Modern applications and methods," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 11, pp. 7279–7300, Nov. 2024.

[3] X. Qin, Y. Niu, H. Zhou, X. Li, W. Jia, and Y. Zheng, "Driver drowsiness eeg detection based on tree federated learning and interpretable network," *International Journal of Neural Systems*, vol. 33, no. 3, p. 2350009, Mar. 2023.

[4] K. Fujiwara, H. Iwamoto, K. Hori, and M. Kano, "Driver drowsiness detection using rr interval of electrocardiogram and self-attention autoencoder," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2956–2965, Aug. 2023.

[5] I. Garcia, S. Bronte, L. M. Bergasa, J. Almazán, and J. Yebes, "Vision-based drowsiness detector for real driving conditions," in *IEEE Intelligent Vehicles Symposium (IV)*, June 2012, pp. 618–623.

[6] D. Tran, J. Du, W. Sheng, D. Osipychev, Y. Sun, and H. Bai, "A human-vehicle collaborative driving framework for driver assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3470–3485, Dec. 2018.

[7] Z. Zhang, H. Ning, and F. Zhou, "A systematic survey of driving fatigue monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 19 999–20 020, Nov. 2022.

[8] E. Perkins, C. Sitaula, M. Burke, and F. Marzbanrad, "Challenges of driver drowsiness prediction: The remaining steps to implementation," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1319–1338, Nov. 2022.

[9] A. Singh and R. Thakur, "A cognitive-intelligence-based personalized federated approach for monitoring driver behavior," *IEEE Internet of Things Journal*, vol. 12, no. 14, pp. 26 116–26 127, July 2025.

[10] V. P. Chellapandi, L. Yuan, C. G. Brinton, S. H. Żak, and Z. Wang, "Federated learning for connected and automated vehicles: A survey of existing approaches and challenges," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 119–137, Nov. 2023.

[11] Z. Chen, S. Yu, F. Chen, F. Wang, X. Liu, and R. H. Deng, "Lightweight privacy-preserving cross-cluster federated learning with heterogeneous data," *IEEE Transactions on Information Forensics and Security*, vol. 19, pp. 7404–7419, July 2024.

[12] M. E. Shaik, "A systematic review on detection and prediction of driver drowsiness," *Transportation Research Interdisciplinary Perspectives*, vol. 21, p. 100864, Sep. 2023.

[13] S. Mittal, S. Gupta, A. Shamma, I. Sahni, N. Thakur *et al.*, "Driver drowsiness detection using machine learning and image processing," in *International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, Sep. 2021, pp. 1–8.

[14] L. Mou, C. Zhou, P. Xie, P. Zhao, R. Jain, W. Gao, and B. Yin, "Isotropic self-supervised learning for driver drowsiness detection with attention-based multimodal fusion," *IEEE Transactions on Multimedia*, vol. 25, pp. 529–542, Nov. 2021.

[15] N. N. Pandey and N. B. Muppalaneni, "Dumodds: Dual modeling approach for drowsiness detection based on spatial and spatio-temporal features," *Engineering Applications of Artificial Intelligence*, vol. 119, p. 105759, Mar. 2023.

[16] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, Apr. 2018.

[17] S. Usmani, B. Chandwani, and D. Sadhya, "Driver drowsiness detection using vision transformer," in *International Conference on Computer Vision and Image Processing (CVIP)*, Nov. 2023, pp. 445–454.

[18] G. S. Krishna, K. Supriya, J. Vardhan *et al.*, "Vision transformers and yolov5 based driver drowsiness detection framework," *arXiv preprint arXiv:2209.01401*, Sep. 2022.

[19] C. Zhao, Z. Gao, Q. Wang, K. Xiao, Z. Mo, and M. J. Deen, "Fedsup: A communication-efficient federated learning fatigue driving behaviors supervision approach," *Future Generation Computer Systems*, vol. 138, pp. 52–60, Jan. 2023.

[20] J. Cui, Z. Lan, O. Sourina, and W. Müller-Wittig, "Eeg-based cross-subject driver drowsiness recognition with an interpretable convolutional neural network," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7921–7933, Oct. 2023.

[21] L. Zhang, H. Saito, L. Yang, and J. Wu, "Privacy-preserving federated transfer learning for driver drowsiness detection," *IEEE Access*, vol. 10, pp. 80 565–80 574, June 2022.

[22] Y. Albadawi, A. AlRedhaei, and M. Takruri, "Real-time machine learning-based driver drowsiness detection using visual features," *Journal of Imaging*, vol. 9, no. 5, Apr. 2023. [Online]. Available: https://www.mdpi.com/2313-433X/9/5/91

[23] C. Yang, Z. Yang, W. Li, and J. See, "Fatigueview: A multi-camera video dataset for vision-based drowsiness detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 1, pp. 233–246, 2023.

[24] Tesla Inc., "Tesla ai day 2021," 2021, https://www.tesla.com/AI.

[25] Mark Harris, "Tesla's autopilot depends on a deluge of data," 2022, https://spectrum.ieee.org/tesla-autopilot-data-deluge.

[26] Adam Geitgey, "Face Recognition Framework." [Online]. Available: https://pypi.org/project/face-recognition/

[27] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005, pp. 886–893.

[28] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2002.

[29] I. Kim, W. Baek, and S. Kim, "Spatially attentive output layer for image classification," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020, pp. 9533–9542.

[30] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[31] Y. M. Saputra, D. Nguyen, H. T. Dinh, Q.-V. Pham, E. Dutkiewicz, and W.-J. Hwang, "Federated learning framework with straggling mitigation and privacy-awareness for AI-based mobile application services," *IEEE Transactions on Mobile Computing*, vol. 22, no. 9, pp. 5296–5312, Sep. 2023.

[32] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International Conference on Machine Learning (ICML)*, June 2019, pp. 7354–7363.

[33] K. Xia, J. Huang, and H. Wang, "Lstm-cnn architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56 855–56 866, Mar. 2020.

[34] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: Lstm cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, July 2019.

[35] A. Graves, *Long Short-Term Memory*. Springer Berlin Heidelberg, 2012, pp. 37–45.

[36] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, Apr. 2017, pp. 1273–1282.

[37] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *Conference on Machine Learning and Systems (MLSys)*, Mar. 2020.

[38] R. Ghoddoosian, M. Galib, and V. Athitsos, "A realistic dataset and baseline temporal model for early drowsiness detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2019, pp. 178–187.

[39] "Uta real-life drowsiness dataset," https://sites.google.com/view/utarldd/home.

[40] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, June 2006.

[41] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, pp. 37–63, Oct. 2011.

[42] M. M. Bin Mohamad Azmi and F. H. Kamaru Zaman, "Driver drowsiness detection using vision transformer," in *IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, May 2024, pp. 329–336.

[43] P. Liu, H.-L. Chi, X. Li, and J. Guo, "Effects of dataset characteristics on the performance of fatigue detection for crane operators using hybrid deep neural networks," *Automation in Construction*, vol. 132, p. 103901, Dec. 2021.

[44] R. Tamanani, R. Muresan, and A. Al-Dweik, "Estimation of driver vigilance status using real-time facial expression and deep learning," *IEEE Sensors Letters*, vol. 5, no. 5, pp. 1–4, May 2021.

[45] M. Mohammadi, R. Allocca, D. Eklund, R. Shrestha, and S. Sinaei, "Privacy-preserving federated learning system for fatigue detection," in *IEEE International Conference on Cyber Security and Resilience (CSR)*, Aug. 2023, pp. 624–629.