Efficient Federated Learning with Spike Neural Networks for Traffic Sign Recognition

Kan Xie, Zhe Zhang, Bo Li, Jiawen Kang, Dusit Niyato, *Fellow, IEEE*, Shengli Xie, *Fellow, IEEE*, Yi Wu

Abstract—With the gradual popularization of self-driving, it is becoming increasingly important for vehicles to smartly make the right driving decisions and autonomously obey traffic rules by correctly recognizing traffic signs. However, for machine learning-based traffic sign recognition on the Internet of Vehicles (IoV), a large amount of traffic sign data from distributed vehicles is needed to be gathered in a centralized server for model training, which brings serious privacy leakage risk because of traffic sign data containing lots of location privacy information. To address this issue, we first exploit privacy-preserving federated learning to perform collaborative training for accurate recognition models without sharing raw traffic sign data. Nevertheless, due to the limited computing and energy resources of most devices, it is hard for vehicles to continuously undertake complex artificial intelligence tasks. Therefore, we introduce powerful Spike Neural Networks (SNNs) into traffic sign recognition for energy-efficient and fast model training, which is the next generation of neural networks and is practical and well-fitted to IoV scenarios. Furthermore, we design a novel encoding scheme for SNNs based on neuron receptive fields to extract information from the pixel and spatial dimensions of traffic signs to achieve high-accuracy training. Numerical results indicate that the proposed federated SNN outperforms traditional federated convolutional neural networks in terms of accuracy, noise immunity, and energy efficiency as well.

Index Terms—Federated learning, spike neural networks, traffic sign recognition, Internet of vehicles

I. INTRODUCTION

According to the forecast report from well-known analysis organization Gartner, by 2023, there are more than 740,000 autonomous-ready vehicles added all around the world [1]. It is obvious that autonomous driving is becoming more and more popular and shows huge potential to realize high-level intelligent transportation [2]. When driving on the roads, traffic signs are detected and recognized by vehicles with powerful sensors and cameras. These signs play important roles to guide vehicles in an orderly manner and also improving driving safety, which provides significant assistance for vehicle control without interruption [3]. For example, warning signs remind vehicles to avoid obstacles and be aware of dangerous bends. The indicator signs can help the vehicles to perform pre-processing control and to follow road instructions, e.g, the speed limit on the highway. Therefore, the correct recognition and precise application of traffic signs can provide more perfect assistance for driver assistance systems and even autonomous driving [4].

Traditionally, to achieve precious and intelligent traffic sign recognition, the traffic signs are collected by vehicles and then gathered in a centralized server for machine learning-based analysis and recognition [5]. However, these existing schemes have overlooked the privacy concerns of vehicles (e.g., some sensitive location information mapping to specific traffic signs on a particular location), which hinders data gathering and the training collaboration among vehicles [6]. Motivated by the advantages of privacy protection and breaking the data barrier, Federated Learning (FL) has been introduced into various vehicular applications [7], [8] and IoV scenarios [9], [10]. Here, federated learning is a promising privacy-preserving machine learning paradigm that allows numerous vehicles to train a globally shared recognition model and only share a local training model without revealing individual raw data (i.e., collected traffic sign pictures) [11], [12].

For traffic sign recognition in IoV, due to limited energy resources and intermittent communications of vehicles, the energy efficiency, scalability, and robustness of model training need to be further enhanced for wide deployment [13]. However, traditional FL methods pose challenges to the energy overhead and scalability with the rapid increase of vehicles in IoV. In addition, there exist blurred sensing traffic sign pictures caused by high-speed mobility of vehicles or bad weather conditions, it is increasingly important to improve the robustness of traffic sign recognition in IoV. All these

Kan Xie is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Key Laboratory of Intelligent Information Processing and System Integration of IoT, Ministry of Education, Guangzhou 510006, China (e-mail: kxie@gdut.edu.cn).

Bo Li is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong-HongKong-Macao Joint Laboratory for Smart Discrete Manufacturing, Guangzhou 510006, China (e-mail: libo999@gdut.edu.cn).

Jiawen Kang is with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the 111 Center for Intelligent Batch Manufacturing Based on IoT Technology, Guangzhou 510006, China (e-mail: kavinkang@gdut.edu.cn).

Shengli Xie is with School of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the Guangdong Key Laboratory of IoT Information Technology, Guangzhou 510006, China (e-mail: shlxie@gdut.edu.cn).

Z. Zhang and Y. Wu are with the School of Data Science and Technology, Heilongjiang University, Harbin 150080, China, and Y. Wu is also with the Institute for Cryptology & Network Security, Heilongjiang University, Harbin 150080, China (e-mail: 2202517@s.hlju.edu.cn, 1995050@hlju.edu.cn).

D. Niyato is with School of Computer Science and Engineering, Nanyang Technological University (e-mail: DNIYATO@ntu.edu.sg).

issues inspire us to propose a new FL framework in this paper, where the emerging Spike Neural Network (SNN) technology is naturally integrated into IoV systems. The SNNs, which are 3rd generation artificial neural networks inspired by information processing in biology, have been introduced to achieve efficient deep neural network implementation with favorable properties including low power consumption, fast inference, good robustness, and event-driven information processing [14]. Therefore, SNNs are a promising choice and highly suitable for training many machine learning tasks in IoV, especially traffic sign recognition [15]. To this end, as shown in Fig. 1, we design an efficient SNN-empowered federated learning framework for low-power and privacy-preserving traffic sign recognition. In each communication round, the networked vehicle uses the local traffic sign dataset to train the SNN model, which is then uploaded to a nearby Road Side Unit (RSU) for aggregation to obtain the global model of the next round. However, this framework still faces a challenge in encoding traffic sign images for efficient SNNs.

In SNNs, the training data needs to be encoded as a set of spike sequences with time steps. Rate-based encoding [16] has been widely used such as in [17]–[19], which utilizes the pixel intensities of images to extract input features. The greater the intensity of a single pixel of an image, the more informative the spike sequence it encodes. This method generally requires very high pixel values to be encoded into a sequence of high-frequency spikes for quick forward propagation in SNNs. Thus, rate-based coding has limitations when facing blurry traffic sign images captured by vehicles in bad weather conditions. Inspired by the biological vision, we find that the degree of stimulation of an image on organisms is not only affected by the color depth of the pixels but also related to the different areas observed by the biological retina. To this end, we propose a novel encoding method for SNNs called Neuronal Receptive Field Encoding (NRFE), which also extracts features from the spatial dimension to achieve high-precision traffic sign recognition. More specifically, the main contributions of this paper are summarized as follows.

- We propose a new federated learning framework with spike neural networks for efficient and distributed traffic sign recognition. The spike neural networks are particularly suitable and practical for federated learning in IoV because of the advantages of low power consumption, fast inference, and event-driven information processing.
- We propose a new encoding scheme for SNNs based on the neuronal receptive field to extract features from pixels and different regions of traffic signs. Meanwhile, we design a Gaussian receptive field matrix to simulate the area detected by biological neurons.
- We perform extensive experiments to show the performance of adaptability, robustness, and energy efficiency of the federated SNN algorithm on a real traffic sign recognition dataset by comparing it with baseline schemes. Numerical results demonstrate the advantages of the proposed algorithm regarding noise resistance and energy efficiency.

The rest of this paper is organized as follows. Section II introduces related work and Section III presents the methodology of FL and SNN. Section IV describes the proposed federated SNN framework for traffic sign recognition. Numerical results are presented in Section V and conclusions are drawn in Section VI.

II. RELATED WORK

A. Autonomous Driving Approaches for Internet of Vehicles

With advanced embedded devices and communication technologies, more and more intelligent vehicle applications have emerged, such as automatic driving, emergency collision avoidance, and target tracking. In particular, vehicle automatic driving applications can display warning signs on the dashboard, actively sense information about the surrounding environment outside the driver's field of vision and, effectively provide target detection and collision avoidance assistance. For instance, autonomous driving applications require vehicles to quickly and accurately recognize traffic signs on the roadside. The authors in [20] proposed a road sign intelligence system based on Support Vector Machines (SVMs) for detecting and recognizing road signs. The authors in [21] adopted a Region-Convolutional Neural Network (R-CNN) approach for large-scale detection and recognition of traffic signs through automatic end-to-end learning. The authors in [22] added an attention network to the Fast R-CNN model to extract latent features of traffic signs. The authors in [23] proposed a traffic sign recognition system based on a space transformer, and quantitatively compared different adaptive and non-adaptive gradient descent methods.

Although these methods improve the perception and navigation strategies of autonomous driving in smart cars, they still face two severe challenges: data privacy and limited computing resources.

B. Federated Learning for Internet of Vehicles

In IoV, vehicles normally generate a large amount of data through advanced sensors and vehicular applications containing abundant user privacy information, e.g., location information, user preference, and driving trajectory [24]. With the help of powerful Machine Learning (ML) methods, these data are analyzed and utilized to achieve intelligent transportation systems [25]. Traditional data analysis using centralized ML approaches has high risks of user privacy leakage [11]. As a distributed privacy-preserving machine learning approach, FL has attracted increasing attention and become an important method to achieve multi-vehicle communication with privacy protection in IoV scenarios. Researchers have introduced federated learning into autonomous driving, in which vehicles locally train models and an RSU aggregate trained model parameters for AI-based decisions in IoV. Such as the authors in [7] designed a federated learning-based gated recurrent unit neural network algorithm for efficient traffic flow prediction in IoV. The authors in [26] proposed a Differential Privacy-based Asynchronous Federated Learning (DPAFL) scheme for resource sharing and secure local model updates in IoV. The authors in [27] designed an FL-based vehicle cooperative
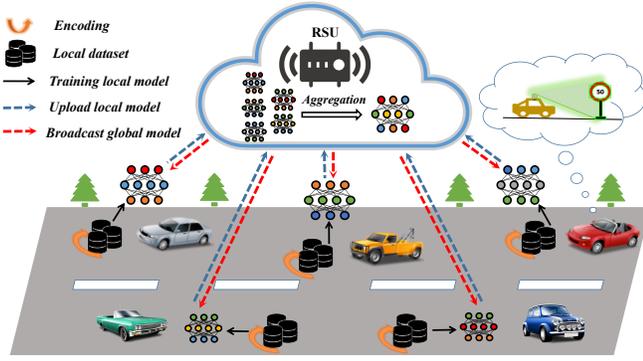
Fig. 1. Overview of federated spike neural networks in autonomous driving for road traffic sign recognition.

positioning system to achieve high-precision coordinated vehicle positioning while ensuring user privacy. To ensure secure and reliable FL, Kang *et al.* in [9] utilized blockchain to achieve secure reputation management, and then proposed a reputation-based worker selection scheme to encourage high-quality workers to participate in FL. Furthermore, Lu *et al.* in [10] designed a blockchain empowered FL architecture to achieve secure data sharing without heavy transmission load and privacy risks. Hu *et al.* in [28] build a two-layer federated structure by coordinating multiple RSUs, which reduces the computational overhead of traffic flow and protects data privacy during model training.

Due to the limited computing resource and energy constraints for vehicular networks, the above FL methods do not work well in resource-hungry and complex vehicular applications.

C. Low-power Spike Neural Network

Deep Convolutional Neural Networks (CNNs) provide efficient tools for the accurate and effective processing of vehicular data. However, it requires a tremendous amount of computing resources, time, and energy for task training, which may be not practical and feasible for IoV scenarios with limited computing resources and strict energy consumption restrictions [29], [30]. To address these challenges, spike neural networks, which are next-generation artificial neural networks inspired by information processing in biology, have been introduced to efficiently achieve deep neural network implementation with favorable properties including low-power consumption, fast inference, and event-driven information processing [14]. Chandarana *et al.* in [31] proposed a method of encoding a static image into a time pulse sequence and achieving adaptive signal sampling for low-power edge computing through SNNs. Kim *et al.* in [32] proposed a time batch normalization technique to train high-precision and low-latency SNN, which brings huge energy efficiency benefits to the hardware. Therefore, SNNs are becoming a promising alternative to traditional convolutional neural networks in IoV. For example, the author in [33] proposed Reward-modulated Spike-Timing-Dependent Plasticity (R-STDP), which combines reinforcement learning and spike-timing-dependent plasticity in IoV for end-to-end

TABLE I
SUMMARIES OF APPLIED SYMBOLS.

Symbols	Description
$\mathcal{K}_r$	The set of clients selected in round r
$\mathcal{D}_k$	The k-th client local dataset
$\mathcal{O}_k$	The dataset of the k-th client after encoding
$\mathcal{L}_k$	The k-th client local loss function
$L(\cdot)$	The cross-entropy function
ω_r	The global model of round r
ω_r^k	The local model of the k client in round r
ω^*	The global optimal model
$f_\omega(\cdot)$	The prediction function on model ω
x_i	The i-th sample
o_i	The i-th sample to be encoded
y_i	The label of the i-th sample
p_k	The contribution rate of the k-th client
c	The total number of classes
K	The total number of clients
F	The client participation rate
B	The local min-batch size
E	The number of local epochs
M	The side length of the output feature map
N_l	The number of neurons in layer l
$E(l)$	The energy consumption of layer l
r	The round of communication
ks	The convolution kernel size
s	The input feature size
C	The number of channels
$I(t)$	The input current at time t
$u(t)$	The membrane potential at time t
u_{rest}	The resting potential
u_i^t	The membrane potential of the i-th neuron at time step t
o_i^t	The spike sequence value of neuron i at time step t
γ_i^t	The learned weight of the BN layer
θ	The threshold voltage of spike neural network
t	The time step of spike neural network
λ	The leakage factor
α	The substitution rate
w_{ij}	The weight of the connection between neuron i and neuron j
W_l	The weight of the l-th layer
$R_s(l)$	The spike rate of the l-th layer
τ_m	The time constant of membrane potential decay
μ	The parameters of Dirichlet distribution function

training SNN. The author in [34] proposed a spike neural network based on visual coding inspired by silicon retina, which controls the end-to-end lane following vehicle behaviors.

Inspired by previous work, this paper studies the traffic sign recognition tasks, which require strong privacy protection, and considers vehicles with limited computing resources and energy in IoV. To this end, we propose a federated spike neural network framework, which provides low power consumption and high-precision traffic sign recognition while ensuring user privacy. Furthermore, the symbols of this paper are summarized in Tab. I.

III. METHODOLOGY

A. Federated Learning

Federated Learning (FL) is a distributed machine learning framework with privacy protection, which requires the transmission of model parameters rather than raw data between clients and the server [7]. In FL, we assume that there are K clients and a server $\mathcal{S}$, which cooperate to train a global optimal model ω^*, where the server does not hold any raw data while each client holds a local dataset $\mathcal{D}_k$. The detailed training process of FL is described in the following.

In each communication round r, the server randomly selects a subset of clients $\mathcal{K}_r \subseteq \mathcal{K}$ to send the current global model ω_r. Then, the selected clients use Stochastic Gradient Descent

(SGD) to train the received global model ω_r on their local dataset $\mathcal{D}_k = \{(x_1, y_1), (x_2, y_2), \cdots, (x_i, y_i)\}$. For the k-th client, the following objective function needs to be minimized:

$$\mathcal{L}_k(\omega) = \frac{1}{|\mathcal{D}_k|} \sum_{x_i, y_i \in \mathcal{D}_k} L_k(y_i, f_{\omega_k}(x_i)), \qquad (1)$$

where $L_k(\cdot)$ generally is the cross-entropy function, y_i is the ground-truth of x_i, $f_{\omega_k}(\cdot)$ is the probability vector predicted by the model. Meanwhile, the local update of the k-th client is as follows:

$$\omega_{r+1}^k = \omega_r - \eta \bigtriangledown \mathcal{L}_k(\mathcal{D}_k; \omega), \qquad (2)$$

where η is learning rete. After completing local training, each client uploads its local model update ω_{r+1}^k to the server.

Next, all uploaded model updates will be aggregated into the current global model by the server using the federated average algorithm (i.e., FedAvg [35]), which is expressed as:

$$\omega_{r+1} = \sum_{k \in \mathcal{K}_r} p_k \omega_{r+1}^k, \qquad (3)$$

where $p_k = \frac{|D_k|}{\sum_{k \in \mathcal{K}_r} |D_k|}$ represents the contribution rate of the k-th client to the current global model, $\mathcal{K}_r$ represents the client set selected in the r-th round.

In the whole FL training process, with the increase of rounds, the model will tend to converge to obtain the optimal global model ω^*.

B. Spiking Neural Networks

Spiking Neural Networks (SNNs) have emerged as a promising computing paradigm using discrete action potentials (i.e., spikes) that highly mimic the physiological mechanisms by which biological neurons propagate information [36]. Unlike traditional Artificial Neural Networks (ANNs) [37]–[39], SNNs use binary spikes to transmit information between layers and layers of the network with less computational cost and extremely low power consumption [40]. The low power consumption of SNNs is reflected in the fact that the floating-point product operation in ANNs can be simplified to the accumulation operation of multiple additions [41]. Specifically, each neuron maintains a membrane potential level, which integrates the input spike signal [42]. When a neuron receives a sequence of spike signals within a time step t, its membrane potential will increase, decrease, or remain unchanged. Once the membrane potential of the neuron exceeds a threshold θ, it generates an output spike signal 1 to the next layer of neurons. Next, the neuron enters a transient refractory period. At this time, no matter whether any spike information is received, the membrane potential will not change. If the neuron does not reach the threshold potential within the time step, it will output a spike signal 0. In addition, the membrane potential will also leak during the accumulation process. Fig. 2(a) shows the change of membrane potential of a spiking neuron in time step t.

To simulate the spike generation process of neurons, researchers widely use Leakage-Integration-and-Fire (LIF) model [43], as shown in Fig. 2(b), which is similar to a physical structure composed of a capacitor C, a resistor R, power supply u, and an input current I. Here, we regard the
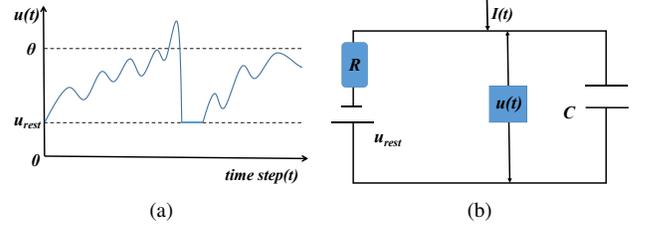

Fig. 2. (a) Changes in membrane potential; (b) Physical structure of LIF model.

input current $I(t)$ as a spike sequence received by neurons, and the process of accumulating neuronal membrane potential as a charging capacitor C. Namely, when capacitor C accumulates sufficient charge and reaches the threshold voltage θ, it will enter the discharge state, i,e., a spike signal will be generated. If the input current $I(t)$ disappears, the voltage of the entire capacitor is provided by u_{rest}, which is the resting potential. Moreover, the charge leaks from the cell membrane when neurons accumulate membrane potential, so a leakage resistance R can be used to simulate this biological phenomenon. According to the law of conservation of current, the input current can be divided into two parts: $I(t) = I_R(t) + I_C(t)$. The LIF can be expressed as a differential equation:

$$\tau_m \frac{du(t)}{dt} = -(u(t) - u_{rest}) + RI(t), u(t) < \theta, \qquad (4)$$

where $\tau_m = RC$ is the time constant of membrane potential decay, $u(t)$ represents the neuron membrane potential at time t, u_{rest} is the resting potential, θ is the threshold voltage. Also, R and $I(t)$ represent the resistance and input current at time t, respectively.

As illustrated in Fig. 3, specific to a single neuron i, we can express the membrane potential u_i^t at time step t as:

$$u_i^t = \lambda u_i^{t-1} + \sum_j w_{ij} o_j^t, \qquad (5)$$

where $\lambda \in (0, 1)$ is the leakage factor, j represents the number of neurons in the previous layer, w_{ij} represents the weight of neuron i's connection to neuron j in the previous layer. Here, o_j^t indicates that neuron i receives the spike sequence of neuron j at time t. At this point, the spike sequence output by neuron i at time t can be expressed as:

$$o_i^t = \begin{cases} 1, & if\ u_i^t \geq \theta \\ 0, & otherwise \end{cases}. \qquad (6)$$

Note that the spike neural network has no bias b in the forward propagation since it adds redundant voltage to the membrane potential of SNNs [32].

For the output layer, we discard the function of neurons triggering spikes so that neurons do not generate any spikes. Specifically, when the spike signal propagates to the last layer of the SNN, the membrane potential in the accumulation time step t will be output. Similar to ANNs, the number of neurons in the last layer of SNNs is also determined by the total number of classes in the training dataset. In this way, the output layer can predict the probability distribution of samples
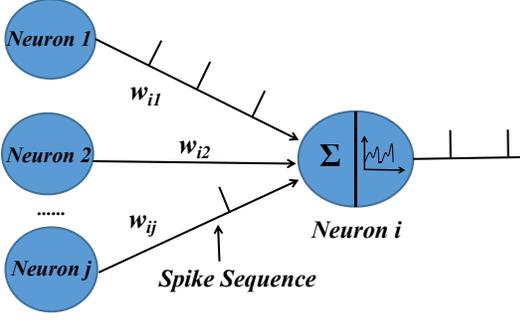
Fig. 3. Spike neuron propagation process.

without causing information loss. Thus, the cross-entropy loss function of SNNs can be expressed as:

$$L = -\sum_{j=1}^{c} y_j log(\frac{e^{u_j^t}}{\sum_{j=1}^{c} e^{u_j^t}}), \quad (7)$$

where y is the ground-truth label, c is the total number of classes in the training dataset, u_j^t is the membrane potential accumulated by the j-th neuron in the output layer of the SNN during a time step t.

For model updates, SNNs usually use gradient descent to backpropagate the loss value to update the weights of all layers. Unlike ANNs, SNNs need to calculate the gradient of a time step. According to the chain derivation rule, the gradient of the l-th layer can be defined as:

$$\triangle W_l = \begin{cases} \sum_{t=1}^{t} \frac{\partial L}{\partial O_l^t} \frac{\partial O_l^t}{\partial U_l^t} \frac{\partial U_l^t}{\partial W_l^t}, & if \ l = hidden \ layer \\ \sum_{t=1}^{t} \frac{\partial L}{\partial U_l^t} \frac{\partial U_l^t}{\partial W_l^t}, & otherwise \end{cases}, \quad (8)$$

where U_l is the membrane potential of neurons in the l-th layer of the SNN, O_l is the spike sequence of the l-th layer, and W_l is the weight of the l-th layer. However, in SNNs, neurons can only produce spike output when the membrane potential exceeds the threshold, resulting in non-differentiable. Thus, to overcome this, researchers generally use some alternative functions to approximate the derivative of the threshold function, the most common of which is the piece-wise linear function, such as:

$$\frac{\partial o_i^t}{\partial u_i^t} = \alpha \max\{0, 1 - |\frac{u_i^t - \theta}{\theta}|\}, \quad (9)$$

where α is the substitution rate that controls the update amplitude of SNNs, θ is the threshold voltage. The reason for this design is that when the membrane potential of a neuron is closer to the voltage threshold, the membrane potential is easier to be activated and send spikes to the next layer of neurons. On the other hand, since the gradient is accumulating at each time step, we guess that the value of the hyperparameter α is also related to the time step t. For a larger time step t, a smaller hyperparameter α should be set to avoid gradient explosion. Overall, the backpropagation method of SNNs is similar to ANN, except that the alternative function is used to approximate the non-differentiable threshold function. Therefore, the model update of the l-th layer of SNNs can be expressed as: $W_l = W_l - \eta \triangle W_l$.

C. Training Spiking Neural Networks with Batch Normalization Through Time

Batch Normalization (BN) [44] has the advantages of accelerating the speed of network training and convergence, controlling gradient explosion, and preventing gradient disappearance, which has been widely used in the deep learning model. However, the BN layer only normalizes a batch of inputs, ignoring the information in the time dimension. Thus, the naive application of the BN layer in SNN cannot capture the time representation of the spike sequence. Recently, the authors in [32] propose a method to update the internal parameters of the BN layer with the time step to improve the SNN training performance, which is called Batch Normalization Through Time (BNTT). Specifically, after the spike sequence passes through the convolution or linear layer, it will enter the BN layer that expands with the time step t, i.e., t BN layers. The mathematical formula of BNTT is given as follows:

$$u_i^t = \lambda u_i^{t-1} + BNTT_{\gamma^t} \sum_j w_{ij} o_j^t,$$
$$= \lambda u_i^{t-1} + \gamma_i^t(\frac{\sum_j w_{ij} o_j^t - \mu_i^t}{\sqrt{(\sigma_i^t)^2 + \epsilon}}), \quad (10)$$

where γ_i^t is the learned weight of the BN layer, ϵ is a constant that prevents the denominator from being 0, μ_i^t and σ_i^t are the mean and variance from the samples in a mini-batch B for time step t, respectively. For the update of the BN layer, given $x_i^t = \sum_j w_{ij} o_j^t$, the gradient of weight γ^t in a batch b can be expressed as:

$$\triangle \gamma^t = \frac{\partial L}{\partial u^t} \frac{\partial u^t}{\partial \gamma^t} = \sum_{i=1}^{b} \frac{\partial L}{\partial u^t} \hat{x}_i^t. \quad (11)$$

Here, $\hat{x}_i^t = \frac{x_i^t - \mu^t}{\sqrt{(\sigma^t)^2 + \epsilon}}$. Therefore, a batch of weight updates for the t-th BN layer can be calculated as:

$$\gamma^t = \gamma^t - \eta \triangle \gamma^t. \quad (12)$$

IV. FEDERATED SPIKE NEURAL NETWORK ENCODED BY NEURONAL RECEPTIVE FIELD

A. Neuronal Receptive Field Encoding

The attention mechanism originates from the fact that humans can reasonably utilize limited visual information processing resources to selectively focus on visual focus areas, and has been widely used in the field of deep learning. In recent years, many attention models such as Transformer [45], DIN [46] and AFM [47] have been proposed in natural language processing and image recognition. Inspired by the attention mechanism, we believe that information in the central region of the neuron receptive field is more likely to activate that neuron to emit spikes, while it is more difficult for those at the edges. To this end, we propose a novel encoding method for SNNs called Neuronal Receptive Field Encoding (NRFE) for fast and efficient training. Specifically, we design a Gaussian distribution matrix to simulate the regions observed by neurons, and then combine the pixel features of the image to jointly encode the local traffic sign dataset of vehicles. Next, we provide more details about the NRFE method as follows.

- **Step 1, Normalization:** The input feature of an image is usually composed of three channels of RGB pixels. Note that the range of RGB pixel values is between 0 and 255. However, if the input features are large, the gradient passed to the input layer during backpropagation will become large, resulting in a suboptimal model. Therefore, in order to accelerate the model convergence and prevent gradient explosion, we normalize the input features. For each channel's input $X = [x_{ij}]_{s \times s}$, we normalize it to the range 0 to 1 (i.e., $X \to X^*$).

$$x_{ij}^* = \frac{x_{ij} - x_{min}}{x_{max} - x_{min}}, \qquad (13)$$

where x_{max} and x_{min} are the minimum and maximum pixel values in matrix X, respectively. Here, s is the size of the input image.

- **Step 2, Gaussian receptive field matrix:** The receptive field is the output feature of each layer in SNNs, which is mapped to the size of the area on the input image, i.e., a point on the output feature corresponds to an area on the input image. In the receptive field, the pixel is closer to the center, and the greater contribution to the calculation of output features will be. In other words, the closer the pixel is to the center of the receptive field, the more times it will be convolved. However, the contribution of all pixels in the receptive field to the output feature is not all the same. In many cases, the influence of pixels in the receptive field area on the model follows a Gaussian distribution [48]. Inspired by [49], we design a Gaussian receptive field matrix $G_{s \times s} \sim N(\mu_g, \sigma_g^2)$ to simulate the area detected by biological neurons, as follows.

$$\mu_g = \frac{1}{s} \sum_{i=1}^{s} \frac{2i - 1}{2(s - ks)}, \qquad (14)$$

$$\sigma_g = \frac{BCE}{s - ks}, \qquad (15)$$

where ks represents the size of the convolution kernel, C represents the number of input channels, B is the local mini-batch size, and E is the local epoch.

- **Step 3, Coding:** For an image, we consider that pixels in the center of the receptive field and with large values are more likely to activate neurons to emit spike signals. Thus, we encode such pixels as 1. Secondly, for the pixels with small values at the edge of the receptive field, we encode them as -1. In other cases, it is coded as 0. Given $Q = X^* + G$, the coding formula of the Gaussian receptive field is as follows:

$$input = \begin{cases} 1, & q_{ij} \geqslant x_{ij}^* \cap q_{ij} \geqslant g_{ij} \\ -1, & q_{ij} \leq x_{ij}^* \cap q_{ij} \leq g_{ij}, \\ 0, & otherwise \end{cases} \qquad (16)$$

where $q_{ij} \in Q$, $x_{ij}^* \in X^*$, $g_{ij} \in G$.

1) Discussion: Different coding schemes have different principles for extracting features from the raw data. The choice of a specific encoding method depends on whether the data has relevant information in the temporal and spatial domains, and whether there is noise. In rate-based coding of SNNs [17],

---

Algorithm 1 FedSNN-NRFE algorithm.

Input: Set of $\mathcal{K}$ clients, mini-batch size B, local training epoch number E, time step T, network layer L, and learning rate η.

Output: The convergent SNN model ω^*.

1: **Server executes:**
2: Initialize a global SNN model ω_0
3: **for** each round $r = 0, 1, 2, ...$ **do**
4:    $S_r \leftarrow$ randomly select $m = \max(F \cdot K, 1)$ clients from a set of $\mathcal{K}$ clients
5:    **for** each client $k \in S_r$ **in parallel do**
6:       $\omega_{r+1}^k \leftarrow$ **ClientUpdate**(k, ω_r)
7:    **end for**
8:    $\omega_{r+1} = \sum_{k \in \mathcal{S}_r} p_k \omega_{r+1}^k$
9: **end for**
10: **ClientUpdate**(k, ω_r)**://** For the k-th client
11: $\mathcal{O}_k \leftarrow$ NRFE $(\mathcal{D}_k)$
12: **for** each local epoch $e \leftarrow 1$ to E **do**
13:    **for** mini-batch B **do**
14:       **for** layer l from 1 to $L - 1$ **do**
15:          **for** time step t from 1 to T **do**
16:             $U_l^t = \lambda U_l^{t-1} + BNTT_{\gamma^t}(W_l, O_{l-1}^{t-1})$
17:          **end for**
18:       **end for**
19:       //Accumulation of membrane potential at layer L
20:       $\omega_{r+1}^k = \omega_r - \eta \bigtriangledown \mathcal{L}_k(k, \mathcal{O}_k; \omega_r)$
21:    **end for**
22: **end for**
23: **return** ω^* to server.

---

the input of each image is coded as a sequence of Poisson spikes according to its pixel intensity. The probability of a spike in this sequence is positively correlated with the pixel intensity of the image. However, it generally requires very high pixel values to be encoded into a sequence of high-frequency spikes for quick propagation forward in the network. Therefore, rate coding has limitations in the face of noisy and low-pixel traffic sign data. Another common encoding method is temporal-based [50], which generates the precise time of each spike according to changes in the input signal. Although this approach has been shown to be effective in practice, it is generally suitable for the fast processing of time series and streaming data. Thus, it is also inappropriate for traffic sign recognition. Compared with the above two encoding methods, our proposed encoding scheme based on neuron receptive fields extracts features from pixel and spatial dimensions and is especially suitable for encoding traffic sign data.

B. Federated Spike Neural Networks for IoV

Autonomous driving applications in IoV may involve multiple fields and need to integrate data from various agencies and departments. However, the data collected by vehicles usually include private information such as driver location and traveling trajectory. Therefore, it is generally unrealistic to collect all data on a centralized server. Federated learning is essentially an encrypted distributed machine learning technology, which

provides an effective computing paradigm for executing multi-vehicle autonomous driving tasks. Furthermore, due to the limited computing and storage resources of most IoV devices (e.g., vehicles), it is impossible to achieve complex artificial intelligence computing tasks. In contrast, SNNs achieve comparable performance to deep neural networks with relatively low computational cost and extreme energy efficiency, which has been widely used in IoV, such as image classification, target detection, navigation, and motion control. Thus, based on the automatic driving scenarios of IoV, we propose a federated spike neural network algorithm called FedSNN-NRFE, which focuses on a fast and effective coding scheme to achieve low-power and high-performance FL.

Considering that a group of wirelessly connected vehicles perform FL autonomous driving tasks on the same road, as shown in Fig. 1. These vehicles use on-board cameras to scan traffic signs in front of the road and then make appropriate driving decisions. The entire FL-base IoV system consists of a roadside unit (RSU, i.e., server) and K vehicles working as clients, in which each vehicle has its own private local dataset $\mathcal{D}_k$. The RSU aggregates the SNN models uploaded by the vehicle and sends the global model. Each vehicle uses the local dataset to train the SNN model. Specifically, for communication round r, the iterative process of FedSNN-NRFE is based on the following protocol:

1) The RSU randomly selects a certain proportion of F ($0 < F < 1$) vehicles from all vehicles to deliver the global model parameter ω_r.
2) For the k-th vehicle, it first utilizes the NRFE method to encode a local private dataset, i.e.,

$$NRFE(\mathcal{D}_k) \to \mathcal{O}_k, \tag{17}$$

where $\mathcal{O}_k = \{(o_1, y_1), (o_2, y_2), \cdots, (o_i, y_i)\}$. Afterwards, the encoded dataset $\mathcal{O}_k$ is used to train the received global SNN model ω_r. The loss function is defined as follows:.

$$\mathcal{L}_k(\omega) = \frac{1}{|\mathcal{O}_k|} \sum_{o_i, y_i \in \mathcal{O}_k} L_k(y_i, f_{\omega_k}(o_i)), \tag{18}$$

where $L_k(\cdot)$ is the cross-entropy function in Equation (7), $f_{\omega_k}(o_i)$ is the $softmax$ output of the accumulated membrane potential in the last layer of the SNN model, i.e., $f_{\omega_k}(o_i) = \{f_{\omega_k}(o_i)_1, \cdots, f_{\omega_k}(o_i)_j, \cdots, f_{\omega_k}(o_i)_c\}$. Specifically, the $softmax$ of $f_{\omega_k}(o_i)_j$ is expressed as follows:

$$f_{\omega_k}(o_i)_j = \frac{e^{u_j^t}}{\sum_{j=1}^{c} e^{u_j^t}}. \tag{19}$$

Given a time step $t = 1, 2, \cdots, T$, the membrane voltage U_l^t for each layer $l = 1, 2, \cdots, L-1$ can be expressed as:

$$U_l^t = \lambda U_l^{t-1} + BNTT_{\gamma_l^t}(W_l, O_{l-1}^{t-1}). \tag{20}$$

Similarly, when U_l^t is greater than the threshold voltage θ, $O_l^t = 1$; otherwise, $O_l^t = 0$. Then, the vehicle computes the gradient change of the local model (Equations (8) and (11)) based on the loss to update the model weights.

$$\omega_{r+1}^k = \omega_r - \eta \bigtriangledown \mathcal{L}_k(k, \mathcal{O}_k; \omega_r). \tag{21}$$

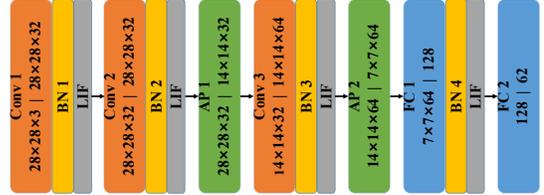
Fig. 4. Overview of BelgiumTS image dataset.

Fig. 5. Description of SNN network structure. Here, Conv represents the convolution layer, AP represents the average pooling layer, and FC represents the full connection layer.

3) The RSU collects the model parameters uploaded by the local vehicle and uses the federated average aggregation algorithm (Equation (3)) to update the global model.

Note that FedSNN-NRFE repeats all the above steps until the global model converges or reaches a predetermined accuracy. The proposed FedSNN-NRFE algorithm training details are given in Algorithm 1. Furthermore, for the image recognition task of traffic signs, the complexity mainly depends on the convolutional layers of SNNs. Thus, given the number of convolutional layers l, the size of the convolutional kernel ks, the side length M of the output feature map, and the number of input and output channels C_{in} and C_{out}, the complexity of this algorithm is $O(rFK \sum_l ks_l^2 M_l^2 C_{in}^l C_{out}^l T)$.

V. NUMERICAL RESULTS

In IoV, autonomous vehicles correctly perceive their surrounding environment through advanced sensors, radar, and Global Position System (GPS), and then make appropriate navigation strategies according to sensing traffic information and recognized traffic signs [7], [51]. Therefore, we experimentally validate our method FedSNN-NRFE and baselines through traffic sign recognition tasks on a real-world dataset named Belgium traffic sign (BelgiumTS) Dataset [52].

For all experiments, we evaluate the performance of the proposed FedSNN-NRFE and baseline schemes on a desktop computer with an i9-9900K CPU and RTX 2080Ti GPU. For the federated learning tasks, we consider that there are K vehicles acting as clients and one RSU working as the parameter server to execute traffic sign recognition tasks.

A. Experiment Setup

1) Dataset under IID and non-IID settings: As shown in Fig. 4, the BelgiumTS dataset has 62 classes of 28×28 RGB

color traffic sign images, including 4575 training images and 2520 test images. For the IID setting of data distribution in FL, each client holds 186 training samples, in which there are 3 samples for each class and a total of 20 clients. For the non-IID settings of data distribution, we simulated class imbalances in the local data distribution of different clients. Specifically, we generate different non-IID levels by sampling the training set by adjusting parameter μ of the Dirichlet distribution function [53], [54]. The smaller the parameter μ, the more obvious the class imbalance of the local training set between different clients; on the contrary, the less obvious it will be. Here, we use $\mu = 0.5$ as the non-IID setting of FL.

2) Model architecture: As shown in Fig. 5, the model structure of the SNN used in our experiment consists of three convolution layers, four BN layers, two pooling layers, and two fully connected layers. Since the SNN layer transmits a binary sequence among layers, maximum pooling will cause a large amount of information to be lost in the next layer. Therefore, we use the average pool during training.

3) Baselines: Our baselines include: 1) Centralized SNN: traditional and centralized spike neural network; 2) FedCNN: federated convolutional neural network with the same model architecture; 3) FedSNN: federated spike neural network coded by standard normal distribution of receptive fields (i.e., $G_{s \times s} \sim N(0,1)$).

4) Training details: In the experiment, we set the total number of clients $K = 20$, the participation rate $F = 0.5$, local training epoch number $E = 2$, mini-batch size $B = 8$ and learning rate $\eta = 1e-1$. For the SNN, the time step is $T = 10$, the leakage rate $\lambda = 0.9$, the substitution rate $\alpha = 0.3$, and the threshold voltage $\theta = 1.0$.

B. Experiment Results

1) The performance impacts of IID and non-IID setting: In Table II, we show the average accuracy and fluctuation range of test results of centralized and federated learning methods under IID and non-IID settings. Obviously, the centralized SNN achieves the best performance thus validating that the spike neural network can be well applied to the image classification tasks in IoV. However, for IoV scenarios, it is impractical to gather all the sensing traffic signs from distributed vehicles into a centralized database because of the high risk of user privacy leakage. Therefore, we introduce the paradigm of federated learning to effectively solve the privacy protection issues. Fig. 6(a) shows that, compared with centralized SNN, FedSNN and FedSNN-NRFE almost achieved the same test accuracy. Meanwhile, from Fig. 6(b), we observe that the spike neural network has a significant advantage at the level of non-IID data distribution. Specifically, the accuracy levels of FedSNN-NRFE and FedSNN exceed that of FedCNN by 7% and 4%, respectively. These results show that FedSNN-NRFE is very effective on the BelgiumTS dataset and hence well-suited for the traffic image recognition task in IoV scenarios.

2) Sensitivity of hyperparameters: To validate and analyze the optimal hyperparameter settings of the proposed FedSNN-NRFE, we compare the performance under different time

TABLE II
PERFORMANCE COMPARISON OF DIFFERENT METHODS UNDER IID AND NON -IID SETTINGS.

BelgiumTS with 20 clients (K=20, F=0.5)		
Methods	IID	non-IID
Centralized SNN	95.86±0.15	N/A
FedCNN	92.89±0.23	87.75±0.36
FedSNN	92.93±0.18	91.62±0.54
FedSNN-NRFE	**94.95±0.28**	**94.88±0.36**

steps, substitution rates, and leak rates. As shown in Fig. 7(a), the substitution rate is negatively correlated with the model performance. The larger substitution rate brings the lower test accuracy of the final convergence of the model. Meanwhile, it is worth noting that a larger substitution rate speeds up the convergence of the model. The reason is that the large substitution rate leads to too large model updates, and it is easier to achieve the local optimum, but miss the global optimum. Conversely, the lower substitution rate makes the model update slowly and achieves higher convergence accuracy. Therefore, in our experiments, we set the substitution rate α as 0.3.

For SNNs, too small a time step T and leak rate λ make it difficult to activate neurons to emit spikes, but too large a time step will cause the model to have a gradient explosion during the update process. Therefore, it is necessary to explore the impact of these two hyperparameters on the global model performance. As we can observe from Fig. 7(b) and Fig. 7(c), the best test accuracy is achieved with a time step of $T = 10$ and a leak rate of $\lambda = 0.9$. Meanwhile, in Fig. 8, regardless of the value of the time step t, the performance of leakage rate $\lambda = 0.9$ is significantly or approximately better than that of the leakage rate $\lambda = 0.8$. This is because the leakage rate is positively correlated with the probability of neurons being activated. In detail, the larger leakage rate of the spiking neuron brings the more membrane voltage it retains at the previous moment, and the easier it to activate the neuron to emit the spike. Moreover, the setting of time steps T is also important for SNNs. It is difficult for the membrane potential of spike neurons to exceed the threshold voltage if the time step is too small. An excessively large time step increases the computational complexity and is prone to gradient explosion and over-fitting problems. Through experimental verification, we observe that the time step $T = 10$ and the leak rate $\lambda = 0.9$ are the optimal settings for the proposed FedSNN-NRFE.

3) The performance impacts on the number of participating clients: In FL, there are a large number of dropped devices due to system heterogeneity factors such as network connection (3G, 4G, 5G, WiFi), hardware (CPU, RAM), battery capacity, etc. To reduce computational complexity and communication overhead, we prefer to obtain a high-quality global model in the setting of fewer clients participating in each training round. To this end, in the non-IID case, we explore the performance impact of different numbers of clients participating in training. Fig. 9(a) presents the maximum accuracy level converged by our method and the baseline FedCNN after 200 rounds with the participation rate $F = 5\%, 25\%, 50\%, 75\%, 100\%$ settings. Obviously, as the number of participating clients increases, the
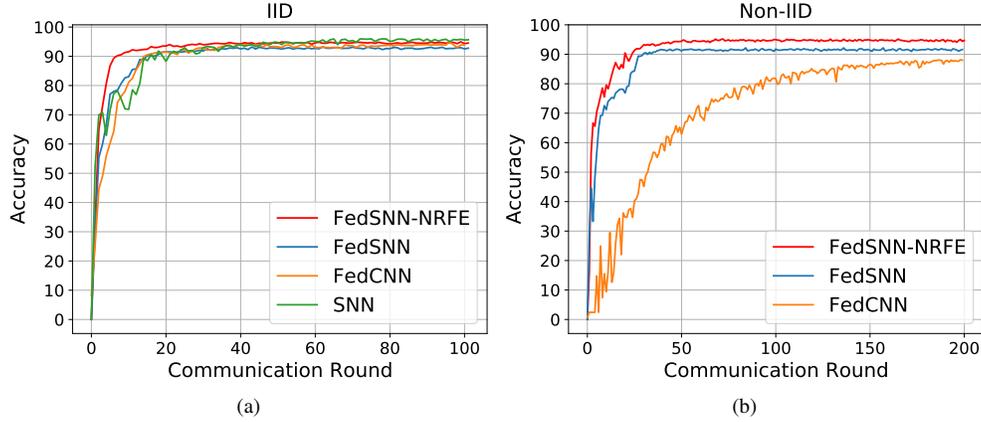
Fig. 6. Test accuracy curves of (a) IID and (b) non-IID settings of data distribution.
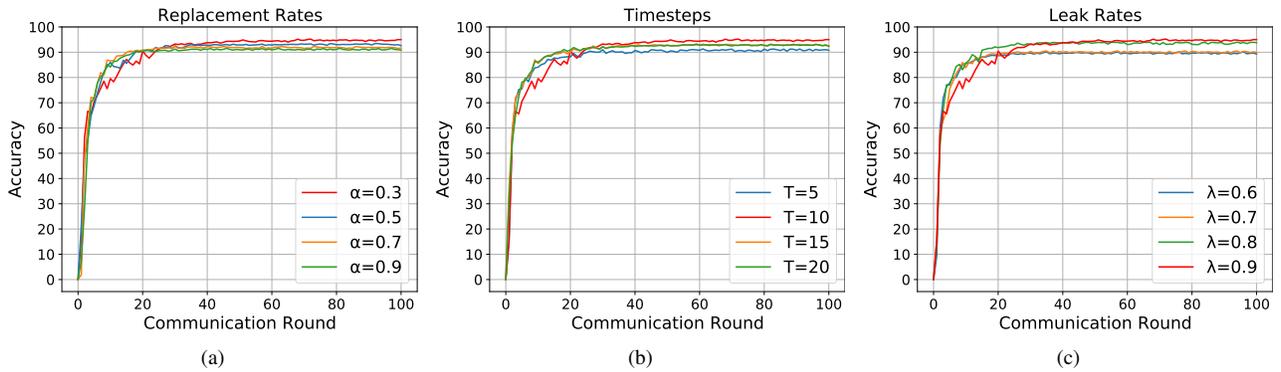

Fig. 7. Test accuracy curves of (a) different substitution rates, (b) different time steps, and (c) different leak rates.
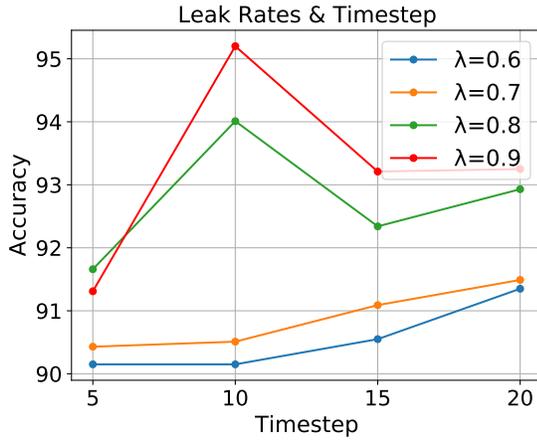

Fig. 8. Performance comparison of different time steps and leak rates.

performance also increases gradually. Especially for FedSNN-NRFE, when the number of participations is increased from 1 to 10, the performance improvement is significant and much higher than the baseline FedCNN. This implies that it is feasible to integrate SNNs into FL.

4) The performance impacts of the number of training sets: In traditional ANNs, a high-precision traffic sign recognition model usually requires a large number of training samples. However, this increases the cost of collecting data and the time complexity of training the model to a certain extent. SNNs are the third generation of artificial neural networks inspired by biological information processing, which have the advantages of fast reasoning and event-driven information processing. To this end, we compare the performance of CNN and SNN for traffic sign image recognition under the FL paradigm. Fig. 9(b) shows the performance comparison of 200, 150, and 100 training samples per client under non-IID settings. Obviously, for FedCNN, as the number of local training samples held by the client decreases, the test accuracy of the global model drops by 15%. In contrast to our FedSNN-NRFE, when the number of training samples is reduced from 200 to 100, the accuracy is only reduced by 5%. Meanwhile, we observe that the accuracy of our method is better than that of the baseline (i.e., FedCNN) when the number of training samples is only 100. This proves that the SNN has the characteristics of strong adaptability under the condition of few training samples.

5) The robustness performance to image noise and different non-IID data distributions: In the real world, due to external environmental factors (e.g., sun exposure, rain erosion, weathering, etc.), traffic signs will become blurred. Thus, to simulate this phenomenon, we add different intensities of salt-and-pepper noise [55] to the BelgiumTS training dataset to verify the robustness of our FedSNN-NRFE model. Specifically, for an image, we randomly change the pixel value of a fixed proportion from the original to 0 or 255. The observation
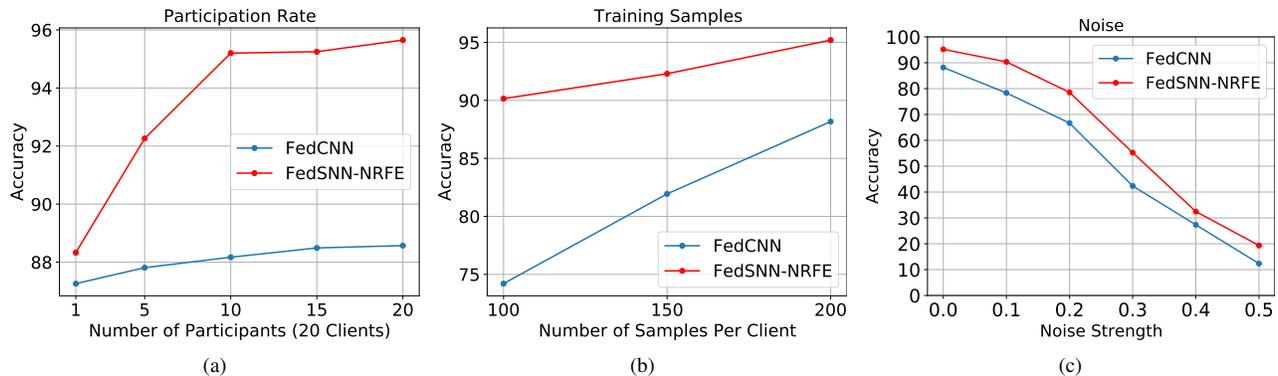
Fig. 9. Performance comparison of (a) different of participating clients, (b) different training samples, and (c) different noise strengths.
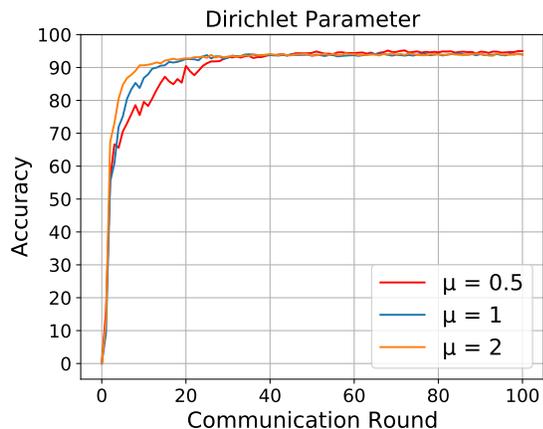

Fig. 10. Test curves for different non-IID levels.

result is shown in Fig. 9(c). As the proportion of added noise increases, the performance of FedSNN-NRFE and FedCNN decreases significantly. We observe that when the noise proportion increased from 0.2 to 0.4, the performance of both our method and baseline decreased by more than 35%. It is worth noting that under the same noise proportion, the test accuracy of FedSNN-NRFE is better than that of FedCNN. Furthermore, when the salt-and-pepper noise ratio is 0.1, the performance of our proposed method still reaches 91%. In short, compared to CNNs, SNNs are more robust to noisy images in IoV.

In the FL-based IoV scenarios, the traffic signs captured by vehicles using different types of in-vehicle cameras in different areas usually have large differences in features and categories. Therefore, the data distribution among vehicles is non-IID. Fig. 10 shows the test accuracy of FedSNN-NRFE on traffic sign data with different non-IID levels of vehicles. The difference is that the higher non-IID level results in the slower convergence of the global model. Interestingly, our proposed method achieves stable performance at all three non-IID levels and differs very little from each other.

6) The performance of energy consumption: To better evaluate the computational cost of CNNs and SNNs during the training process, we use the energy consumption metric described in [56]. Table III shows the energy consumption

of 32-bit integer arithmetic operations based on the 45nm CMOS process. In the propagation process of neural networks, the computational cost is proportional to the calculation of floating-point numbers. In our experiment, we only consider multiplication and addition operations. Specifically, for layer l in CNNs, the number of floating-point operations (FLOPs) is defined as:

$$FLOPs(l) = \begin{cases} ks^2 \times M_{out}^2 \times C_{in} \times C_{out}, & Conv \\ C_{in} \times M_{in}^2, & BN \ or AP \ , \\ N_{in} \times N_{out}, & FC \end{cases}$$
(22)

where ks is kernel size. C_{in} is input channel and C_{out} is output channel. M_{in} and M_{out} are the side lengths of the input and output feature maps. N_{in} is input neuron numbers and N_{out} is output neuron numbers.

For SNNs, floating-point operations are only generated when the neuron outputs pulses. Meanwhile, binary propagation performs is an accumulation (AC) operation, resulting in significant energy efficiency [41]. We define the spike rate R_s of each layer, i.e., the number of spikes n_l^s sent by neurons in the l-th layer within the time step T divided by the number of neurons N_l in the l-th layer. The calculation formula of spike rate $R_s(l)$ of the l-th layer is given as follows:

$$R_s(l) = \frac{n_l^s}{N_l}.$$
(23)

Next, we calculate the energy consumption of each layer of CNN and SNN separately:

$$E_{CNN}(l) = FLOPs(l) \times (E_{Mult} + E_{Add}),$$
(24)

$$E_{SNN}(l) = FLOPs(l) \times R_s(l) \times T \times E_{AC}.$$
(25)

Because a larger time step will inevitably lead to higher computational complexity, we set the time step $T = 10$ in our experiment. Fig. 11 shows the estimated energy consumption of each layer of CNNs and SNNs for the model of a client trained on the BelgiumTS dataset. Intuitively, the energy consumption of floating-point operations is mainly concentrated in the convolutional layer. Compared with CNNs, we observe that the energy consumption of SNNs mainly depends on the ratio of activated neurons in each layer of the network. Table IV provides the details about the energy consumption and spike
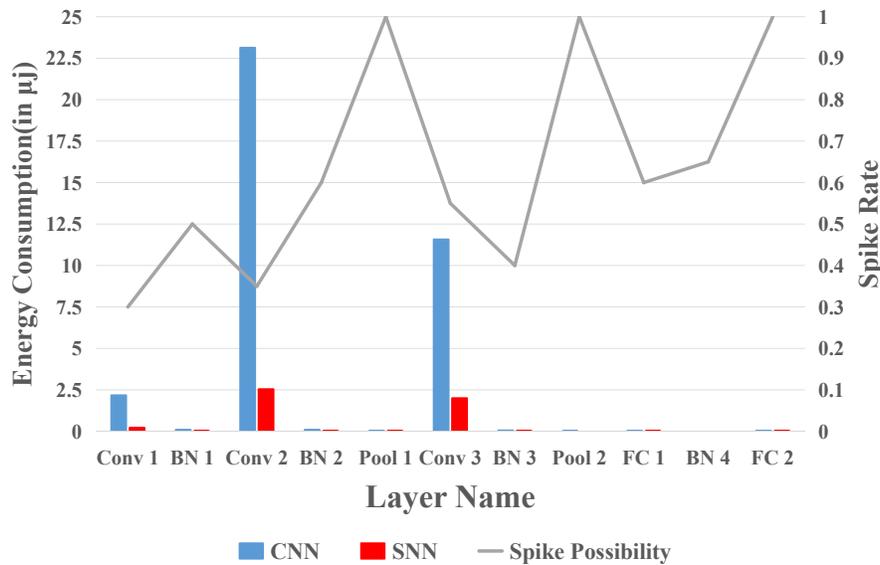
Fig. 11. Comparison of the energy consumption of each layer of SNN and CNN after 100 rounds of communication.

TABLE III
ESTIMATE THE ENERGY OF EACH MULTIPLICATION AND ADDITION OPERATION.

Operation	Estimated Energy (pJ)
32-bit Multiply (E_{Mult})	3.1
32-bit Add (E_{Add})	0.1
32-bit Accumulate (E_{AC})	0.1

TABLE IV
COMPARISON OF THE ENERGY CONSUMPTION OF EACH LAYER OF CNN AND SNN.

Layer Name	CNN (pJ)	SNN (pJ)	Spike Rate
Conv 1	2,167,603.2	203,212.8	0.3
BN 1	80,281.6	12,544	0.5
Conv 2	23,121,100.8	2,528,870.4	0.35
BN 2	80,281.6	15,052.8	0.6
AP 1	21,952	6,860	1.0
Conv 3	11,560,550.4	1,986,969.6	0.55
BN 3	40,140.8	5,017.6	0.4
AP 2	10,579.2	3,306	1.0
FC 1	20,070.4	3,763.2	0.6
BN 4	409.6	83.2	0.65
FC 2	25,395.2	7,936	1.0

rate of each layer. The total energy estimated by CNN is about **37.096** μ**J** while that of SNN is about **4.773** μ**J**, which is **7.772**$\times$ more efficient than of CNN.

VI. CONCLUSION

In this paper, we conduct a comprehensive study on the feasibility of training spike neural networks with the federated learning paradigm in the Internet of Vehicles scenarios. Specifically, we propose a FedSNN-NRFE scheme based on the neuronal receptive field encoding, which realizes the high-precision recognition of road traffic signs in autonomous driving. Furthermore, as far as we know, we are the first to train FedSNN under the traffic sign dataset, and conduct a comprehensive analysis of performance, adaptability, robustness, and energy efficiency. Through numerical results,

we conclude that in the federated learning scenario of the Internet of Vehicles, SNN has become a viable alternative to CNN with its superior energy efficiency and noise resistance. Nevertheless, FL still faces challenges in terms of expensive communication overhead, unreliable devices, and security. There are several possible directions that are worth being studied: I) Due to the binary spike sequences passed between neurons in SNNs, it is promising to improve the aggregation algorithm of FL for efficient communication overhead. II) Considering that client selection strategies based on SNN-related metrics (such as spike rate) should be designed to improve the performance and convergence speed of FL. III) It is still necessary to strengthen the security of the FedSNN-NRFE training algorithm to ensure strict confidentiality of user privacy data. In future work, we will further compare and evaluate these methods for integrating SNNs into FL.

REFERENCES

[1] Gartner Forecasts More Than 740,000 Autonomous-Ready Vehicles to Be Added to Global Market in 2023 (2021). [Online] Available: https://www.gartner.com/en/newsroom/press-releases/ 2019-11-14-gartner-forecasts-more-than-740000-autonomous-ready-vehicles-to-be-ad

[2] M. Xu, D. T. Hoang, J. Kang, D. Niyato, Q. Yan, and D. I. Kim, "Secure and reliable transfer learning framework for 6g-enabled internet of vehicles," *IEEE Wireless Communications*, pp. 1–8, 2022.

[3] M. Li, J. Gao, L. Zhao, and X. Shen, "Adaptive computing scheduling for edge-assisted autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5318–5331, 2021.

[4] Z. Wang, H. Zhu, M. He, Y. Zhou, X. Luo, and N. Zhang, "Gan and multi-agent drl based decentralized traffic light signal control," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2021.

[5] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, 2020.

[6] B. Liu, C. Han, X. Liu, and W. Li, "Vehicle artificial intelligence system based on intelligent image analysis and 5g network," *International Journal of Wireless Information Networks*, pp. 1–17, 2021.

[7] Y. Liu, J. James, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7751–7763, 2020.

[8] X. Huang, P. Li, R. Yu, Y. Wu, K. Xie, and S. Xie, "Fedparking: A federated learning based parking space estimation with parked vehicle assisted edge computing," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9355–9368, 2021.

[9] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[10] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4298–4311, 2020.

[11] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[12] W. Zhang, D. Yang, W. Wu, H. Peng, N. Zhang, H. Zhang, and X. Shen, "Optimizing federated learning in distributed industrial iot: A multi-agent approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 12, pp. 3688–3703, 2021.

[13] M. Shen, J. Zhang, L. Zhu, K. Xu, and X. Tang, "Secure svm training over vertically-partitioned datasets using consortium blockchain for vehicular social networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5773–5783, 2020.

[14] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: opportunities and challenges," *Frontiers in neuroscience*, vol. 12, p. 774, 2018.

[15] N. Skatchkovsky, H. Jang, and O. Simeone, "Federated neuromorphic learning of spiking neural networks for low-power edge intelligence," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8524–8528.

[16] R. V. Rullen and S. J. Thorpe, "Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex," *Neural Computation*, vol. 13, no. 6, pp. 1255–1283, 2001.

[17] Y. Wu, L. Deng, G. Li, J. Zhu, Y. Xie, and L. Shi, "Direct training for spiking neural networks: Faster, larger, better," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1311–1318.

[18] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: Vgg and residual architectures," *Frontiers in neuroscience*, vol. 13, p. 95, 2019.

[19] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in computational neuroscience*, vol. 9, p. 99, 2015.

[20] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road-sign detection and recognition based on support vector machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 264–278, 2007.

[21] D. Tabernik and D. Skočaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1427–1440, 2020.

[22] T. Yang, X. Long, A. K. Sangaiah, Z. Zheng, and C. Tong, "Deep detection network for real-life traffic sign in vehicular networks," *Computer Networks*, vol. 136, pp. 95–104, 2018.

[23] Á. Arcos-García, J. A. Alvarez-Garcia, and L. M. Soria-Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," *Neural Networks*, vol. 99, pp. 158–165, 2018.

[24] R. Yu, J. Kang, X. Huang, S. Xie, Y. Zhang, and S. Gjessing, "Mixgroup: Accumulative pseudonym exchanging for location privacy enhancement in vehicular social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 93–105, 2016.

[25] F. Tang, Y. Kawamoto, N. Kato, and J. Liu, "Future intelligent and secure vehicular network toward 6g: Machine-learning approaches," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 292–307, 2020.

[26] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Differentially private asynchronous federated learning for mobile edge computing in urban informatics," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 2134–2143, 2020.

[27] X. Kong, H. Gao, G. Shen, G. Duan, and S. K. Das, "Fedvcp: A federated-learning-based cooperative positioning scheme for social internet of vehicles," *IEEE Transactions on Computational Social Systems*, pp. 1–10, 2021.

[28] X. Hu, G. Wang, L. Jiang, S. Ding, and X. He, "Towards efficient learning using double-layered federation based on traffic density for internet of vehicles," in *International Conference on Web Information Systems and Applications*. Springer, 2021, pp. 287–298.

[29] I. M. Comsa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, and J. Alakuijala, "Temporal coding in spiking neural networks with alpha synaptic function," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8529–8533.

[30] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "Stdp-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, 2018.

[31] P. Chandarana, J. Ou, and R. Zand, "An adaptive sampling and edge detection approach for encoding static images for spiking neural networks," in *2021 12th International Green and Sustainable Computing Conference (IGSC)*, 2021, pp. 1–8.

[32] Y. Kim and P. Panda, "Revisiting batch normalization for training low-latency deep spiking neural networks from scratch," *Frontiers in neuroscience*, p. 1638, 2021.

[33] Z. Bing, C. Meschede, G. Chen, A. Knoll, and K. Huang, "Indirect and direct training of spiking neural networks for end-to-end control of a lane-keeping vehicle," *Neural Networks*, vol. 121, pp. 21–36, 2020.

[34] J. Kaiser, J. C. Vasquez Tieck, C. Hubschneider, P. Wolf, M. Weber, M. Hoff, A. Friedrich, K. Wojtasik, A. Roennau, R. Kohlhaas, R. Dillmann, and J. M. Zöllner, "Towards a framework for end-to-end control of a simulated vehicle with spiking neural networks," in *2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2016, pp. 127–134.

[35] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[36] K. Roy, A. Jaiswal, and P. Panda, "Towards spike-based machine intelligence with neuromorphic computing," *Nature*, vol. 575, no. 7784, pp. 607–617, 2019.

[37] Y. Zhong, T. Q. S. Quek, and X. Ge, "Heterogeneous cellular networks with spatio-temporal traffic: Delay analysis and scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 6, pp. 1373–1386, 2017.

[38] X. Ge, B. Yang, J. Ye, G. Mao, C.-X. Wang, and T. Han, "Spatial spectrum and energy efficiency of random cellular networks," *IEEE Transactions on Communications*, vol. 63, no. 3, pp. 1019–1030, 2015.

[39] W. Zhang, D. Yang, H. Peng, W. Wu, W. Quan, H. Zhang, and X. Shen, "Deep reinforcement learning based resource management for dnn inference in industrial iot," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7605–7618, 2021.

[40] M. Bouvier, A. Valentian, T. Mesquida, F. Rummens, M. Reyboz, E. Vianello, and E. Beigne, "Spiking neural networks hardware implementations and challenges: A survey," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 15, no. 2, pp. 1–35, 2019.

[41] Y. Venkatesha, Y. Kim, L. Tassiulas, and P. Panda, "Federated learning with spiking neural networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 6183–6194, 2021.

[42] M. Xiao, Q. Meng, Z. Zhang, Y. Wang, and Z. Lin, "Training feedback spiking neural networks by implicit differentiation on the equilibrium state," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[43] P. Dayan and L. F. Abbott, *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2005.

[44] S. Santurkar, D. Tsipras, A. Ilyas, and A. Mądry, "How does batch normalization help optimization?" in *Proceedings of the 32nd international conference on neural information processing systems*, 2018, pp. 2488–2498.

[45] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," *Advances in neural information processing systems*, vol. 28, pp. 2017–2025, 2015.

[46] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1059–1068.

[47] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: learning the weight of feature interactions via attention networks," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 3119–3125.

[48] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 4905–4913.

[49] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1-4, pp. 17–37, 2002.

[50] I.-M. Comşa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, and J. Alakuijala, "Temporal coding in spiking neural networks with alpha synaptic function: Learning with backpropagation," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021.

[51] S. Luo, L. Yu, Z. Bi, and Y. Li, "Traffic sign detection and recognition for intelligent transportation systems: A survey," *Journal of Internet Technology*, vol. 21, no. 6, pp. 1773–1784, 2020.

[52] A. Jain, A. Mishra, A. Shukla, and R. Tiwari, "A novel genetically optimized convolutional neural network for traffic sign recognition: A new benchmark on belgium and chinese traffic sign datasets," *Neural Processing Letters*, vol. 50, no. 3, pp. 3019–3043, 2019.

[53] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7252–7261.

[54] T.-M. H. Hsu, H. Qi, and M. Brown, "Measuring the effects of non-identical data distribution for federated visual classification," *arXiv preprint arXiv:1909.06335*, 2019.

[55] R. H. Chan, C.-W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transactions on image processing*, vol. 14, no. 10, pp. 1479–1485, 2005.

[56] M. Horowitz, "1.1 computing's energy problem (and what we can do about it)," in *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, 2014, pp. 10–14.

Jiawen Kang received the Ph.D. degree from the Guangdong University of Technology, China in 2018. He has been a postdoc at Nanyang Technological University, Singapore from 2018 to 2021. He currently is a full professor at Guangdong University of Technology, China. His research interests mainly focus on blockchain, security, and privacy protection in wireless communications and networking.

Dusit Niyato (M'09-SM'15-F'17) is currently a professor in the School of Computer Science and Engineering, at Nanyang Technological University, Singapore. He received B.Eng. from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand in 1999 and the Ph.D. in Electrical and Computer Engineering from the University of Manitoba, Canada in 2008. His research interests are in the areas of the Internet of Things (IoT), machine learning, and incentive mechanism design.

Kan Xie received the Ph.D. degree in Control Science and Engineering from the Guangdong University of Technology, Guangzhou, China, in 2017. He joined the Institute of Intelligent Information Processing, Guangdong University of Technology, where he is currently an Associate Professor. His research interests include machine learning, non-negative signal processing, blind signal processing, smart grid, and Internet of Things.

Shengli Xie (M'01, SM'02, F'19) received the B.S. degree in Mathematics from Jilin University, China in 1983, the M.S. degree in Mathematics from Central China Normal University, China in 1995, and the Ph.D. degree in Control Theory and Applications from South China University of Technology, China in 1997. He is currently a Full Professor and the head of the Institute of Intelligent Information Processing, Guangdong University of Technology. He has co-authored two books and over 150 research papers in refereed journals and conference proceedings and was awarded Highly Cited Researcher in 2020. His research interests include blind signal processing, machine learning, and the Internet of Things. He was awarded the Second Prize of the National Natural Science Award of China in 2009. He is an Associate Editor of IEEE TSMC.

Zhe Zhang received the B.S. degree from the North China University of Water Resources and Electric Power, in 2020. He is currently working toward a Master's Degree in Information Statistics Technology at Heilongjiang University. His research interests are mainly federated learning, semi-supervised learning, and spiking neural networks.

Bo Li received the Ph.D. degree in Computer Science and Technology from the School of Intelligent Systems Engineering at Sun Yat-sen University, Guangzhou, China, in 2021. From September 2014 to June 2016, he was a master student in the school of engineering, at Sun Yat-sen University. He is currently a Post-Doctoral Research Fellow with the School of Automation, Guangdong University of Technology, Guangzhou, China. His current research interests include intelligent transportation systems, traffic information processing as well as big data technology. Dr. Li is currently a very active reviewer for some international journals and a member of the program committee for many international conferences.

Yi Wu received the B.S. and M.S. degrees in computer science from Heilongjiang University, Harbin, China, in 1985 and 2005, respectively. He is a Professor at the School of Data Science and Technology, Heilongjiang University. His current research interests include large-scale data processing, data mining, and database application.