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, internet of vehicles, spike neural networks, traffic sign recognition.

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].
We propose a new encoding scheme for SNNs based on the rate-based coding has limitations when facing blurry traffic sign images. Thus, we need a new encoding method to achieve high-precision 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 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 the challenge of recognizing traffic signs in various weather conditions and lighting environments.

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 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.
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 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 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 Table I.

### Table I

| Symbols | Description |
|---------|-------------|
| $K_r$   | The set of clients selected in round $r$ |
| $D_k$   | The $k$-th client local dataset |
| $C_k$   | The set of the $k$-th client after encoding |
| $L_k$   | The $k$-th client local loss function |
| $L(r)$  | The cross-entropy function |
| $\omega_r$ | The global model of round $r$ |
| $\omega_k$ | The local model of the $k$-th client |
| $f_\omega(r)$ | The prediction function on model $\omega$ |
| $s_i$   | The $i$-th sample |
| $y_i$   | 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 |
| $F_p$   | The client participation rate |
| $B$     | The number of local epochs |
| $E$     | The number of local epochs |
| $M$     | The side length of the output feature map |
| $N_l$   | The number of neurons in layer $l$ |
| $K_l$   | The energy consumption of layer $l$ |
| $s$     | The convolution kernel size |
| $c$     | The input feature size |
| $N_l$   | The number of channels |
| $t(l)$  | The input current at time $t$ |
| $u(l)$  | The membrane potential at time $t$ |
| $u_{rest}$ | The resting potential |
| $u_i$   | The membrane potential of the $i$-th neuron at time step $t$ |
| $o_i$   | The spike sequence value of neuron $i$ at time step $t$ |
| $\gamma_i$ | The learned weight of the BN layer |
| $\theta$ | The threshold voltage of spike neural network |
| $t$     | The time step of spike neural network |
| $\lambda$ | The leakage factor |
| $\alpha$ | The substitution rate |
| $w_{ij}$ | The weight of the connection between neuron $i$ and neuron $j$ |
| $w_{ij}$ | The weight of the $l$-th layer |
| $R_l$   | The spike rate of the $l$-th layer |
| $\tau_m$ | The time constant of membrane potential decay |
| $\mu$   | The parameters of Ditzhiet distribution function |

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 $\omega^*$, 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 $\omega_r$. Then, the selected clients use Stochastic Gradient Descent (SGD) to train the received global model $\omega_r$ on their local dataset $\mathcal{D}_k = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, y_i)\}$. For the $k$-th client, the following objective function needs to be minimized:

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

(1)

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

$$
\omega_{r+1}^k = \omega_r - \eta \triangledown L_k(\mathcal{D}_k; \omega),
$$

(2)

where $\eta$ is learning rate. After completing local training, each client uploads its local model update $\omega_{r+1}^k$ 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:

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

(3)

where $p_k = \frac{|\mathcal{D}_k|}{\sum_{k \in \mathcal{K}_r} |\mathcal{D}_k|}$ 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 $\omega^*$.

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 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 $\theta$, 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 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 $\theta$, 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 $u_{\text{rest}}$, 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:

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

(4)

where $\tau_m = RC$ is the time constant of membrane potential decay, $u(t)$ represents the neuron membrane potential at time $t$, $u_{\text{rest}}$ is the resting potential, $\theta$ 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 $u^t_i$ at time step $t$ as:

$$
\sum_{j} w_{ij} o_j^{t-1} + \frac{u_{\text{ref}} - u^{t-1}_i}{\tau_m} = \sum_{j} w_{ij} o_j^t + \frac{I_t}{\tau_m},
$$

(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, $o_{ij}$ 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:

$$o_{ij} = \begin{cases} 1, & \text{if } u_{ij} \geq \theta \\ 0, & \text{otherwise} \end{cases}.$$  \hspace{1cm} (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 without causing information loss. Thus, the cross-entropy loss function of SNNs can be expressed as:

$$L = - \sum_{j=1}^{c} y_j \log \left( \frac{e^{u_{ij}}}{\sum_{j=1}^{c} e^{u_{ij}}} \right),$$  \hspace{1cm} (7)

where $y$ is the ground-truth label, $c$ is the total number of classes in the training dataset, $u_{ij}$ 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:

$$\Delta W_{l} = \begin{cases} \sum_{i=1}^{t} \frac{\partial L}{\partial u_{ij}} \frac{\partial o_{ij}}{\partial u_{ij}} \frac{\partial u_{ij}}{\partial W_{ij}}, & \text{if } l = \text{hidden layer} \\ \sum_{i=1}^{t} \frac{\partial L}{\partial u_{ij}} \frac{\partial o_{ij}}{\partial W_{ij}}, & \text{otherwise} \end{cases}.$$  \hspace{1cm} (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:

$$\frac{\partial o_{ij}}{\partial u_{ij}} = \alpha \max \left\{ 0, 1 - \frac{u_{ij} - \theta}{\theta} \right\},$$  \hspace{1cm} (9)

where $\alpha$ is the substitution rate that controls the update amplitude of SNNs, $\theta$ 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 $\alpha$ is also related to the time step $t$. For a larger time step $t$, a smaller hyperparameter $\alpha$ should be set to avoid gradient explosion. Overall, the back-propagation 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: $W_{l} = W_{l} - \eta \Delta W_{l}$.

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 (BNNT).

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 BNNT is given as follows:

$$u_{ij} = \lambda u_{ij}^{t-1} + \text{BNNT}_{t} \sum_{j} w_{ij} o_{ij},$$

$$= \lambda u_{ij}^{t-1} + \gamma_{ij} \left( \sum_{j} w_{ij} o_{ij} - \mu_{ij} \right) \frac{1}{\sqrt{\sigma_{ij}^{2} + \epsilon}},$$  \hspace{1cm} (10)

where $\gamma_{ij}$ is the learned weight of the BN layer, $\epsilon$ is a constant that prevents the denominator from being 0, $\mu_{ij}$ and $\sigma_{ij}$ 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 $x_{ij} = \sum_{j} w_{ij} o_{ij}$, the gradient of weight $\gamma$ in a batch $b$ can be expressed as:

$$\Delta \gamma = \frac{\partial L}{\partial u_{ij}} \frac{\partial u_{ij}}{\partial \gamma} = \sum_{i=1}^{b} \frac{\partial L}{\partial u_{ij}} \frac{\partial o_{ij}}{\partial \gamma} \frac{\partial u_{ij}}{\partial \gamma}.$$  \hspace{1cm} (11)

Here, $\gamma = \frac{x_{ij} - \mu_{ij}}{\sqrt{\sigma_{ij}^{2} + \epsilon}}$. Therefore, a batch of weight updates for the $t$-th BN layer can be calculated as:

$$\gamma^{t} = \gamma^{t-1} - \eta \Delta \gamma^{t}.$$  \hspace{1cm} (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., \( X \rightarrow X' \)).

\[
x'_{ij} = \frac{x_{ij} - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}},
\]

where \( x_{\text{max}} \) and \( x_{\text{min}} \) 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 \( G_{s \times s} \sim N(\mu_g, \sigma^2_g) \) to simulate the area detected by biological neurons, as follows.

\[
\mu_g = \frac{1}{s} \sum_{i=1}^{s} \frac{2i - 1}{2(s - ks)} ,
\]

\[
\sigma_g = \frac{BCE}{s - ks},
\]

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:

\[
input = \begin{cases} 
1, & q_{ij} \geq x_{ij}' \cap q_{ij} \geq g_{ij} \\
-1, & q_{ij} \leq x_{ij}' \cap q_{ij} \leq g_{ij} \\
0, & \text{otherwise}
\end{cases},
\]

where \( g_{ij} \in G, x_{ij}' \in X', q_{ij} \in Q \).

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], 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 in the 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-based IoV system consists of a roadside unit (RSU, i.e., server) and \( K \) vehicles working as
Algorithm 1: FedSNN-NRFE Algorithm.

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

**Output:** The convergent SNN model $\omega^*$.

1. **Server executes:**
   1. Initialize a global SNN model $\omega_0$
   2. For each round $r = 0, 1, 2, \ldots$
   3. For each client $k \in S_r$ in parallel do
   4. $\omega_{r+1} = \sum_{k \in S_r} \rho_k \omega^k_{r+1}$
   5. For each local epoch $e \leftarrow 1$ to $E$
   6. **for** mini-batch $B$ **do**
   7. **for** layer $l$ from 1 to $L - 1$
   8. **for** time step $t$ from 1 to $T$
   9. $U^l_t = \lambda U^l_{t-1} + BNTT(l) (W_l, O^{l-1}_t)$
   10. **end** for
   11. **end** for
   12. **end** for
   13. **end** for
   14. **end** for
   
2. **ClientUpdate**($k$, $\omega^*$):
   1. For the $k$-th client
   2. $O_k \leftarrow$ NRFE ($\mathcal{D}_k$)
   3. For each local epoch $e \leftarrow 1$ to $E$
   4. For mini-batch $B$ do
   5. **for** layer $l$ from 1 to $L - 1$
   6. **for** time step $t$ from 1 to $T$
   7. $U^l_t = \lambda U^l_{t-1} + BNTT(l) (W_l, O^{l-1}_t)$
   8. **end** for
   9. **end** for
   10. **end** for
   11. **end** for
   12. **end** for
   13. **end** for
   14. **end** for
   
3. $\omega^k_{r+1} = \omega_r - \eta \nabla L_k(k, O_k; \omega_r)$

4. **end** for
5. **end** for
6. **end** for
7. **end** for
8. **end** for
9. **end** for
10. **end** for
11. **end** for
12. **end** for
13. **end** for
14. **end** for
15. **end** for
16. **end** for
17. **end** for
18. **end** for
19. **end** for
20. **end** for
21. **end** for
22. **end** for
23. **return** $\omega^*$ to server.

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 i9-9900 K 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.

### V. NUMERICAL RESULTS

**A. Experiment Setup**

1. **Dataset Under IID and non-IID Settings:** As shown in Fig. 4, the BelgiumTS dataset has 62 classes of $28 \times 28$ 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 $\mu$ of the Dirichlet distribution function [53].
In the experiment, we set the total number of clients \( K = 20 \), mini-batch size \( B = 8 \), learning rate \( \eta = 1 \times 10^{-3} \), and the threshold voltage \( \theta = 1.0 \). 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. As shown in Fig. 6(a), the 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 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 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 \( \alpha = 0.3 \).

For SNNs, too small a time step \( T \) and leak rate \( \lambda \) 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
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. (c) Different leak rates.

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

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 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 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.

Comparing the performance of different non-IID levels, we observe 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 45 nm 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:

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

(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 an accumulation (AC) operation, resulting in significant energy efficiency [41]. We define the spike rate $R_s(l)$ of each layer, i.e., the number of spikes $n_i^l$ 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
The $l$-th layer is given as follows:

$$R_s(l) = \frac{n_l^s}{N_l}.$$  \hspace{1cm} (23)

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

$$E_{CNN}(l) = FLOPs(l) \times (E_{Mul} + E_{Add}),$$  \hspace{1cm} (24)

$$E_{SNN}(l) = FLOPs(l) \times R_s(l) \times T \times E_{AC}.$$ \hspace{1cm} (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 rate of each layer. The total energy estimated by CNN is about 37.096 $\mu$J while that of SNN is about 4.773 $\mu$J, which is 7.772× more efficient than that of CNN.

### Table IV

| Layer Name | CNN (pJ) | SNN (pJ) | Spike Rate |
|------------|----------|----------|------------|
| Conv 1     | 2,167,603.2 | 203,212.8 | 3.0        |
| BN 1       | 80,281.6   | 12,344   | 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        |

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.

### VI. Conclusions

This paper presents 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] M. Rimol, “Gartner forecasts more than 740,000 autonomous-ready vehicles to be added to global market in 2023,” 2020. [Online] Available: https://www.gartner.com/en/newsroom/press-releases/2019-11-14-gartner-forecasts-more-than-740000-autonomous-ready-vehicles-to-be-added-to-global-market-in-2023

[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 Commun., vol. 7, no. 2, pp. 1992–1999, Apr. 2022.

[3] M. Li, J. Gao, L. Zhao, and X. Shen, “Adaptive computing scheduling for edge-assisted autonomous driving,” IEEE Trans. Veh. Technol., vol. 70, no. 6, pp. 5318–5331, Jun. 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 Trans. Veh. Technol., vol. 71, no. 2, pp. 1333–1348, Feb. 2022.

[5] Y. Ma, Z. Wang, H. Yang, and L. Yang, “Artificial intelligence applications in the development of autonomous vehicles: A survey,” IEEE J. Automat. Sinica, vol. 7, no. 2, pp. 315–329, Mar. 2020.

[6] B. Liu, C. Han, X. Liu, and W. Li, “Vehicle artificial intelligence system based on intelligent image analysis and 5G network,” Int. J. Wireless Inf. Netw., 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 Things J., vol. 7, no. 8, pp. 7751–7763, Aug. 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 Trans. Veh. Technol., vol. 70, no. 9, pp. 9355–9368, Sep. 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 Things J., vol. 6, no. 6, pp. 10700–10714, Dec. 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 Trans. Veh. Technol., vol. 69, no. 4, pp. 4298–4311, Apr. 2020.
[11] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, “Reliable federated learning for mobile networks,” IEEE Wireless Commun., vol. 27, no. 2, pp. 72–80, Apr. 2020.

[12] W. Zhang et al., “Optimizing federated learning in distributed industrial IoT: A multi-agent approach,” IEEE J. Sel. Areas Commun., vol. 39, no. 12, pp. 3688–3703, Dec. 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 Trans. Veh. Technol., vol. 69, no. 6, pp. 5773–5783, Jun. 2020.

[14] M. Pfeiffer and T. Pfeil, “Deep learning with spiking neurons: Opportunities and challenges,” Front. Neurosci., vol. 12, 2018, Art. no. 774.

[15] N. Skatchkovsky, H. Jang, and O. Simeone, “Federated neuromorphic learning of spiking neural networks for low-power edge intelligence,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2020, pp. 8524–8528.

[16] R. V. Rullien and S. J. Thorpe, “Rate coding versus temporal order coding: What the retinal ganglion cells tell the visual cortex,” Neural Comput., 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 Proc. AAAI Conf. Artif. Intell., 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,” Front. Neurosci., vol. 13, 2019, Art. no. 95.

[19] P. U. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” Front. Comput. Neurosci., vol. 9, 2015, Art. no. 99.

[20] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, “Road-signal detection and recognition based on support vector machines,” IEEE Trans. Intell. Transp. Syst., vol. 8, no. 2, pp. 264–278, Jun. 2007.

[21] D. Tabernik and D. Skočaj, “Deep learning for large-scale traffic-sign detection and recognition,” IEEE Trans. Intell. Transp. Syst., vol. 21, no. 4, pp. 1427–1440, Apr. 2020.

[22] T. Yang, X. Long, A. K. Sangaiah, Z. Zheng, and C. Tong, “Deep detection and recognition for DNN inference in Industrial IoT,” IEEE Trans. Veh. Technol., vol. 69, no. 8, pp. 8695–8718, Aug. 2021.

[23] M. Bouvier et al., “Spiking neural networks hardware implementations and challenges: A survey,” ACM J. Emerg. Technol. Comput. Syst., vol. 15, no. 2, pp. 1–35, 2019.

[24] Y. Venkatesha, Y. Kim, L. Tassiulas, and P. Panda, “Federated learning with spiking neural networks,” IEEE Trans. Signal Process., vol. 69, pp. 6183–6194, 2021.

[25] M. Xiao, Q. Meng, Z. Zhang, Y. Wang, and Z. Lin, “Training feedback spiking neural networks by implicit differentiation on the equilibrium state,” Adv. Neural Inf. Process. Syst., vol. 34, pp. 14516–14528, 2021.

[26] P. Dayan and L. F. Abbott, Theoretical Neurosciences: Computational and Mathematical Modeling of Neural Systems. Cambridge, MA, USA: MIT Press, 2005.

[27] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, “How does batch normalization help optimization?,” in Proc. 32nd Int. Conf. Neural Inf. Process. Syst., 2018, pp. 2488–2498.

[28] M. Jaderberg et al., “Spatial transformer networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28, 2015, pp. 2017–2025.

[29] G. Zhou et al., “Deep interest network for click-through rate prediction,” in Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2018, pp. 1059–1068.

[30] 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 Proc. 26th Int. Joint Conf. Artif. Intell., 2017, pp. 3119–3125.

[31] W. Luo, Y. Li, R. Urtasun, and R. Zemel, “Understanding the effective receptive field in deep convolutional neural networks,” in Proc. 30th Int. Conf. Neural Inf. Process. Syst., 2016, pp. 4905–4913.

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

[33] J.-M. Comsa, 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 Trans. Neural Netw. Learn. Syst., to be published, doi: 10.1109/TNNLS.2021.3071976.

[34] S. Luo, L. Yu, Z. Bi, and Y. Li, “Traffic sign detection and recognition for intelligent transportation systems: A survey,” J. Internet Technol., vol. 21, no. 6, pp. 1773–1784, 2020.

[35] 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 Process. Lett., vol. 50, no. 3, pp. 3019–3043, 2019.

[36] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, “Bayesian nonparametric federated learning of neural networks,” in Proc. Int. Conf. Mach. Learn., 2019, pp. 7252–7261.

[37] T.-M. H. Hsu, Q. Qi, and M. Brown, “Measuring the effects of non-identical data distribution for federated visual classification,” 2019, arXiv:1909.06335.

[38] R. H. Chan, C.-W. Ho, and M. Nikolova, “Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization,” IEEE Trans. Image Process., vol. 14, no. 10, pp. 1479–1485, Oct. 2005.

[39] M. Hornowitz, “‘I computing’s energy problem (and what we can do about it),” in Proc. IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, 2014, pp. 10–14.
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 the Internet of Things.

Zhe Zhang received the B.S. degree from the North China University of Water Resources and Electric Power, Zhengzhou, China, in 2020. He is currently working toward the master’s degree in information statistics technology from Heilongjiang University, Harbin, China. His research interests include 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, Sun Yat-sen University, Guangzhou, China, in 2021. From September 2014 to June 2016, he was working toward the master’s degree with the School of Engineering, Sun Yat-sen University. He is currently a Postdoctoral Research Fellow with the School of Automation, Guangdong University of Technology, Guangzhou, China. His current research interests include intelligent transportation systems, traffic information processing, and Big Data technology. He is currently a very active Reviewer for some international journals and a member of the program committee for many international conferences.

Jiawen Kang received the Ph.D. degree from the Guangdong University of Technology, Guangzhou, China in 2018. From 2018 to 2021, he was a Postdoc with Nanyang Technological University, Singapore. He is currently a Full Professor with the Guangdong University of Technology. His research focuses on blockchain, security, and privacy protection in wireless communications and networking.

Dusit Niyato (Fellow, IEEE) received the B.Eng. degree from the King Mongkut’s Institute of Technology Ladkrabang, Bangkok, Thailand, in 1999 and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008. He is currently a Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include the Internet of Things (IoT), machine learning, and incentive mechanism design.

Shengli Xie (Fellow, IEEE) received the B.S. degree in mathematics from Jilin University, Changchun, China in 1983, the M.S. degree in mathematics from Central China Normal University, Wuhan, China in 1995, and the Ph.D. degree in control theory and applications from the South China University of Technology, Guangzhou, 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 coauthored two books and more than 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 the recipient of the Second Prize of the National Natural Science Award of China in 2009. He is an Associate Editor for IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS.

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 currently a Professor with the School of Data Science and Technology, Heilongjiang University. His current research interests include large-scale data processing, data mining, and database application.