IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 December 2023, accepted 21 January 2024, date of publication 2 February 2024, date of current version 13 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3361837

RESEARCH ARTICLE

CNN-CLFFA: Support Mobile Edge Computing in Transportation Cyber Physical System

ASHOK BHANSALI[1], RAJ KUMAR PATRA[2], PARAMESHACHARI BIDARE DIVAKARACHARI[3], (Senior Member, IEEE), PRZEMYSŁAW FALKOWSKI-GILSKI[4], GANDLA SHIVAKANTH[5], AND SUJATHA N. PATIL[6]

[1]Department of Computer Engineering and Applications, GLA University, Mathura 281406, India
[2]Department of ECE, J.J. College of Engineering and Technology, Tiruchirappalli 620009, India
[3]Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bengaluru 560064, India
[4]Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, 80-233 Gdansk, Poland
[5]Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswaram 520002, India
[6]Department of Radiation Oncology (Pathology Laboratory), Thomas Jefferson University, Philadelphia, PA 19144, USA

Corresponding author: Przemysław Falkowski-Gilski (przemyslaw.falkowski@eti.pg.edu.pl)

ABSTRACT In the present scenario, the transportation Cyber Physical System (CPS) improves the reliability and efficiency of the transportation systems by enhancing the interactions between the physical and cyber systems. With the provision of better storage ability and enhanced computing, cloud computing extends transportation CPS in Mobile Edge Computing (MEC). By inspecting the existing literatures, the cloud computing cannot fulfill the requirements in transportation CPS like lower context-awareness and latency. For enhancing the context-awareness and reducing the latency in a realistic MEC environment, an efficient portable deep learning model: Convolutional Neural Network (CNN) with Chaotic Lévy Flight based Firefly Algorithm (CLFFA) is implemented in this article. In the CNN model, the CLFFA selects the appropriate hyper-parameters or reduces the redundant parameters that results in minimal model size and inference latency than the traditional CNN models. Additionally, the CNN-CLFFA model significantly outperformed the existing models by means of recall, accuracy, F1-score, and precision on the benchmark datasets like German Traffic Sign Recognition Benchmark (GTSRB), MIOvision Traffic Camera Dataset (MIO-TCD) classification, and VCifar-100 datasets. The numerical analysis demonstrates that the CNN-CLFFA model obtained maximum accuracy of 99.02%, 99.11%, and 99.03% on the VCifar-100, MIO-TCD, and GTSRB-T datasets, which are superior to the traditional models.

INDEX TERMS Convolutional neural network, cyber physical system, firefly algorithm, intelligent transportation system, mobile edge computing.

I. INTRODUCTION

In recent decades, the integration of transportation systems with the advanced communication technologies gained attention among the researchers [1]. The transportation CPS is expected to bring more improvements in the present transportation systems by means of sustainability, efficiency, and safety [2]. In the transportation CPS, Internet of Things (IoT) facilitates the interaction among computing platforms and physical environments, where it helps in extracting the useful information from the huge volume of IoT data to make timely and intelligent decisions [3], [4]. Currently, the deep learning and machine learning algorithms are efficient in extracting the useful information from the IoT data [5]. The deep learning algorithms have more stringent demands in the computing-devices, but the majority of the IoT devices have not fulfilled the demands, because of the computing and storage limitations [6]. These limitations are overcome with the introduction of cloud computing in the IoT devices by performing off-loading computing tasks in the cloud servers [7].

In the transportation CPS, cloud computing has better computing ability and data storage, but it cannot fulfill the requirements of context-awareness and low latency [8], [9]. Potentially, the MEC overcomes the cloud computing

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

limitations by performing off-loading tasks in the edge servers, which are deployed at access points and base stations. In this article, the traffic signs and vehicle recognition are considered as an example in the Intelligent Transportation System (ITS) and transportation CPS [10], [11]. The traffic signs and vehicle recognition plays a crucial role in the transportation CPS. Here, the user requests are redirected to the nearest edge servers instead of remote cloud servers, which are initiated from the mobile devices [12]. In the similar way, the context-awareness is greatly improved and latency is reduced by implementing a portable CNN model [13], [14]. Furthermore, the mobile devices in transportation CPS have limited storage capacity and computing power. Conventional deep learning models are ineffective for deployment on mobile devices, because of their resource constraints. Whereas, the portable CNN model is efficient and lightweight, which is well suited for deployment on resource constrained devices. Additionally, transportation CPS need real time processing for tasks like traffic signs and vehicle recognition. Without the need for communication with remote servers, the portable CNN model enables timely decision making on mobile devices. In this article, the hyper-parameters of the CNN model are selected by CLFFA that results in maximal classification accuracy with minimum model size and inference latency. The contributions are as follows:

- We used Principal Component Analysis (PCA) jittering and data augmentation (color, rotation, and Gaussian augmentation) techniques for reducing the overfitting and class imbalance problems. Specifically, PCA jittering is robust to viewpoint changes and illumination variations, which affect the appearance of vehicles and traffic signs. On the other hand, data augmentation increases the image variability that enables the CNN-CLFFA model in learning more generalized features, which makes this model more resilient to the variations experienced in the real world scenario.
- We integrated CLFFA with the CNN model to choose the optimal hyper-parameters that reduces the redundant parameters and improves the efficacy of the activation function. Additionally, the selection of appropriate hyper-parameters diminishes the model size and it results in limited inference latency and computational time. Furthermore, during the training process, the hyper-parameter selection leads to faster convergence rate.
- The effectiveness of the proposed CNN-CLFFA model is validated on two modules such as MEC module (Jetson TX2) and personal computer. The evaluation measures: accuracy, recall, F1-score, model size, inference latency, and precision confirmed that the proposed CNN-CLFFA model obtained better results in the personal computer than the Jetson TX2 module.

This article is prepared in the following manner, a few papers on the topic of ITS are surveyed in Section II. The details about undertaken methodologies, results, and the conclusion of the CNN-CLFFA model are presented in Sections III, IV, and V, respectively.

II. RELATED WORKS

Chen et al. [15] have presented a Deep Belief Network (DBN) model for detecting attacks in MEC. Here, the active feature learning improves accuracy in the DBN model, and it is better related to other comparative machine learning models. In this study, the DBN model uses 512 hidden units for learning the attack features, and further, a contrastive divergence method was employed to update the parametric values in the DBN model. In the experimental evaluation, ten datasets were utilized for conducting experiments, and the presented DBN model gained 6% higher accuracy than the comparative models. However, the presented DBN model was computationally costly, because it includes complex data models. Additionally, Shen et al. [16] have implemented a genetic algorithm in green ITS for optimizing the locations of public parking. In the ITS, the presented genetic algorithm not only supports parking guidance and assists in public parking allocations.

In the ITS, Ashraf et al. [17] have presented an effective Intrusion Detection System (IDS) for finding the suspicious activities in vehicles to infrastructure networks, in-vehicles networks, and vehicles to vehicles communications. Here, the autoencoder model was integrated with the Long Short Term Memory (LSTM) network for recognizing the intrusive events in the Autonomous Vehicles (AVs). The presented IDS's performance was tested on two online datasets, and the obtained results confirmed that the presented IDS has higher detection accuracy than the existing techniques. On the other hand, the presented IDS has high time complexity and suffers from the vanishing gradient problems.

Zhou et al. [18] have presented a new light-weighted stacked CNN model for supporting MEC in transportation CPS. The presented stacked CNN model includes compression layers with factorization convolutional layers to improve the context-awareness and reduce the latency in MEC. The results showed that the presented stacked CNN model decreases the unneeded parameters by diminishing the size of the model, and maintains higher accuracy than the traditional CNN models. Here, the effectiveness of the stacked CNN model was evaluated by conducting an experiment in a real time MEC platform. The results showed that the stacked CNN model effectively preserves the model size and maintains high accuracy in a real-time MEC platform. However, the stacked CNN model was computationally costly.

In the application of ITS, Kumar et al. [19] have presented an effective system for controlling the traffic lights. The presented system runs in three modes, such as: emergency, priority, and fair. Here, the Fuzzy Inference System (FIS) was used to select an appropriate mode (emergency, priority, and fair) based on the traffic condition, and the deep reinforcement learning approach switches the traffic lights (yellow, green, and red) in dissimilar phases. The obtained

results showed the effectiveness of the presented system over the existing systems in light of different evaluation measures. The reinforcement learning causes an overload and its maintenance cost was too high.

Rathore et al. [20] have presented a new transport control framework, which exploits sensor technology and CPS for effective decision-making. The presented framework utilizes the information of road conditions, speed of the vehicles, traffic intensity, and travel time for constructing the city graphs and it represents the road networks. The conventional graph method needs authorities and commuters for developing an optimal and smart transportation system. Here, an Apache GraphX software tool was used for processing the data streams. The presented framework's performance was analyzed by means of processing time and throughput, and the achieved results were superior to the existing systems.

Currently, the CPS is incorporated with several physical systems (industries, ITS, and cities) for improving its comfort, efficiency, energy, and intelligence. In the present scenario, the ITS highly relied on the Traffic Flow Prediction (TFP). Jain et al. [21] have used Adaptive Neuro FIS (ANFIS) model for effective TFP and energy management in ITS. In this literature, the ANFIS model computes engine torque, and further, the combination of a fuzzy wavelet neural network with sailfish optimization algorithm was employed for estimating the traffic flow in the ITS. The experimental findings on a benchmark dataset demonstrated that the presented model outperformed the existing models. However, the ANFIS model has four concerns in the ITS such as curse of dimensionality, difficulty in the selection of membership function, computationally expensive, and the interpretability loss.

In addition to this, the Traffic Sign Detection (TSD) is crucial in ITS, but it is quite challenging. A robust, fast, and real time TSD supports and relieves the drivers, where it helps in improving the driving comfort and safety. Rai et al. [22] have incorporated the You Only Look Once (YOLO) v5 model with the CLEAR framework for effective TSD in adverse climatic conditions. The evaluation performed on the open image dataset showed the efficacy of the presented model over the existing models by means of recall and mean average precision. The undertaken YOLO v5 model was extremely fast, but computationally costly.

Liu et al. [23] have integrated reinforcement learning with the bidirectional LSTM network for improving the energy efficiency of CPS in ITS. Zhou et al. [24] have presented a Wide Attention Deep Composite (WADC) model for TFP. The presented WADC model has two modules: deep composite module and wide attention module. The wide attention module extracts the features from the traffic flow data, and the deep composite model has LSTM and CNN components for generalizing the extracted features. The results revealed that the presented WADC model obtained superior performance than the existing nine deep learning and machine learning models. However, the incorporation of two deep learning models increases the complexity and processing time.

Wasim et al. [25] have implemented CNN model for academic activity detection in the smart CPS. The CNN model effectively recognizes the academic activities in the real time campus dataset, and its performance was validated by means of memory requirement, computational time, and accuracy. As discussed in previous literature, the CNN model obtained superior performance in academic activity detection, but expensive in other applications like co-operative control of multi-agent systems [26], [27].

The conventional CNN model is computationally intensive because it requires significant memory and processing power. Mobile devices have limited computational resources, in the context of transportation CPS. Resource intensive models like CNN lead to delays that affect tasks like traffic signs and vehicle recognition. The conventional CNN model struggles with the precise detection of smaller objects, and is sensitive to variations in weather, environmental, and lighting conditions. These drawbacks affect the practical deployment of traditional CNN models in this system. In order to highlight the above-stated concerns, a portable deep learning model is proposed for supporting MEC in transportation CPS.

III. METHODOLOGY

In recent times, several transportation CPS applications are employed in the realistic traffic conditions, particularly related to the traffic signs and vehicle recognition. Traditionally, machine and deep learning algorithms are utilized in traffic signs and vehicle recognition. Compared to machine learning algorithms, deep learning algorithms are advanced in computer vision and image processing applications. Usually, deep neural networks contain more parameters with multiple layers that results in larger model size and requires strong processing devices. Therefore, a portable CNN-CLFFA model is proposed in this article for effective traffic signs and vehicle recognition. The steps involved in the proposed framework are as follows:

- Input data are collected from GTSRB, MIO-TCD, and VCifar-100 datasets.
- Image pre-processing is accomplished using PCA Jittering, color augmentation, rotation augmentation, and Gaussian augmentation.
- Traffic signs and vehicle recognition is performed using CNN-CLFFA model.
- Performance analysis is done using the measures like accuracy, F1-score, recall, model size, inference latency and precision.

A. IMAGE PRE-PROCESSING

After acquiring the images from GTSRB, MIO-TCD, and VCifar-100 datasets, the image pre-processing is performed utilizing PCA jittering and data augmentation (color, rotation, and Gaussian augmentation) techniques [28], [29]. In this scenario, the class imbalance and overfitting problems are resolved by adopting data augmentation techniques with an oversampling approach. The oversampling approach

FIGURE 1. Sample-augmented images.

optimizes the images in every class and then averages the class distributions. The oversampling approach efficiently resolves the problem of class-imbalance in the GTSRB, MIO-TCD, and VCifar-100 datasets [30], [31]. The data augmentation techniques avoids overfitting problems by obtaining a number of images with dissimilar effects [32].

- **Gaussian augmentation:** Additional images are generated with the inclusion of Gaussian noise.
- **Rotation augmentation:** Augmented images are generated by rotating the original images clockwise between the degrees of 0 to 360.
- **Color augmentation:** It adjusts the contrast, brightness, and color saturation of the original images.
- **PCA jittering:** Feature vectors and feature values are computed from the images by calculating the standard deviation and mean values from the Red Green Blue channels. The sample-augmented images are shown in Fig. 1.

B. TRAFFIC SIGNS AND VEHICLE RECOGNITION

After pre-processing the acquired images, the traffic signs and vehicle recognition is performed by implementing a stacked CNN model [33], [34]. Due to the higher learning capacity in the image representations, the stacked CNN model is effective in image processing and computer vision applications. In the initial phase of the CNN model, the pre-processed images are passed to the convolutional layers, which have kernel filters for learning and convolving the vectors from the pre-processed traffic sign and vehicle images [35]. In the convolutional layers, the kernel filter performs nonlinear activation, and it is mathematically depicted in (1).

$$a_{i,j} = f\left(\sum_{m=1}^{M}\sum_{n=1}^{N} w_{m,n} \times x_{i+m,j+n} + b\right) \quad (1)$$

where, $w_{m,n}$ is indicated as convolutional weight matrix, b is represented as bias, and f is indicated as nonlinear activation function [36]. In the nonlinear activation $a_{i,j}$, the neurons (i, j) are inter-connected with the upper neurons $x_{i+m,j+n}$. In this scenario, the ReLU activation function $\sigma\ (x)$ is utilized for measuring the nonlinear functions and feature maps in the convolutional layers. The mathematical expression of the ReLU activation function $\sigma\ (x)$ is indicated in (2).

$$\sigma\ (x) = max(0, x) \quad (2)$$

The convolutional kernels and the hidden vectors are extracted from the pre-processed images in the convolutional layers. In this scenario, we considered the kernel size of 3×3, stride of four, and convolutional kernels of 64 for extracting vectors (learning parameters) in the first convolutional layer. The deep feature extraction is accomplished in the next convolutional layer with the kernel size of 3×3, stride of two, and convolutional kernels of 128 [37].

Next is the pooling layer, the nonlinear sub-sampling process is carried out in the pooling layer for reducing the vector's size. In this layer, the two commonly utilized pooling methods are average pooling and max pooling. The maximum values are selected in the max pooling operation, and the average value of the elements are considered in the average pooling operation. Here, the down sampling process is performed in the two pooling layers with max-pooling operation in order to maintain the dominant vectors. The activation set is denoted as P and the pooling region is specified as E. The activation set and the average pooling (AP) operation are mathematically depicted in (3) and (4).

$$P = \{p_k | k \in E\} \quad (3)$$

$$AP = \frac{\sum P_E}{|P_E|} \quad (4)$$

In addition, the mathematical expression of max pooling (MP) operation is indicated in (5). Where, a set of cardinal numbers x is denoted as $|x|$.

$$MP = max(P_E) \quad (5)$$

The output layer is the fully connected or flatten layer, which is used along with the softmax function. In the flatten layer, every node is inter-connected with the nodes of the upper layer for merging the extracted vectors. Here, the softmax function is used for converting the extracted vectors into probability vectors in order to perform classification. The softmax function S_j is mathematically presented in (6). Where, q is denoted as the flatten layers with output vectors, N is indicated as number of classes, and the output vector of j^{th} value is denoted as a_j.

The architecture of the stacked CNN model is specified in Fig. 2. For reducing the size of CNN model, the hyper-parameters are selected by the CLFFA, which are listed as follows: momentum is 0.9, optimizer is Adam, activation function is ReLU, mini batch size is 500, max epochs is 100, learning rate drop period is 5, learning rate drop factor is 0.2, loss function is binary cross entropy, and the initial learning rate is 0.001.

In this context, the learning rate is a crucial hyper-parameter, which influences convergence speed of CNN

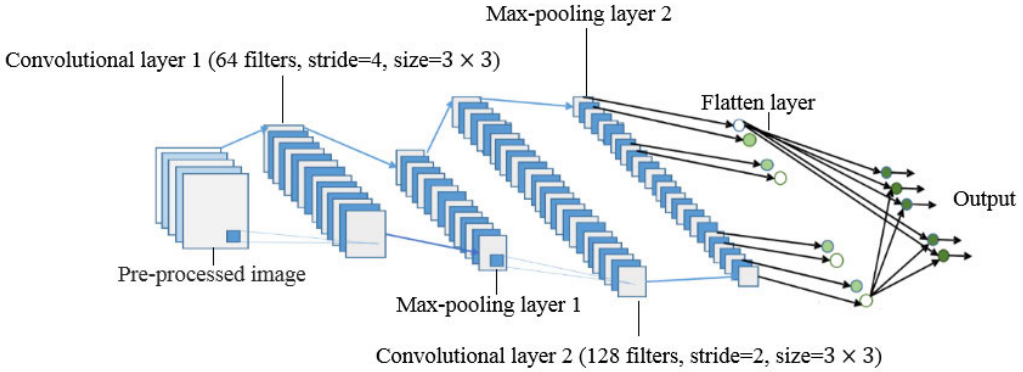

FIGURE 2. Architecture of the stacked CNN model.

model while training the data. Here, the CLFFA finds the optimal learning rate of 0.001 and it avoids problems like overshooting and slow convergence. Additionally, the CLFFA finds the optimal batch size of 500, which balances model performance and efficiency for traffic sign and vehicle recognition. During data training, the batch size directly affects the memory requirements and convergence speed. The regularization techniques like dropout plays a vital role in preventing overfitting problems. The optimal selection of dropout rate/regularization parameter improves the CNN model's generalization ability. During data training, the CLFFA finds the optimal point for early stopping and it overcomes overfitting problems by stooping the training process, while the CNN model begins to exhibit signs of decreasing performance on the validation set.

$$S_j = \frac{e^{q_j}}{\sum_{k=1}^{N} e^{q_k}} \tag{6}$$

Generally, the FA follows the firefly interacting behavior by using its flashing lights. This optimization algorithm considers all the fireflies are unisex, which means brighter fireflies attract residual fireflies [38], [39], [40]. According to the objective function, the firefly's brightness is directly proportional to the attractiveness of the fireflies. Here, the intensity of the light $I(r)$ is computed by performing inverse square law, which is specified in (7).

$$I\,(r) = I_0 \times e^{-\rho \times r^2} \tag{7}$$

where, the actual light intensity is represented as I_0 and the coefficients of light absorption are specified as ρ. Additionally, the attractiveness of the firefly is represented as $\beta(r)$ and it is mathematically expressed in (8). Where, the attractiveness is β_0 in the condition of $\rho = 0$. In the conventional FA, the Cartesian distance metric is utilized for computing the distance between fireflies i and j at x_i and x_j. The mathematical expression of the Cartesian distance metric r_{ij} is denoted in (9). On the other hand, the firefly's positions are updated by utilizing (10).

$$\beta\,(r) = \beta_0 \times e^{-\rho \times r^2} \tag{8}$$

$$r_{ij} = \sqrt{\sum_{p=1}^{u} (x_{i,p} - x_{j,p})^2} \tag{9}$$

$$x_{i,t+1} = x_{i,t} + \beta_0 \times e^{-\rho \times r_{ij}^2} \times \left(x_{j,t} - x_{i,t}\right) + \alpha \times (rand - 0.5) \tag{10}$$

where, *rand* is represented as the uniform random vector, $x_{i,p}$ is denoted as the spatial coordinate of i^{th} firefly at p^{th} component, t is indicated as time, and the term $\rho \in [0, 1]$ at u^{th} dimensional space [41], [42].

The traditional FA and other conventional algorithms like grid search, random search, and genetic algorithm have a drawback of lower exploration and exploitation ability in the global and local search spaces that result in poor convergence rate. To overcome the aforementioned drawback, a new CLFFA is implemented in this article for hyper-parameter selection, here; the fitness function is the hyper-parameters of the CNN model. It is analyzed based on the performance metrics like precision, recall, F1-score, and accuracy. The CLFFA uses the search pattern of Lévy Flight Distribution (LFD) and chaotic map (Gauss/Mouse map) instead of the random parameter β in orderto strengthen the exploration and exploitation ability of the algorithm in the global and local search spaces, where this process efficiently improves the convergence speed. In the CLFFA, the firefly's positions are updated based on LFD, which is mathematically depicted in (11).

$$x_{i,t+1} = x_{i,t} + \beta_0 \times e^{-\rho \times r_{ij}^2} \times \left(x_{j,t} - x_{i,t}\right) + \alpha \times sign(rand - 0.5) \otimes Lévy \tag{11}$$

The parameter β in (11) is replaced by the (12) and (13). Where, the randomization parameter is represented as α and the Hadamard product is denoted as $\otimes$. The $sign(rand - 0.5)$ accomplishes a random direction, if the random step is drawn utilizing the Lévy flights, where it helps in enhancing the ability of global search. The mathematical expressions of Lévy random number and LFD are denoted in (14) and (15).

$$\beta = \left(\beta_{chaos}^{t} - \beta_0\right) e^{-\gamma r_{ij}^2} + \beta_0 \tag{12}$$

$$\beta_{chaos}^{t} = \begin{Bmatrix} 0 & \beta_{chaos}^{t-1} = 0 \\ 1/\beta_{chaos}^{t} mod(1) & otherwise \end{Bmatrix} \tag{13}$$

$$Lévy\,(\eta) \sim \frac{\emptyset \times \mu}{|\vartheta|^{1/\eta}} \tag{14}$$

$$Lévy\,(\eta) \sim \mu = t^{-1-\eta}, (0 \leq \eta \leq 2) \tag{15}$$

where, the terms ϑ and μ are represented as standard distributions, and $\emptyset$ is calculated using (16).

$$\emptyset = [\frac{\tau(1+\eta) \times sin(\pi \times \frac{\eta}{2})}{\tau((1+\frac{\eta}{2}) \times \eta \times 2^{(\eta-1)/2}}]^{1/\eta} \tag{16}$$

where, τ is represented as a gamma function and η= 1.5. The parameters fixed in the CLFFA are: gamma is 1, beta is 1, threshold value is 0.5, theta is 0.97, number of population is 30, and the number of iteration is 100. The proposed CLFFA terminates, once the maximum number of iterations is reached. The numerical analysis of the proposed CNN-CLFFA model is presented in Section IV, and the pseudocode of the CLFFA and the steps involved in the proposed CNN-CLFFA model is given as follows.

Pseudocode of the CLFFA

Input: Set light absorption coefficients, iteration *iter*, uniform random vector, initial attractiveness, and maximum number of iteration *maxiter*.
Output: Selection of best hyper-parameters for the CNN model
Generate random solutions along with the firefly's dimension.
Compute the threshold limit value.
Estimate the fitness value of every firefly.
While (***iter*** $<$ ***maxiter***)
Firefly's positions are updated utilizing (10).
If ***rand*** $<$ 0.5
  Firefly's positions are updated based on LFD utilizing equation (11).
End If
Sort the solutions based on the fitness value and save the present best solution.
Return the best solution.

Steps involved in the proposed CNN-CLFFA model

Step 1: First, set the parameters of the CLFFA like number of iterations, absorption coefficient, initial attractiveness, randomness, and number of fireflies.

Step 2: Compute light intensity for every firefly and further, calculate the firefly's attractiveness utilizing (8).

Step 3: Compute the firefly's movement with less brightness towards the firefly's with more brightness.

Step 4: Then, the exploration and exploitation ability of the FA algorithm is improved by including LFD and chaotic map (instead of the random parameter β) in the global and local search spaces by using (12)-(16).

Step 5: Further, the CLFFA is employed for selecting the hyper-parameters of the CNN model such as initial learning rate, loss function, drop factor of learning rate, drop period of learning rate, number of epochs, batch size, momentum, activation function, and optimizer.

Step 6: Train the CNN model by using the hyper-parameters and evaluate the fitness function.

Step 7: Repeat the steps 2 to 6, until the maximum number of iterations is reached.

TABLE 1. Details about system configuration.

Personal computer	
Operating system	Win-64bit
RAM	128 GB
Processor	Intel core i9 12th generation
GPU	NVIDIA GeForce RTX 3080 Ti
Jetson TX2 module	
Operating system	Win-64bit
RAM	128 GB
Processor	NVIDIA denver2 with ARM Cortex A57
GPU	256-core NVIDIA Pascal

Step 8: Compare the performance of the proposed model with the existing models by using the evaluation measures like accuracy, F1-score, recall, model size, inference latency, and precision.

IV. RESULTS AND DISCUSSION

The proposed CNN-CLFFA model is implemented on two modules: MEC module (Jetson TX2) and personal computer. Here, Keras 2.0 is the software framework used for experimental evaluation, and it runs on the Win-64bit operating system. For MEC, the Jetson TX2 module is developed with lower power consumption and smaller size. The embedded platform includes 128GB LPDDR4 memory, Hex-core ARMv8 Central Processing Unit (CPU), and 256-core NVIDIA Pascal Graphics Processing Unit (GPU). The details about system configuration are specified in Table 1.

A. EVALUATION MEASURES

In this article, the efficacy of the proposed CNN-CLFFA model is evaluated on the VCifar-100, MIO-TCD classification, and GTSRB datasets in light of precision, recall, model size, F1-score, inference latency, and accuracy [43]. The accuracy is determined as the ratio of total number of classifications to the number of correct classifications. Additionally, the file size (MB) and the number of parameters of the trained models are considered for evaluating the model size and inference latency on the VCifar-100, MIO-TCD classification, and GTSRB datasets. The evaluation measure: recall determines the ability of a CNN-CLFFA model in finding the relevant data points of a class. Similarly, the precision determines the ability of a CNN-CLFFA model in returning only the data points of a specific class, and F1-score is a harmonic mean value of recall and precision. The mathematical formulas of accuracy, recall, F1-score, and precision are given in (17)-(20). Where, FN, FP, TP, and TN are indicated as false negative values, false positive values, true positive values, and true negative values.

$$Accuracy = \frac{TP + TN}{TN + TP + FN + FP} \times 100 \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \times 100 \tag{18}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN} \times 100 \tag{19}$$

FIGURE 3. Sample vehicle images in VCifar-100 dataset.

TABLE 2. Statistics about the MIO-TCD classification dataset.

Types of vehicles/classes	Testing images	Training images
Background	40,000	160,000
Single unit truck	1,280	5,120
Articulated truck	2,587	10,346
Work-Van	2,422	9,679
Motorcycle	495	1,982
Non-motorized vehicle	438	1,751
Pickup truck	12,727	50,906
Car	65,131	260,518
Bus	2,579	10,316
Pedestrian	1,565	6,262
Bicycle	571	2,284
Total	129,795	519,164

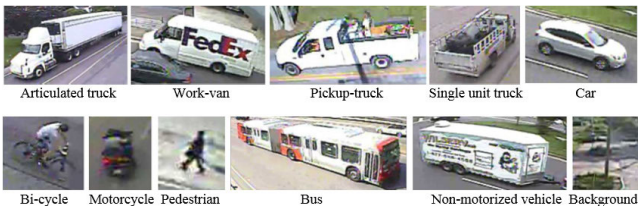

FIGURE 4. Sample images in MIO-TCD classification dataset.

$$Precision = \frac{TP}{TP + FP} \times 100 \tag{20}$$

B. DATASET DESCRIPTION

The VCifar-100 dataset includes vehicle images, which are extracted from the Cifar-100 dataset, where it has 60,000 vehicle images (32×32) and 100 classes. Particularly, the VCifar-100 dataset has five classes such as trains, motorcycles, trucks, buses, and bicycles. In this dataset, the vehicle images in every class are identical, so it avoids class imbalance problems. The sample vehicle images in VCifar-100 dataset are mentioned in Fig. 3.

In recent times, the MIO-TCD classification is one of the largest motorized traffic analysis datasets, where it includes eleven classes: bus, pedestrian, motorcycle, background, articulated truck, bicycle, car, single unit truck, work van, non-motorized vehicle, and pickup truck. The MIO-TCD classification dataset has 129,795 testing images and 519,164 training images, which are recorded using surveillance cameras placed across the United States and Canada. The statistics about the MIO-TCD classification dataset and its sample images are graphically given in table 2 and Fig. 4.

Additionally, the GTSRB is one of the extensively utilized datasets in TSD, which has 50,000 images and 40 classes. In this scenario, we selected three main types such as attention signs, direction signs, and speed limit signs. The sample images in the GTSRB dataset are graphically depicted in Fig. 5. The data augmentation and oversampling techniques resolve the class imbalance problem in the GTSRB dataset. In order to simplify the discussion, the GTSRB dataset is categorized into three types: speed limit sign images are considered as GTSRB-1, direction sign images are represented as GTSRB-2, attention sign images are denoted as GTSRB-3, and the combination of three traffic signs indicates GTSRB-T (Total). Furthermore, the details about the undertaken datasets are given in Table 3.

FIGURE 5. Sample images in GTSRB dataset.

TABLE 3. Details about the datasets.

Datasets	Total images	Classes
VCifar-100	60,000	100
MIO-TCD classification	648,959	11
GTSRB	50,000	40

TABLE 4. Results of CNN-CLFFA model and the existing models by means of model size and inference latency.

Models	Number of parameters	Model size (MB)	Inference latency (seconds)
RNN	4,385,208	36.5	0.40
LSTM	3,900,903	30.4	0.43
Bi-LSTM	3,659,762	27.9	0.39
VGG-16	15,303,222	119.2	0.36
GRU	2,476,829	19.9	0.22
AlexNet	756,382	6	0.19
VGG-19	603,445	4.8	0.08
CNN	600,212	4.7	0.08
CNN-FA	566,342	4.2	0.07
CNN-CLFFA (Jetson TX2)	509,344	3.8	0.02
CNN-CLFFA (Personal computer)			

C. QUANTITATIVE STUDY

In this research article, the CNN-CLFFA model's efficacy is validated with nine existing models such as Recurrent Neural Network (RNN), LSTM, Bi-directional LSTM (Bi-LSTM), Visual Geometry Group (VGG)-16, Gated Recurrent Unit (GRU), AlexNet, VGG-19, CNN, and CNN-FA. The common parameters of RNN, LSTM and Bi-LSTM networks are: 100 hidden layers with 100 units, activation functions are sigmoid and tangent, dropout rate is 0.5, batch size is 64, learning rate is 0.001, and network type is fully connected. The VGG-16 comprises sixteen layers in that three layers

TABLE 5. Results of CNN-CLFFA model and the existing models in light of classification accuracy.

	Classification accuracy (%)					
Models	VCifar-100	MIO-TCD	GTSRB-1	GTSRB-2	GTSRB-3	GTSRB-T
RNN	89.75	89	88.62	89.07	88.44	89.75
LSTM	90.12	89.27	90.80	89.77	88.60	90
Bi-LSTM	90.78	90.05	90.86	90.43	89.76	91.12
VGG-16	91.28	91.20	91.02	91.22	90	92
GRU	92	93.24	92.30	92.40	92.10	94.90
AlexNet	93.25	95.45	97.22	95.63	94.38	95.28
VGG-19	96.44	97.17	98.06	96.35	98.22	98.10
CNN	96.90	97.22	98.40	97.87	98.10	98.45
CNN-FA	97.50	97.23	98.80	98.12	98.55	98.76
CNN-CLFFA	99.02	99.11	99.12	99.32	99.21	99.03

TABLE 6. Results of CNN-CLFFA model and the existing models in light of precision.

	Precision (%)					
Models	VCifar-100	MIO-TCD	GTSRB-1	GTSRB-2	GTSRB-3	GTSRB-T
RNN	88	87.83	87.09	88.25	88.40	88.24
LSTM	88.18	88.10	88.87	88.70	88.68	89.26
Bi-LSTM	89	89.09	89.98	90.40	90.15	90.15
VGG-16	90.26	90.26	90.56	90.80	90.44	90.32
GRU	93.02	94.60	93.42	93.44	92.66	93.96
AlexNet	94.32	96.50	96.28	94.68	94.78	96.88
VGG-19	97.40	98.19	98.78	97.42	98.28	98.42
CNN	97.65	98.20	98.82	97.92	98.16	98.28
CNN-FA	98.11	98.20	98.90	98.33	98.27	98.90
CNN-CLFFA	99.14	99.32	99.08	99.30	99.12	99.10

are fully connected and thirteen layers are convolutional. The AlexNet has eight layers: three fully connected layers with ReLU activation function and five convolutional layers with max-pooling operation. Similarly, the GRU model has 100 hidden layers with 100 units, activation function is ReLU, dropout rate is 0.5, batch size is 32, learning rate is 0.001, optimizer is Adam, loss function is mean absolute error, and network type is fully connected. Lastly, the VGG-19 has nineteen layers in that three are fully connected layers and sixteen are convolutional layers.

The results of CNN-CLFFA model and the existing models (RNN, LSTM, Bi-LSTM, VGG-16, GRU, AlexNet, VGG-19, CNN, and CNN-FA) by means of model size and inference latency are depicted in Table 4. By viewing Table 4, the CNN-CLFFA model has a minimal size of 3.8 MB and inference latency of 0.02 seconds by reducing the number of parameters to 509,344. However, the comparative models (RNN, LSTM, Bi-LSTM, VGG-16, GRU, AlexNet, VGG-19, CNN, and CNN-FA) have size of 36.5 MB, 30.4 MB, 27.9 MB, 119.2 MB, 19.9 MB, 6 MB, 4.8 MB, 4.7MB, and 4.2 MB.

The obtained results of CNN-CLFFA model and the existing models (RNN, LSTM, Bi-LSTM, VGG-16, GRU, AlexNet, VGG-19, CNN, and CNN-FA) in terms of classification accuracy are denoted in Table 5. By viewing Table 5, the proposed CNN-CLFFA model has high classification accuracy of 99.02%, 99.11%, 99.03%, 99.21%, 99.32%, and 99.12% on the datasets like VCifar-100, MIO-TCD, GTSRB-T, GTSRB-3, GTSRB-2, and GTSRB-1, here, the results are analyzed on the personal computer. The obtained experimental results revealed that the proposed CNN-CLFFA model is feasible and attributable in reducing the redundant and unnecessary parameters than the conventional models. As depicted in Tables 5, 6, and 7, the proposed CNN-CLFFA model not only improves classification accuracy, but also reduces the computational cost and time.

Correspondingly, the numerical results of CNN-CLFFA model and the existing models (RNN, LSTM, Bi-LSTM, VGG-16, GRU, AlexNet, VGG-19, CNN, and CNN-FA) by means of precision, recall, and F1-score are stated in Tables 6, 7, and 8. On the benchmark datasets, the CNN-CLFFA model obtained maximum precision, recall, and F1-score values than the comparative models. Specifically, the CNN-CLFFA model obtained 99.14%, 99.32%, 99.08%, 99.30%, 99.12%, and 99.10% of precision value, 99.18%, 99.26%, 99.17%, 99.25%, 99.14%, and 99.16% of recall value, and 98.98%, 99.10%, 99.18%, 98.97%, 98.98%, and 99.12% on the VCifar-100, MIO-TCD, GTSRB-1, GTSRB-2, GTSRB-3, and GTSRB-T datasets. The selection of hyper-parameters in the CNN model by CLFFA significantly removed the redundant parameters of the trained model, here; the numerical evaluation is carried-out using the personal computer.

D. COMPARATIVE STUDY

The comparative evaluation between the CNN-CLFFA and light-weighted CNN is depicted in Table 9. Zhou et al. [18]

TABLE 7. Results of CNN-CLFFA model and the existing models in light of recall.

	Recall (%)					
Models	VCifar-100	MIO-TCD	GTSRB-1	GTSRB-2	GTSRB-3	GTSRB-T
RNN	89.65	89.22	89.44	89.24	88.59	90
LSTM	89.70	89.92	89.88	89.40	89.08	90.22
Bi-LSTM	90.76	90	90.36	90.88	90.47	90.25
VGG-16	91.66	91.32	91.58	91.84	91.28	91.92
GRU	94.72	95.68	94.90	94.30	93.40	94.82
AlexNet	95.80	97.46	97.52	95.69	95.82	97.44
VGG-19	98.43	99	98.90	98.21	97.92	98.90
CNN	98.55	99.06	98.98	98.57	98.22	98.66
CNN-FA	98.80	99.12	99.02	98.97	98.72	98.55
CNN-CLFFA	99.18	99.26	99.17	99.25	99.14	99.16

TABLE 8. Results of CNN-CLFFA model and the existing models in light of F1-score.

	F1-score (%)					
Models	VCifar-100	MIO-TCD	GTSRB-1	GTSRB-2	GTSRB-3	GTSRB-T
RNN	88.70	90.55	90.50	90.65	89.50	91.30
LSTM	88.98	90.96	90.79	90.90	90.74	91.86
Bi-LSTM	89.10	91.04	91.12	91.68	91.40	92.20
VGG-16	90.25	92.87	92.20	92.80	92.77	93.42
GRU	92.70	94.12	95.76	95.42	94.46	95.80
AlexNet	94.33	96.44	96.96	96.60	96.90	96.73
VGG-19	96.80	97.66	97.48	97.24	97.72	97.94
CNN	97.56	98.32	97.94	97.50	97.98	97.98
CNN-FA	97.78	98.69	98.86	98.85	98.64	98.95
CNN-CLFFA	98.98	99.10	99.18	98.97	98.98	99.12

TABLE 9. Comparative results of CNN-CLFFA model and light-weighted CNN model.

Models	Model size	Classification accuracy (%)				
		VCifar-100	GTSRB-1	GTSRB-2	GTSRB-3	GTSRB-T
Light-weighted CNN [18]	4.9 MB	96.98	98.43	98.61	97.91	98.96
CNN-CLFFA	3.8 MB	99.02	99.12	99.32	99.21	99.03

have introduced a light-weighted CNN for supporting MEC in transportation CPS by enhancing the context-awareness and decreasing the latency. The experiments conducted on the online datasets revealed that the size of light-weighted CNN is 4.9 MB by reducing the number of parameters to 602,475. Additionally, the light-weighted CNN has a maximal classification accuracy of 96.98%, 98.96%, 97.91%, 98.61%, and 98.43% on the VCifar-100, GTSRB-T, GTSRB-3, GTSRB-2, and GTSRB-1 datasets. Whereas, the size of CNN-CLFFA is 3.8 MB, and it selects appropriate parameters of 509,344. The CNN-CLFFA obtained 99.02%, 99.03%, 99.21%, 99.32%, and 99.12% of classification accuracy on the VCifar-100, GTSRB-T, GTSRB-3, GTSRB-2, and GTSRB-1 datasets. The obtained numerical results revealed that the CNN-CLFFA efficiently maintains high accuracy and preserves the model's size in the MEC applications.

E. DISCUSSION

In this article, a novel CNN-CLFFA model is proposed for supporting MEC applications in transportation CPS. The proposed CNN-CLFFA model has a main benefit of a smaller model size and inference latency than the traditional CNN models. The model with a smaller size effectively utilizes resources that ensures optimal performance even on mobile devices with restricted computing power. Furthermore, the smaller model size reduces storage requirements, which is more vital in edge devices. The numerical analysis demonstrated that the proposed CNN-CLFFA model maintained higher accuracy in vehicle and traffic sign classification, and reduced computational time and cost. The selection of optimal hyper-parameters in CNN model by CLFFA reduced computational cost to 234,565, where the light-weighted CNN with FA model has a computational cost of 335,872, and the computational complexity of the CNN-CLFFA model is linear $O(N)$, where N indicates size of input data and O denotes order of magnitude. Furthermore, the CNN-CLFFA model has limited computational time of 22.10 seconds, 43.10 seconds, and 14.42 seconds on the VCifar-100, MIO-TCD, and GTSRB-T datasets, which are minimal than the existing models (RNN, LSTM, Bi-LSTM, VGG-16, GRU, AlexNet, VGG-19, CNN, and CNN-FA). The performance analysis by means of computational time is specified in

TABLE 10. Performance analysis by means of computational time.

Models	Computational time (seconds)					
	VCifar-100	MIO-TCD	GTSRB-T	GTSRB-1	GTSRB-2	GTSRB-3
RNN	31.45	82.48	45.28	35.40	32.48	32.06
LSTM	30.14	80.30	44.46	33.24	31.60	31.36
Bi-LSTM	29.33	76.59	42.42	30.12	31.22	30.14
VGG-16	28.27	72.33	40.35	28.41	30.26	30.38
GRU	44.24	55.20	28.32	26.24	22.09	20
AlexNet	28.20	47.12	22.44	18.54	20.36	20.25
VGG-19	27.12	48.05	20.34	17.22	18.18	17.26
CNN	26.10	46.03	19.22	16.49	17.10	17.08
CNN-FA	25.08	45.50	17.44	15.54	16.52	16.33
CNN-CLFFA	22.10	43.10	14.42	11.10	12.03	11.54

Table 10. The efficacy of the CNN-CLFFA model is clearly described in Tables 4, 5, 6, 7, 8, 9, and 10.

In this paper, we have evaluated the reliability and accuracy of the proposed system in traffic sign and vehicle recognition under a real world scenario. Scenario: a four way urban intersection with multiple lanes, pedestrian crossings, and traffic lights. Here, different vehicles are considered, including bicycles, buses, cars, vans, etc., and analyzed the performance under two conditions (varied traffic density and daytime with normal weather conditions). Pass criteria: proposed system achieved predefined accuracy level in traffic sign and vehicle recognition. Fail criteria: proposed system does not respond within a timeframe and it is not properly aligned with actual traffic conditions and movements.

V. CONCLUSION

In the mobile applications, the deep learning models face numerous challenges, particularly at MEC devices and IoT nodes. The deep learning models with heavy size results in computing and storage problems in the mobile devices, therefore, it is essential in developing a portable CNN model to support transportation CPS. In this article, an efficient CNN-CLFFA model is implemented in transportation CPS for enhancing the context-awareness and reducing the latency in a realistic MEC environment. Here, the size of CNN-CLFFA model is smaller and maintains higher classification accuracy. For instance, the size of CNN-CLFFA model is 3.8 MB, which is very smaller compared to the existing models like RNN, LSTM, Bi-LSTM, VGG-16, GRU, AlexNet, VGG-19, CNN, and CNN-FA. Further, the proposed CNN-CLFFA model achieved higher classification accuracy of 99.02%, 99.11%, and 99.03% on the VCifar-100, MIO-TCD, and GTSRB-T datasets. Correspondingly, the precision and recall of the CNN-CLFFA model also outperforms the comparative models (RNN, LSTM, Bi-LSTM, VGG-16, GRU, AlexNet, VGG-19, CNN, and CNN-FA).

By inspecting experimental results, the conventional CNN model failed in recognizing smaller distant vehicles and traffic signs, particularly when they are present in a smaller portion of the image. Furthermore, the conventional CNN model struggled to adapt to illumination changes, weather conditions, and other external or environmental variations. Therefore, as a future extension, advanced objection recognition models like You Only Look Once (YOLO) can be developed for the effective recognition of smaller objects, even in diverse conditions. Models like YOLO capture contextual information and relations between objects for the accurate recognition of smaller objects. Specifically, the YOLO model effectively reduces false positives, which is vital while handling smaller objects. The YOLO model improves overall accuracy and reduces false detections by optimizing the localization process. In addition to this, the proposed CNN-CLFFA model can be applied in other transportation CPS domains like parking systems, aircraft transportation systems, industrial automation, healthcare, and battlefield surveillance.

Data Availability: The datasets generated during and/or analyzed during the current study are available in the [VCifar-100 dataset], [MIO-TCD classification dataset] and [GTSRB dataset] repositories.

- **VCifar-100 dataset:** https://git-disl.github.io/GTDLBench/datasets/cifar-100_datasets/
- **MIO-TCD classification dataset:**https://www.kaggle.com/datasets/yash88600/miotcd-dataset-50000-imagesclassification
- **GTSRB dataset:** https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

REFERENCES

[1] A. Haydari and Y. Yilmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 11–32, Jan. 2022, doi: 10.1109/TITS.2020.3008612.

[2] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1840–1852, Mar. 2021, doi: 10.1109/TITS.2020.3025687.

[3] F. Zhu, Y. Lv, Y. Chen, X. Wang, G. Xiong, and F. -Y. Wang, "Parallel transportation systems: Toward IoT-enabled smart urban traffic control and management," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 10, pp. 4063–4071, Oct. 2020, doi: 10.1109/TITS.2019.2934991.

[4] W.-H. Ko, B. Satchidanandan, and P. R. Kumar, "Dynamic watermarking-based defense of transportation cyber-physical systems," *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 1, pp. 1–21, Nov. 2019, doi: 10.1145/3361700.

[5] G. Xiong, F. Zhu, X. Liu, X. Dong, W. Huang, S. Chen, and K. Zhao, "Cyber-physical-social system in intelligent transportation," *IEEE/CAA J. Autom. Sinica*, vol. 2, no. 3, pp. 320–333, Jul. 2015, doi: 10.1109/JAS.2015.7152667.

[6] C. Chen, X. Liu, T. Qiu, and A. K. Sangaiah, "A short-term traffic prediction model in the vehicular cyber-physical systems," *Future Gener. Comput. Syst.*, vol. 105, pp. 894–903, Apr. 2020, doi: 10.1016/j.future.2017.06.006.

[7] L. Deka, S. M. Khan, M. Chowdhury, and N. Ayres, "Transportation cyber-physical system and its importance for future mobility," in *Transportation Cyber-Physical Systems*. Amsterdam, The Netherlands: Elsevier, 2018, pp. 1–20, doi: 10.1016/B978-0-12-814295-0.00001-0.

[8] D. P. F. Möller and H. Vakilzadian, "Cyber-physical systems in smart transportation," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, Grand Forks, ND, USA, May 2016, pp. 0776–0781, doi: 10.1109/EIT.2016.7535338.

[9] D. Trentesaux, T. Knothe, G. Branger, and K. Fischer, "Planning and control of maintenance, repair and overhaul operations of a fleet of complex transportation systems: A cyber-physical system approach," in *Studies in Computational Intelligence*, vol. 594. Cham, Switzerland: Springer, 2015, pp. 175–186, doi: 10.1007/978-3-319-15159-5_17.

[10] M. M. Hussain and M. M. S. Beg, "Using vehicles as fog infrastructures for transportation cyber-physical systems (T-CPS): Fog computing for vehicular networks," *Int. J. Softw. Sci. Comput. Intell.*, vol. 11, no. 1, pp. 47–69, Jan. 2019, doi: 10.4018/ijssci.2019010104.

[11] D. B. Rawat, C. Bajracharya, and G. Yan, "Towards intelligent transportation cyber-physical systems: Real-time computing and communications perspectives," in *Proc. SoutheastCon*, Fort Lauderdale, FL, USA, Apr. 2015, pp. 1–6, doi: 10.1109/SECON.2015.7132923.

[12] Y. Hou, Y. Zhao, A. Wagh, L. Zhang, C. Qiao, K. F. Hulme, C. Wu, A. W. Sadek, and X. Liu, "Simulation-based testing and evaluation tools for transportation cyber-physical systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1098–1108, Mar. 2016, doi: 10.1109/TVT.2015.2407614.

[13] T. Zhang, Y. Zou, X. Zhang, N. Guo, and W. Wang, "Data-driven based cruise control of connected and automated vehicles under cyber-physical system framework," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 10, pp. 6307–6319, Oct. 2021, doi: 10.1109/TITS.2020.2991223.

[14] H. Liu, D. Sun, and W. Liu, "Lattice hydrodynamic model based traffic control: A transportation cyber-physical system approach," *Phys. A, Stat. Mech. Appl.*, vol. 461, pp. 795–801, Nov. 2016, doi: 10.1016/j.physa.2016.06.069.

[15] Y. Chen, Y. Zhang, S. Maharjan, M. Alam, and T. Wu, "Deep learning for secure mobile edge computing in cyber-physical transportation systems," *IEEE Netw.*, vol. 33, no. 4, pp. 36–41, Jul. 2019, doi: 10.1109/MNET.2019.1800458.

[16] T. Shen, K. Hua, and J. Liu, "Optimized public parking location modelling for green intelligent transportation system using genetic algorithms," *IEEE Access*, vol. 7, pp. 176870–176883, 2019, doi: 10.1109/ACCESS.2019.2957803.

[17] J. Ashraf, A. D. Bakhshi, N. Moustafa, H. Khurshid, A. Javed, and A. Beheshti, "Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4507–4518, Jul. 2021, doi: 10.1109/TITS.2020.3017882.

[18] J. Zhou, H.-N. Dai, and H. Wang, "Lightweight convolution neural networks for mobile edge computing in transportation cyber physical systems," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 6, pp. 1–20, Oct. 2019, doi: 10.1145/3339308.

[19] N. Kumar, S. S. Rahman, and N. Dhakad, "Fuzzy inference enabled deep reinforcement learning-based traffic light control for intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 8, pp. 4919–4928, Aug. 2021, doi: 10.1109/TITS.2020.2984033.

[20] M. M. Rathore, S. A. Shah, A. Awad, D. Shukla, S. Vimal, and A. Paul, "A cyber-physical system and graph-based approach for transportation management in smart cities," *Sustainability*, vol. 13, no. 14, p. 7606, Jul. 2021, doi: 10.3390/su13147606.

[21] D. K. Jain, S. Neelakandan, T. Veeramani, S. Bhatia, and F. H. Memon, "Design of fuzzy logic based energy management and traffic predictive model for cyber physical systems," *Comput. Electr. Eng.*, vol. 102, Sep. 2022, Art. no. 108135, doi: 10.1016/j.compeleceng.2022.108135.

[22] M. Rai, B. Khosla, Y. Dhawan, H. Kharotia, N. Kumar, and A. Bandi, "CLEAR: An efficient traffic sign recognition technique for cyber-physical transportation systems," in *Proc. IEEE/ACM 15th Int. Conf. Utility Cloud Comput. (UCC)*, Vancouver, WA, USA, Dec. 2022, pp. 418–423, doi: 10.1109/UCC56403.2022.00072.

[23] T. Liu, B. Tian, Y. Ai, and F.-Y. Wang, "Parallel reinforcement learning-based energy efficiency improvement for a cyber-physical system," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 617–626, Mar. 2020, doi: 10.1109/JAS.2020.1003072.

[24] J. Zhou, H.-N. Dai, H. Wang, and T. Wang, "Wide-attention and deep-composite model for traffic flow prediction in transportation cyber-physical systems," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3431–3440, May 2021, doi: 10.1109/TII.2020.3003133.

[25] M. Wasim, I. Ahmed, J. Ahmad, and M. M. Hassan, "A novel deep learning based automated academic activities recognition in cyber-physical systems," *IEEE Access*, vol. 9, pp. 63718–63728, 2021, doi: 10.1109/ACCESS.2021.3073890.

[26] Q. Hou and J. Dong, "Robust adaptive event-triggered fault-tolerant consensus control of multiagent systems with a positive minimum interevent time," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 7, pp. 4003–4014, Jul. 2023, doi: 10.1109/TSMC.2023.3238709.

[27] Q. Hou and J. Dong, "Cooperative fault-tolerant output regulation of linear heterogeneous multiagent systems via an adaptive dynamic event-triggered mechanism," *IEEE Trans. Cybern.*, vol. 53, no. 8, pp. 5299–5310, Aug. 2023, doi: 10.1109/TCYB.2022.3204119.

[28] J. Talukdar, A. Biswas, and S. Gupta, "Data augmentation on synthetic images for transfer learning using deep CNNs," in *Proc. 5th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Noida, India, Feb. 2018, pp. 215–219, doi: 10.1109/SPIN.2018.8474209.

[29] S. T. M. Ataky, J. de Matos, A. D. S. Britto, L. E. S. Oliveira, and A. L. Koerich, "Data augmentation for histopathological images based on Gaussian–Laplacian pyramid blending," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8, doi: 10.1109/IJCNN48605.2020.9206855.

[30] E. Goceri, "Image augmentation for deep learning based lesion classification from skin images," in *Proc. IEEE 4th Int. Conf. Image Process., Appl. Syst. (IPAS)*, Genova, Italy, Dec. 2020, pp. 144–148, doi: 10.1109/IPAS50080.2020.9334937.

[31] B. L. Geeraert, M. Chamberland, R. M. Lebel, and C. Lebel, "Multimodal principal component analysis to identify major features of white matter structure and links to reading," *PLoS One*, vol. 15, no. 8, Aug. 2020, Art. no. e0233244, doi: 10.1371/journal.pone.0233244.

[32] A. Djerida, Z. Zhao, and J. Zhao, "Background subtraction in dynamic scenes using the dynamic principal component analysis," *IET Image Process.*, vol. 14, no. 2, pp. 245–255, Feb. 2020, doi: 10.1049/iet-ipr.2018.6095.

[33] J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, and A. Valenzuela, "A review of convolutional neural network applied to fruit image processing," *Appl. Sci.*, vol. 10, no. 10, p. 3443, May 2020, doi: 10.3390/app10103443.

[34] J. Bernal, K. Kushibar, D. S. Asfaw, S. Valverde, A. Oliver, R. Martí, and X. Lladó, "Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: A review," *Artif. Intell. Med.*, vol. 95, pp. 64–81, Apr. 2019, doi: 10.1016/j.artmed.2018.08.008.

[35] N. Zhang, Y.-X. Cai, Y.-Y. Wang, Y.-T. Tian, X.-L. Wang, and B. Badami, "Skin cancer diagnosis based on optimized convolutional neural network," *Artif. Intell. Med.*, vol. 102, Jan. 2020, Art. no. 101756, doi: 10.1016/j.artmed.2019.101756.

[36] D. J. Hemanth, O. Deperlioglu, and U. Kose, "An enhanced diabetic retinopathy detection and classification approach using deep convolutional neural network," *Neural Comput. Appl.*, vol. 32, no. 3, pp. 707–721, Feb. 2020, doi: 10.1007/s00521-018-03974-0.

[37] P. Yao, H. Wu, B. Gao, J. Tang, Q. Zhang, W. Zhang, J. J. Yang, and H. Qian, "Fully hardware-implemented memristor convolutional neural network," *Nature*, vol. 577, no. 7792, pp. 641–646, Jan. 2020, doi: 10.1038/s41586-020-1942-4.

[38] N. Dey, J. Chaki, L. Moraru, S. Fong, and X. -S. Yang, "Firefly algorithm and its variants in digital image processing: A comprehensive review," in *Applications of Firefly Algorithm and Its Variants: Case Studies and New Developments*. Singapore: Springer, 2020, pp. 1–28.

[39] N. Bacanin, T. Bezdan, K. Venkatachalam, and F. Al-Turjman, "Optimized convolutional neural network by firefly algorithm for magnetic resonance image classification of glioma brain tumor grade," *J. Real-Time Image Process.*, vol. 18, no. 4, pp. 1085–1098, Aug. 2021, doi: 10.1007/s11554-021-01106-x.

[40] A. Sharma, R. Chaturvedi, and A. Bhargava, "A novel opposition based improved firefly algorithm for multilevel image segmentation," *Multimedia Tools Appl.*, vol. 81, no. 11, pp. 15521–15544, May 2022, doi: 10.1007/s11042-022-12303-6.

[41] J. Nayak, B. Naik, P. Dinesh, K. Vakula, and P. B. Dash, "Firefly algorithm in biomedical and health care: Advances, issues and challenges," *Social Netw. Comput. Sci.*, vol. 1, no. 6, p. 311, Sep. 2020, doi: 10.1007/s42979-020-00320-x.

[42] S. Garg and B. Jindal, "Skin lesion segmentation using K-mean and optimized fire fly algorithm," *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 7397–7410, Feb. 2021, doi: 10.1007/s11042-020-10064-8.

[43] Q. Pan, J. Wu, A. K. Bashir, J. Li, W. Yang, and Y. D. Al-Otaibi, "Joint protection of energy security and information privacy for energy harvesting: An incentive federated learning approach," *IEEE Trans. Ind. Informat.*, vol. 18, no. 5, pp. 3473–3483, May 2022, doi: 10.1109/TII.2021.3105492.

ASHOK BHANSALI is currently a Professor and the Dean of the Department of Computer Engineering and Applications, GLA University, Mathura, India. He possesses approximately 25 years of amalgamated experience in the industry and academia, during which he has worked with top MNCs and universities, such as TechMahindra, NELCO, Symbiosis University, and OP Jindal University. He has published more than 40 papers in different journals/conferences on various topics, including social network mining, eLearning, ad-hoc networks, the IoT, and blockchain. He has a special interest in education 4.0 and implementing and promoting eLearning. He was an Advisor of the Government of Chhattisgarh for the Virtual Education Project of MHRD, Government of India. He is an active IEEE volunteer and has been the general chair of many IEEE conferences. He has delivered many keynote sessions and lectures at FICCI, CII, and national/international conferences and workshops.

RAJ KUMAR PATRA received the Ph.D. degree in engineering, in 2017. Currently, he is a Professor with the Department of Computer Science and Engineering, CMR Technical Campus, Hyderabad, India. He has more than 21 years of teaching and research experience. He has published more than 50 papers in reputed peer-reviewed national and international journals and conferences. He has more than 15 publications in SCI and Scopus-indexed journals.

PARAMESHACHARI BIDARE DIVAKARACHARI (Senior Member, IEEE) is currently a Professor with the Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, India, affiliated to Visvesvaraya Technological University (VTU), Belagavi, Karnataka, India. He has around 19 years of experience and has published more than 200 articles in SCI, Scopus, and other indexed journals and conferences. He serves as an editorial board member, an associate editor, an academic editor, a guest editor, and a reviewer for various reputed indexed journals. He is the Founder Chair of the IEEE Information Theory Society Bangalore Chapter and the IEEE Mysore Subsection. He is the SAC Chair of the IEEE Bangalore Section.

PRZEMYSŁAW FALKOWSKI-GILSKI received the B.Sc. degree in Polish and the M.Sc. degree in English, in 2012 and 2013, respectively, and the Doctor of Technical Sciences degree (Hons.), in 2018. During the Ph.D. studies, he pursued his interests sin the field of electronic media, particularly digital broadcasting systems and the quality of networks and services, from 2013 to 2017. He is currently a Graduate Member of the Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology. He is also an Assistant Professor. He is a member of several organizational, scientific, technical, and program committees of national and international conferences, indexed in DBLP, IEEE Xplore, Scopus, Springer, and Web of Science. He is a reviewer of several national Polish and international English-language journals from publishing houses, including Elsevier, IEEE, Inderscience, IPPT-PAN, MDPI, Springer, Taylor and Francis, and Wiley.

GANDLA SHIVAKANTH received the B.Tech. and M.Sc. degrees in CSE from JNTUH University, Hyderabad, in 2012 and 2016, respectively, and the Ph.D. degree from Madhav University, Rajasthan, India, in July 2022. He is currently an Associate Professor with the Department of Computer Science and Engineering, K. L. University, Vijayawada. His current research interests include image processing, cloud computing, computer networks, data mining, and machine learning.

SUJATHA N. PATIL received the B.E. degree from Karnataka University Dharwad, the M.Tech. degree from the B. V. Bhoomreddy College of Engineering and Technology, Hubli, and the Ph.D. degree in inter-disciplinary research with AI and ML applications for predicting the grade of human IVF embryo image to transfer a single embryo in a women's womb. Currently, she is working with many machine learning algorithms for predicting the early detection of prostate cancer using convolutional neural networks (CNNs). Her research interest includes applying AI and ML for neurological disorders and early detection of Alzheimer's disease. She was the PI of theVision Group of Science and Technology (VGST) in setting up the Center of Excellence in Artificial Intelligence, KLE Dr. MSSCET. She has received the Competitive Research Grant of Rs. 2 lakhs under TEQIP for the project titled, "Neural Network Based Test Marks Reader and Acquisition System," as a Co-Principal Investigator.