IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 October 2023, accepted 5 December 2023, date of publication 7 December 2023,
date of current version 14 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3340719

RESEARCH ARTICLE

ps-CALR: Periodic-Shift Cosine Annealing Learning Rate for Deep Neural Networks

OLANREWAJU VICTOR JOHNSON[1], CHEW XINYING[1], KHAI WAH KHAW[2], AND MING HA LEE[3]

[1]School of Computer Sciences, Universiti Sains Malaysia, Gelugor, Penang 11800, Malaysia
[2]School of Management, Universiti Sains Malaysia, Gelugor, Penang 11800, Malaysia
[3]School of Engineering, Swinburne University of Technology, Sarawak Campus, Kuching 93350, Malaysia

Corresponding author: Chew Xinying (xinying@usm.my)

This work was supported by the Ministry of Higher Education Malaysia, Fundamental Research Grant Scheme, through the Project titled ''Efficient Joint Process Monitoring Using a New Robust Variable Sample Size and Sampling Interval Run Sum Scheme'' under Grant FRGS/1/2022/STG06/USM/02/4.

ABSTRACT There Are Continued Efforts to Build on the Performance of Deep Learning (DL) Models in Various Fields of Application. Developing New DL Models Continues to Open Unprecedented Opportunities in Diverse Application Areas Despite the Enormous Resources Required. Generally, the Learning Mechanism of DL Models Depends on the Term ''Cost Function'' (CF) or ''Loss Function'' (LF), and DL Models Require Varied Hyperparameter Settings and, Precisely, Parameters That Can Help the Model to Continually Minimize the Cost Function Until Faster Convergence, With Better Generalization Over the Data in the Loss Landscape, Is Assumed. The Learning Rate (LR) Update Seeks to Find the Optimal Solution for DL Models Through Relative Cost Function Minimization. Therefore, Selecting the Appropriate LR Is Essential to the Performance of DL Models. Despite Its Demonstration for Fast Model Convergence, the Existing Cosine Annealing LR Lacks Complete Loss Landscape Exploration of the Flat Minima, Hence Limiting Its Ability to Model Better Generalization. To Address This, the Paper Proposes a Period-Shift Cosine Annealing Learning Rate With Warm-up Epochs (Ps-CALR) to Perturb the LR Update. Six Publicly Available Datasets Were Used to Benchmark the Proposed LR Method by Experimenting With Custom DL (multilayer Perceptron and Convolutional Neural networks) and Pre-Trained DL Models. The Proposed Ps-CARL Enhances Model Generalization and Convergence, Pushing the Solution to Notably Better Performance Than Fixed LR and the Existing Cosine Annealing Method.

INDEX TERMS Cosine annealing, convergence, flat minima, learning rate, loss function, optimizers.

I. INTRODUCTION

OVER a Decade, Deep Learning (DL) Models Have Continued to Spur Research Interest Due to Diverse Cutting-Edge Breakthroughs in Artificial Intelligence (AI) Applications Successfully Developed and Deployed. Aside From the Foundational Application Areas Such as Image Classification, Pattern Recognition, and Regression Problem [1], DL Has Gained a Momentum Effect in Modern-Day Applications, Including Advanced Image Segmentation [2], Traffic Pattern Analysis [3], Computer Vision [4], Motion Recognition [5], Voice Recognition [6], Natural Language Processing (NLP) [7], [8], Drug Discovery, Genomics and Protein Sequencing [9] and Many Other Domains [1].

Compared With Conventional Machine Learning (ML) Models, DL Is Computationally Expensive [10], [11]. However, Technological Advancements, Such as High-Performance Computing (HPC) and Numerical Computation, Have Graciously Helped to Push Behind the Narratives [12], [13]. Meanwhile, DL Is Not a One-in-All ML Algorithm Without Its Intricacies. A DL Model Consists of Different Hyperparameters Requiring Tuning in One Way or Another, Which Inform the Model's Performance Under Training [14]. Two Vital of These Hyperparameters Are the Learning

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

© 2023 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

Rate (LR) and Batch Size [15], [16]. Others Include but Are Not Limited to Filters, Kernel Size, Number of Neurons in the Hidden Layers, L1/2 Regularizers, and Optimizers [17], [18]. Moreover, There Are Well-Known Hyperparameter Search Algorithms Proposed in Research Studies, Including Random, Grid, Bayesian, Genetic, and Hyperband [17], [19], [20] for Parameter Tuning Solutions. However, This Applies Only to the Searching Space of the Hyperparameters and Not Specifically to Perturb the LR Dynamically. Research Studies Also Rely on Ablation Studies of Batch Size and Epochs to Showcase the Effect of Hyperparameter Tuning [18].

The Learning Mechanism of ML/DL Models Generally Depends on the Term Called the Cost Function (CF) or Loss Function (LF), Which Culminates in the Model Performance Over the Learning Period. The Idea Is to Continually Minimize This CF Until the Model Achieves Better Generalization Over the Data. The Iterative Update of the Model's Parameter to Find the Optimal Solution Set Is Called Learning, While the Small Term That Allows the CF Minimization Is the LR [21]. LR Is Very Critical to DL Performance, and Finding a Suitable LR Value to Achieve This Objective Has Been Intensified in Research Studies. Moreover, the Complexity of the DL Models as a Non-Deterministic Polynomial (NP) Problem Requires Some Design Mechanism Leading to the Capability of the Training Process, Adjusting the Weights and Biases of the Model to Minimize the CF Using the Optimizer.

As a Result, Various Optimizers Are Studied in the Literature, Starting With Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSprop), Adaptive Moment Estimation (Adam), and Many Others [22], [23]. Achieving the Optimization Goal of DL by Considering the Peculiarities of the Optimizers Implemented for Stepwise Weights Update, Therefore, Requires LR Schedules. Fixed LR Such as 0.01 and 0.001 Are Also Explored in Research Studies With Good Performance Recorded for Some DL Model Tasks [21]. On the Other Hand, Other LR Schedules, Including Polynomial Decay, Step Decay, Time-Based Decay, ReduceLROnPlateau, and Cyclical LR [21], [24], Have Remained Popular Due to Their Capacity to Scale up the Learning Process in Non-Protuberant and Rough Loss Situations With Several Local Minima as Well as Potential Saddle-Points. Recent Studies Have Also Focused on Using Polyak's Method for Generalizing the Step Size of Step Decay LR With SGD [25], [26], [27]. In Modern-Day Networks, Many LR Schedules Are Adaptive or Elastic to the Optimizers to Strike a Balance Between Fast Initial Progress and Stable Convergence During Training.

A High LR Aids DL Models in Achieving Strong Generalization. In Contrast, a Lower LR Aids the Model (occasionally Failing to Converge Because of Being Trapped at the Local minimum) for a Global Minimum Search [22]. Hence, a Good LR Should Help the Model Converge Faster and Escape Local Minima to a Global Minimum, as Shown in Fig. 1. A Flat Minima of the Global Minimum for Good Generalization Is Strongly Argued for in the Literature [28], [29], [30], [31], and [32]; However, Some Studies Proved That

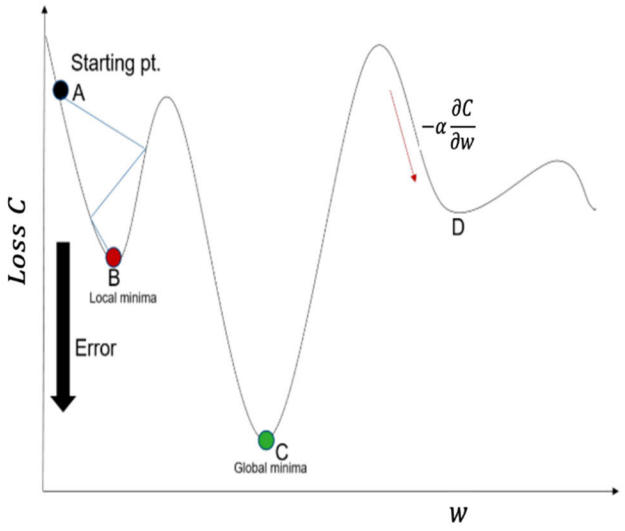

FIGURE 1. Gradient descent process with different minima points adapted from [55].

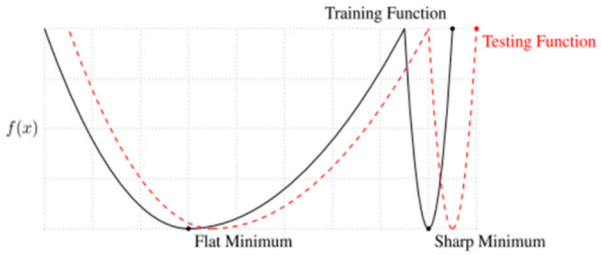

FIGURE 2. An illustration of flat and sharp minima [24].

Sharp Minima Also Provide Good Generalization [28], [33]. Fig. 2 Illustrates These Global and Local Minima for DL Model Generalization. In Any Case, an LR Schedule That Provides Faster Convergence Is Desirable.

Addressing, Therefore, the Perennial Challenge of Getting Stuck in a Sharp Minima, Loshchilov and Hutter [10] Introduced a Cosine Annealing LR That Uses the Cosine Function as a Basis to Allow LR Oscillation During Training, Typically Starting From a High LR to a Smaller Value to Achieve Faster Convergence. Liu [34] Proposed a Linearly Warm-up on the Cosine Annealing LR (CALR). The Method Initiates the Model With a Very Small LR to a Large Value and Then Uses a Cosine Function to Anneal the High LR to a Relatively Small Value [35]. The Result of the Study Indicates That Improved Model Convergence Was Achieved. However, the Two Approaches Are Subject to the Same Point Decay (point, π) Before a Warm Restart to the Initial High LR as a Starting Point in the Next Cycle and, Hence, Have a Shortened Lifespan in the Loss Landscape.

Consequently, the Approaches Are Limited to Provide Better Model Generalization. This Paper, Therefore, Proposes a Periodic Shift-Based Approach to the Cosine Annealing LR With Warm-up Epochs (Ps-CALR) for the LR Schedule Implementation. The Proposed Policy Seeks to Thoroughly

Explore the Region Where Loss Is Minimal (flat minima) to Achieve Better Model Generalization and Maintain Stability Along Faster Convergence, Even With Fewer Epochs. The Contributions of the Paper Are as Follows:

i. Firstly, We Introduce a Periodic Shift, Which Allows Models to Better Explore the Loss Landscape During Training While Maintaining a Warm-up Epoch for Aggressive LRs on the Model's Early Training Phases to Mitigate Adverse Effects in the Model.
ii. Secondly, We Used an Offset Value in the Cosine Function to Avoid Potential Numerical Precision Bottlenecks.
iii. Lastly, Several Experiments Were Conducted With the Proposed Method on Image Datasets as Obtainable in Past Studies and Non-Image Datasets. The Non-Image Datasets Used Demonstrate the Support Crusading and Recent Efforts Towards Using DL Classifiers for Non-Image Datasets [36], [39].

The Rest of the Paper Is as Follows. Section II Discusses the Literature. Section III Focuses on the Concept of Cosine Function and the Proposed Method. The Experiment and Empirical Results With the Discussion Are Presented in Section IV, and the Conclusion Is Discussed in Section V.

II. LITERATURE REVIEW

As DL keeps advancing, studies on LR schedules have received more attention, primarily due to the need to constantly develop a fast converging and good generalizing model [40]. Several works suggest a heuristic method of LR schedule to accomplish such a purpose.

Mishra and Sarawadekar [41] developed a warm restart technique on a polynomial LR policy. The proposed method was based on a cyclical LR and SGD with a single warm-restart [10]. The suggested LR strategy has a greater classification accuracy and accelerates the DNN's convergence. However, the polynomial LR with such a single warm-restart may limit DL models' ability to explore the landscape better. In Wang et al. [42], a traditional LR decay method was identified to adopt manual mannerism during training; hence the small LR produced causes slow convergence in training the DL models. An automatic LR decay approach was proposed using SGD and momentum to alleviate this challenge. However, momentum with SGD optimizer suffers from accumulating velocity and overshooting the minimum in flat regions of loss landscape. Hence, our approach is pivotal to this loss landscape exploration. In another paper, there was an effort to solve the symmetric optimization or initialize the parameters symmetrically while searching for the best solution. This concept inspired the authors in the study to suggest a changeable LR instead of the monotonically lowering approach by providing a CF technique to identify the ideal parameters that modify the LR adaptively [43].

Li et al. [44] studied the Gaussian Process Regression (GPR) on LR optimization to increase classification accuracy. In particular, the link between LR and corresponding accuracy and the GPR model was examined. The GRP is responsible for predicting the next potential LR. This approach may be challenging due to the GRP's overhead cost. Also, Nakamura et al. [45] presented an LR schedule using the annealing approach that combines warm-up and the sigmoid function. Not only did the method shown increase DL model performance, but it was competitive with both existing adaptive methods and the other LR schedules in accuracy.

Moreover, the approaches in [44] and [45] have their peculiarities demonstrating improved model convergence but are not primarily cosine annealing based. Aside from the LR schedules mentioned above, Sadr et al. [7] studied deep networks for sentiment analysis using varying fixed LR of 0.025 and 0.01 with ADADELTA, respectively. Coupled with other model parameters for hyperparameter tuning, they demonstrated that carefully chosen LR can improve model performance. The adaptive LR crusade, where LR schedules are implemented using adaptive optimizers such as the SGD, ADAM, Adaptive mean square gradient (AMSGrad), and many others, is rigorously pursued in literature [46], [47], [48], [49], [50], [51], [52], and [53].

The closest work to the proposed approach in this paper was inspired by Loshchilov and Hutter [10]. The authors developed a cosine annealing to help oscillate the LR from a high value to the minimum within the defined number of iterations before oscillating back to the initial high LR at the next cycle. Huang et al. [54] further used the LR technique to develop snapshot ensemble DL models, where a singular DL model was trained as a multiple-like DL model. The cosine annealing was iteratively used (since it can run for i-th cycles) within the DL model to form multiple networks. The prediction outputs in each cycle are then combined as a final prediction. In application, Nie et al. [55] used cosine annealing LR in the DL model to classify skin cancer. The authors used the annealing method to demonstrate their proposed deep model and compared the result with a fixed LR schedule. They further experimented the cosine annealing using well-known pre-trained models. Results show that the LR schedule helps facilitate faster and better convergence than the Fixed LR. Also, Howard and Ruder [56] used cosine annealing for fine-tuning a universal language model for text classifiers. The method was applied to aggressively perturb LR to achieve good performance as obtainable in computer vision. Furthermore, cosine annealing was used for a time-series-based DL model comprising dual attention Recurrent Neural Networks (RNN). The model was designed to solve the electric load forecasting task. The cosine annealing improved the model prediction with faster convergence compared with conventional forecasting techniques [57].

Meanwhile, Liu [34] proposed a warm-up technique to modify the cosine annealing to achieve improved convergence compared with the existing method. The proposed approach first uses linearly/non-linearly warm-up to initiate the DL model with a very small LR to a high value. Then, it uses the cosine annealing to oscillate the high LR to a relatively small value for faster convergence.

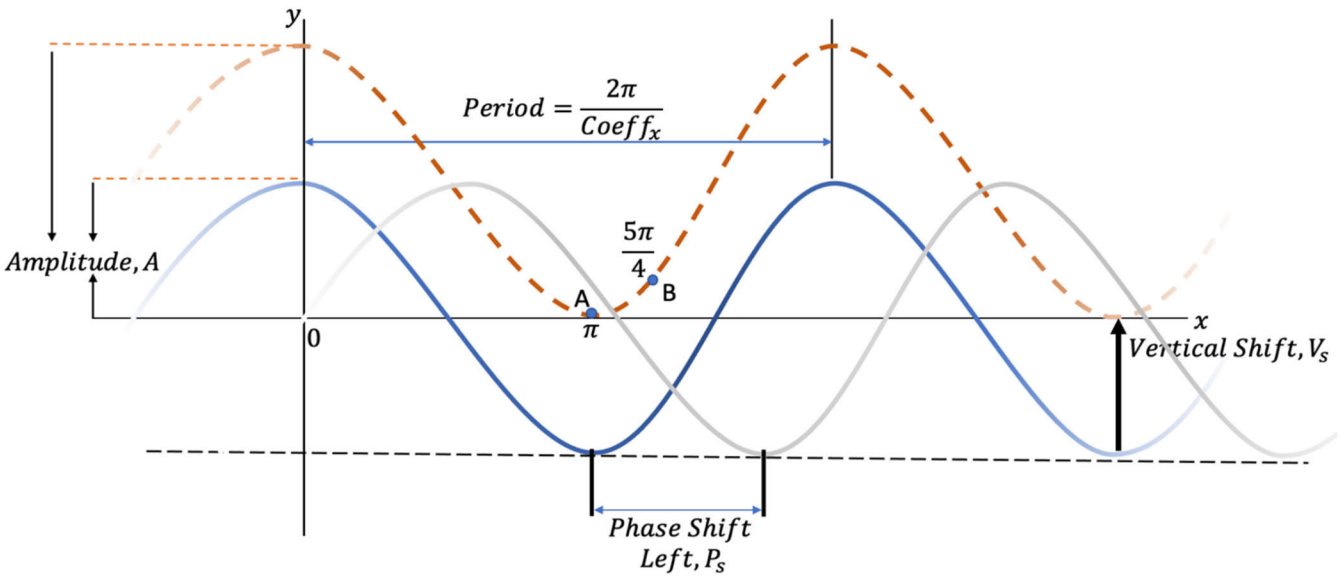

FIGURE 3. Conceptual concept of cosine function for modeling learning rate schedule.

While demonstrating the proposed approach as an improvement on the existing cosine annealing, it is subject to limited good generalization of the loss landscape. The focus of this paper, therefore, is to enhance the existing cosine annealing method to explore the entire loss region of the global minimum to improve model generalization and, at the same, maintain its faster convergence.

III. METHODS

This section discusses the underlying concept of cosine annealing from the basic principle of cosine function as foundational building blocks upon which the cosine annealing was built and the proposed improvement. In addition, we discuss optimizers, datasets, and DL models used along the experimental setup to benchmark the proposed method.

A. PROPOSED METHOD

The proposed method in this paper follows the suggestion of Loshchilov and Hutter [10] and, in addition, provides modifications to solve the challenge of ''good model generalization.'' The fundamental building block of cosine annealing is based on the concept of cosine function. The examined cosine function and its properties to support LR annealing can be expressed as:

$$y = A\,cos\,(C_x\,(x + P_s)) + V_s, \tag{1}$$

where A is the amplitude, P_s represents the phase shift, V_s represents the vertical shift and the period is defined as:

$$P = \frac{2\pi}{C_x}\,^{1} \tag{2}$$

[1]There are 2π radians in a full rotation

In the case of cosine annealing, as illustrated by the cosine function in Fig. 3, the following deductions are satisfied as follows:

i. The vertical shift allows bounding the LR update on the positive y-axis while seeking the minimum value indicated with the dotted orange curve.
ii. The amplitude A represents the initial high LR.
iii. The warm restart takes place at point π

Following the concept of cosine annealing, we introduce three ideas in the proposed method:

i. First, a periodic shift is indicated at point B in Fig. 3, allowing the model to better explore the loss landscape region during training. The loss region established by the periodic shift is further illustrated in Fig. 4 (Just as the descent from a mountaintop to the valley reveals a greener landscape enriched with minerals, skillful navigation through the hurdles to the valley amplifies the magnitude of achievements). The red arrow indicates the loss landscape exploration-exploitation in Fig. 4.
ii. The second is using warm-up epochs to enhance the stability of the DL model during training. The warm-up epoch is used at the initial training phase to gradually increase the LR from a very small value to the initial LR value. The process helps the DL model stabilize and settle into a reasonable region of the loss landscape before oscillating the high LR with the cosine annealing. The warn-up epochs further aid the model to converge on lesser iterations than total iterations and use the acceleration gained to explore the loss landscape, as shown in Fig. 5. This process is called an aggressive learning technique for cosine annealing.

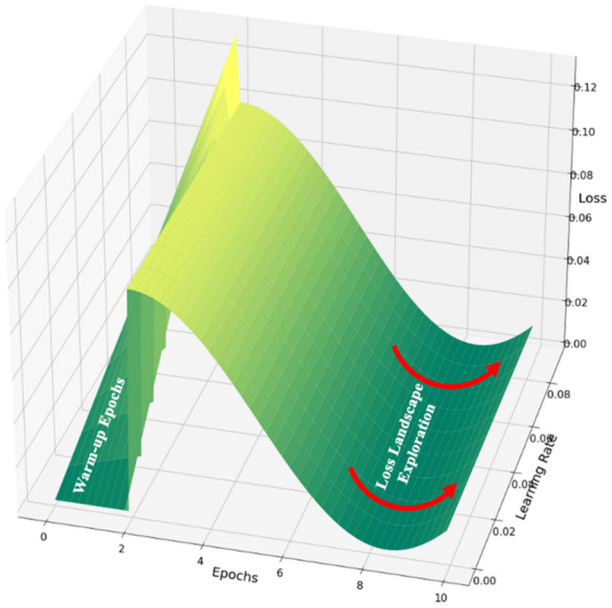

FIGURE 4. 3D view of the proposed method showing the loss landscape exploration.

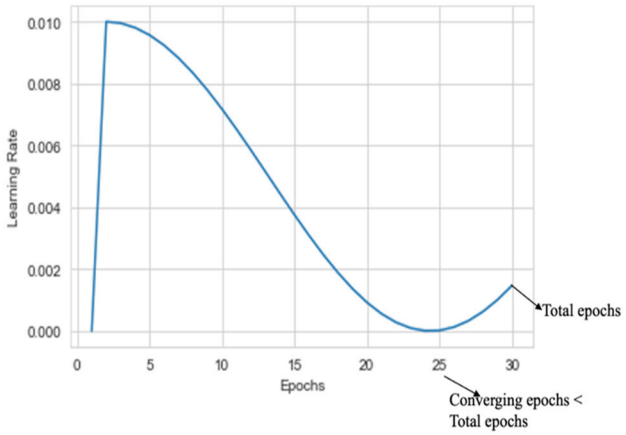

FIGURE 5. The proposed method annealing process during model training.

iii. Lastly, an offset value in the cosine function is introduced to avoid the potential numerical precision of dividing by zero.

Based on the existing cosine annealing in (3) and (4) [10], the proposed method is expressed in (5), (6), and (7) as follows.

$$\alpha(t) = f\left(mod\left(t,\ \lceil T/M \rceil\right)\right) \tag{3}$$

$$\alpha(t) = \frac{\alpha_0}{2}\left(cos\left(\frac{\pi\, mod\left(t-1,\ \lceil T/M \rceil\right)}{\lceil T/M \rceil}\right)+1\right) \tag{4}$$

$$P_s = \frac{5\pi}{4} \tag{5}$$

For a better exploration of the loss landscape, we replace point π with P_s in (5) to (4). We also injected the warm-up epoch in the existing iterations to complete the enhancement of (4). The offset value, a relatively small insignificant value used to avoid the potential numerical precision of dividing by zero, is finally subtracted from the enhancement as expressed in (6). The proposed method runs conditionally in two parts when used in model training. Under the given iterations, the warm-up epoch execution condition is satisfied in the first part before the oscillation of cosine annealing takes effect, as in (7).

$$C_a = cos\left(\frac{P_s(mod\,(t-1,\ \lceil T/M \rceil) - E_w)}{\lceil T/M - E_w \rceil} - offset\right) \tag{6}$$

$$\alpha(t)_{CALR_{PS,EW}} = \begin{cases} \alpha_0 \cdot \dfrac{t+1}{E_w}, & E_w < t,\ set\ initial\ E_w \\ \dfrac{\alpha_0}{2}\left(C_a+1\right), & E_w \geq t \end{cases} \tag{7}$$

where t is the iteration number, T is the total number of training iterations, M is the number of cycles as in the original function, f is a monotonically decreasing function, E_w represents the number of warm-up epochs, an *offset* represents the offset value, and α_0 defines the initial LR. Notably, the periodic shift P_s and *offset* are constants, which may not necessarily impede the model's hyperparameter tuning, whereas the warm-up epoch initialization must be carefully chosen. We discuss further in the result section the choice of warm-up epoch used in the experiment.

B. ADAPTIVE OPTIMIZERS

As mentioned earlier, DL models as an optimization problem require both an optimizer and an LR schedule for faster convergence solutions. A simple gradient descent (slope) technique for minimizing the cost function of the optimization DL problem is expressed as:

$$w_{i+1} = w_i - \alpha \frac{\partial C}{\partial w}\left(w_i\right), \tag{8}$$

where w is the weights parameter of the network, C is the cost function of the model, and α represents the LR. When the gradient is large, gradient-based approaches are challenged for lack of parameter optimization. Hence, in the parameter update rule, the gradient is multiplied by a little constant known as the LR to resolve this issue. Increasing learning efficiency, even in a non-convex case, requires constant LR and methods for modifying the value at each step update, which LR schedules with optimizers are meant to do.

1) SGD

The SGD update employs (8) for every w_i at each time T step expressed as follows:

$$w_{T+1,i} = w_{T,i} - \alpha \cdot G_{T,i} \tag{9}$$

2) RMSprop

RMSProp, one of the adaptive optimizers, modifies the LR in each step using the square-root of the gradient's squared

TABLE 1. Characteristics of datasets used in the paper.

Dataset	Feature	Class	Dimension	Channel	#Instances
MNIST	Handwritten 0...9	10	28x28	gray	70,000
CIFAR10	Natural pictures	10	32x32	color	70,000
CIFAR100	Natural pictures	100	32x32	color	70,000
TinyImageNet	Natural images	200	64x64	color	120,000
Iranian Telecom	Numeric/categorical	2	16	-	3,150
German Credit	Numeric/categorical	2	21	-	1,000

Exponential Moving Mean (EMM). The RMSprop update rule is expressed as:

$$w_{i+1} = w_i + \alpha \frac{G_i}{\sqrt{EMM_i}} \tag{10}$$

$$EMM_i = \beta \cdot EMM_{i-1} + (1-\beta)\, G_i^2 \tag{11}$$

where G_i represents the ith step in the gradient, and β represents the coefficient of EMM.

3) ADAM

RMSProp strategy was combined with another well-known optimizer called adaptive gradient (adagrad) to form Adam, thereby exploring the inherent capabilities of the two and, consequently, overcoming their weakness. Adam's update rule is expressed as:

$$w_{i+1} = w_i - \alpha \frac{\hat{\boldsymbol{m}}_i}{\sqrt{\hat{\boldsymbol{v}}_i + \epsilon}}, \tag{12}$$

where

$$\boldsymbol{m}_i = \gamma_1 \boldsymbol{m}_{i-1} + (1-\gamma_1)\frac{\partial C(w_i)}{\partial w} \tag{13}$$

$$\boldsymbol{v}_i = \gamma_2 \boldsymbol{v}_{i-1} + (1-\gamma_2)\left(\frac{\partial C(w_i)}{\partial w}\right)^2 \tag{14}$$

Hence, $\hat{\boldsymbol{m}}_i = \boldsymbol{m}_i / \left(1-\gamma_1^i\right)$, and $\hat{\boldsymbol{v}}_i = \boldsymbol{v}_i / \left(1-\gamma_2^i\right)$. In order to update the LR based on the ratios of the gradients, the Adam approach computes the EMM for each gradient and its square.

C. DATASETS

Five different datasets: three image datasets (MNIST, CIFAR10, and CIFAR100) and two unbalanced non-image datasets (Iranian Telecom and German Credit) were first used in the paper. At the same time, an additional experiment was performed using TinyImageNet to further benchmark model performance. Contrary to past studies, we used the non-image datasets in the viewpoint of this paper to embrace recent research support and the possibility of using DL classifiers to analyze such datasets. In the case of the former, MNIST consists of 70,000 28×28 grayscale handwriting digits labeled 0 to 9, with 7000 images per digit. CIFAR10 and CIFAR-100 are also image datasets considered in this paper. There are 60000 images for both datasets with 10 and 100 class labels, respectively. Each image in the dataset is 32×32 with RGB channels. The summary of the datasets is presented in Table 1.

The Iranian Telecom and German credit datasets were initially passed through data pre-processing. These included converting the categorical features using dummy encoding, normalizing the numeric feature using a standard scaler, and reshaping the dataset to fit the DL classifiers.

D. EXPERIMENTAL SETUP

Two basic DL classifiers were employed in the first part of the experiment to evaluate the performance of the proposed method in comparison with the existing cosine annealing method and the fixed LR. In the second part of the experiment, we further benchmarked the proposed method on selected pre-trained models. The DL classifiers are Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN). Each classifier is a simple model met to experiment with the paper's objective and is not necessarily a state-of-the-art model. In the case of the CNN classifier, 2-dimensional (2D-CNN) was used for the image, whereas 1-dimensional (1D-CNN) was used for the non-image datasets. This convention allows the non-image to be passed into the convolution layer as sequence-like data. The summary of the models' hyperparameter tuning with respect to each dataset used is presented in Table 2.

Most experiments used a split ratio of 80:20 for train and test. The splitting process ensures that models are trained to the unseen data [45]. One crucial point with the two non-image datasets is that they are unbalanced. Hence, model training may be biased towards the majority class as against the minority class if the imbalance problem is not resolved. A GSMOTE [58] data resampling technique was then employed in the experiment to solve this challenge.

This paper focuses on experimenting with the proposed LR schedule, ps-CALR, using the Adam optimizer. Meanwhile, further experiments were carried out to include the use of SGD and RMSprop optimizers as well as fixed LR of 0.01 and 0.001 to benchmark model performance. We used an initial LR of 0.01, warm-up epochs of 2, and set the offset constant to 0.0002. The choice of warm-up epochs depends on the initial LR. If the initial LR is high, a choice of small warm-up epochs is desirable, whereas a low initial LR requires a high value of warm-up epochs. Since we used a high LR throughout the experiments, a small value of warm-up epochs was chosen to prevent the model from learning too quickly and diverging from the optimal solution [59].

TABLE 2. Hyperparameter tuning of MLP and CNN models used.

Model	Dataset	Input	Model Hyperparameters							Layers		Output	
			Batch Size	Epoch	Kernel Size	Filter	AF	Dropout	BN	Layers Con [a], fc [b]	#Neurons	#Classes	AF
MLP	MNIST	(784,)	128	30			relu	0.2	✓	fc:3	128,64,32	10	Softmax
	CIFAR10	(32*32*3,)	✓	✓			✓	✓	✓	✓	512,256, 128	✓	✓
	CIFAR100	✓	✓	✓			✓	✓	✓	✓	✓	100	✓
	Iranian Telecom	(23,)	32	✓			✓	✓	✓	✓	64,32,16	2	Sigmoid
	German Credit	(48,)	✓	✓			✓	✓	✓	✓	✓	✓	✓
CNN 2D [c] 1D [d]	MNIST	(28,28,1)	128	✓	3	3	✓			1,2	32, 64 64	10	Softmax
	CIFAR10	(32,32,3)	✓	✓	✓	✓	✓			✓	✓	✓	✓
	CIFAR100	✓	✓	100	✓	✓	✓	✓	✓	2,2	32, 128 1024, 512	100	✓
	Iranian Telecom	(23,1)	32	30	6	6	✓	✓	✓	3,2	64,128,256 32,16	2	Sigmoid
	German Credit	(48,1)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

[a]Con: represents convolutional layer, [b]fc: represents fully connected layer, [c]2D CNN represents 2-dimensional CNN for image classification,
[d]1D CNN represents 1-dimensional CNN for non-image classification

TABLE 3. Comparative results of the proposed method with fixed LR and optimizers using MLP (part a).

Datasets/LR	0.01	0.001	Adam	Sgd	Rmsprop	Adam+ calr	Adam+ ps-calr
MNIST	0.9782	0.9797	0.9786	0.9782	0.9807	0.9804	**0.9837**
CIFAR10	0.4884	0.4838	0.5120	0.4894	0.4742	0.5278	**0.5306**
CIFAR100	0.2284	0.2435	0.2442	0.2164	0.2369	**0.2683**	0.2664
Iranian Telco	0.9286	0.9349	0.9397	0.8523	0.9365	0.9413	**0.9429**
German Credit	0.6933	0.7133	0.7133	0.6000	0.7000	0.7133	**0.7200**

TABLE 4. Comparative results of the proposed method and optimizers using MLP (part b).

Datasets/LR	Adam+ calr	Sgd+ calr	Rmsprop+ calr	Sgd+ ps-calr	Rmsprop+ ps-calr	Adam+ ps-calr
MNIST	0.9804	0.9790	0.9807	0.9778	0.9816	**0.9837**
CIFAR10	0.5278	0.5246	0.5232	0.5157	0.5249	**0.5306**
CIFAR100	**0.2683**	0.2118	0.2620	0.2011	0.2533	0.2664
Iranian Telco	0.9413	0.6841	0.9238	0.6222	0.9349	**0.9429**
German Credit	0.7133	0.6067	0.6933	0.6267	0.7133	**0.7200**

Subsequent to the two classifiers used in the first part of the experiment, the following pre-trained models, including VGG16 [60], VGG19 [60], RestNet50 [61], and InceptionV3 [62], were experimented with the proposed method. The pre-trained classifiers' accuracies were compared in all experiments.

IV. EMPIRICAL RESULTS AND DISCUSSION

A. MLP CLASSIFIER

The MLP classifiers used in the experiment generally consist of the input layer (I/L), hidden layers (H/L), and the output layer (O\L) with Softmax or sigmoid activation function, depending on the classification task. Dropout and batch normalization layers were added to reduce overfitting and enhance model generalization in some model training. The MLP experimental results are presented in Tables 3 and 4. Table 3 compares fixed LR schedules and adaptive optimizers without incorporating specific LR annealing with Adam+Cosine Annealing Learning Rate (CALR) and the proposed method. Additionally, the existing CALR and the proposed method were compared by incorporating the optimizers as presented in Table 4. The proposed method shows competitive performance in all the experiments over the other methods for all datasets except the cifar100.

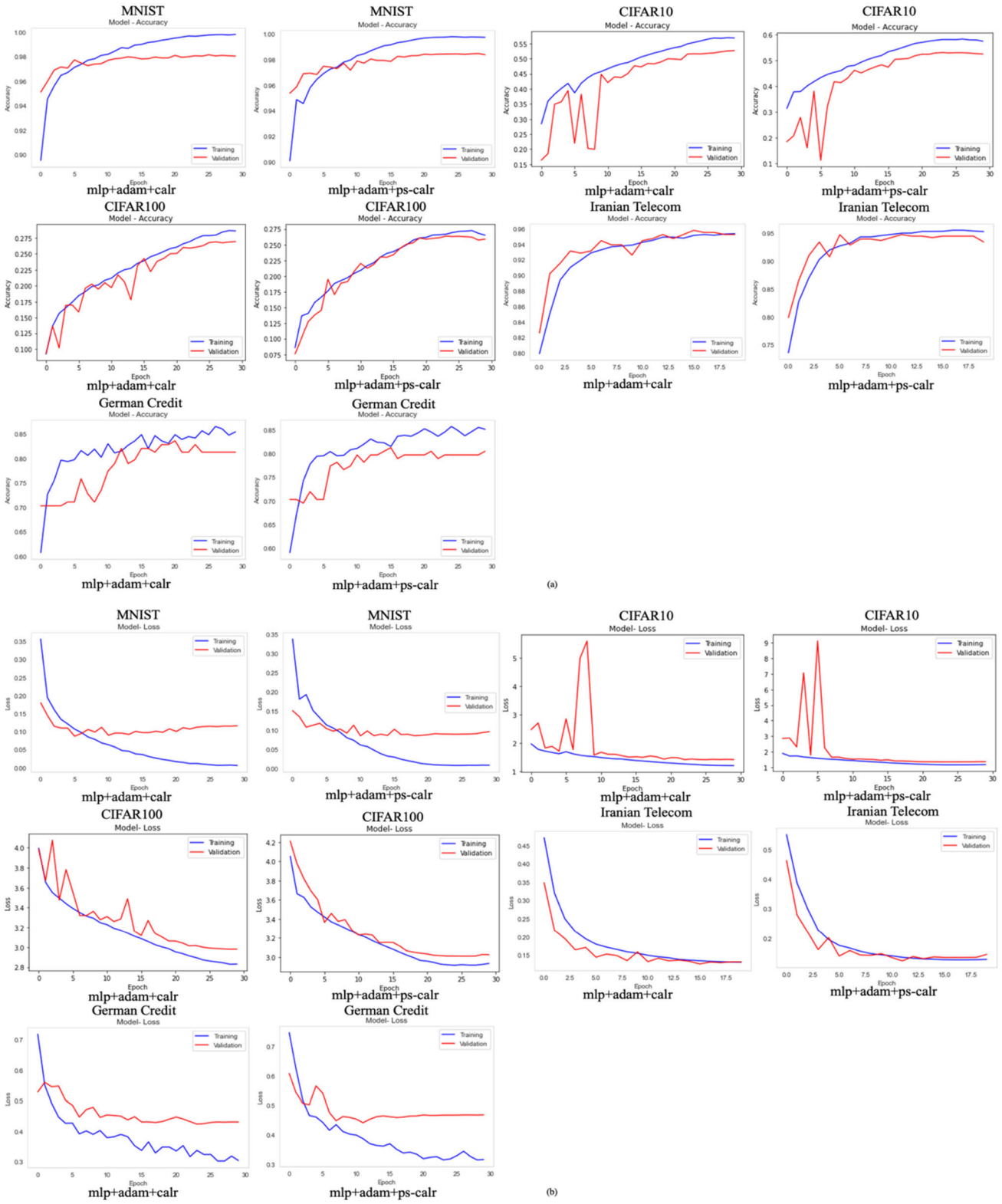

FIGURE 6. MLP classifier's plots for cosine annealing and the proposed method (a) training and validation accuracy, (b) training and validation loss.

Specifically, the proposed approach offers an improvement over the existing CALR for Adam optimizer. The proposed method yielded, as against the existing CALR, an accuracy of 98.37% against 98.04% for MNIST, 53.06% against 52.78% for cifar10, 94.29% against 94.13% for Iranian Telecom, and finally 72.00% against 71.33% for German credit. The MLP model accuracy and loss result for both Adam+CALR and Adam+ps-CALR for each dataset is shown in Fig. 6.

TABLE 5. Comparative results of the proposed method with fixed LR and optimizers using CNN (part a).

Datasets/LR	0.01	0.001	Adam	Sgd	Rmsprop	Adam+ calr	Adam+ ps-calr
MNIST	0.9858	0.9916	0.9896	0.9860	0.9914	0.9919	**0.9929**
CIFAR10	0.4283	0.7035	0.6675	0.6334	0.6966	0.7084	**0.7178**
CIFAR100	0.5425	0.5807	0.5820	0.5807	0.5789	0.5794	**0.5849**
Iranian Telco	0.9524	0.9556	0.9556	0.9635	0.9556	0.9651	**0.9667**
German Credit	0.7267	0.7533	0.7600	0.7600	0.7533	0.7667	**0.7733**

TABLE 6. Comparative results of the proposed method and optimizers using CNN (part a).

Datasets/LR	Adam+ calr	Sgd+ calr	Rmsprop+ calr	Sgd+ ps-calr	Rmsprop+ ps-calr	Adam+ ps-calr
MNIST	0.9919	0.9816	0.9914	0.9802	0.9926	**0.9929**
CIFAR10	0.7084	03867	0.7098	03787	0.6946	**0.7178**
CIFAR100	0.5794	0.2683	0.5691	0.2060	0.5656	**0.5849**
Iranian Telco	0.9651	0.9635	0.9619	0.9587	0.9635	**0.9667**
German Credit	0.7667	0.7333	0.7667	0.7333	0.7267	**0.7733**

TABLE 7. Comparative results of the proposed method with fixed lr and existing cosine using pre-trained models.

Pre-trained Models	Datasets/LR	0.01	0.001	adam+calr	adam+ps-calr
VGG16	MNIST	0.9726 (0.1123)[e]	0.9743 (0.0814)	0.9785 (0.0814)	**0.9812 (0.0734)**
	CIFAR10	0.6215 (1.1479)	0.6260 (1.2695)	0.6411 (1.1668)	**0.6424 (1.0938)**
	CIFAR100	0.3602 (3.3527)	0.3745 (3.6430)	0.3894 (3.8711)	**0.3908 (3.6492)**
VGG19	MNIST	0.9716 (0.1103)	0.9751 (0.1011)	0.9810 (0.0743)	**0.9813 (0.0678)**
	CIFAR10	0.6068 (1.1919)	0.6126 (1.2243)	0.6328 (1.1478)	**0.6350 (1.0990)**
	CIFAR100	0.3375 (3.2659)	0.3548 (3.4557)	0.3652 (3.6644)	**0.3657 (3.4538)**
ResNet50	MNIST	0.9879 (0.0411)	0.9923 (0.0310)	0.9919 (0.0284)	**0.9942 (0.0228)**
	CIFAR10	0.7158 (1.0449)	0.7377 (1.3989)	0.6893 (0.8963)	**0.7383 (0.8083)**
	CIFAR100	0.6957 (1.1654)	0.7466 (0.8594)	0.7785 (0.7763)	**0.7799 (0.7549)**
InceptionV3	MNIST	0.9685 (0.0831)	0.9855 (0.0505)	0.9867 (0.0527)	**0.9926 (0.0282)**
	CIFAR10	0.8456 (0.4312)	0.8541 (0.4272)	0.8904 (0.3305)	**0.8925 (0.3166)**
	CIFAR100	0.7299 (0.9397)	0.7402 (0.8830)	0.7806 (0.7579)	**0.7834 (0.7420)**

[e]Validation loss of the models for each dataset is indicated in the bracket

We observed that the MLP classifier performs significantly for MNIST and, surprisingly, also for Iranian Telecom and German credit datasets. Despite the fact that the MLP performance was not quite impressive with both cifar10 and 100, we observed that the proposed method aids model improvement compared with fixed LR. The indications in summary are: -

i. The MNIST is a grayscale image set, and with proper reshaping, as done in this paper, the MLP classifier trains very well on the data with welcoming performance.
ii. MLP also trained well on non-image data, indicating a comparable performance with conventional ML models if used in the same experiment.

B. CNN CLASSIFIER

A CNN model works for image and pattern classification tasks. It leverages convolution and pooling techniques to

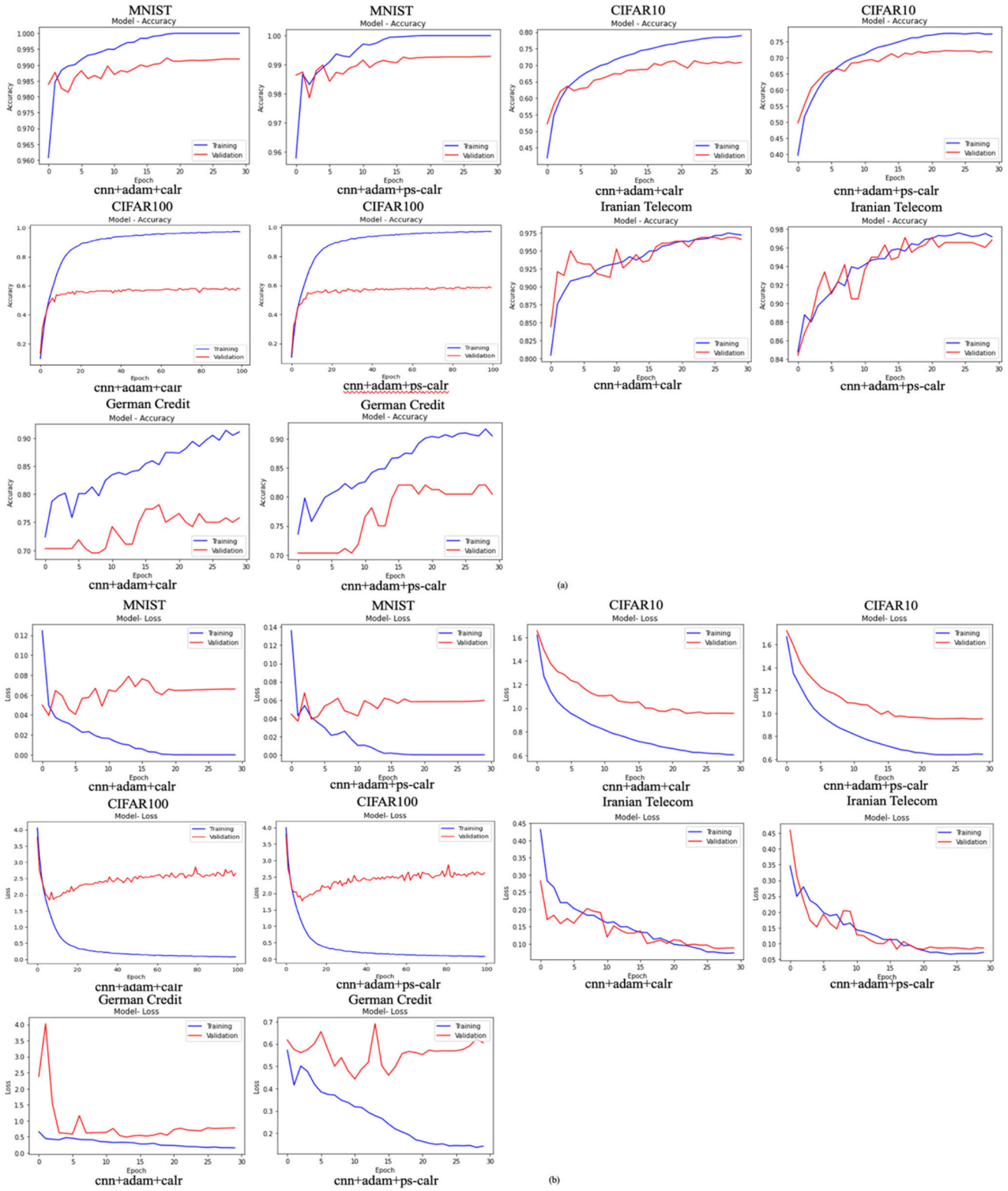

FIGURE 7. CNN classifier's plots for cosine annealing and the proposed method (a) training and validation accuracy, (b) training and validation loss.

provide automatic feature extraction from the data input and reliably predict categories to which the label classes belong. Two variant types of the CNN model were implemented in the experiment, including conventional 2D-CNN for analyzing the image data and 1D-CNN for analyzing the non-image datasets. The CNN classifier experimental results are

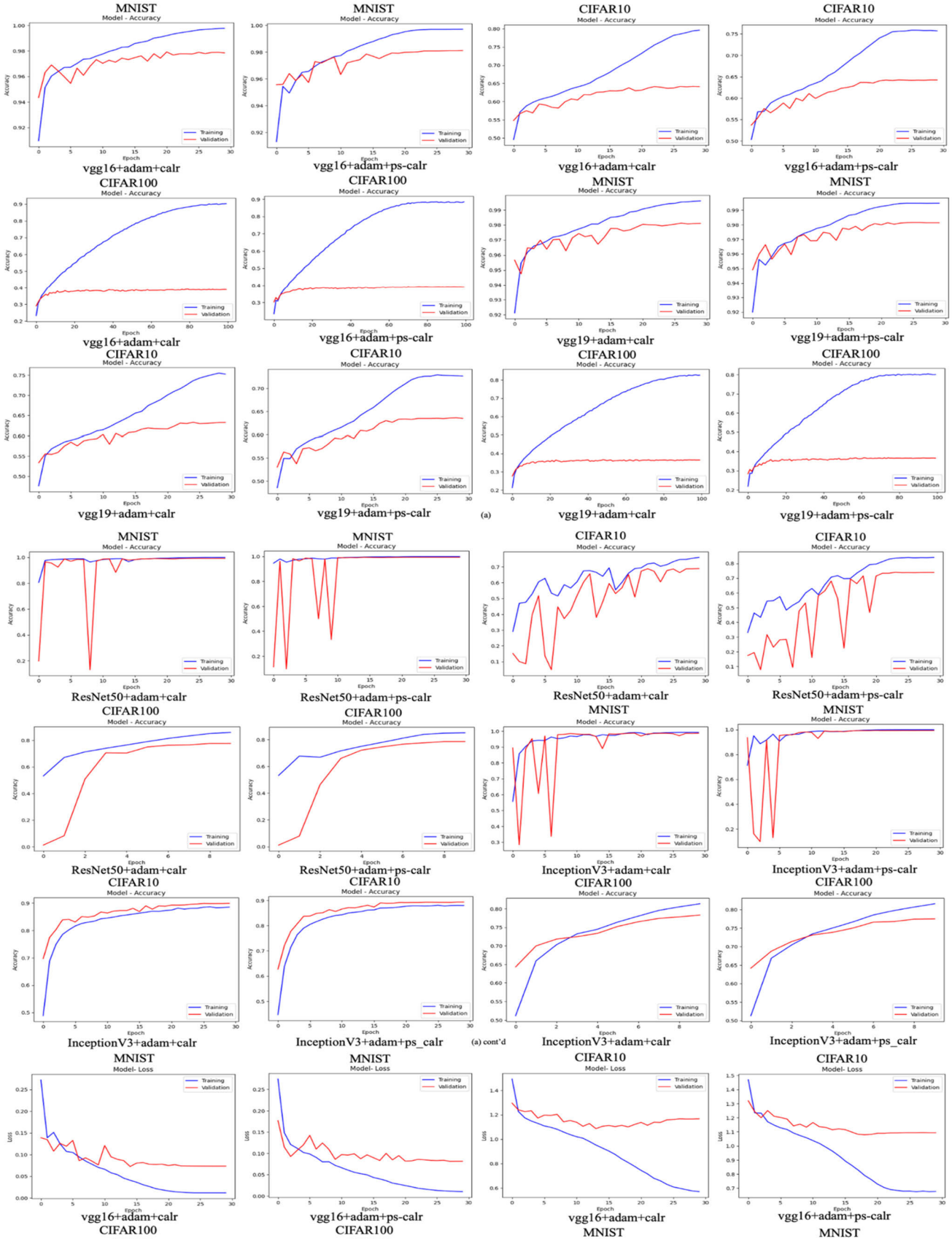

FIGURE 8. Pre-trained models' plots for cosine annealing and the proposed method (a) training and validation accuracy.

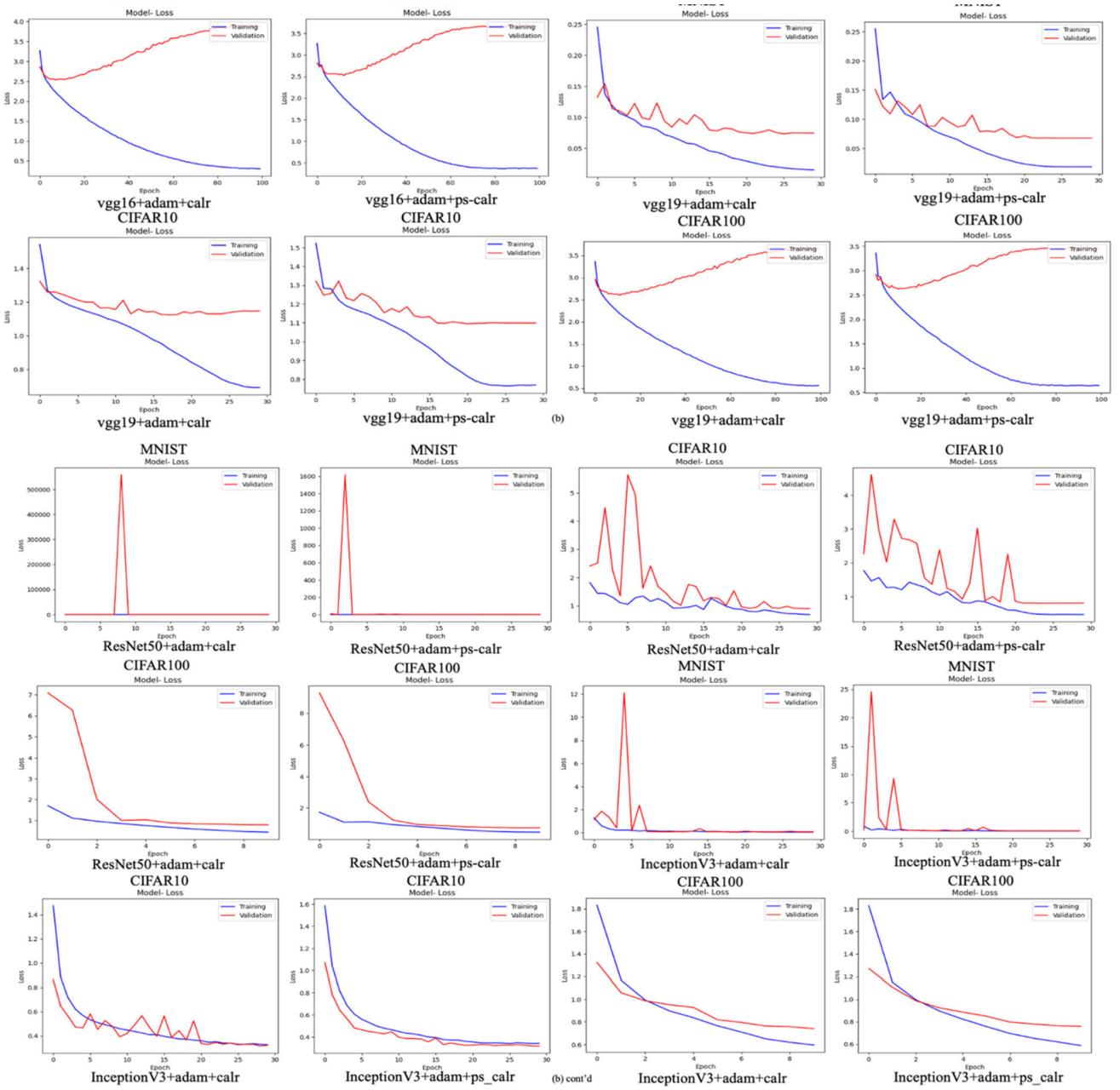

FIGURE 8. *(Continued.)* **Pre-trained models' plots for cosine annealing and the proposed method (a) training and validation accuracy, (b) training and validation loss.**

presented in Tables 5 and 6. The same comparative approach implemented for the MLP classifier earlier discussed is used for the CNN classifier for the selected optimizers and LR schedules.

In all the experiments, we observed that the proposed method shows a consistent and better performance over the rest of the methods in all the datasets used. The Adam optimizer with the proposed method offers a better improvement compared with the existing CALR and fixed LR using SGD and RMSprop optimizers. The proposed method yielded, as against the existing CALR, an accuracy of 99.29% against 99.19% for MNIST, 71.78% against 70.84% for cifar10, 58.13% against 57.94% for cifar100, 96.67% against 96.51% for Iranian Telecom, and finally 77.33% against 76.67% for German credit. Fig. 7 shows each dataset's CNN model accuracy and loss results for Adam+CALR and Adam+ps-CALR.

Notably, results indicated that the CNN model performs better than its counterpart MLP model across all the datasets used. Unlike the MLP classifier, the CNN with the proposed method in the case cifar100 shows consistency and improved performance. We observed that both cifar10 and

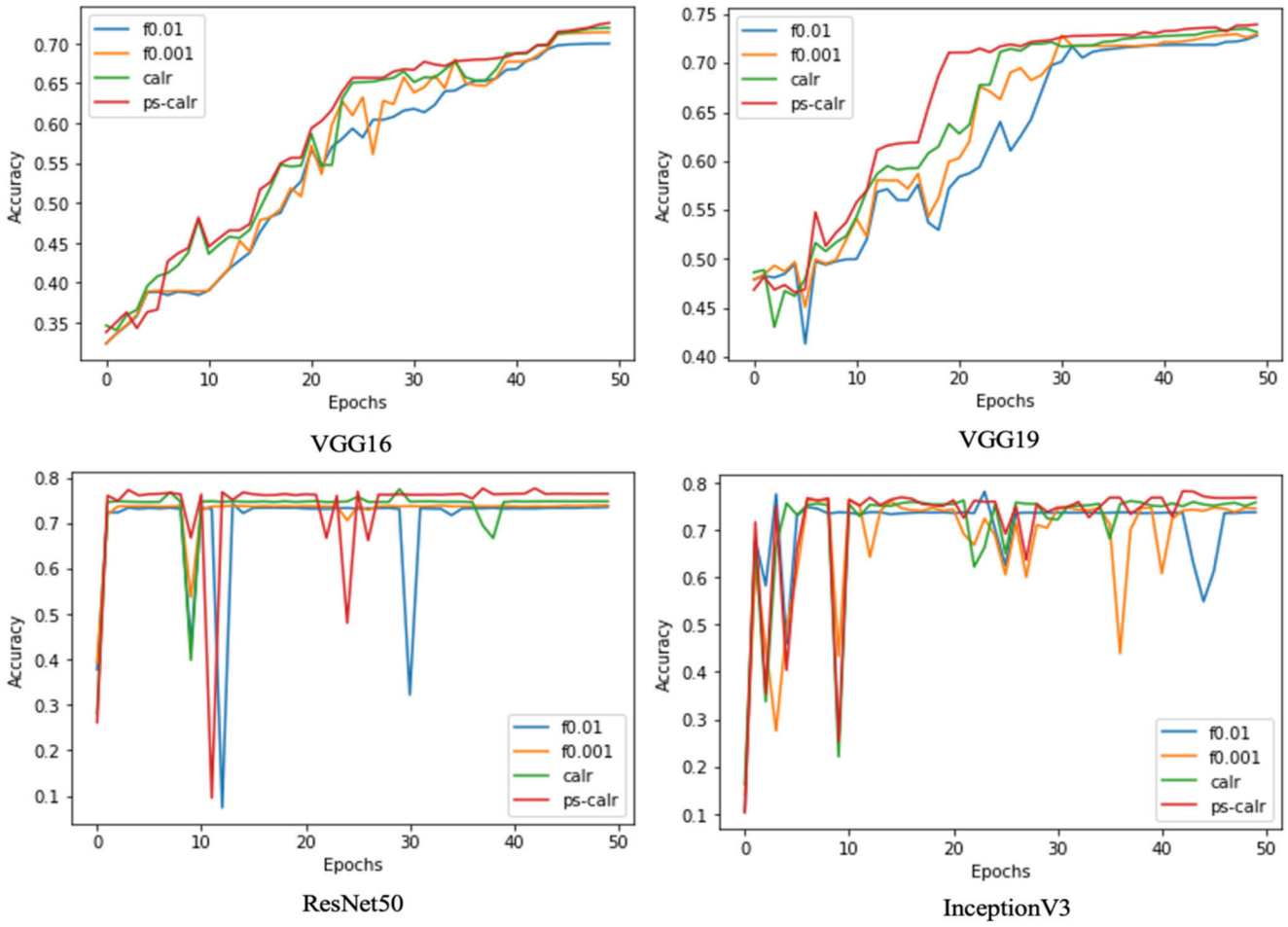

FIGURE 9. Pre-trained models' performance on the TinyImageNet using the proposed LR Method, existing cosine annealing, and the fixed LR.

100 performed better than the previous MLP model. The indications are:

i. CNN uses its inherent automatic feature extraction that helps to learn complex non-linear interactions to yield better classification.
ii. 1D CNN is adept at providing automatic feature extraction for DL models employed for non-image datasets rather than handcrafting or over-dependence on feature selection techniques in conventional ML models.

C. PRE-TRAINED MODELS

In the second part of the experiment, pre-trained models, including VGG16, VGG19, ResNet50, and InceptionV3, were trained on the three image datasets used in the previous experiments. The LR schedule policy considers fixed LR, existing CALR, and the proposed approach. The Adam optimizer for the LR schedules was only considered based on initial experiments and the fact that the paper focuses on cosine annealing with Adam. The experimental result of the pre-trained models is presented in Table 7.

In all the experiments, we observed that the proposed method demonstrates consistent and better performance over the rest of the LR schedules in all the datasets used. The Adam optimizer with the proposed method competes favorably with Adam+CALR, with a notable improvement recorded. For VGG16, the proposed method (Adam+ps-CALR) obtains an average test accuracy of 98.12% for MNIST, 64.24% for cifar10, and 39.08% for cifar100. For VGG19, the proposed method obtains an average test accuracy of 98.13% for MNIST, 63.50% for cifar10, and 36.57% for cifar100. In the case of ResNet50, the average test accuracy for MNIST is 99.42%, cifar10 is 73.83%, and cifar100 is 77.99% for cifar100. Whereas, in the case of InceptionV3, the proposed method obtains an average test accuracy of 99.26% for MNIST, 89.25% for cifar10, and 78.34% for cifar100. The pre-trained models' accuracies and losses for each dataset are shown in Fig. 8.

In addition to the previous comparison experiments, the TinyImageNet is used further to justify the performance of the proposed LR method. The TinyImageNet dataset is a subset of the ImageNet dataset with 200 classes. The dataset consists

of 100,000 images in a training set and 10,000 images each for validation and testing. The image size is 64×64 pixels.

To train the TinyImageNet on the previous pre-trained models, we resized the image to 96×96 and used a batch size 64 for 50 epochs. In addition, we used a custom data augmentation technique to avoid overfitting in order to achieve better accuracy. Our experimental findings, as illustrated in Fig. 9, show that not only did image classification training improve with using TinyImageNet, but also the proposed LR method consistently outperforms the existing CALR and fixed LR used.

V. CONCLUSION

LR schedule, as one of the key factors affecting the DL optimization problem, has been discussed in this paper. It is identified as the small term to help adjust the weights and biases of the model to minimize the CF along with an optimizer to obtain an optimal solution. We further mentioned that a high LR aids DL models in achieving strong generalization. In contrast, a lower LR aids in the model's minimum search, occasionally failing to converge because it is trapped at the local minimum. Consequently, a good LR helps the model converge faster and escape local minima to a global minimum in all cases.

We also discussed that a CALR was introduced to solve the trapping to the local minimum problem but did not provide a holistic exploration of the loss landscape for good model generalization. Hence, we proposed a ps-CALR to improve the existing CALR schedule. We demonstrated the proposed method with different classifiers and compared it with CALR and fixed LR. Findings show that the proposed method notably improved the performance of the MLP and CNN models, including the pre-trained models. In addition, we used TinyImageNet to further experiment with the proposed method, and the finding shows that the proposed method's performance was consistent. We observed that the proposed method is less adaptable to SGD compared with Adam and RMSprop optimizers. However, as earlier stated, recent studies provide a significant improvement to accelerate the performance of SGD with momentum using an effective step size method called Polyak [25], [26], [27]. Moreover, this paper focuses on existing CALR and improving it to model generalization; hence, it did not compare with SGD+Polyak or other LR schedules (such as time-based, polynomial, exponential). We suggest a comparative approach to benchmark the performance of the proposed method with other LR schedules.

We also noted that the warm-up epoch may change depending on the complexity of applications or designs, especially for models requiring initial lower LR. In this instance, additional fine-tuning of the warm-up epoch to a higher value may be performed. Availability of required computing resources also limited experimenting with ImageNet or specific applications in state-of-the-art NLP tasks, segmentation and complex object detections. We suggest future work to experiment the proposed ps-CALR with NLP task.

REFERENCES

[1] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[2] D. Jha, P. H. Smedsrud, M. A. Riegler, D. Johansen, T. D. Lange, P. Halvorsen, and H. D. Johansen, "ResUNet++: An advanced architecture for medical image segmentation," in *Proc. IEEE Int. Symp. Multimedia (ISM)*, Dec. 2019, pp. 2250–2255, doi: 10.1109/ISM46123.2019.00049.

[3] S. Soleymanpour, H. Sadr, and M. N. Soleimandarabi, "CSCNN: Cost-sensitive convolutional neural network for encrypted traffic classification," *Neural Process. Lett.*, vol. 53, no. 5, pp. 3497–3523, Oct. 2021, doi: 10.1007/s11063-021-10534-6.

[4] Y. Bao, Z. Tang, H. Li, and Y. Zhang, "Computer vision and deep learning-based data anomaly detection method for structural health monitoring," *Struct. Health Monitor.*, vol. 18, no. 2, pp. 401–421, Mar. 2019, doi: 10.1177/1475921718757405.

[5] P. Wang, H. Liu, L. Wang, and R. X. Gao, "Deep learning-based human motion recognition for predictive context-aware human–robot collaboration," *CIRP Ann.*, vol. 67, no. 1, pp. 17–20, 2018, doi: 10.1016/j.cirp.2018.04.066.

[6] S. Suparatpinyo and N. Soonthornphisaj, "Smart voice recognition based on deep learning for depression diagnosis," *Artif. Life Robot.*, vol. 28, no. 2, pp. 332–342, May 2023, doi: 10.1007/s10015-023-00852-4.

[7] H. Sadr, M. M. Pedram, and M. Teshnehlab, "Multi-view deep network: A deep model based on learning features from heterogeneous neural networks for sentiment analysis," *IEEE Access*, vol. 8, pp. 86984–86997, 2020, doi: 10.1109/ACCESS.2020.2992063.

[8] H. Sadr, M. M. Pedram, and M. Teshnehlab, "A robust sentiment analysis method based on sequential combination of convolutional and recursive neural networks," *Neural Process. Lett.*, vol. 50, no. 3, pp. 2745–2761, Dec. 2019, doi: 10.1007/s11063-019-10049-1.

[9] C. Cao, F. Liu, H. Tan, D. Song, W. Shu, W. Li, Y. Zhou, X. Bo, and Z. Xie, "Deep learning and its applications in biomedicine," *Genomics, Proteomics Bioinf.*, vol. 16, no. 1, pp. 17–32, Feb. 2018, doi: 10.1016/j.gpb.2017.07.003.

[10] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–16.

[11] Z.-J. Wang, H.-B. Gao, X.-H. Wang, S.-Y. Zhao, H. Li, and X.-Q. Zhang, "Adaptive learning rate optimization algorithms with dynamic bound based on Barzilai–Borwein method," *Inf. Sci.*, vol. 634, pp. 42–54, Jul. 2023, doi: 10.1016/j.ins.2023.03.050.

[12] T. Carneiro, R. V. Medeiros Da NóBrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. R. Filho, "Performance analysis of Google colaboratory as a tool for accelerating deep learning applications," *IEEE Access*, vol. 6, pp. 61677–61685, 2018, doi: 10.1109/ACCESS.2018.2874767.

[13] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters," in *Proc. 26th ACM SIGKDD Knowl. Discovery Data Mining (KDD)*, New York, NY, USA, 2020, pp. 3505–3506, doi: 10.1109/SC41405.2020.00024.

[14] H. Zhao, F. Liu, H. Zhang, and Z. Liang, "Research on a learning rate with energy index in deep learning," *Neural Netw.*, vol. 110, pp. 225–231, Feb. 2019, doi: 10.1016/j.neunet.2018.12.009.

[15] J. Wei, X. Zhang, Z. Zhuo, Z. Ji, Z. Wei, J. Li, and Q. Li, "Leader population learning rate schedule," *Inf. Sci.*, vol. 623, pp. 455–468, Apr. 2023, doi: 10.1016/j.ins.2022.12.039.

[16] F. He, T. Liu, and D. Tao, "Control batch size and learning rate to generalize well: Theoretical and empirical evidence," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–10.

[17] H. Alibrahim and S. A. Ludwig, "Hyperparameter optimization: Comparing genetic algorithm against grid search and Bayesian optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 5, p. 740, Jun. 2020, doi: 10.1109/TETCI.2020.3020707.

[18] Y. Kim and M. Chung, "An approach to hyperparameter optimization for the objective function in machine learning," *Electronics*, vol. 8, no. 11, p. 1267, Nov. 2019, doi: 10.3390/electronics8111267.

[19] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020, doi: 10.1016/j.neucom.2020.07.061.

[20] V. Nguyen, "Bayesian optimization for accelerating hyper-parameter tuning," in *Proc. IEEE 2nd Int. Conf. Artif. Intell. Knowl. Eng. (AIKE)*, Jun. 2019, pp. 302–305, doi: 10.1109/AIKE.2019.00060.

[21] Y. Wu and L. Liu, "Selecting and composing learning rate policies for deep neural networks," *ACM Trans. Intell. Syst. Technol.*, vol. 14, no. 2, pp. 1–25, Apr. 2023, doi: 10.1145/3570508.

[22] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.

[23] S. You, W. Gao, Z. Li, Q. Yang, M. Tian, and S. Zhu, "Dynamic adjustment of the learning rate using gradient," in *Human Centered Computing*, vol. 13795. Cham, Switzerland: Springer, 2022, doi: 10.1007/978-3-031-23741-6_6.

[24] J. Jepkoech, D. M. Mugo, B. K. Kenduiywo, and E. C. Too, "The effect of adaptive learning rate on the accuracy of neural networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 8, pp. 736–751, 2021, doi: 10.14569/ijacsa.2021.0120885.

[25] X. Wang, M. Johansson, and T. Zhang, "Generalized Polyak step size for first order optimization with momentum," 2023, *arXiv:2305.12939*.

[26] N. Loizou, S. Vaswani, I. Laradji, and S. Lacoste-Julien, "Stochastic Polyak step-size for SGD: An adaptive learning rate for fast convergence," in *Proc. Mach. Learn. Res.*, vol. 130, 2021, pp. 1306–1314.

[27] M. Prazeres and A. M. Oberman, "Stochastic gradient descent with Polyak's learning rate," *J. Scientific Comput.*, vol. 89, no. 1, pp. 1–14, Oct. 2021, doi: 10.1007/s10915-021-01628-3.

[28] N. S. Keskar, J. Nocedal, P. T. P. Tang, D. Mudigere, and M. Smelyanskiy, "On large-batch training for deep learning: Generalization gap and sharp minima," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–16.

[29] C. Baldassi, F. Pittorino, and R. Zecchina, "Shaping the learning landscape in neural networks around wide flat minima," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 1, pp. 161–170, Jan. 2020, doi: 10.1073/pnas.1908636117.

[30] J. Kaddour, L. Liu, R. Silva, and M. J. Kusner, "When do flat minima optimizers work?" 2022, *arXiv:2202.00661*.

[31] Z. Xie, I. Sato, and M. Sugiyama, "A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima," in *Proc. 9th Int. Conf. Learn. Represent.*, 2021, pp. 1–28.

[32] C. Baldassi, C. Lauditi, E. M. Malatesta, G. Perugini, and R. Zecchina, "Unveiling the structure of wide flat minima in neural networks," *Phys. Rev. Lett.*, vol. 127, no. 27, Dec. 2021, Art. no. 278301, doi: 10.1103/physrevlett.127.278301.

[33] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," in *Proc. 34th Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, 2017, pp. 1019–1028.

[34] Z. Liu, "Super convergence cosine annealing with warm-up learning rate," in *Proc. 2nd Int. Conf. Artif. Intell., Big Data Algorithms (CAIBDA)*, Nanjing, China, Jun. 2022, pp. 1–7.

[35] Y. Li, C. Wei, and T. Ma, "Towards explaining the regularization effect of initial large learning rate in training neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–12.

[36] B. Garimella, G. V. S. N. R. V. Prasad, and M. H. M. K. Prasad, "Churn prediction using optimized deep learning classifier on huge telecom data," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 3, pp. 2007–2028, Mar. 2023, doi: 10.1007/s12652-021-03413-4.

[37] V. Haridasan, K. Muthukumaran, and K. Hariharanath, "Arithmetic optimization with deep learning enabled churn prediction model for telecommunication industries," *Intell. Autom. Soft Comput.*, vol. 35, no. 3, pp. 3531–3544, 2023, doi: 10.32604/iasc.2023.030628.

[38] S. J. Haddadi, M. O. Mohammadi, M. Bahrami, E. Khoeini, M. Beygi, and M. H. Khoshkar, "Customer churn prediction in the Iranian banking sector," in *Proc. Int. Conf. Appl. Artif. Intell. (ICAPAI)*, May 2022, pp. 1–6, doi: 10.1109/ICAPAI55158.2022.9801574.

[39] S. Li, G. Xia, and X. Zhang, "Customer churn combination prediction model based on convolutional neural network and gradient boosting decision tree," in *Proc. 5th Int. Conf. Algorithms, Comput. Artif. Intell.*, pp. 1–6, Dec. 2022, doi: 10.1145/3579654.3579666.

[40] S. Zubair, A. K. Singha, N. Pathak, N. Sharma, S. Urooj, and S. R. Larguech, "Performance enhancement of adaptive neural networks based on learning rate," *Comput., Mater. Continua*, vol. 74, no. 1, pp. 2005–2019, 2023, doi: 10.32604/cmc.2023.031481.

[41] P. Mishra and K. Sarawadekar, "Polynomial learning rate policy with warm restart for deep neural network," in *Proc. IEEE Region 10 Conf. (TENCON)*, Oct. 2019, pp. 2087–2092, doi: 10.1109/TENCON.2019.8929465.

[42] K. Wang, Y. Dou, T. Sun, P. Qiao, and D. Wen, "An automatic learning rate decay strategy for stochastic gradient descent optimization methods in neural networks," *Int. J. Intell. Syst.*, vol. 37, no. 10, pp. 7334–7355, Oct. 2022, doi: 10.1002/int.22883.

[43] J. Park, D. Yi, and S. Ji, "A novel learning rate schedule in optimization for neural networks and it's convergence," *Symmetry*, vol. 12, no. 4, p. 660, Apr. 2020, doi: 10.3390/sym12040660.

[44] Y. Li, Q. Zhang, and S. W. Yoon, "Gaussian process regression-based learning rate optimization in convolutional neural networks for medical images classification," *Expert Syst. Appl.*, vol. 184, Dec. 2021, Art. no. 115357, doi: 10.1016/j.eswa.2021.115357.

[45] K. Nakamura, B. Derbel, K.-J. Won, and B.-W. Hong, "Learning-rate annealing methods for deep neural networks," *Electronics*, vol. 10, no. 16, p. 2029, Aug. 2021, doi: 10.3390/electronics10162029.

[46] H. Iiduka, "Appropriate learning rates of adaptive learning rate optimization algorithms for training deep neural networks," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13250–13261, Dec. 2022, doi: 10.1109/TCYB.2021.3107415.

[47] K. Itakura, K. Atarashi, S. Oyama, and M. Kurihara, "Adapting the learning rate of the learning rate in hypergradient descent," in *Proc. Joint 11th Int. Conf. Soft Comput. Intell. Syst. 21st Int. Symp. Adv. Intell. Syst. (SCIS-ISIS)*, Dec. 2020, pp. 1–6, doi: 10.1109/SCISISIS50064.2020.9322765.

[48] Z. Hao, Y. Jiang, H. Yu, and H. D. Chiang, "Adaptive learning rate and momentum for training deep neural networks," in *Machine Learning and Knowledge Discovery in Databases. Research Track* (Lecture Notes in Computer Science, Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12977. Cham, Switzerland: Springer, 2021, pp. 381–396, doi: 10.1007/978-3-030-86523-8_23.

[49] Q. Tong, G. Liang, and J. Bi, "Calibrating the adaptive learning rate to improve convergence of Adam," *Neurocomputing*, vol. 481, pp. 333–356, Apr. 2022, doi: 10.1016/j.neucom.2022.01.014.

[50] Z. Xie, X. Wang, H. Zhang, I. Sato, and M. Sugiyama, "Adaptive inertia: Disentangling the effects of adaptive learning rate and momentum," in *Proc. Mach. Learn. Res.*, vol. 162, 2022, pp. 24430–24459.

[51] J. Yang and F. Wang, "Auto-ensemble: An adaptive learning rate scheduling based deep learning model ensembling," *IEEE Access*, vol. 8, pp. 217499–217509, 2020, doi: 10.1109/ACCESS.2020.3041525.

[52] W. Zhu and Y. Tang, "DALU: Adaptive learning rate update in distributed deep learning," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI)*, Oct. 2021, pp. 203–209, doi: 10.1109/SWC50871.2021.00036.

[53] G. Ioannou, T. Tagaris, and A. Stafylopatis, "AdaLip: An adaptive learning rate method per layer for stochastic optimization," *Neural Process. Lett.*, vol. 55, no. 5, pp. 6311–6338, Oct. 2023, doi: 10.1007/s11063-022-11140-w.

[54] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get M for free," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–14.

[55] Y. Nie, M. Carratú, M. O'Nils, P. Sommella, A. U. Moise, and J. Lundgren, "Skin cancer classification based on cosine cyclical learning rate with deep learning," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2022, pp. 1–6, doi: 10.1109/I2MTC48687.2022.9806568.

[56] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2018, pp. 328–339, doi: 10.18653/v1/p18-1031.

[57] M. Nagubandi, R. Walia, A. Karanath, and G. N. Pillai, "Electric load forecasting using dual-stage attention network with cosine annealed warm restart schedule," in *Proc. Int. Conf. Emerg. Techn. Comput. Intell. (ICETCI)*, Aug. 2022, pp. 141–146, doi: 10.1109/ICETCI55171.2022.9921361.

[58] G. Douzas and F. Bacao, "Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE," *Inf. Sci.*, vol. 501, pp. 118–135, Oct. 2019, doi: 10.1016/j.ins.2019.06.007.

[59] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *Proc. 8th Int. Conf. Learn. Represent.*, 2020, pp. 1–14.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–14.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[62] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

OLANREWAJU VICTOR JOHNSON received the M.Sc. degree from the University of Benin, Benin City, Nigeria. He is currently pursuing the Ph.D. degree with Universiti Sains Malaysia. He studied advanced ICT management with the renowned Galilee International Management Institute, Israel, in 2014. His research interests include machine learning, artificial intelligence, data mining, decision-making analysis, big data, and cloud computing. He received the Research Project Management Certificate from the Cambridge Moller Institute, University of Cambridge, U.K., in 2020.

CHEW XINYING received the Ph.D. degree in statistical quality control from Universiti Sains Malaysia. She is currently a Senior Lecturer with the School of Computer Sciences, Universiti Sains Malaysia. She is also a Certified Trainer with the Human Resources Development Fund (HRDF) and a Professional Technologist with the Malaysia Board of Technologists (MBOT). Her research interests include advanced analytics and statistical quality/process control.

KHAI WAH KHAW received the Ph.D. degree from the School of Mathematical Sciences, Universiti Sains Malaysia (USM). He is currently a Senior Lecturer with the School of Management, USM. His research interests include statistical process control and advanced analytics.

MING HA LEE received the M.Sc. degree in applied statistics from Universiti Putra Malaysia. Prior to the Ph.D. degree, she was a Lecturer in a private college, where she taught mathematics for matriculation and business statistics at the bachelor's level. She joined the School of Engineering, Swinburne University of Technology, Sarawak Campus, as a Lecturer, in January 2007. Her research interest includes statistical process control.