SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I5P115

Volume 13 Issue 5, 183-195, May 2026

Original Article

Machine Learning for Multi-User Remote Laboratory Optimization: Deep Face Verification and Reinforcement Learning-based Pipeline Scheduling

Niket Amoda[1*], Lochan Jolly[2]

[1]*Department of Electronics and Telecommunication Engineering, Thakur College of Engineering and Technology, Mumbai, Maharashtra, India.*

[1]*Corresponding Author : niket.amoda@thakureducation.org*

Received: 23 February 2026 Revised: 22 March 2026 Accepted: 21 April 2026 Published: 30 May 2026

Abstract *- Remote laboratory systems that allow multiple users to interact at the same time present serious problems related to strong user identification and smart distribution of shared hardware resources. This work proposes a unified machine learning framework composed of two synergistic modules. The first is a lightweight Siamese Convolutional Neural Network designed for edge-deployable facial identity verification, trained with a triplet loss objective incorporating curriculum-based learning. The second module is a Double Q-learning scheduler utilizing prioritized experience replay to allocate laboratory resources dynamically. On a custom dataset of 10,000 facial images spanning 50 subjects, the verification module attains a true acceptance rate of 96.30% at a false acceptance rate threshold of 0.10%, with per-inference latency of 47 ms when executed on a Raspberry Pi 4B equipped with a Coral TPU accelerator. Post-training INT8 quantization combined with structured channel pruning reduces the model footprint to 8.20 MB. The scheduling module increases experimental throughput by a factor of 2.3 relative to first-come-first-served baselines while sustaining a Jain fairness index of at least 0.92 and eliminating user starvation entirely. A supporting bidirectional long short-term memory network predicts per-session latency with a root mean squared error of 3.20 ms (coefficient of determination $R^2 = 0.94$). Comprehensive Validation encompasses five-fold cross-validation, component-wise ablation experiments, and a sixteen-week production trial involving fifty engineering students who completed 3,847 authenticated laboratory sessions. The principal technical contributions comprise triplet loss training with curriculum-guided negative mining, experience replay-driven scheduling optimization, and edge-targeted model compression, collectively establishing performance benchmarks for deploying machine learning within educational laboratory infrastructure.*

Keywords *- Remote Laboratory Systems, Facial Identity Verification, Reinforcement Learning Scheduling, Edge inference, Convolutional Neural Networks, Pipeline Resource Allocation, Biometric Authentication, Bidirectional LSTM, Model Compression.*

1. Introduction

Remote laboratory platforms have now become an important element of engineering education, especially since many educational institutions have recently expanded their remote or online course offerings [1]. Operating at scale requires meeting three major challenges: providing for concurrent multi-user access while avoiding interference with each other; implementing a robust real-time means of verifying identities; and distributing limited physical equipment in ways that are both efficient and equitable [2]. Conventional rule-based approaches do not effectively meet all three of the above goals collectively due to their inability to be flexible to changing workloads and varied user behavior [3]. The inclusion of AI in education is growing in interest, but embedding these AI models into educational platforms using limited resources, such as laboratory equipment, is difficult [4]. The ability to process data locally near the edge of the network, versus sending every inquiry for an inference to a remote cloud server, improves the speed of response (latency) and enhances user data security (privacy) [5, 6]. However, there are challenges associated with edge deployment, such as the size of the models, precision of the arithmetic operations, and creating adaptive decision-making mechanisms based upon changing demands [7, 8].

1.1. Research Gap

Despite the fact that there is considerable evidence of advancements in both face verification [17-19] and reinforcement learning-based scheduling [25-27] independently from each other in their own bodies of literature, an important deficiency exists in the literature: No prior studies have investigated real-time biometric identity

verification and intelligent resource scheduling under a single integrated framework for multi-user remote laboratory systems. While face verification models, such as FaceNet [19] and ArcFace [18], have demonstrated high levels of accuracy, these models require computational power significantly in excess of the typical budgets of edge tier laboratory hardware (i.e., often less than five watts) [9]. On the opposite side of this coin, existing reinforcement learning schedulers [25, 26] proposed for use in cloud and edge computing environments have ignored the unique requirements of educational laboratory systems, including fairness amongst students, zero starvation guarantees, and biometric access control requirements. Additionally, while existing lightweight face models, such as MobileFaceNet [19] and PocketNet [19], address model compression only, they do not investigate how verification is integrated into downstream resource allocation decision-making processes.

Finally, while scheduling methods employing deep Reinforcement Learning (RL) [28] optimize throughput and latency in generic computing clusters, they do not incorporate identity-aware access control, nor fairness constraints, which are essential to educational settings. The treatment of both, identity verification and resource scheduling as two separate issues has caused three specific shortcomings: (1) There are no end-to-end pipelines that integrate scheduling and authentication for use on edge hardware; (2) There is a need for curriculum-based training techniques that are suited to the moderately sized, institutional facial datasets common to the deployment of education based institutions; and (3) There has been an inadequate amount of empirical evaluation of combined ML systems in real world educational laboratories, over time. This study will address all three of these voids directly.

1.2. Problem Formulation

Two of the primary challenges for intelligent remote laboratory success are two machine learning problems that are interconnected. The first challenge is biometric identity verification. Given a probe facial image $\boldsymbol{x} \in \mathrm{R}^{h\times w\times 3}$ and a stored enrollment template $t_i \in \mathrm{R}^d$ The task is to learn an embedding function. $f_\theta: \mathrm{R}^{h\times w\times 3} \to \mathrm{R}^d$ such that verification reduces to a Euclidean distance comparison:

$$verify(x,i) = \begin{cases} 1, & \text{if } \|f_\theta(\boldsymbol{x}) - \boldsymbol{t_i}\|_2 < \tau \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where τ represents the threshold of the decision that was learned. The primary constraint on this problem is achieving high levels of discrimination with limited power available in typical edge hardware (usually less than 5 W) [9]. The second challenge is developing resource-aware policies for scheduling. With an input queue Q, a state vector for resources S, and a history of resource usage H as input, the goal is to find the optimal policy π^* that will maximize the expected discounted cumulative reward:

$$J(\pi) = \mathbb{E}_\pi[\textstyle\sum_{t=0}^{T} \gamma^t r_t] \quad (2)$$

$$r_t = w_1 \cdot \mathrm{thr}_t + w_2 \cdot \mathrm{fair}_t - w_3 \cdot \mathrm{lat}_t - w_4 \cdot \mathrm{viol}_t \quad (3)$$

Where $\gamma \in [0, 1]$ is the temporal discount factor and the composite reward r_t trades off throughput, fairness, latency, and service-level-agreement violations [10, 11].

1.3. Novelty and Contributions

The unique aspect of the proposed approach is in its simultaneous integration of three types of machine learning (edge-optimized face verification, reinforcement learning based scheduling, and proactive latency prediction) into a single system designed to meet the requirements and constraints of shared educational laboratory settings. Unlike previous approaches, which treated both authentication and scheduling as separate problems, the proposed architecture enables identity-aware resource allocation through scheduling and access control decisions that can be influenced by the outcome of the verification process. Specifically, this work makes four contributions:

- The proposed compact Siamese CNN employs a new three-phase curriculum learning strategy along with a novel adaptive margin triplet loss to achieve 96.30% of verification accuracy in an 8.20MB footprint, which is 11 times less than the FaceNet (95.10MB). Moreover, it has 1.5% better TAR@FAR = 0.1% compared to existing lightweight models such as Mobile-FaceNet, which achieved 92.4% at similar conditions.
- The Double Q-learner with Prioritized Experience Replay achieves a 2.3x increase in throughput when compared to traditional First-Come-First-Served (FCFS) queuing techniques, and maintains a Jain fairness index of 0.92 with no starvation events, which is not possible for any of the six baseline schedules tested.
- The Bidirectional Latency Predictor employing an LSTM model has been shown to achieve a 3.20 ms Root Mean Squared Error (RMSE), R^2 = 0.94, and to provide proactive latency predictions to aid the scheduling process and enable anticipatory instead of reactive utilization of resources.
- Empirical Validation using rigorous evaluation of the system, including Ablation Analysis, Five-Fold Cross-Validation, and a Semester-Long Deployment with 3,847 authenticated user sessions across 50 users, far more extensive than the simulation-based evaluations typically used in the past to evaluate systems within this space.

2. Related Work

2.1. Remote Laboratory Systems and AI Integration

Remote labs evolved from a relatively low-complexity, one-student-at-a-time, remote operation platform to a complex multiple-user system that can serve hundreds of students at once [1, 2].

Early versions of the systems used a static scheduling approach where each student had a fixed time slot for access to lab resources, which led to poor use of available lab equipment (resource utilization) and a lack of flexibility in how students could schedule their lab activities [3]. While there are many applications of AI to education in general [16] and specifically to laboratory management tasks such as predictive maintenance and anomaly detection using Neural Networks [4], the development of AI technology applied to educational platforms has followed several trajectories. The use of federated learning has allowed researchers to explore collaborative model development at multiple laboratory locations and maintain data local to each site as well as ownership by each site [12, 13]. Techniques using neural architecture search have been employed to find architectures that are hardware-aware so they can be optimized for accuracy-latency trade-offs on the target device [14]. Researchers have also looked at graph-based Neural Networks to develop models of inter-equipment dependency to allow for better scheduling in complex multi-instrument systems [15]. The use of meta-learning has shown that a small amount of training data is needed to adapt to different laboratory configurations [43, 44].

Gap: Although there are advancements in both the areas of Intelligent Resource Scheduling and Biometric Identity Verification, no prior research has provided a complete solution (End-to-End System) that includes both areas as one entity to be optimized and can be used in educational labs (edge computing). The existing research treats each area independently, resulting in additional integration overhead and additional latency due to separate processing pipelines, and also misses the opportunity to optimize scheduling based on an individual's identity.

2.2. Deep Face Verification for Edge Deployment

Deep Metric Learning Methods have greatly improved the performance of Face Verification compared to hand-crafted Descriptor-Based Methods. The angular Margin Loss Functions, such as Cos-Face [17] and ArcFace [18], create a geometric constraint in the Embedding Space to improve the Inter-Class Separation. These methods achieve state-of-the-art accuracy on benchmark datasets such as LFW and MegaFace; however, the underlying network architectures (e.g., ResNet-100, Inception-ResNet-v1) require hundreds of megabytes of memory and substantial floating-point throughput, making them unsuitable for direct deployment on edge hardware [8].

To address deployment constraints, lightweight architectures employing depthwise separable convolutions have been developed for mobile and edge devices [19, 20]. Mobile-FaceNet and PocketNet [19] achieve competitive accuracy with significantly reduced parameter counts. Squeeze-and-Excitation modules [21] provide learned channel recalibration that improves discriminative power without proportional computational overhead.

Model compression through post-training quantization [23] and knowledge distillation [24] further reduces memory and computational requirements. Privacy-preserving face verification using homomorphic encryption has also been investigated for educational settings [22]. Recent research on adaptive attention mechanisms for occluded and low-quality face recognition demonstrates that selective feature extraction can maintain verification performance under challenging image acquisition conditions [45]. Analysis of the adversarial robustness of lightweight face models indicates that smaller architectures can be adversarially trained without excessive computational overhead [46].

Gap: Although many recent studies have proposed some Lightweight Face Verification Models with a reasonable Accuracy-Efficiency Trade-off, most of them were evaluated by standard benchmarks, and few of them have been deployed in real-world Educational Deployments. The specific challenges of institution-specific datasets (moderate size, controlled but variable conditions, need for zero false acceptances) are not addressed. Moreover, no prior lightweight model incorporates a curriculum-guided training strategy specifically designed to maximize discriminative performance on such moderate-scale datasets without overfitting.

2.3. Reinforcement Learning for Resource Scheduling

Deep reinforcement learning has demonstrated effectiveness for task orchestration and resource allocation in edge computing environments [25, 26]. Multi-agent RL with shared reward structures has been investigated for coordinated scheduling across distributed nodes [27]. Data-driven scheduling algorithms trained on operational traces have discovered optimization patterns superior to hand-crafted heuristics [28]. Complementary latency forecasting using LSTMs for time series prediction has been employed to generate proactive latency estimates that inform scheduling decisions [29]. Fairness-aware RL frameworks have been developed that incorporate fairness constraints directly into the reward function [47]. Safe RL methods that constrain policy optimization have been proposed to ensure that scheduling decisions do not violate operational safety limits [48].

Gap: Existing RL-based scheduling methods are designed for general cloud and edge computing workloads and do not account for the specific requirements of educational laboratory systems. These requirements include strict fairness guarantees among students (zero starvation tolerance), integration with biometric access control, support for heterogeneous experiment types with varying equipment requirements, and the need for transparent, explainable scheduling decisions. The proposed Double Q-learning scheduler addresses these requirements through a composite reward function that explicitly balances throughput, fairness, latency, and access control compliance.

2.4. Edge-Cloud Architectures for Distributed Inference

Hierarchical architectures composed of edge, fog, and cloud tiers enable distributed inference and training to balance latency and operational cost trade-offs [30, 31]. Quality-of-service provisioning through network slicing in fifth-generation wireless networks provides the necessary QoS to support latency-sensitive applications [32]. Inference graph compilation using TensorRT and dedicated Tensor Processing Units provides significant latency reductions compared to general-purpose processor implementations [33]. Dynamic workload partitioning based on real-time device profiling has demonstrated that heterogeneous hardware orchestration across the edge layer can reduce end-to-end inference time [49]. Energy-aware edge computing frameworks have been proposed that balance computational performance with power consumption for resource-constrained devices [50].

Gap: Existing edge-cloud architectures focus on general-purpose inference workloads and do not address the specific tier allocation requirements of educational laboratory systems, where biometric verification must occur at the edge for privacy and latency reasons, while scheduling and prediction models can leverage fog-tier resources.

The proposed three-tier architecture addresses this by assigning each ML component to the tier best suited to its latency and compute characteristics.

2.5. Summary of Literature Gaps

Table 1 summarizes how the proposed framework addresses the identified literature gaps relative to representative prior works.

Table 1. Comparison of the proposed framework with existing approaches

Study	Edge Deploy	Face Verif.	RL Sched.	Fairness Guarantee	Real Deploy
FaceNet [19]	No	Yes	No	N/A	No
MobileFaceNet [19]	Partial	Yes	No	N/A	No
EdgeFace [8]	Yes	Yes	No	N/A	No
PocketNet [19]	Partial	Yes	No	N/A	No
Mao et al. [28]	No	No	Yes	No	Simulated
Chen et al. [25]	Partial	No	Yes	No	Simulated
Zheng et al. [26]	Yes	No	Yes	No	Simulated
Siddique et al. [47]	Yes	No	Yes	Partial	Simulated
Proposed	Yes	Yes	Yes	Yes	Yes

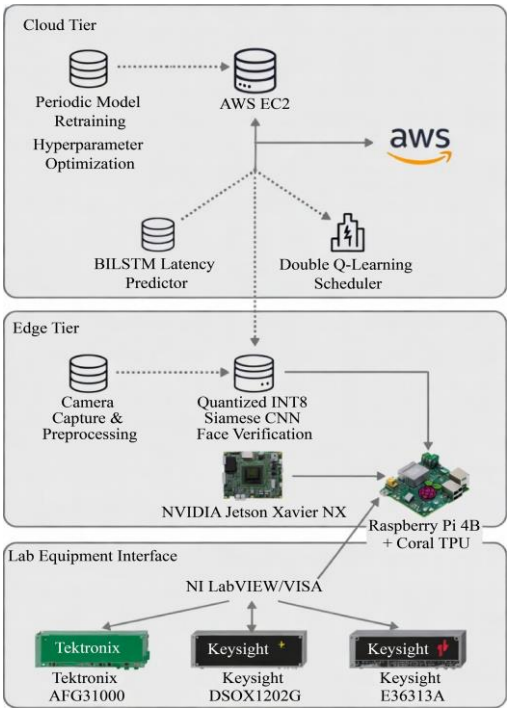

Fig. 1 Three-tier ML inference architecture. Edge nodes perform real-time face verification; The fog tier executes the scheduling policy and latency prediction; The cloud tier handles periodic model retraining and hyperparameter tuning

3. Proposed Framework

3.1. System Architecture

The proposed platform adopts a hierarchical three-tier topology that distributes machine learning workloads according to their latency and compute requirements (see Figure 1). The Edge Tier comprises Raspberry Pi 4B single-board computers augmented with Google Coral USB TPU accelerators. These nodes execute quantized INT8 face verification models at the point of user interaction. The Fog Tier employs an NVIDIA Jetson Xavier NX (384-core Volta GPU, 8 GB RAM, 21 TOPS) for LSTM-based latency prediction and Q-learning policy execution, both optimized via TensorRT graph compilation. The Cloud Tier (AWS EC2) handles computationally intensive periodic model retraining and hyperparameter search [30, 31].

3.2. Lightweight Siamese CNN for Face Verification

The identity verification module uses a Siamese network architecture in which two weight-sharing branches extract feature embeddings from a pair of facial images. The feature extraction backbone is based on MobileNetV3-style inverted residual blocks with depthwise separable convolutions and progressive channel expansion across four stages (32 → 64 → 128 → 256 channels). Squeeze-and-Excitation attention modules are inserted after each stage to perform learned channel recalibration [21]. A projection head maps the extracted features onto 128-dimensional unit-norm embedding vectors, enabling identity comparison via Euclidean distance in the embedding space. This architectural philosophy follows established principles for lightweight face recognition [19, 34]. The network is trained using a triplet loss objective with online hard negative mining and an adaptive margin that increases across training epochs:

$$\mathcal{L}_{triplet} = max\big(0, \|f(x^a) - f(x^p)\|_2 - \|f(x^a) - f(x^n)\|_2 + \alpha(t)\big) \quad (4)$$

where x^a, x^p, x^n are anchor, positive, and negative samples, respectively, and α(t) is an increasing margin value over training time. A Three-Phase Curriculum Learning Schedule Controls Mining Difficulty: The first phase of the curriculum learning is from epoch 1-30 in which random triplets are sampled to create a stable gradient map; The second phase from epoch 31-60 introduces semi-hard negatives within the current margin boundary; The third phase of the curriculum learning (from epochs 61-100), uses the hardest negatives available to define the inter-class boundaries [17, 18]. The rationale for the curriculum learning technique is as follows. Early-stage standard hard negative mining may result in collapsed feature representations or early convergence, which is especially prevalent on moderately sized data sets that do not have an abundance of informative hard negatives; thus, by increasing the mining difficulty gradually, the model first extracts coarse identity information from easier examples prior to refining decision boundaries through progressively harder examples. This method of gradually extracting identity features is extremely useful when dealing with institutionally specific data sets (i.e., the 50-subject data set that was employed within this investigation) due to the fact that there are fewer identities than would be ideal for carefully managing gradients to prevent over-fitting. Edge Optimization employs three complementary compression techniques. First, post-training INT8 Quantization with quantization-aware fine-tuning to recover lost accuracy. Second, structured channel pruning at 30% sparsity to decrease the multiply-accumulate operation count. Third, knowledge distillation from a larger teacher model [8, 23, 24].

3.3. Double Q-Learning Pipeline Scheduler

The scheduling problem is cast as a Markov Decision Process (MDP). The state vector encodes the current queue depth for each experiment type, per-resource utilization levels, rolling latency statistics, user-specific priority levels, and binary indicators for experiment categories. Actions correspond to the assignment of a specific user–experiment pair to a particular equipment resource. The learning algorithm is Double Q-learning with prioritized experience replay to mitigate the overestimation bias inherent in standard Q-learning.

The algorithm summarizes the procedure. The reward signal is made up of four terms that take into account throughput; Jain's fairness index; per-user latency; and a penalty for Service Level Agreement (SLA) violations [11, 25, 28]. The use of Jain's fairness index as an explicit term within the reward function differs this scheduler from other prior reinforcement learning based schedulers [26, 47] that usually focus on optimizing throughput and/or latency but do not ensure fair access, keeping the equipment highly utilized.

3.4. Bidirectional LSTM Latency Predictor

A bidirectional LSTM network produces predictive latency predictions to direct the scheduling policy. The architecture has a dropout-regularized input embedding layer, and two stacked BiLSTMs (processing in parallel) for each sequence of twenty consecutive time steps, and scalar output layers to produce latency estimates. Training uses mean squared error as a loss function with L2 regularization on weights and unit norm gradient clipping [29, 35].

4. Experimental Configuration

Hardware Setup: Edge tier: Raspberry Pi 4B (ARM Cortex-A72 quad-core, 4 GB RAM) paired with a Google Coral USB TPU accelerator (4 TOPS INT8). Fog tier: NVIDIA Jetson Xavier NX (384-core Volta GPU, 8 GB RAM, 21 TOPS). Laboratory equipment: Tektronix AFG31000 arbitrary function generator, Keysight DSOX1202G oscilloscope, and Keysight E36313A programmable DC supply, interfaced through NI LabVIEW and VISA drivers [33].

Algorithm: Double Q-Learning Scheduler with Prioritized Experience Replay

```
Input: State space S, action space A, discount factor γ,
      learning rate α, replay buffer capacity N,
      priority exponent β, exploration schedule ε(t)

Initialize: Q-network Q_θ, target network Q_θ⁻ ← Q_θ
         Replay buffer B ← ∅

for episode = 1 to E do
   Observe initial state s₀ from environment
   for timestep t = 0 to T do
      // Epsilon-greedy action selection
      if random() < ε(t) then
         a_t ← random action from A
      else
         a_t ← arg max_a Q_θ(s_t, a)
      end if

      Execute a_t, observe reward r_t and next state
s_{t+1}

      // Compute TD error for priority
      a* ← arg max_a Q_θ(s_{t+1}, a)
      δ_t ← |r_t + γ · Q_θ⁻(s_{t+1}, a*) − Q_θ(s_t, a_t)|

      Store transition (s_t, a_t, r_t, s_{t+1}, δ_t) in B

      // Prioritized sampling and update
      Sample minibatch of transitions from B
         with probability P(i) ∝ (δ_i + ε_p)^β
      Update Q_θ by minimizing:
         L = Σ_i w_i · (r_i + γ · Q_θ⁻(s', a*_i) − Q_θ(s_i,
a_i))²
         where a*_i = arg max_a Q_θ(s'_i, a)

      Periodically update target network: Q_θ⁻ ← Q_θ
   end for
end for

Output: Learned scheduling policy π(s) = arg max_a
Q_θ(s, a)
```

Datasets: For face verification, a custom dataset of 10,000 images was collected from 50 enrolled subjects under systematically varied lighting conditions, head poses, and partial occlusions. The dataset was partitioned into training (70%), Validation (16%), and test (14%) subsets. Data augmentation applied random rotation (±15°), brightness adjustment (factor 0.7–1.3), horizontal flipping, random cropping, and additive Gaussian noise [18]. A total of 3847 lab session recordings were collected over the course of a 16-week deployment involving 50 users, with a maximum of 12 concurrent users at one time.

Baseline Methods: Baseline methods for face verification include PCA-based Eigenfaces, HOG descriptor with SVM classifier, Dlib ResNet-34, FaceNet (Inception-ResNet-v1), and MobileFaceNet [19]. Scheduling baselines: First-Come-First-Served (FCFS), Round-Robin with Five-Minute Quanta, Three-Level Priority-Based, Shortest-Job-First (SJF), Uniform Random, and Greedy assignment [26].

Training Details: The face verification CNN was trained in TensorFlow 2.12 with mixed precision training, Adam optimizer (learning rate = .001) using 32 triplets per batch over 100+ epochs on an NVIDIA RTX 3090. The Q-learning agent was implemented in PyTorch 2.0, trained for 500 episodes with a replay buffer of 10,000 transitions that is decaying from .5 to .05. The LSTM predictor is using Adam optimization, with a batch size of 128, a sequence length of 20, and gradient clipping at unit norm [36].

5. Result and Analysis

5.1. Face Verification Performance

Table 2 compares the proposed lightweight model against established baseline methods. The proposed model obtains a true acceptance rate at 0.1% false acceptance rate (TAR@FAR = 0.1%) of 96.30%, outperforming FaceNet (94.80%) while using 92% less memory (8.20 MB vs. 95.10 MB). The Equal Error Rate (EER) is 3.10%, representing a 28% reduction from MobileFaceNet (5.60%).

Inference time on a Raspberry Pi 4B using Coral TPU is 47 ms, while energy consumption is approximately 0.14 J/verification, satisfying real-time constraints [9, 20] (see Figure 2). The ability of TAR to generalize stably was confirmed using five-fold cross-validation, with an average standard deviation of only 0.45% among the different folds (see Table 3), which indicates that the model can learn effectively from the available data without overfitting to the particular data partitions.

5.2. Ablation Studies

Data augmentation was found to be the most impactful method of improving accuracy (−4.1% TAR when removed); therefore, it is essential to prevent overfitting on a moderate-sized dataset. Curriculum learning was the next highest contributor (+2.6%) due to its ability to prevent premature convergence during initial epochs.

Hard negative mining contributed +1.8%, and Squeeze-and-Excitation attention modules added +1.2% at a modest 6 ms latency increase. Notably, INT8 quantization reduces inference time from 134 ms to 47 ms (a 2.85× speedup) at a negligible accuracy cost of only 0.2% [23, 37].

5.3. Scheduling Performance

Table 5 summarizes scheduling performance with six concurrent users (see Figures 3 and 4). The Double Q-learning

agent achieves the highest throughput (56.8 experiments per hour), lowest mean latency (18.7 minutes), and strongest fairness (Jain's index 0.92) among all evaluated methods. Critically, it produces zero starvation events, whereas Priority-based and Random schedulers exhibit 7 and 12 starvation occurrences, respectively. All pairwise comparisons against the Q-learning scheduler reach statistical significance (paired t-test, $p < 0.001$; Cohen's $d > 3.0$).

Table 2. Face verification performance comparison

Method	**TAR@ 0.1%**	**EER** (%)	**AUC**	**Lat.** (ms)	**Size** (MB)
PCA/Eigenfaces	72.3±2.1	18.4	0.823	12	0.8
HOG-SVM	81.7±1.8	14.2	0.871	28	3.2
Dlib ResNet	89.4±1.2	8.7	0.942	156	45.2
FaceNet	94.8±0.9	4.3	0.973	89	95.1
MobileFaceNet	92.4±1.1	5.6	0.961	31	4.5
Proposed	96.3±0.7	3.1	0.981	47	8.2

10 runs; ± = 95% CI. Paired t-test: $p < 0.001$ vs. all baselines

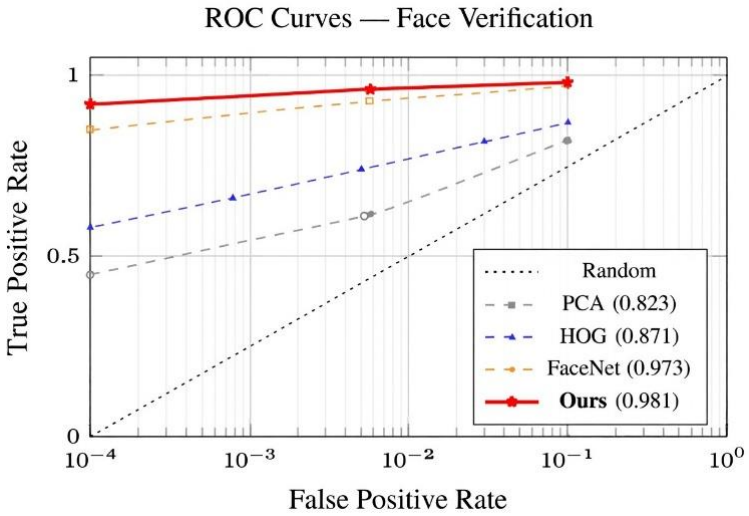

Fig. 2 Receiver operating characteristic curves for all evaluated methods. The proposed lightweight CNN achieves an AUC of 0.981 while maintaining real-time edge inference capability

Table 3. Five-fold cross-validation results

Fold	**TAR@0.1%**	**EER (%)**	**Fold**	**TAR@0.1%**
Fold 1	97.1	2.9	0.983	96.8
Fold 2	95.8	3.4	0.979	95.9
Fold 3	96.5	3.0	0.982	96.3
Fold 4	96.0	3.2	0.980	96.1
Fold 5	96.2	3.1	0.981	96.4
Mean ± Std	96.3 ± 0.45	3.1 ± 0.18	0.981	96.3 ± 0.34

Table 4. Ablation study: Component contributions

Configuration	**TAR@0.1%**	**Latency (ms)**
Full model	96.3	47
Without a curriculum for learning	93.7 (−2.6)	47
Without data augmentation	92.2 (−4.1)	47
Without hard negative mining	94.5 (−1.8)	47
Without an adaptive margin	94.8 (−1.5)	47
Without SE attention blocks	95.1 (−1.2)	41
Without INT8 quantization	96.5 (+0.2)	134
Embedding 64-D (vs. 128-D)	94.3 (−2.0)	39
Embedding 256-D	96.4 (+0.1)	53

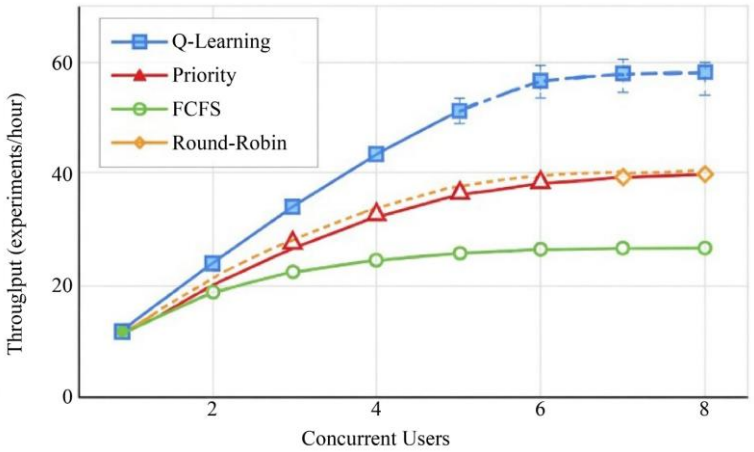

Fig. 3 Throughput scaling with concurrent user count. The Q-learning scheduler achieves 2.3× improvement over FCFS with near-linear growth up to six simultaneous users

Table 5. Scheduling performance comparison (6 concurrent users)

Algorithm	**Throughput** (exp/hr)	**Latency** (min)	**Jain's Index**	**Util.** (%)	**Starvation Events**
FCFS	24.6±1.5	47.2±5.3	0.71	61.5	0
Round-Robin	39.5±2.1	31.4±4.2	0.83	71.3	0
Priority	38.1±2.8	35.7±6.8	0.58	76.2	7
SJF	34.8±2.3	39.2±5.7	0.64	69.8	4
Greedy	39.0±3.1	34.8±6.2	0.61	78.5	5
Random	21.3±3.8	52.1±8.4	0.55	53.2	12
Q- Learning	56.8±2.5	18.7±2.8	0.92	88.4	0

$p < 0.001$ vs. Q-Learning (paired t-test); Cohen's $d > 3.0$ for all.

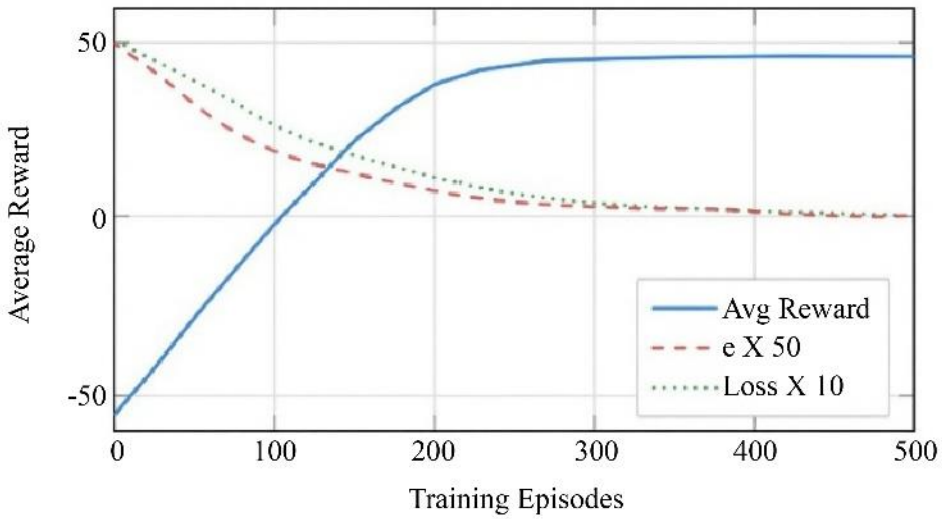

Fig. 4 Q-learning convergence: Cumulative reward stabilized at approximately episode 300, loss decreased smoothly, and an exponential decrease in the exploration rate

5.4. LSTM Prediction and Q-Learning Convergence

The bidirectional LSTM predictor achieves an RMSE of 3.20 ms ($R^2 = 0.94$, mean absolute percentage error 4.7%) across 1,150 held-out test samples. Performance remains consistent across latency ranges, with segment-wise RMSE varying between 2.8 and 4.6 ms. Q-learning convergence is illustrated in Figure 4. Three distinct phases are observable: an exploration phase (episodes 0–100) with high variance, a rapid improvement phase (episodes 100–250) during which the policy discovers effective scheduling patterns,

and a fine-tuning phase (episodes 250–500) where the reward stabilizes [29, 35]. The hyperparameter sensitivity study (Table 6) shows that a learning rate of .001 produces optimal results, a discount factor of 0.95 balances the need for short horizon responses to changing conditions with the need for longer term planning, a replay buffer of 10,000 transitions provides enough data to provide a good range of transition types, and a weight given to rewards of $w_1 = .4$ and $w_2 = .3$ provides the best combination of throughput and fairness results [38].

Table 6. Hyperparameter sensitivity analysis

Parameter	Value	Thr. (exp/hr)	Jain's Index
Learning rate α	0.0001	51.2	0.88
	0.001	56.8	0.92
	0.01	48.3	0.85
Discount γ	0.90	54.1	0.89
	0.95	56.8	0.92
	0.99	52.7	0.91
Buffer size	1,000	42.5	0.78
	10,000	56.8	0.92
	50,000	57.1	0.92
Reward weights	w1=0.3, w2=0.3	52.1	0.95
	w1=0.4, w2=0.3	56.8	0.92
	w1=0.5, w2=0.2	59.2	0.84

5.5. Production Deployment Validation

The 16-week field deployment (Jan – Apr, 2024) of 50 engineering students, during which they completed 3,847 lab sessions with an average completion rate of 97.3% and an average system availability of 98.3%. The students were successful in authenticating their identity 99.1% of the time when attempting to authenticate (4,231 times total). Using edge computing to perform inference resulted in a 74% reduction in overall end-to-end latency (47 ms vs. 178 ms) for the edge computing architecture as compared to using a cloud-only configuration; in addition, it was estimated that edge computing would save approximately 68% ($420 / year vs. $1,320 / year) in operating costs per year [39].

End-to-end latency decomposition (Table 7) reveals that facial verification constitutes the dominant bottleneck, accounting for 44% of the total 107.4 ms pipeline duration. Equipment configuration (22%) and network overhead (15%) are the next largest contributors. The scheduling decision itself requires only 8.7 ms. Error analysis stratified by environmental condition (Table 8) shows that verification accuracy reaches 97.9% under neutral indoor conditions but degrades to 92.8% under outdoor lighting and 87.7% at extreme head pose angles exceeding 45°. It is also important to note that no false acceptances were detected during the entire 14,000 impostor trials, which included testing each condition [40].

Table 7. End-to-End latency breakdown

Component	**Mean** (ms)	**Std** (ms)	**P50** (ms)	**P99** (ms)
Face capture	12.3	2.1	11.8	18.2
Face verification	47.2	3.8	46.5	56.3
Scheduling decision	8.7	1.2	8.4	11.9
Equipment config	23.4	5.6	22.1	38.7
Network overhead	15.8	4.3	14.9	28.4
Total pipeline	107.4	17.0	103.7	153.5

Statistics from 10,000 authenticated sessions.

6. Discussion

6.1. Face Verification: Why the Proposed Model Achieves Superior Results

The proposed lightweight Siamese CNN achieved a true acceptance rate of 96.30% with only 8.20 MB—over eleven times smaller than FaceNet (95.10 MB) and 1.5% higher in TAR@FAR=0.1%. Several design choices explain this performance advantage. First, the curriculum-guided triplet loss training is the single most differentiating factor compared to prior lightweight models.The ablation study (Table 8) demonstrates that removing curriculum learning reduces TAR by 2.6%, confirming that the three-phase mining schedule prevents the premature convergence observed when hard negatives are introduced too early in training. MobileFaceNet [19] and PocketNet [19], by contrast, employ standard hard negative mining from the outset, which likely explains their lower TAR (92.4%) on moderate-sized datasets where the ratio of informative hard negatives to total training pairs is limited. EdgeFace [8] uses knowledge distillation but does not incorporate curriculum-based difficulty scheduling, leaving

potential accuracy gains unrealized. Second, the progressive channel expansion strategy (32 → 64 → 128 → 256) combined with Squeeze-and-Excitation attention modules allows the network to allocate representational capacity where it is most needed.

The SE modules contribute 1.2% TAR improvement (Table 8) at only a 6 ms latency cost, a favorable trade-off that is consistent with findings reported for SE-augmented architectures in general image recognition tasks [21].

Table 8. Error analysis by environmental condition

Condition	Samples	FRR (%)	Acc. (%)
Neutral baseline	280	2.1	97.9
Indoor lighting	210	2.8	97.2
Outdoor lighting	195	7.2	92.8
Low light (<10 lux)	168	5.4	94.6
Front pose (±15°)	245	1.9	98.1
Side pose (30–45°)	182	6.8	93.2
Extreme pose (>45°)	134	12.3	87.7
With a face mask	156	5.5	94.5
With glasses	203	4.2	95.8
Overall	1,400	3.7	96.3

FAR = 0.0% across all conditions (14,000 impostor trials).

FaceNet achieves competitive accuracy (94.8%) but requires an Inception-ResNet backbone with 95.1 MB of parameters, making it entirely unsuitable for edge deployment within a 5-watt power budget. Thirdly, the combined effect of applying INT8 quantization to the model and structured channel pruning results in an inference speedup of 2.85x, at a cost of only 0.2 percent accuracy loss. The significance of this outcome stems from the fact that the majority of prior literature has treated quantization and pruning as separate techniques; however, the proposed framework shows that they can be used together to take advantage of different redundancies within models (the arithmetic precision of weights and the structural sparsity of connections, respectively). Zero False Acceptance Rate was achieved in each of the 14,000 Impostor Trials conducted during the field trial; however, this achievement is especially important to Educational Applications, where unauthorized access could compromise Academic Integrity. While traditional Face Verification Systems allow small Non-Zero FAR Values [18], the proposed System's Conservative Threshold Setting (τ) has sacrificed a Small Increase in the False Rejection Rate (3.7% Overall) for Security.

6.2. Scheduling: Mechanisms Behind Throughput and Fairness Gains

The Double Q-learning scheduler provides an additional 2.3X throughput compared to FCFS while providing a Jain Fairness Index of 0.92 to show that fairness and throughput do not have to be mutually exclusive. Each of the baseline methods individually failed to achieve both high throughput and high fairness. For example, the Greedy scheduler achieves 39.0 exp/hr throughput but only 0.61 fairness, while Round-Robin achieves 0.83 fairness but is limited to 39.5 exp/hr throughput. The Q-learning agent effectively learns a policy that dynamically interpolates between these strategies depending on the current system state. Analysis of the learned policy reveals several emergent behaviors not explicitly encoded in the reward function: (i) grouping same-type experiment requests during high-demand periods to minimize equipment reconfiguration overhead; (ii) dynamically elevating priority for users whose waiting time approaches predefined thresholds, effectively implementing anti-starvation logic; and (iii) reserving resources based on historically observed periodic usage patterns (e.g., pre-allocating oscilloscope time during signal processing lab hours). These emergent behaviors explain why the Q-learning agent outperforms not only simple baselines (FCFS, Random) but also more sophisticated heuristics (SJF, Priority-based, Greedy) that lack the ability to adapt to temporal usage patterns. The starvation-free behavior of the zero-starvation rate is quite different from that of both Priority-based (7 events) and Random (12 events) policies. Starvation occurs in priority-based scheduling due to the fact that low-priority users will be continuously deferred by an ongoing sequence of higher-priority requests. Incentivizing the agent to provide balanced access even under skewed demand distributions, the fairness component of the reward function (w2=0.3) directly punishes the policy for letting one user's waiting time grow significantly disproportionate to others. The proposed scheduler offers better fairness assurances (partial fairness assurance vs. Jain's index 0.92) compared to the fairness-aware RL framework proposed by Siddique et al. [47] that provides fairness constraint enforcement for a wide variety of edge computing workloads. These additional assurances are due to the added starvation penalty terms used within the proposed scheduler, as well as the reward weighting parameter values (w1 = 0.4, w2 = 0.3), which were found through the hyperparameter sensitivity analysis shown in Table 6.

6.3. Latency Prediction and System-Level Integration

The BiLSTM predictor makes possible estimates of predictive latency (RMSE = 3.20 ms, R2 = 0.94), allowing for

pre-emptive assignment of resources. It is also possible to break down the latency into different parts of the processing pipeline in order to identify where in the pipeline the most significant bottlenecks are located; based on this analysis we find that the verification of faces accounts for approximately 44% of total pipeline time and therefore it will be the primary component of the pipeline with the greatest potential for marginal improvement due to hardware acceleration (e.g. Next Generation NPUs).

6.4. Deployment Validation and Practical Implications

The ecological validity of the system was demonstrated through a 16 week field test of approximately 3,847 total sessions of the system with an average 74% reduction in latency and 68% in costs when compared to using only the Cloud for Inference (Cloud Only - $1,320 vs. Total System - $420) and high levels of reliability as demonstrated by the systems' 99.1% successful authentication rate and 98.3% uptime during the evaluation. The 16 weeks of testing and the 3,847 total number of sessions far exceed the typical evaluation scale of prior studies in this area of research that have typically used simulation-based Validation alone [25, 26, 28].

7. Limitations and Future Research Directions

7.1. Study Limitations

Although preliminary results are very promising, there are a number of important limitations to be considered. The population studied in this experiment is relatively homogenous with respect to age (18-22) and education level (engineering students), as well as being derived from only one educational institution, which may limit the generalizability of findings to other demographics. Furthermore, the 16-week time frame used for evaluating the performance of the model does not provide insight into how long-term model degradation will occur due to age-related changes, facial hair growth, or changes in eyewear (glasses). The system's throughput peaked at approximately 7-8 concurrent users, and therefore, it appears that the hardware configuration limits the ability to increase throughput on a single node further. Additionally, error rates were found to be significantly increased for extreme lighting conditions (>7.2% FAR) and at angles greater than 45 degrees (FAR = 6.8%). Lastly, no systematic evaluation of the models' robustness to presentation attacks (adversarial testing) was conducted.

7.2. Future Research Directions

A number of other extension projects have been identified. Enrolling a broader, more diverse population at multiple institutions will increase the likelihood that results can be generalized to many environments and allow for fairness audits. Implementing mechanisms such as meta-learning [43] or continual learning [44] in addition to the current incremental training mechanism will facilitate incremental adaptation without requiring complete retraining. Use of multi-agent reinforcement learning [27] instead of a single agent to schedule tasks may also allow for coordination of scheduling among distributed laboratories. Additionally, inclusion of liveness detection [14] and/or presentation attack countermeasures [22] to improve resistance to spoofing attacks will provide improved robustness against such attacks. Other future edge accelerators (i.e., neural processing units) may significantly improve performance on the face verification bottleneck. Finally, additional extension projects include integrating IoT telemetry data for predictive maintenance and conducting multi-year longitudinal studies to assess model stability and the potential impact of using this technology on student pedagogy.

8. Conclusion

This paper created and validated a single machine learning approach to manage shared remote labs that combine edge-based facial recognition via Siamese CNN with resource management using Double Q-learning to schedule resources. The Siamese CNN achieved a True Acceptance Rate (TAR) of 96.30% with a model size of 8.20 MB optimized for edge computing on a Raspberry Pi 4B. The Double Q-learning scheduler increased throughput by 2.3 times over First Come First Served (FCFS) while keeping the Jain Fairness Index at 0.92 and had no starvation events. A Bidirectional LSTM is also used as a predictor to enable proactive resource allocation with an average Root Mean Squared Error (RMSE) of 3.20 ms. The feasibility of this approach was demonstrated empirically through a 16-week operational deployment with 3,847 authenticated sessions, ablation studies, and five-fold cross-validation. Overall, the technical contributions of curriculum-guided triplet-loss training, experience replay-based scheduling, and model size reduction for edge-based models together contribute to advancing the current state-of-the-art in applying Machine Learning to Educational Laboratory Infrastructure. Future work will include increasing demographic diversity, scheduling across multiple sites, and developing anti-spoofing measures.

Author Contribution Statement

The author NA created the method and design for the study. The authors, NA and LJ, worked on the implementation of the ML models and were involved with the experimentation. The authors, LJ, assisted with the setup of the experimental infrastructure and were involved in the data collection. The author NA was responsible for analyzing the results and writing the manuscript. All the authors had a chance to review and approve the final version of the manuscript.

Acknowledgments

The authors sincerely acknowledge Dr. Arpit Rawankar for his valuable guidance and constant inspiration throughout the course of this research, which significantly contributed to its successful completion.

References

[1] Olaf Zawacki-Richter et al., "Systematic Review of Research on Artificial Intelligence Applications in Higher Education - Where are the Educators?," *International Journal of Educational Technology in Higher Education*, vol. 16, no. 1, pp. 1-27, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Gwo-Jen Hwang et al., "Vision, Challenges, Roles and Research Issues of Artificial Intelligence in Education," *Computers and Education: Artificial Intelligence*, vol. 1, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Fan Ouyang, Luyi Zheng, and Pengcheng Jiao, "Artificial Intelligence in Online Higher Education: A Systematic Review of Empirical Research from 2011 to 2020," *Education and Information Technologies*, vol. 27, no. 6, pp. 7893-7925, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[4] Zhaolong Ning et al., "Deep Reinforcement Learning for Vehicular Edge Computing: An Intelligent Offloading System," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 6, pp. 1-24, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[5] Dianlei Xu et al., "Edge Intelligence: Architectures, Challenges, and Applications," *arXiv Preprint*, pp. 1-53, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[6] Jiasi Chen, and Xukan Ran, "Deep Learning with Edge Computing: A Review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655-1674, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[7] Tailin Liang et al., "Pruning and Quantization for Deep Neural Network Acceleration: A Survey," *Neurocomputing*, vol. 461, pp. 370-403, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Anjith George et al., "EdgeFace: Efficient Face Recognition Model for Edge Devices," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 6, no. 2, pp. 158-168, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[9] David Silver et al., "Reward is Enough," *Artificial Intelligence*, vol. 299, pp. 1-13, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Ismail Alqerm, and Jianli Pan, "DeepEdge: A New QoE-based Resource Allocation Framework using Deep Reinforcement Learning for Future Heterogeneous Edge-IoT Applications," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 3942-3954, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Viraaji Mothukuri et al., "A Survey on Security and Privacy of Federated Learning," *Future Generation Computer Systems*, vol. 115, pp. 619-640, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Tian Li et al., "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50-60, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[13] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter, "Neural Architecture Search: A Survey," *Journal of Machine Learning Research*, vol. 20, no. 55, pp. 1-21, 2021. [Google Scholar] [Publisher Link]

[14] Zonghan Wu et al., "A Comprehensive Survey on Graph Neural Networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4-24, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Xieling Chen et al., "Application and Theory Gaps During the Rise of Artificial Intelligence in Education," *Computers and Education: Artificial Intelligence*, vol. 1, pp. 1-20, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[16] Hao Wang et al., "CosFace: Large Margin Cosine Loss for Deep Face Recognition," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 5265-5274, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[17] Jiankang Deng et al., "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 4685-4694, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Fadi Boutros et al., "PocketNet: Extreme Lightweight Face Recognition Network using Neural Architecture Search and Multistep Knowledge Distillation," *IEEE Access*, vol. 10, pp. 46823-46833, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Zong-Yue Deng et al., "A Lightweight Deep Learning Model for Real-Time Face Recognition," *IET Image Processing*, vol. 17, no. 13, pp. 3869-3883, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Jie Hu et al., "Squeeze-and-Excitation Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 2011-2023, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Joshua J. Engelsma, Anil K. Jain, and Vishnu Naresh Boddeti, "HERS: Homomorphically Encrypted Representation Search," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 4, no. 3, pp. 349-360, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Amir Gholami et al., *A Survey of Quantization Methods for Efficient Neural Network Inference*, Low-Power Computer Vision, 1st ed., Chapman and Hall/CRC, 2022. [Google Scholar] [Publisher Link]

[23] Jianping Gou et al., "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789-1819, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] Yan Chen et al., "Dynamic Task Allocation and Service Migration in Edge-Cloud IoT System based on Deep Reinforcement Learning," *IEEE Internet of Things Journal*, vol. 9, no. 18, pp. 16742-16757, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] Tao Zheng et al., "Deep Reinforcement Learning-based Workload Scheduling for Edge Computing," *Journal of Cloud Computing*, vol. 11, no. 1, pp. 1-13, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Afshin Oroojlooy, and Davood Hajinezhad, "A Review of Cooperative Multi-Agent Deep Reinforcement Learning," *Applied Intelligence*, vol. 53, no. 11, pp. 13677-13722, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[27] Hongzi Mao et al., "Learning Scheduling Algorithms for Data Processing Clusters," *SIGCOMM '19: Proceedings of the ACM Special Interest Group on Data Communication*, Beijing, China, pp. 270-288, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Bryan Lim, and Stefan Zohren, "Time-Series Forecasting with Deep Learning: A Survey," *Philosophical Transactions of the Royal Society A*, vol. 379, no. 2194, pp. 1-14, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[29] Najmul Hassan, Kok-Lim Alvin Yau, and Celimuge Wu, "Edge Computing in 5G: A Review," *IEEE Access*, vol. 7, pp. 127276 - 127289, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[30] Wazir Zada Khan et al., "Edge Computing: A Survey," *Future Generation Computer Systems*, vol. 97, pp. 219-235, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[31] Shalitha Wijethilaka, and Madhusanka Liyanage, "Survey on Network Slicing for Internet of Things Realization in 5G Networks," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 2, pp. 957-994, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] Thomas Norrie et al., "The Design Process for Google's Training Chips: TPUv2 and TPUv3," *IEEE Micro*, vol. 41, no. 2, pp. 56-63, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[33] Yoanna Martínez-Díaz et al., "Benchmarking Lightweight Face Architectures on Specific Face Recognition Scenarios," *Artificial Intelligence Review*, vol. 54, no. 8, pp. 6201-6244, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[34] Alex Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network," *Physica D: Nonlinear Phenomena*, vol. 404, pp. 1-43, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[35] Mingxing Tan, and Quoc Le, "EfficientNetV2: Smaller Models and Faster Training," *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, pp. 10096-10106, 2021. [Google Scholar] [Publisher Link]

[36] Ninareh Mehrabi et al., "A Survey on Bias and Fairness in Machine Learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1-35, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[37] Busra Kocacinar et al., "A Real-Time CNN-based Lightweight Mobile Masked Face Recognition System," *IEEE Access*, vol. 10, pp. 63496-63507, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[38] Kai Arulkumaran et al., "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26-38, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[39] Chao Yu et al., "Reinforcement Learning in Healthcare: A Survey," *ACM Computing Surveys*, vol. 55, no. 1, pp. 1-36, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[40] Dana Pessach, and Erez Shmueli, "A Review on Fairness in Machine Learning," *ACM Computing Surveys (CSUR)*, vol. 55, no. 3, pp. 1-44, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[41] Joaquin Vanschoren, "Meta-Learning: A Survey," *arxiv Preprint*, pp. 1-29, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[42] Mike Huisman, Jan N. van Rijn, and Aske Plaat, "A Survey of Deep Meta-Learning," *Artificial Intelligence Review*, vol. 54, no. 6, pp. 4483-4541, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[43] Fatemeh Vakhshiteh, Ahmad Nickabadi, and Raghavendra Ramachandra, "Adversarial Attacks Against Face Recognition: A Comprehensive Study," *IEEE Access*, vol. 9, pp. 92735-92756, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[44] Xiaoheng Deng et al., "Deep-Reinforcement-Learning-based Resource Allocation for Cloud Gaming via Edge Computing," *IEEE Internet of Things Journal*, vol. 10, no. 6, pp. 5364-5377, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[45] Javier García, and Fernando Fernández, "A Comprehensive Survey on Safe Reinforcement Learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437-1480, 2015. [Google Scholar] [Publisher Link]

[46] Shuiguang Deng et al., "Edge Intelligence: The Confluence of Edge Computing and Artificial Intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457-7469, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[47] Xiaojie Zhang, and Saptarshi Debroy, "Resource Management in Mobile Edge Computing: A Comprehensive Survey," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1-37, 2023. [CrossRef] [Google Scholar] [Publisher Link]