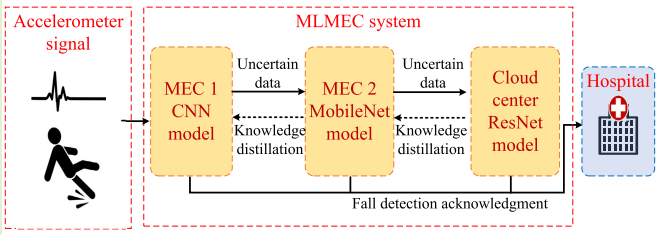
MECKD: Deep Learning-Based Fall Detection in Multilayer Mobile Edge Computing With Knowledge Distillation

Wei-Lung Mao, Chun-Chi Wang, Po-Heng Chou, *Member, IEEE*, Kai-Chun Liu, *Member, IEEE*, and Yu Tsao, *Senior Member, IEEE*

***Abstract*—In recent decades, the rising aging population has increased the importance of fall detection (FD) systems as assistive technology. To enhance accuracy, deep learning (DL) techniques are widely applied to these systems. A typical FD system involves placing edge devices (EDs) on individuals to collect real-time data, which is transmitted to a cloud center (CC) or processed by the ED itself. However, this architecture faces challenges such as limited ED model size and data transmission latency to the CC. Mobile edge computing (MEC) has been explored to overcome these challenges, which allows computations at MEC servers deployed between EDs and the CC. A multilayer mobile edge computing (MLMEC) framework is proposed to manage the tradeoff between accuracy and latency. MLMEC divides the architecture into multiple stations, each equipped with a neural network model. If the front-end equipment lacks deterministic detection capability, data are transmitted to a station with robust back-end computing for detection. The knowledge distillation (KD) approach is employed to improve front-end detection accuracy. The KD approach allows high-power back-end stations to provide additional learning experiences, enhancing precision and reducing latency and processing load. Simulation results demonstrate improved accuracy with the KD approach (11.65% average improvement for the SisFall dataset and 2.78% for the FallAllD dataset). The proposed MLMEC with the KD approach outperforms without the KD approach regarding data latency rate by 54.15% for the FallAllD dataset and 46.67% for the SisFall dataset. In summary, the MLMEC FD system exhibits improved accuracy and reduced latency.**



***Index Terms*— Accelerometer, deep learning (DL), fall detection (FD), knowledge distillation (KD), multilayer mobile edge computing (MLMEC).**

I. Introduction

FALL is the major cause of critical injury and/or death for the elderly of the world. According to the World Population Prospects 2022 [1], the number of people aged 65 and above was 80 million (10% of the total population of the world). According to research reports, approximately half of them were unable to get back up without assistance due to fall-related injuries or a lack of physical fitness and strength, and more than half of those injured have a high risk of death within six months of falling [2], [3]. In the past two decades, wearable-based fall detection (FD) systems have been developed as an assistive technology [4], [5], [6], [7]. The primary goal of FD systems is to automatically detect critical fall events and immediately alert medical professionals or caregivers. In recent years, various sensor devices have been used in FD system research, including smartphones [6], accelerometers [4], [7], radio-frequency identification (RFID) [8], cameras [9], and pressure sensors [10]. Among these technologies, FD systems that use accelerometers have many advantages, such as high sampling rates, low cost, high efficiency, and portability [11]. For example, Moulik and Majumdar [4] proposed a fuzzy-based FD inference system to fuse the data of triaxial accelerometers from multiple sensors for accuracy improvement of FD.

Since a large amount of data from these sensors, machine learning (ML) and deep learning (DL) technologies have

Received 2 August 2024; accepted 1 September 2024. Date of publication 13 September 2024; date of current version 13 December 2024. This work was supported in part by Academia Sinica under Grant 235g Postdoctoral Scholar Program and in part by the National Science and Technology Council (NSTC) of Taiwan under Grant 113-2926-I-001-502-G. The associate editor coordinating the review of this article and approving it for publication was Dr. Zhenghua Chen. *(Corresponding author: Po-Heng Chou.)*

Wei-Lung Mao and Chun-Chi Wang are with the Department of Electrical Engineering and the Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology, Douliu, Yunlin 64002, Taiwan (e-mail: wlmao@yuntech.edu.tw; d11010202@gemail.yuntech.edu.tw).

Po-Heng Chou, Kai-Chun Liu, and Yu Tsao are with the Research Center for Information Technology Innovation (CITI), Academia Sinica, Taipei 11529, Taiwan (e-mail: d00942015@ntu.edu.tw; t22302856@gmail.com; yu.tsao@citi.sinica.edu.tw).

Digital Object Identifier 10.1109/JSEN.2024.3456577

been widely applied to FD problems [7], [12], [13], [14], [15], [16], [17], [18], [19]. Choudhury and Soni [12] reviewed and discussed the latest state-of-the-art (SOTA) human activity recognition (HAR) models with different applications, including ML, DL, and fuzzy algorithms. FD is regarded as a classified problem also known as HAR, which contains two possible outcomes: falls and activities of daily living (ADLs). ADLs may exhibit peak data similar to falls. Therefore, it may lead to a false alarm (the true is ADL but the detection is fall) or miss detection (the true is fall but the detection is ADL), resulting in a waste of medical resources and a missed opportunity to treat injuries, respectively. Thus, the accuracy performance of FD is a critical issue. The main factors that impact the accuracy performance of the DL model include the type of used DL model, the quality of the dataset, data preprocessing, and the computational capability of the sensors or back-end computing. In general, the computational capability of hardware is the key factor in training the DL model and supporting the computational cost for performance optimization [20]. The cloud center (CC) has powerful computation capability and power supply to sufficient the requirement of DL technologies for solving FD problems with high accuracy [8], [20], [21].

Most of the HAR studies only focused on the accuracy measurements. However, the main disadvantage of relying significantly on the CC leads to high costs of transmission in high latency, network bandwidth, and privacy concerns [12], [20]. Choudhury and Soni [12] indicated real-time HAR system development is costly and lacks exploration because of the heavy bandwidth usage and real-time cloud or server architecture requirements. The response latency of FD is a critical measurement to be considered in a real-time HAR system. Therefore, Choudhury and Soni [13] proposed an adaptive batch size-based model and combined hybrid convolutional neural networks (CNNs) and long short-term memory (LSTM) as CNN-LSTM framework for a smartphone sensor-based HAR system to improve recognition accuracy and computational efficiency. Without data preprocessing and data augmentation, Choudhury and Soni [14] also proposed an efficient and lightweight CNN-LSTM model for enhanced ADL classification on raw data of sensors in an uncontrolled environment. However, these HAR works [13], [14] only used the computational resource of edge devices (EDs) and neglected the latency between the ED and the mobile edge computing (MEC) (e.g., access point) or CC. Furthermore, several works [22], [23], [24], [25], [26] only utilized the resources of the MEC that directly communicate with the EDs and make the response latency shorter than the CC computing. Nevertheless, the computational capability of MEC is weaker than CC in general. Therefore, the tradeoff between accuracy and response latency needs to be considered to design the overall EDs–MECs–CC architecture is an essential issue.

To address the issues above, we adopt the multilayer mobile edge computing (MLMEC) framework that was proposed in [27] to trade off the accuracy and latency of FD systems. Fig. 1 illustrates typical two-tier and the proposed MLMEC FD frameworks. In Fig. 1, a typical two-tier framework only relies on the computational capability of the ED or CC for FD and neglects the computational resources of MECs between ED and CC. In contrast, the MLMEC framework is proposed to exploit the computing capabilities of the MECs in [27] and [28]. Therefore, the proposed MLMEC framework for the FD system utilizes the computational resources of MECs between ED and CC and strikes a balance between detection performance and computational complexity. The authors investigated the performance analysis of MLMEC systems in the 6G networks in [28] to enhance performance and efficiency, reduce task execution latency, and optimize resource utilization. In addition, there are typically MECs (e.g., small data centers) with varying computational capabilities in the MLMEC framework. To bridge the gap in performance between different computational capabilities of MECs, we use knowledge distillation (KD) [29], [30], [31] to extract knowledge and enhance the accuracy performance of lighter sized DL models (student) from relatively more powerful pretrained DL models (teacher). KD is a common technique for model compression, aiming to reduce model complexity while maintaining accuracy [32]. KD compresses the knowledge of a larger and computationally expensive neural model (teacher) to a smaller computationally efficient neural model (student). The idea of KD is to train the student model, on a transfer set with soft targets provided by the teacher model. The main contributions of this work are as follows.

1) First, we propose an MLMEC framework that manages the accuracy–latency tradeoff by a threshold-based judgment for FD systems.
2) The KD approach is used to bridge the performance gap among MECs and CC for greatly improving accuracy and reducing latency.
3) The floating-point operations (FLOPs) computation is adopted to show how much computational cost can be saved by the proposed MLMEC framework compared to the typical two-tier cloud-device architecture.
4) Two public FD datasets (FallAllD and Sisfall) were employed to validate the effectiveness of the proposed model for achieving a higher detection accuracy.
5) The source codes (Python) are released to train the three adopted DL-based models (ResNet18, MobileNetV3, and CNN models) on GitHub.[1]

The remaining parts of this article are as follows. In Section II, the related works of a typical two-tier FD system are reviewed. In Section III, the proposed MLMEC framework for FD is introduced. In Section IV, the DL-based FD system with the KD approach is investigated. Section V investigates the metrics to evaluate the performance of the proposed MLMEC systems. Section VI shows the simulation results of dual-layer and triple-layer MLMEC systems with/without KD. Finally, Section VII contains the summary.

II. Related Works

Several works [33], [34], [35], [36] about DL-based FD with the KD approach are reviewed as follows. Most literatures [33], [34], [35], [36] only consider the single-stage KD approach. Hoa et al. [33] proposed a novel 3DKD model

[1]The source codes are available at https://github.com/BoneZhou/MECKD.

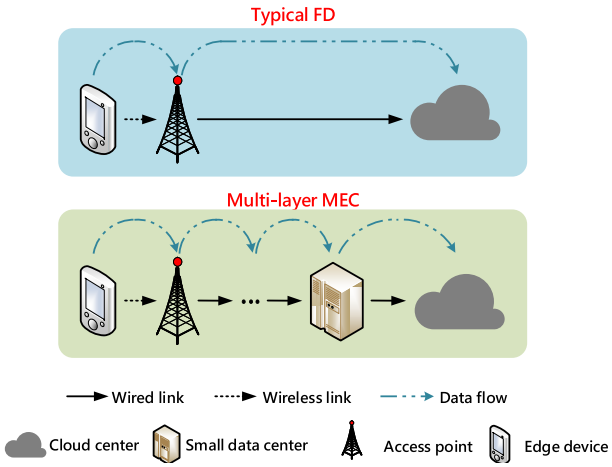


Fig. 1. Typical two-tier and the proposed MLMEC FD frameworks.

combining KD and attention mechanisms to contribute more precise deployment models capable of operating efficiently on weak devices in computer vision tasks for FD. Ni et al. [34] proposed a multimodal approach in a vision sensor-to-wearable sensor with KD (VSKD) framework and designed a novel KD loss function to mitigate the modality variations between the vision and the sensor domain. It not only reduces the computational demands on wearable devices but also develops a DL-based model that closely matches the performance of the computationally expensive multimodal approach. Rashid and Mohsenin [35] proposed a deep neural network (DNN) model with the KD and low bit-width quantization to fit models within lower memory hierarchy levels, reducing latency and enhancing energy efficiency on resource-constrained EDs. Quan et al. [36] proposed a novel semantic-aware multimodal transformer fusion decoupled KD (SMTDKD) to guide video data recognition through the information interaction among different wearable sensor data and visual sensor data.

The most related SOTA work is our previous study [37], which proposed a novel preimpact FD via CNN-vision transformer (ViT) KD, namely, PreFallKD, to strike a balance between detection performance and computational complexity. The CNN-ViT trains a CNN student model using a ViT [38] teacher model to enhance the performance of the CNN student model through the KD approach. The experimental results in [37] show that the latency of CNN is much less than other SOTA models, such as CNN-LSTM and ViT models. However, the training cost of the ViT model requires three types of sensor data including triaxial acceleration data, triaxial gyroscope data, and triaxial Euler angle data from KFall dataset [39] for convergence. In contrast, the training costs of ResNet, MobileNet, and CNN models in the proposed MLMEC require only a single type of sensor data from FallAllD [40] and SisFall [41] datasets. Therefore, it is difficult to compare the proposed MLMEC with KD and PreFallKD [37] under unequal conditions.

However, when there is a significant gap in performance between the teacher and student models, the effectiveness of the single-stage KD is limited [30], [42]. Saleknia and Ayatollahi [42] proposed an innovative multistage KD framework that takes advantage of a mid-size teacher assistant (TA) network to narrow the computational gap between student and teacher networks for computer vision task. Especially, the computational capabilities of edge servers are inconsistent in general heterogeneous networks [27], [28]. Thus, the gap in performance between the teacher and the student models is difficult to anticipate accurately. For more robust KD effectiveness, the multistage KD approach is proposed to bridge the performance gap between teacher and student models by the TA models [30].

To the best of authors' knowledge, this work is the first study to integrate the multilayer MEC framework and the multistage KD approach [30] for FD in wearable devices. The tradeoff between accuracy and latency is considered as the evaluation metric for FD systems. Accuracy improvement reduces the likelihood of false alarms and missed detections, and increases medical resource utilization and treatment opportunities. Even with high accuracy, the latency of the response reduction is also a crucial issue. The proposed MLMEC with KD is to enhance both metrics simultaneously. By the KD training from the upper layer MEC or CC, the accuracy of the lower layer MEC is improved in the MLMEC framework. When the accuracy of the lower layer is increased, the computation of the upper layers is reduced, resulting in the latency of response is also reduced. Therefore, the hyperparameters design of the neural model in the different layers of MEC is a crucial issue.

III. Proposed MLMEC Framework for FD

The proposed MLMEC framework with the KD approach for FD includes the bottom-layer EDs, the middle-layer MEC servers, and the top-layer CC server. All the components in the MLMEC are equipped with different sizes of DL models. Based on the computational capability of the component, a DL model with relative accuracy performance is equipped. In general, the top-layer CC server is equipped with the most computationally expensive DL model, the lowest-layer MEC is equipped with the lightest DL model, and the middle-layer MECs are equipped with the middle-size DL model. The MECs and CC server are connected via wired communications. The KD training process is adopted in the MLMEC framework to enhance the accuracy performance of the lighter sized DL models (student) by the relatively more powerful pretrained DL models (teacher). Hinton et al. [31] demonstrated that the KD training process is highly effective in transferring knowledge from either an ensemble or a large, extensively regularized model to a smaller, distilled model. It also showed substantial improvements obtained through distilling a student model of equivalent size, which is much easier for deployment in the MLMEC framework. However, the effectiveness of KD is limited, particularly when there is a significant gap in performance between the teacher and student models. Therefore, a teacher-assistant KD (TAKD) is proposed in [30], which involves incorporating intermediate models as TAs between the teacher and the student to bridge the performance gap. Mirzadeh et al. [30] showed theoretical analysis to prove the TAKD (teacher–TA–student model)

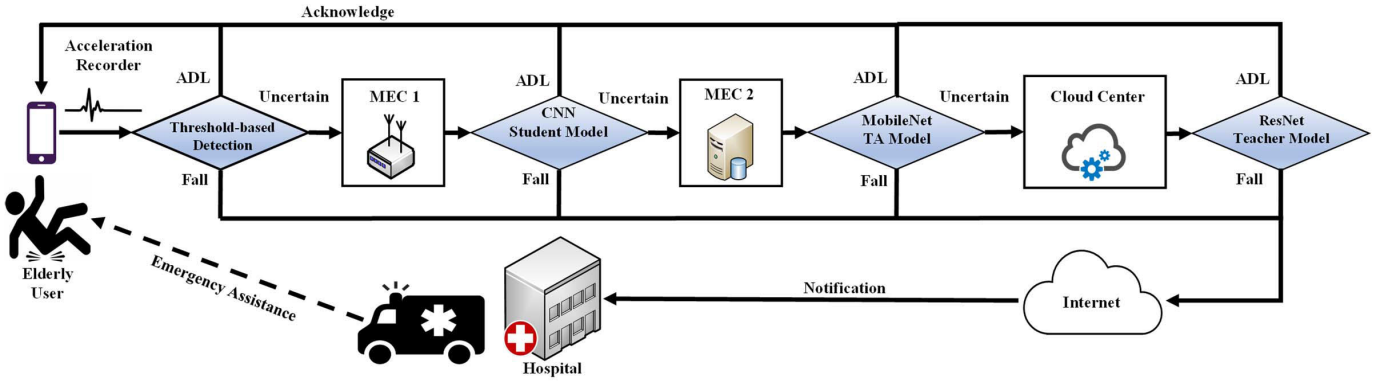


Fig. 2. Application scenario of the proposed MLMEC FD system.

outperforms the baseline KD (teacher–student model) and without KD (original student model).

Fig. 2 illustrates the operation flow and application scenario of the proposed FD system in the MLMEC system with KD. Our goal is to enhance the accuracy of FD and reduce the latency of notification. A smartphone equipped with the threshold-based FD [43], [44] is carried by the user. The threshold-based detection has low complexity properties and is easy to implement in wearable sensors. The details of the threshold-based FD are introduced in Section IV-B. The smartphone reads acceleration values from the accelerometer and assesses whether the detected value corresponds to a fall type, ADLs, or remains uncertain. In the case of a fall-type detection, an alert is sent to the hospital for emergency ambulance assistance via the Internet. For ADL detection, the system records only the intense signal in the acceleration recorder. In the case of an uncertain detection, the signal is forwarded to MEC 1 for a more accurate determination. If the detection of MEC 1 remains uncertain, the signal is then transmitted to MEC 2. If the detection of MEC 2 is still uncertain, the signal is routed to the upper layer MECs for a more precise determination than the lower layer MECs. If all the detection results of MECs are uncertain, then it is sent to the CC for the final decision. The proposed FD system architecture is deployed in the MLMEC with KD. In Fig. 2, MEC 1, MEC 2, and CC are equipped with DL models, specifically ResNet [45] (large), MobileNetV3 [46] (medium), and basic CNNs [47] (small), respectively.

By adopting the KD technique, the accuracy of the TA model in MEC 2 is enhanced by using the teacher model in CC, and the accuracy of the student model in MEC 1 is improved by the TA model in MEC 2. In addition, the TA models can be deployed to different MEC layers between the teacher model and the student model. It effectively bridges the performance gap among CC, MECs, and EDs. By TA-based KD, the amount of data labels is reduced, enhancing the usability of the FD model in real-world applications. In particular, the MLMEC framework is very suitable for deploying TA-based KD [30]. Compared to the traditional two-tier cloud-device architecture, the proposed MLMEC framework saves significant computational costs in our experiments.

IV. DL-Based FD With KD in MLMEC Framework

Our goal is to propose an MLMEC framework to trade off between the accuracy and latency of the FD system. Based on the MLMEC framework, the TAKD is introduced to improve the accuracy performance of the lower MEC layer from the upper MEC layer, while the latency of the FD system is improved.

The proposed MLMEC framework for the FD system is illustrated in Fig. 3. The approaches consist of four stages: 1) data preprocessing; 2) threshold-based feature extraction; 3) DL-based FD in the proposed dual-layer and triple-layer MEC system with the KD approach; and 4) threshold-based upward judgment for uncertain data. In the data preprocessing stage, the data are segmented using an impact-defined window with a fixed window size in Section IV-A. The classification is performed at the ED based on the maximum and minimum spatial momentum values (threshold) between ADLs and fall states, extracting the distinguished features of fall data. The advantage of this approach is the ability to provide real-time feedback and subsequent processing for clear fall situations. The details of threshold-based classification of ED for feature extraction are investigated in Section IV-B. Then, the uncertain detection results of the previous MEC layers are processed by using DL models in the subsequent MECs and CC for detection. The loss function design of the DL model in the proposed dual-layer and triple-layer MEC system with the KD approach is investigated in Section IV-C. As for the decision to transfer upward to the next MEC layer or not, we proposed the threshold-based judgment for MECs in Section IV-D.

The impact-defined window is a common method for data detection. The window size is determined by the timing of the impact point to obtain falling patterns (before, during, and after the fall). Fig. 4 illustrates the impact-defined window for the fall signal. Since a peak signal is generated at the timing of impact, the periods before and after the peak point are searched to locate the periods before and after the fall, which are used as a fixed-size window to search for the determined falling patterns.

A. Data Preprocessing

1) Impact-Defined Window: Assuming that the signal from the accelerometer is defined as $S = \{s_j | j = 1, 2, \ldots, n_s\}$,

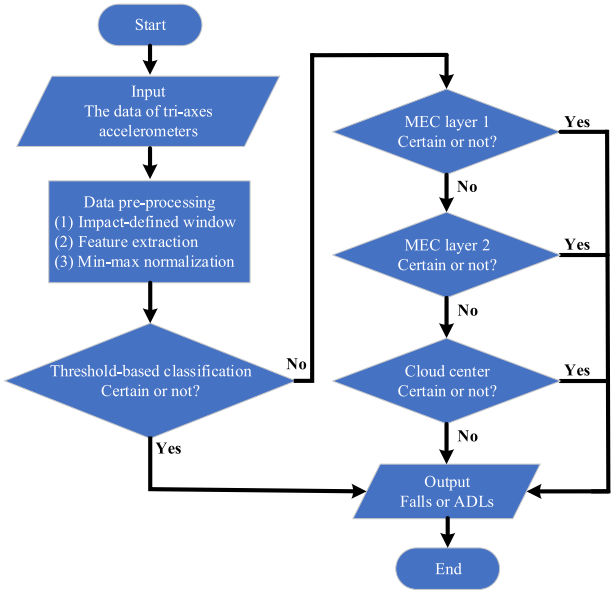


Fig. 3. Flowchart of the proposed MLMEC FD system.

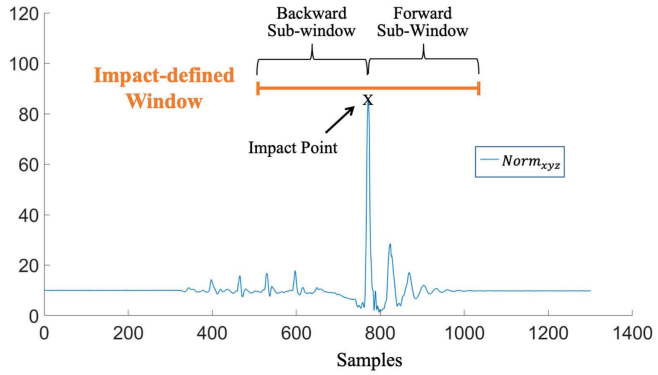


Fig. 4. Impact-defined window on a fall trial [44], [49], where the *X*-axis represents the sampling time (unit: s), and the *Y*-axis represents the acceleration momentum (unit: g).

where n_s is the total number of samples. Each sample point includes XYZ three-axis acceleration $s_j = \{a_{x_j}, a_{y_j}, a_{z_j}\}$. First, the sample with the maximum Norm_{xyz} value is identified as the impact point s_I, and the Norm_{xyz} formula for the jth sample s_j is shown as

$$\text{Norm}_{xyz}(s_j) = \sqrt{a_{x_j}^2 + a_{y_j}^2 + a_{z_j}^2}. \tag{1}$$

The boundary maximum value for identifying falls and the boundary minimum value for identifying nonfalls are extracted based on the Norm_{xyz} distribution of falls and ADLs. Based on the impact point s_I, the subwindows before and after the impact point are, respectively, defined as $W_f = \{S_{I+1}, \ldots, S_{I+\text{WS}_f-1}, S_{I+\text{WS}_f}\}$ and $W_b = \{S_{I-\text{WS}_b}, S_{I+\text{WS}_b+1}, \ldots, S_{I-1}\}$. The WS_f and WS_b are described the window sizes of W_f and W_b. The details of window determination were introduced in [43], [44], and [48].

Therefore, it is feasible to obtain the corresponding impact-defined window by searching for the peak signal.

TABLE I
LIST OF THE EXTRACTED FEATURES [44], [49]

No.	Time-domain statistical features
f_1-f_6	Mean of a_x, a_y, a_z, a_{norm}, a_{verti}, a_{hori}
f_7-f_{12}	Standard deviation of a_x, a_y, a_z, a_{norm}, a_{verti}, a_{hori}
f_{13}-f_{18}	Variance of a_x, a_y, a_z, a_{norm}, a_{verti}, a_{hori}
f_{19}-f_{24}	Maximum of a_x, a_y, a_z, a_{norm}, a_{verti}, a_{hori}
f_{25}-f_{30}	Minimum of a_x, a_y, a_z, a_{norm}, a_{verti}, a_{hori}
f_{31}-f_{36}	Range of a_x, a_y, a_z, a_{norm}, a_{verti}, a_{hori}
f_{37}-f_{42}	Kurtosis of a_x, a_y, a_z, a_{norm}, a_{verti}, a_{hori}
f_{43}-f_{48}	Skewness of a_x, a_y, a_z, a_{norm}, a_{verti}, a_{hori}
f_{49}-f_{51}	Correlation coefficient between each pair of a_x, a_y, a_z
f_{52}-f_{54}	Correlation coefficient between each pair of a_{norm}, a_{verti}, a_{hori}

2) Feature Extraction: For robust FD, there are two critical issues that need to be tackled as follows.

1) *Variability:* Falls may happen suddenly in various directions and forms in activity transitions of daily living. For example, the person who is standing or moving may frequently fall while exiting a chair or bed [50].
2) *Ambiguity:* Since some features of ADLs are similar to the features of falls, which may confuse the FD system, such as the person jumping or running in daily living leads to strong impact and energy, which may be misidentified as falls, and vice versa [11].

Based on [11], we adopt the common nine time-domain statistical features as the extracted features from the segments, including mean, standard deviation, variance, maximum, minimum, range, kurtosis, skewness, and correlation coefficient [51]. These nine features are extracted from the Euclidean norm of triaxial acceleration, the Euclidean norm of acceleration on the coronal plane, and the Euclidean norm of acceleration on a horizontal plane. Table I shows the 54 features used in this study. These nine features are widely used in FD [6] and HAR [52].

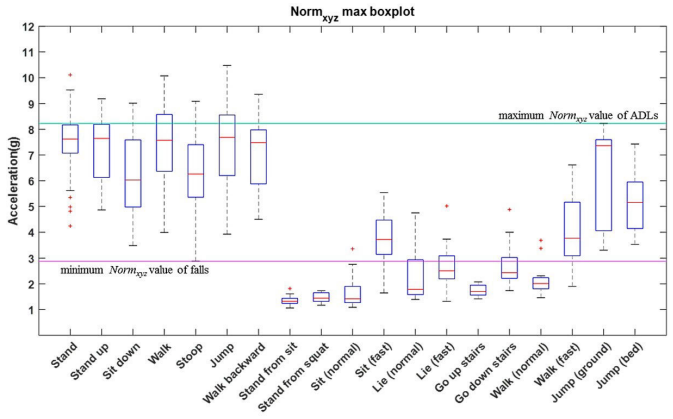
3) Min–Max Normalization: When the impact-defined window is determined, the data are processed using min–max normalization to reduce the scaling effects during the training phase. By applying the min–max normalization, each element of the data sequence $S = \{s_i | i = 1, 2, \ldots, I\}$ is normalized to a range between 0 and 1, which is shown as

$$S_i^{\text{nom}} = \frac{s_i - s_{\min}}{s_{\max} - s_{\min}} \tag{2}$$

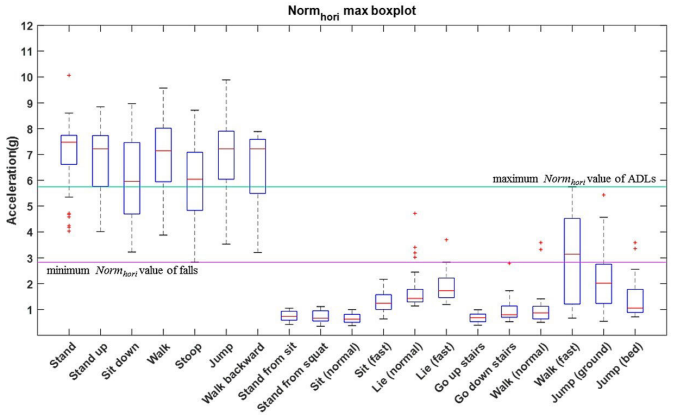
where $s_{\max}$ and $s_{\min}$ are the maximum and minimum values of S, respectively. In addition, we have compared min–max normalization and z-score standardization (standard scalar standardization) in our experiments for FD.

B. Threshold-Based Classification for Feature Extraction

In this study, we categorized all human actions into ADLs and falls. The features of falls and ADLs are extracted by the threshold-based classification. An initial threshold-based detection using maximum and minimum spatial momentum values is employed. The maximum and minimum spatial momentum values are determined by two kinds of data features Norm_{xyz} and $\text{Norm}_{\text{hori}}$ that are extracted from each data frame. Norm_{xyz} is calculated by (1) and characterizes the spatial variation of acceleration during the falling interval. $\text{Norm}_{\text{hori}}$



(a)



(b)

Fig. 5. Illustration of the Norm_{xyz} and $\text{Norm}_{\text{hori}}$ distribution of falls and ADLs [43], [44]. (a) Boxplots of maximum and minimum values of Norm_{xyz} for falls and ADLs. (b) Boxplots of maximum and minimum values of $\text{Norm}_{\text{hori}}$.

is defined as the Euclidean norm of acceleration for capturing the change of velocity in the horizontal plane of the body and calculated by

$$\text{Norm}_{\text{hori}}(S_j) = \sqrt{a_{y_j}^2 + a_{z_j}^2}. \tag{3}$$

The threshold is determined by the maximum and minimum values of Norm_{xyz} and $\text{Norm}_{\text{hori}}$ [43], [44], as shown in Fig. 5. Fig. 5 shows the Norm_{xyz} and $\text{Norm}_{\text{hori}}$ distributions of falls and ADLs, where the left seven activities are falls and the right twelve activities are ADLs, the green line is determined by the maximum value of Norm_{xyz} of ADLs, and the purple line is determined by the minimum value of $\text{Norm}_{\text{hori}}$ of falls.

First, a training set of a data frame is given by $F_{\text{train}} = \{(f_i^{\text{train}})|i = 1, 2, \ldots, N_{\text{train}}\}$, where N_{train} is the total number of the set F_{train}. Second, a testing set of data frame $F_{\text{test}} = \{(f_i^{\text{test}})|i = 1, 2, \ldots, N_{\text{test}}\}$, where N_{test} is the total number of the set F_{test}; there are two sets of maximum Norm_{xyz} and $\text{Norm}_{\text{hori}}$ corresponding to F_{train} that are presented as $V_{\text{train}} = \{(v_i^{\text{train}}|i = 1, 2, \ldots, N_{\text{train}})\}$ and $W_{\text{train}} = \{(w_i^{\text{train}}|i = 1, 2, \ldots, N_{\text{train}})\}$, respectively. Based on the distribution of V_{train} and W_{train}, two thresholds, $T_{\text{fall}}^{\text{Norm}_{xyz}}$ and $T_{\text{fall}}^{\text{Norm}_{\text{hori}}}$, for absolute falls are determined, where $T_{\text{fall}}^{\text{Norm}_{xyz}}$ and $T_{\text{fall}}^{\text{Norm}_{\text{hori}}}$ are the maximum Norm_{xyz} and $\text{Norm}_{\text{hori}}$ values of ADLs, respectively. Additionally, two thresholds, $T_{\text{ADL}}^{\text{Norm}_{xyz}}$ and $T_{\text{ADL}}^{\text{Norm}_{\text{hori}}}$, for the absolute ADL identification are determined, where $T_{\text{ADL}}^{\text{Norm}_{xyz}}$ and $T_{\text{ADL}}^{\text{Norm}_{\text{hori}}}$ are the minimum Norm_{xyz} and $\text{Norm}_{\text{hori}}$ values of falls, respectively. Similarly, the maximum and minimum values of Norm_{xyz} and $\text{Norm}_{\text{hori}}$ corresponding to F_{test} are presented as $V_{\text{test}} = \{(v_i^{\text{test}})|i = 1, 2, \ldots, N_{\text{test}}\}$ and $W_{\text{test}} = \{(w_i^{\text{test}})|i = 1, 2, \ldots, N_{\text{test}}\}$, respectively. Finally, the threshold-based classifier applied to the testing set is defined as follows:

$$\text{TC}(v_i^{\text{test}}, w_i^{\text{test}}) = \begin{cases} \text{Fall}, & \text{if } v_i^{\text{test}} > T_{\text{fall}}^{\text{Norm}_{xyz}} \text{ and } w_i^{\text{test}} > T_{\text{fall}}^{\text{Norm}_{\text{hori}}} \\ \text{ADL}, & \text{if } v_i^{\text{test}} < T_{\text{fall}}^{\text{Norm}_{xyz}} \text{ and } w_i^{\text{test}} < T_{\text{fall}}^{\text{Norm}_{\text{hori}}} \\ \text{Uncertain}, & \text{others.} \end{cases} \tag{4}$$

Therefore, three regions are distinguished: the absolute boundary of falls and ADLs, as well as the range of uncertain data that will be transferred to the MEC.

C. Loss Functions of Dual-Layer and Triple-Layer MECs

In this section, the loss functions of dual-layer and triple-layer MECs are analyzed. The architectures of dual-layer and triple-layer MECs with the KD approach are shown in Fig. 6.

As shown in Fig. 6(a), the teacher model is pretrained based on the dual-layer MECs architecture. Then, we use the pretrained teacher model and ground truth (label) to train the student model. By this KD approach, the weights of the teacher models are distilled into the student model. This KD approach improves the performance of the student model.

In general, supervised learning uses cross-entropy loss to measure the mismatch between the output of the student model and the ground truth (label) as follows:

$$\text{loss} = \mathcal{H}(g, \text{Student}) \tag{5}$$

where g represents the ground-truth value, Student represents the output of the student model, and $\mathcal{H}(p, q) = -\mathbb{E}_p(\log(q))$.

In the dual-layer MEC architecture (teacher–student model), we use the Kullback–Leibler (KL)-divergence loss to match the softened outputs of teacher and student models as follows:

$$\text{KL}\left(\frac{\text{Teacher}}{T}, \frac{\text{Student}}{T}\right) = \left(\frac{\text{Student}}{T}\right)\left[\log\left(\frac{\text{Student}}{T}\right) - \log\left(\frac{\text{Teacher}}{T}\right)\right] \tag{6}$$

where Teacher represents the predicted values of the teacher model, Student represents the predicted values of the student model, KL represents the KL-divergence between the output probabilities of the Teacher and Student, and T is referred to as additional control between the ground truth (label) and the soft targets from the output of the teacher model.

The loss function of the student model is as follows:

$$\text{loss}_{\text{dual}} = \lambda T^2 \times \text{KL}\left(\frac{\text{Teacher}}{T}, \frac{\text{Student}}{T}\right) + (1 - \lambda)\text{loss} \tag{7}$$

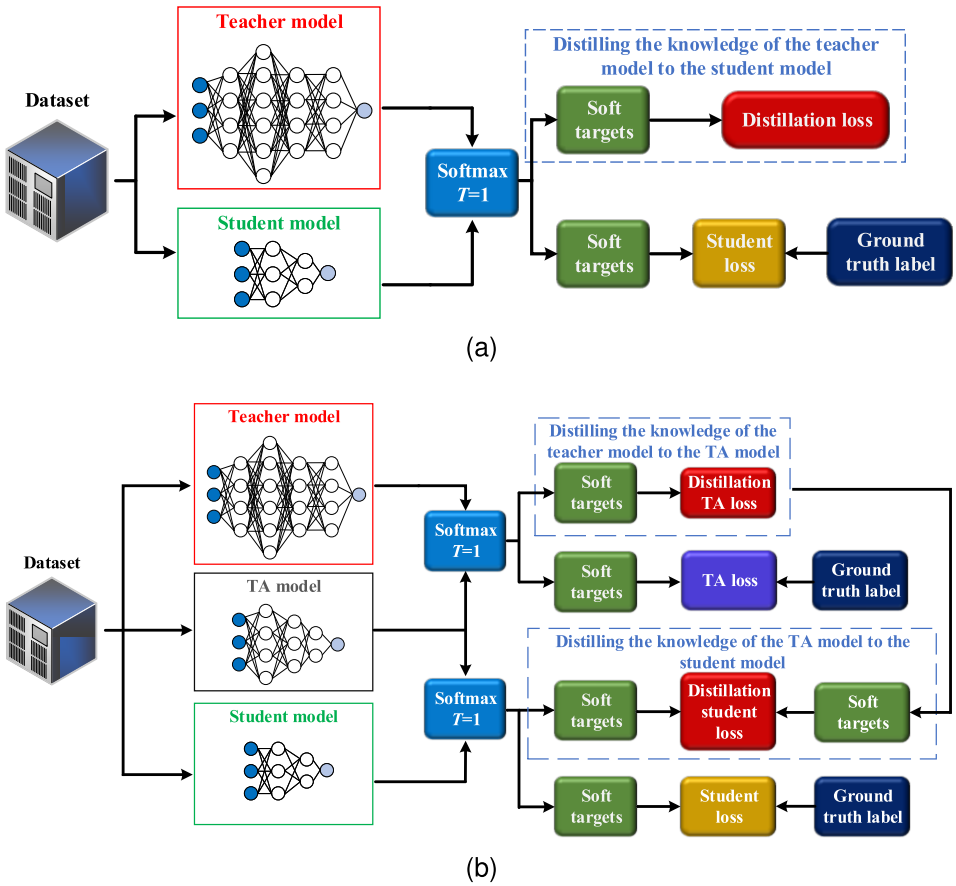


Fig. 6. Multilayer MECs with the KD approach. (a) Dual-layer. (b) Triple-layer.

where λ is a hyperparameter that adjusts the tradeoff between two loss functions.

According to [30], the TAKD is used to improve the performance of the student model of the teacher–TA–student model, when the performance gap between teacher and student is large. As shown in Fig. 6(b), the weights of the pretrained teacher model are extracted and distilled into the TA model. Then, the weights of the pretrained TA model are further distilled into the student model.

In the triple-layer MEC architecture, the KD loss function in the teacher–TA–student model is formulated. First, we formulate the first KL divergence between the teacher model and the TA model. Subsequently, the second KL divergence is calculated for the distilled TA model and the student model. Substitute the first KL divergence into the second KL divergence, as follows:

$$\mathrm{KL}\left(\frac{\text{Teacher}}{T},\frac{\text{TA}}{T},\frac{\text{Student}}{T}\right)=\left(\frac{\text{Student}}{T}\right)\times\left\{\log\left(\frac{\text{Student}}{T}\right)-\log\left(\frac{\text{TA}}{T}\right)\left[\log\left(\frac{\text{TA}}{T}\right)-\log\left(\frac{\text{Teacher}}{T}\right)\right]\right\} \tag{8}$$

where TA represents the predicted values of the TA model. Then, we substitute the obtained values into the following equation to calculate the loss function of the student model in the triple-layer MEC architecture as follows:

$$\text{loss}_{\text{tri}}=\lambda T^2\times\mathrm{KL}\left(\frac{\text{Teacher}}{T},\frac{\text{TA}}{T},\frac{\text{Student}}{T}\right)+(1-\lambda)\text{loss}. \tag{9}$$

D. Threshold-Based Judgment for Uncertain Data

For processing the uncertain data of the previous layer MEC, threshold-based judgment is proposed to identify between the absolute falls, ADL, and uncertain data. We define two threshold values, namely, maximum and minimum upward thresholds $T_{\text{fall}}^{\text{Max}}$ and $T_{\text{fall}}^{\text{Min}}$ are preset to define the range of the absolute falls, ADL, and uncertain data.

If the result of the judgment is still uncertain, the data are sent to the larger model in the upper layer MEC for more precise judgment. After the softmax function in the DL model of the nth layer MEC, the softened output is calculated by

$$v_n^{\text{test}}=\text{softmax}\left(\frac{\text{Layer}_n}{T}\right) \tag{10}$$

where Layer_n represents the predicted result of the nth layer MEC. Then, we input the softened output of the nth layer MEC to the threshold-based upward judgment that is defined as follows:

$$TQ(v_j^{\text{test}})=\begin{cases}\text{Fall}, & \text{if } v_n^{\text{test}}>T_{\text{fall}}^{\text{Max}}\\ \text{ADL}, & \text{if } v_n^{\text{test}}<T_{\text{fall}}^{\text{Min}}\\ \text{Uncertain}, & \text{if } T_{\text{fall}}^{\text{Max}}>v_n^{\text{test}}>T_{\text{fall}}^{\text{Min}}.\end{cases} \tag{11}$$

TABLE II
NOTATIONS OF LATENCY CALCULATION

Variable	Definition
M_{n-1}	The number of devices on each layer $n-1$
Q_{n-1}^{j}	The number of child nodes connected with the parent node j on layer $n-1$
λ_n^i	Data generation speed of ED i
β_n^i	The total volume of the processed data received by node i on layer n
s_n^i	The (equivalent) task division percentage at node i on layer n
θ_n^i	The computing capacity of node i on layer n
$\varphi_n^{j,i}$	The transmitting capacity of node i on layer n to its parent node j

V. PERFORMANCE METRICS

The performance metrics of the FD system are being carried out using the leave-one-out cross-validation method. One subject's data are used as the test dataset while the remaining subjects' data are used as the training dataset. This process is repeated k times, where k is the total number of subjects until all subjects have been used as the test dataset. The overall test data are then aggregated, and the average performance of the k folds is considered as the output to evaluate the performance of the FD system.

Applying the concept of a confusion matrix, four performance metrics are introduced, including accuracy (ACC), precision (PRE), recall (REC), and $F1$-score ($F1$), to measure the detection performance. The definitions of performance metrics are as follows:

$$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FP}+\mathrm{TN}+\mathrm{FN}} \tag{12}$$

$$\mathrm{PRE} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{13}$$

$$\mathrm{REC} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{14}$$

$$F1 = \frac{\mathrm{PRE}\times\mathrm{REC}}{\mathrm{PRE}+\mathrm{REC}} \tag{15}$$

where TP (true positive), TN (true negative), FP (false positive), and FN (false negative) represent the cases where the fall and ADL signals labeled as such are correctly recognized, the ADL signals labeled as a fall are misclassified, and the fall signals labeled as ADLs are misclassified, respectively.

In [27], the latency evaluation of each MEC of the MLMEC framework is calculated. The latency of the MLMEC architecture is defined as the total computation and transmission time from the ED to the CC. From ED to CC, each layer of the MLMEC framework processes incoming data from the previous layer. Therefore, the latency between ED and CC is expressed as

$$L_n = \sum_{j=1}^{M_{n-1}} \sum_{i\in Q_{n-1}^{j}} \left[\frac{s_n^i b_n^i}{\theta_n^i} + \frac{\rho s_n^i \lambda_n^i + \left(1-s_n^i\right)\lambda_n^i + \beta_n^i}{\varphi_n^{j,i}} \right] \tag{16}$$

where the notation definitions are provided in Table II.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

In our simulations, FallAllD [40] and SisFall [41] datasets are chosen to validate the proposed MLMEC FD system and the KD approach for managing the tradeoff between accuracy and latency. These two datasets are collected from real-life fall types and ADLs, and the sampling rates are 200 Hz. We set the window sizes of the subwindows before and after the impact point (i.e., WS_f and WS_b), respectively, determined to be 2 and 1.23 s for the FallAllD dataset, and 2 and 1.44 s for the SisFall dataset. Our previous study [44] has shown that such window size settings achieved optimal performance of the proposed FD systems for these two public datasets. Pytorch 1.13.0 is used on a 64-bit Windows 10 PC with an Intel 13th Gen I7-13700K CPU 3.40 GHz and an Nvidia RTX4080 GPU with 16-GB dedicated memory. The data preprocessing and DL-based modeling are performed in an Anaconda environment.

A. Open Datasets

The public datasets for wearable FD systems include FallAllD [40], SisFall [41], UMAFall [53], and UPFall [54]. In this study, FallAllD and SisFall datasets were chosen to validate the proposed MLMEC framework and the use of the KD model for managing the tradeoff between accuracy and latency in designing FD systems. These two datasets are closer to real-life fall types and an ADL than other datasets, and they have common detection sites that can be used for reference and comparison. The datasets use accelerometers with a sampling rate of 200 Hz, which is helpful for the model performance.

1) FallAllD Dataset: Saleh et al. [40] proposed the public dataset that involved placing an inertial measurement unit (IMU) on the neck, chest, and waist to capture various types of falls during both nonactivity and activity. Each IMU included a triaxial accelerometer, gyroscope, magnetometer, and barometer. Fifteen healthy young participants took part in this study, providing a total of 1053 ADLs and 425 fall trials. Accelerometer data from the waist were the focus due to its prominence and ease of distinction. Table III shows the details of the FallAllD dataset. FallAllD dataset comprised 15 healthy young participants, encompassing eight males and seven females, aged between 21 and 53 years, weighing between 48 and 85 kg, and measuring in height from 158 to 187 cm. The average age, height, and weight were 32 years, 171 cm, and 67 kg, respectively.

2) SisFall Dataset: Sucerquia et al. [41] combined data from two age groups: 23 young participants and 14 participants aged 62 or older. This study used data from 21 young participants, excluding the older group without fall events. Two young participants were excluded due to incomplete ADL trials. The data, collected using a self-developed IMU fixed at the waist, include 1575 fall and 1659 ADLs trials, sampled at 200 Hz. Table IV shows the details of the SisFall dataset. The SisFall dataset involved 21 healthy young participants, including 10 males and 11 females, with ages ranging from 19 to 30 years. The weight of males varied from 58 to 81 kg, while females weighed between 42 and 63 kg. Heights ranged from 165 to 183 cm for males and from 149 to 169 cm for females. On average, the participants were 25 years old, had a height of 165 cm, and weighed 57.7 kg.

B. Performance Evaluations of MLMEC Systems

In this section, we compare the performances of dual-layer and triple-layer MEC FD systems with/without the KD approach. The candidate models of each MEC and CC include ResNet18, ResNet50, ResNet101, MobileNetV3, and basic

TABLE III
INFORMATION OF PARTICIPATION IN THE FALLALLD DATASET

Sex	Age (years)	Height(m)	Weight(kg)
8 Males	21-53	1.58-1.87	48-85
7 Females			

TABLE IV
INFORMATION OF PARTICIPATION IN THE SISFALL DATASET

Sex	Age (years)	Height(m)	Weight(kg)
10 Males	19-30	1.65-1.83	58-81
11 Females		1.49-1.69	42-63

CNN models. The hyperparameter settings of the ResNet series and MobileNetV3 are the same with [45] and [46], respectively. The hyperparameter details of the CNN model include number filters = 64, filter width = 1, epochs = 200, batch size = 64, and learning rate = 0.001. The details of parameter settings can be found on GitHub.[1] The results of all models are an average of ten experiments. In the setting of the loss function, we set $\lambda = 0.5$ and $T = 20$.

Fig. 7 presents the performances in terms of ACC, REC, PRE, and $F1$. As shown in Fig. 7(a) and (b), compared to the dual-layer MECs, the performance of triple-layer MECs has the improvement in ACC and REC, while PRE and $F1$-scores are reduced slightly based on the FallAllD and SisFall datasets. If the weaker performance DL models are included in the MLMEC system, it impacts the overall performance. It is worth mentioning that Fig. 7(a) and (b) shows all evaluation metrics that are effectively enhanced by using the KD approach whether used in FallAllD or SisFall datasets.

Here, our goal is to find outstanding teacher and TA models for triple-layer MECs with the KD approach. Therefore, we first perform the experiments of dual-layer MECs to observe the results of different student models in combination with the same teacher model. Figs. 8 and 9 present the performance of different student models with the KD approach based on the same teacher model.

As shown in Fig. 8(a), by using the FallAllD dataset, ResNet18 is adopted as a teacher model. We adopt ResNet18, MobileNetV3, and CNN as the student models with the KD approach. The performance of ResNet18 with KD outperforms the other models. Similarly, the same phenomenon is demonstrated in Fig. 8(b) and (c). As shown in Fig. 8(b) and (c), the performance of ResNet50 and ResNet101 with KD outperforms the other models, respectively.

As shown in Fig. 9(a), by using the SisFall dataset, ResNet18 is adopted as a teacher model. We also adopt ResNet18 and MobileNetV3 as the student models with the KD approach. Similarly, the performance of ResNet18 outperforms the other models, while MobileNetV3 performs as the suboptimal student model. As shown in Fig. 9(b), the performance of MobileNetV3 is slightly higher than ResNet50 and CNN. As shown in Fig. 9(c), the performance of ResNet101 is slightly better than MobileNetV3 and CNN. In addition, it is worth noticing that increasing the depth of the ResNet teacher model may not improve the performance of the student model and degrade slightly. Therefore, adopting ResNet18 and MobileNetV3 as the teacher and TA models is an option to priority consider, respectively.

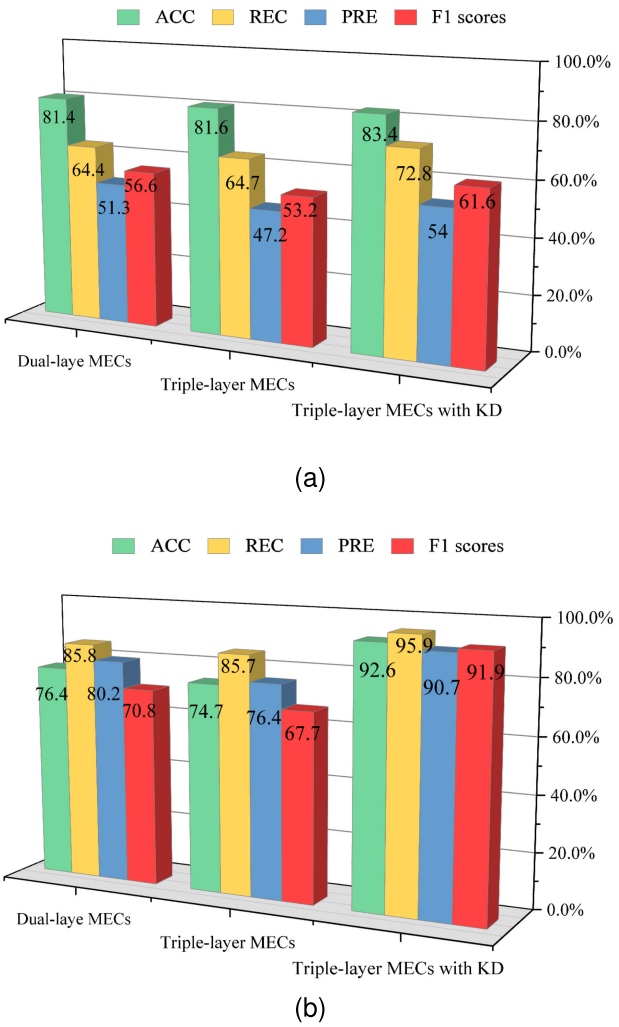


Fig. 7. Comparison of the dual-layer MECs, triple-layer MECs, and triple-layer MECs with KD architectures using two datasets. (a) FallAllD. (b) SisFall.

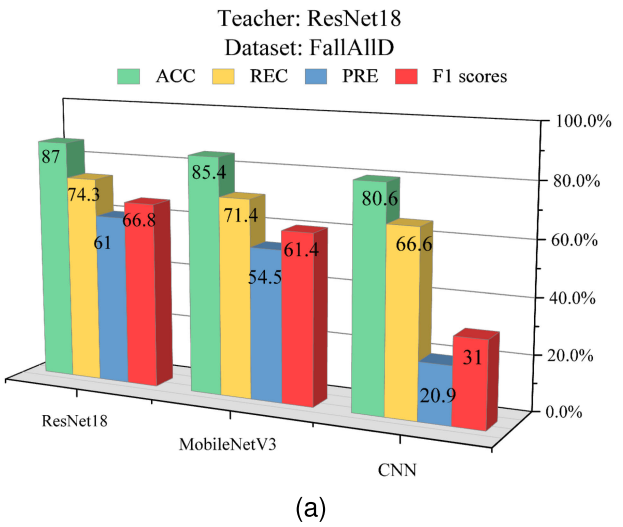
As shown in Fig. 10, by using the FallAllD dataset, ResNet18 and MobileNetV3 are adopted as the teacher and TA models. We adopt ResNet18, MobileNetV3, and CNN as the candidate student models with the KD approach. As shown in Fig. 10(a), the performance of the ResNet18 student model in triple-layer is not superior to the performance of the dual-layer in Fig. 8(a). As shown in Fig. 10(b), the performance of triple-layer KD consistently outperforms the dual-layer KD by using the SisFall dataset.

C. Teacher Model Comparison

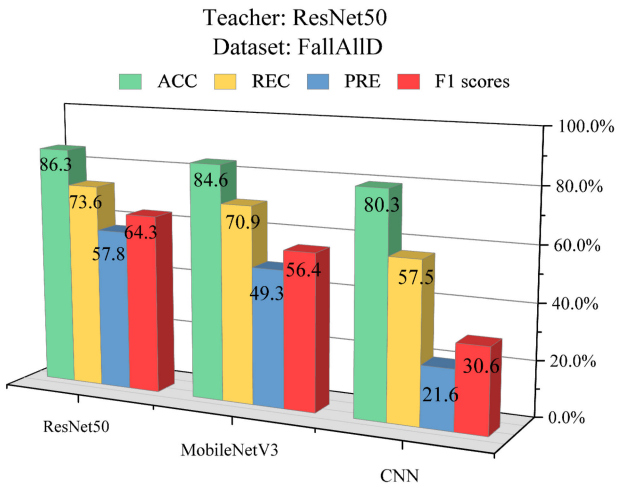
In this section, our goal is to find the outstanding teacher model from ResNet, MobileNetV3, and CNN. For comparing different DL models, the calculations of ACC_{imp}, REC_{imp}, PRE_{imp}, and $F1_{\text{imp}}$ are adopted as shown

$$\text{ACC}_{\text{imp}} = \frac{(\text{Distilled ACC} - \text{Original ACC})}{\text{Original ACC}} \times 100\% \quad (17)$$

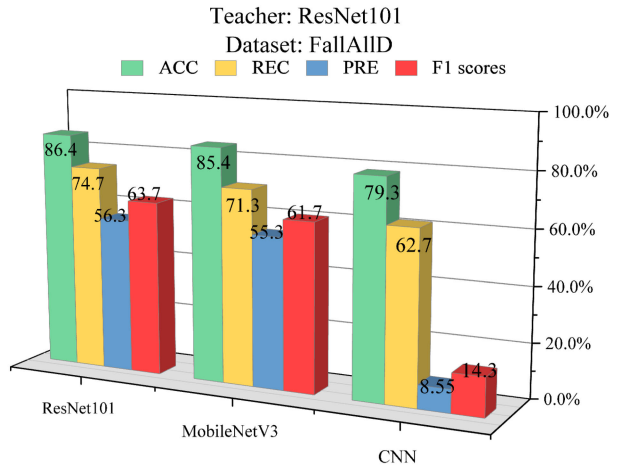
$$\text{REC}_{\text{imp}} = \frac{(\text{Distilled REC} - \text{Original REC})}{\text{Original REC}} \times 100\% \quad (18)$$



(a)

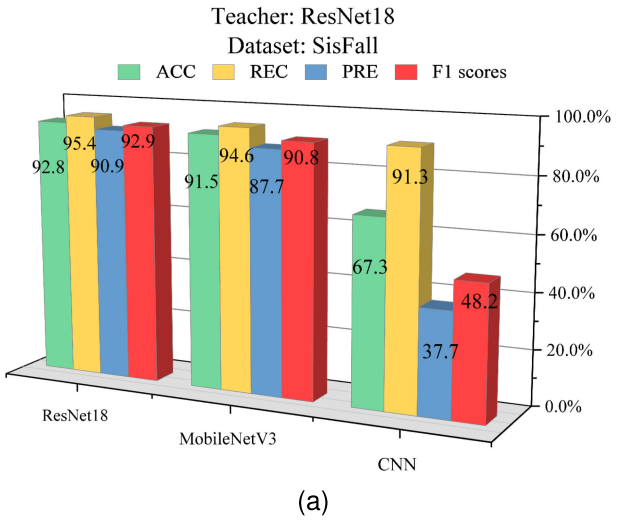


(b)

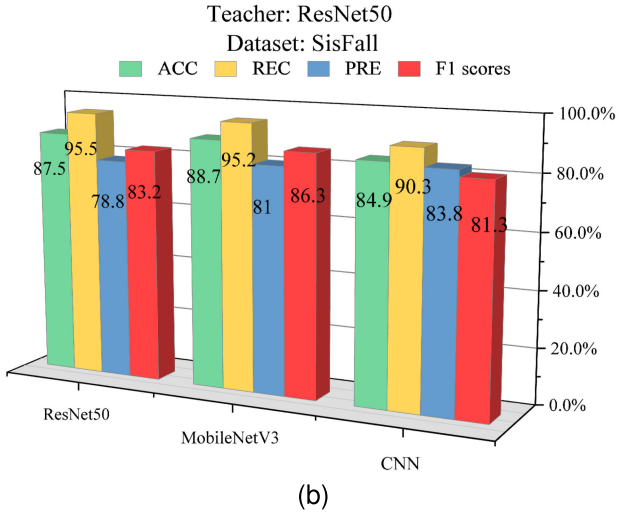


(c)

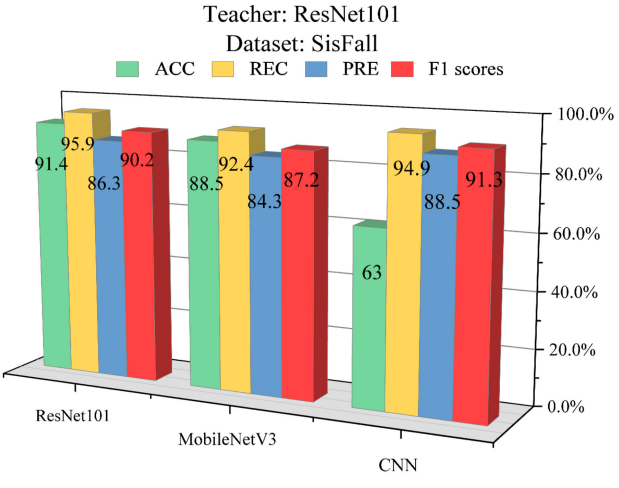
Fig. 8. Different combinations for the dual-layer with KD based on the FallAllD dataset. (a) Teacher model ResNet18. (b) Teacher model ResNet50. (c) Teacher model ResNet101.



(a)



(b)

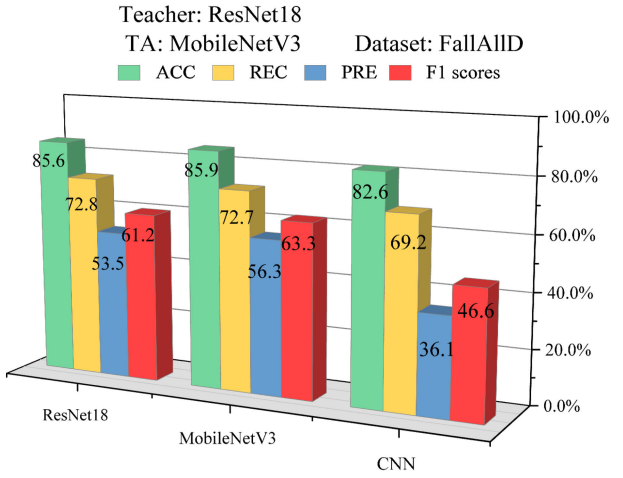


(c)

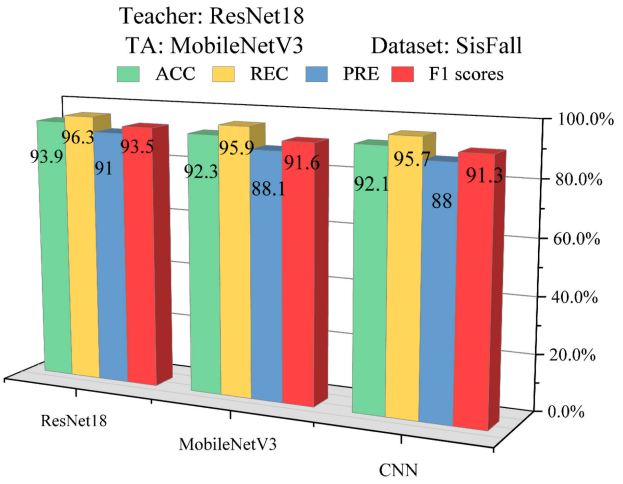
Fig. 9. Different combinations for the dual-layer with KD based on the SisFall dataset. (a) Teacher model ResNet18. (b) Teacher model ResNet50. (c) Teacher model ResNet101.

$$\text{PRE}_{\text{imp}} = \frac{(\text{Distilled PRE} - \text{Original PRE})}{\text{Original PRE}} \times 100\% \quad (19)$$

$$F1_{\text{imp}} = \frac{(\text{Distilled } F1 - \text{Original } F1)}{\text{Original } F1} \times 100\%. \quad (20)$$



(a)



(b)

Fig. 10. Different combinations for the triple-layer with KD. (a) FallAllD dataset. (b) SisFall dataset.

TABLE V
COMPARISON OF CNN AND MOBILENETV3

Teacher Model	Student Model	ACC_{imp}	REC_{imp}	PRC_{imp}	$F1_{imp}$
MobileNetV3	CNN	0.29%	0.62%	3.38%	2.69%
	ResNet18	**0.94%**	1.38%	3.17%	2.81%
	ResNet50	-2.21%	0.78%	1.96%	-2.47%
	ResNet101	0.13%	0.01%	3.71%	2.48%
CNN	MobileNetV3	0.82%	1.27%	**3.84%**	**2.93%**
	ResNet18	-0.07%	-2.88%	-11.10%	-13.71%
	ResNet50	0.37%	**1.81%**	-4.16%	-2.97%
	ResNet101	-2.69%	0.33%	-4.15%	-10.12%

Tables V and VI precisely compare the performances of different teacher models using the KD approach in the dual-layer framework. As shown in Tables V and VI, the distillation effect of using lighter models CNN and MobileNetV3 as a teacher model is significantly worse than that of ResNet as a teacher. Therefore, ResNet is adopted as the teacher model in CC. These results demonstrate that the various depths of the ResNet model have no obvious improvement in the KD approach. Even if the deeper ResNet is adopted, the teacher model may not improve the performance of the student model and degrade slightly. Hence, ResNet18 is adopted as the teacher model in the proposed MLMEC framework. It is worth noticing that the highest improvement

TABLE VI
COMPARISON OF RESNET18, RESNET50, AND RESNET101

Teacher Model	Student Model	ACC_{imp}	REC_{imp}	PRC_{imp}	$F1_{imp}$
ResNet18	CNN	0.20%	11.45%	-32.79%	-23.20%
	ResNet18	4.96%	11.64%	43.13%	29.07%
	MobileNetV3	4.56%	4.36%	**92.17%**	**53.54%**
ResNet50	CNN	-0.57%	-4.95%	-33.31%	-26.94%
	ResNet50	5.04%	11.00%	59.09%	37.63%
	MobileNetV3	3.73%	4.62%	76.34%	42.52%
ResNet101	CNN	-1.55%	4.07%	-71.65%	-63.62%
	ResNet101	**5.73%**	**12.72%**	77.45%	49.23%
	MobileNetV3	4.46%	4.36%	87.99%	50.13%

rate is used by ResNet and MobileNetV3 as teacher and student models, respectively. However, the first purpose of the proposed MLMEC system is to enhance the lighter model that is placed on the lower layer MEC, rather than optimize accuracy performance. Thus, we adopt MobileNetV3 as the TA model in the triple-layer MEC framework.

D. Student Model Comparison

In this section, our goal is to find the light computational complexity student model from ResNet, MobileNetV3, and CNN. We investigate the computational complexity of each layer in terms of FLOP counts, parameters number, and accuracy by using the FallAllD and SisFall datasets, as shown in Tables VII and VIII, respectively. FLOPs and parameter calculations are used to evaluate the computational complexity of the DL models in [55], [56], and [57]. To compute the number of FLOPs, we adopt the THOP package of PyTorch. The source code for the package can be found on GitHub[2] and contain comprehensive details about parameter settings. For example, FLOPs of fully connected layers are computed as

$$\text{FLOPs}_{\text{FC}} = (2I - 1)O \tag{21}$$

where I and O are the dimensions of input and output, respectively. FLOPs of convolution layers are calculated as

$$\text{FLOPs}_{\text{Conv}} = 2\text{HW}(C_{\text{in}}K^2 + 1)C_{\text{out}} \tag{22}$$

where H, W, and C_{in} are the height, width, and the number of channels of the input feature map, respectively, K is the kernel width, and C_{out} is the number of output channels. Chen et al. [57] compare the FLOPs between ResNet18 and MobileNetV3. The KD approach has not changed the calculation amount of the model, but we present FLOPs to completely illustrate the computational complexity of each model and the order of model size placed in each layer of the proposed MLMEC FD system.

To determine the data amount at each layer, we use threshold-based upward judgment to determine the uncertain range of whether the data should be sent to the upper layer model for further detection. In the proposed MLMEC, we adopt ResNet18, MobileNetV3, and CNN as CC, MEC 2, and MEC 1, respectively.

To ensure the applicability of the uncertain data in real-world scenarios, the concept of threshold is proposed during the testing phase utilizing the softmax output judgment

[2]The source codes are available at https://github.com/Lyken17/pytorch-OpCounter.

TABLE VII
COMPARISON OF COMPUTATIONAL COMPLEXITY USING FALLALLD

	MEC 1 CNN	MEC 2 MobileNetV3	CC ResNet18
FLOPs	18.17K	85.87M	1.292G
Parameters	0.058K	1.237M	13.94M
Accuracy	82.59%	85.92%	85.57%

TABLE VIII
COMPARISON OF COMPUTATIONAL COMPLEXITY USING SISFALL

	MEC 1 CNN	MEC 2 MobileNetV3	CC ResNet18
FLOPs	18.17K	85.87M	1.292G
Parameters	0.058K	1.237M	13.94M
Accuracy	89.15%	92.88%	93.89%

TABLE IX
THRESHOLD-BASED FOR UNCERTAIN DATA USING FALLALLD

Maximum Back-ward Threshold	Minimum Back-ward Threshold	MEC 1 CNN (# of data)	MEC 2 MobileNetV3 (# of data)	CC ResNet18 (# of data)
0.6	**0.4**	1373600	55000	0
0.7	**0.3**	1315600	91400	21600
0.8	**0.2**	1233000	148400	47200
0.9	**0.1**	1102400	211800	114400

TABLE X
THRESHOLD-BASED FOR UNCERTAIN DATA USING SISFALL

Maximum Back-ward Threshold	Minimum Back-ward Threshold	MEC 1 CNN (# of data)	MEC 2 MobileNetV3 (# of data)	CC ResNet18 (# of data)
0.6	**0.4**	2270800	171600	20800
0.7	**0.3**	2057600	254800	150800
0.8	**0.2**	1813200	215200	434800
0.9	**0.1**	868400	378400	1216400

TABLE XI
COMPUTATIONAL COMPLEXITY IN DUAL-LAYER MEC WITHOUT KD

Data	MEC 1 CNN (# of data)	CC ResNet18 (# of data)	Latency(ms)
FallAllD	926400	502200	2145.12
SisFall	1813200	650000	2752.54

process. The kinds of ranges are categorized into three groups: fall types, ADLs, and uncertain data. If the detection of the current MEC layer is classified as uncertain data, it is sent to the next MEC layer. The range of maximum and minimum threshold values is discussed based on threshold judgment. For example, the softmax output of the current layer is larger than the maximum threshold value, which is set to 0.8 to represent a fall type, while less than the minimum threshold value is set to 0.2 to represent ADLs.

Tables IX and X present the data amount of each layer in the MLMEC under different threshold values setting by using FallAllD and Sisfall datasets. In the FallAllD dataset, there is a total of 1 428 600 testing data, while the SisFall dataset has 2 463 200 testing data.

The preset threshold values are chosen to maximize the resource efficiency of the MECs. Based on the threshold-based judgment for uncertain data, the data amount in each layer is compared, as shown in Tables XI–XIII. In addition to comparing the processing data amount, the latency of each layer is also calculated. The parameters s_n^i in (16) are the data transmission rate per second, and the parameters θ_n^i are the CPU computation load.

The latencies of the dual-layer and triple-layer MEC architecture without KD are presented in Tables XI and XII, respectively. Since the dual-layer only has the latency from MEC 1 to CC, the latency from MEC 1 to MEC 2 of the triple-layer is used to compare, and the improvement is approximately 4.64%.

TABLE XII
COMPUTATIONAL COMPLEXITY IN TRIPLE-LAYER MEC WITHOUT KD

Data	MEC 1 CNN (# of data)	MEC 2 MobileNetV3 (# of data)	CC ResNet18 (# of data)	MEC 1 to MEC 2 Latency (ms)	MEC 2 to CC Latency (ms)
FallAllD	926400	430200	72000	2045.71	344.48
SisFall	1813200	215200	434800	2691.28	1841.93

TABLE XIII
COMPUTATIONAL COMPLEXITY IN TRIPLE-LAYER MEC WITH KD

Data	MEC 1 CNN (# of data)	MEC 2 MobileNetV3 (# of data)	CC ResNet18 (# of data)	MEC 1 to MEC 2 Latency (ms)	MEC 2 to CC Latency (ms)
FallAllD	1233000	148400	47200	844.86	250.92
SisFall	2012400	322800	128000	1855.35	562.11

TABLE XIV
COMPARISON OF MIN–MAX NORMALIZATION AND Z-SCORE STANDARDIZATION

Methods	DL Models	ACC_{imp}	REC_{imp}	PRC_{imp}	$F1_{imp}$
Min-max	ResNet18	82.64%	67.94%	41.61%	50.85%
	MobileNetV3	81.64%	68.42%	28.37%	39.99%
	CNN	80.44%	59.76%	31.08%	40.39%
Z-score	ResNet18	76.94%	59.38%	32.31%	43.84%
	MobileNetV3	62.71%	47.23%	23.78%	29.61%
	CNN	74.83%	55.14%	36.28%	45.37%

As shown in Tables XII and XIII, we compare the data amount between the without/with the KD approach using two datasets. By using the FallAllD dataset, the data amount of CC is reduced by 34%, while MEC 2 is reduced by 61%. In the SisFall dataset, the data amount of CC is reduced by 71%, while MEC 2 is reduced by 48%. In addition, the latency of the MLMEC architecture is calculated by (16). By comparing Tables XII and XIII, the latency of the MLMEC with KD exhibits a significant reduction compared to that without KD. Specifically, the latency is remarkably reduced by 54.15% for the FallAllD dataset, while it is reduced by 46.67% for the SisFall dataset. Therefore, falls detected at the MEC 1 or MEC 2 in advance, no need to upward the data to CC is a higher computational for further detection, leading to a more efficient.

E. Min–Max Normalization and Z-Score Standardization

Choudhury and Soni [12] and Henderi et al. [58] compared the detection performance by using min–max normalization and z-score standardization. In [12], the obtained test results using min–max normalization and z-score standardization have similar performance in terms of accuracy rate. In [58], the obtained test results using min–max normalization outperform z-score standardization in terms of accuracy rate. Based on the testing result that shows min–max normalization outperforms z-score standardization in terms of ACC, REC, PRE, and $F1$-score, as shown in Table XIV, we chose the min–max normalization in our experiments.

F. Discussion

We summarized the insights from our experiments as follows. Without the KD approach, the ACC performances

of the ResNet, MobileNetV3, and CNN models are 82.64%–88.12%, 67.57%–81.53%, and 72.33%–80.59%, respectively. With the KD approach, the improved ACC performances of combined ResNet, MobileNetV3, and CNN models in the proposed triple-layer MECs are 85.57%–93.89%, 85.92%–92.88%, and 82.99%, respectively. Meanwhile, the FLOPs of ResNet, MobileNetV3, and CNN in the proposed triple-layer MECs are maintained at 1.292G, 85.87M, and 18.17k, which are the same as before combining, respectively.

The generalizability of the proposed MLMEC FD system with the KD approach is observed as follows.

1) *Suitability of the Datasets:* FallAllD and SisFall datasets are the sensor data that are collected from triaxial accelerometers. Thus, as long as the sensor data from triaxial accelerometers can be used to train the proposed MLMEC FD system directly.
2) *Combinations of DL Models:* Comparing different deployed DL model combinations based on FallAllD and SisFall datasets, the proposed triple-layer MLMEC architecture that adopts ResNet18, MobileNetV3, and CNN in the teacher, TA, and student models is the outstanding combination in terms of accuracy rate and latency rate.
3) *Effectiveness of KD:* For FallAllD and SisFall datasets, the performances of FD are improved significantly by the KD approach. The improved accuracy rates by the KD approach are 2.78% and 11.65% for the FallAllD and SisFall datasets, respectively. The improved latency rates by the KD approach are 54.15% and 46.67% for the FallAllD and SisFall datasets, respectively.

The limitations of the proposed MLMEC FD system with the KD approach are investigated as follows.

1) *Deployment of DL Models:* Since the DL-based neural network of the teacher model must have more powerful computational capability than that of the student model, the performance of the student model is improved by the KD approach. Otherwise, the performance of the student model is degraded by using the smaller DL neural network in the teacher model.
2) *Computational Capability of MEC Hardware:* Besides, the performance of the DL-based neural network needs to be considered, whether the hardware of the specific MECs can be installed and perform the proposed DL-based neural network also needs to be considered.
3) *Types of Application:* The proposed MLMEC architecture is used to solve the classification problem. On the other hand, the regression problem cannot be appropriately solved by the proposed MLMEC system.

VII. Conclusion

This article proposes an MLMEC architecture with the KD approach to enhance the accuracy and reduce the latency. The FD detection points are divided into multiple devices (EDs, MECs, and CC) to manage the tradeoff between accuracy and latency. Comparing the performances of dual-layer and triple-layer MECs with/without the KD approach, simulation results show that triple-layer MECs with the KD approach outperform in terms of confusion matrix metrics. We find the appropriate DL model combination for triple-layer MEC architecture, which is ResNet18(teacher)–MobileNetV3(TA)–CNN(student). The threshold-based judgment is introduced for distinguishing uncertain data to quantify detection levels. The performance results demonstrate increased certainty in FD at the front-end MEC, reducing uncertainty during the detection process and decreasing data transmission latency.

For future works, we summarize as follows: 1) applying the proposed MLMEC architecture to other similar physiological signal recognition, such as electroencephalogram (EEG) and electrocardiography (ECG); 2) implementing the critical paradigm of the proposed MLMEC framework on the hardware to verify the feasibility; and 3) developing a more efficient and lightweight neural model for different layers in the proposed MLMEC framework.

References

[1] *Population Division (2022). World Population Prospects 2022: Summary of Results*, document UN DESA/POP/2022/TR/NO. 3, United Nations, 2022.

[2] F. De Backere et al., "Towards a social and context-aware multi-sensor fall detection and risk assessment platform," *Comput. Biol. Med.*, vol. 64, pp. 307–320, Sep. 2015, doi: 10.1016/j.compbiomed.2014.12.002.

[3] D. Wild, U. S. Nayak, and B. Isaacs, "How dangerous are falls in old people at home?" *Brit. Med. J.*, vol. 282, no. 6260, pp. 266–268, Jan. 1981, doi: 10.1136/bmj.282.6260.266.

[4] S. Moulik and S. Majumdar, "FallSense: An automatic fall detection and alarm generation system in IoT-enabled environment," *IEEE Sensors J.*, vol. 19, no. 19, pp. 8452–8459, Oct. 2019, doi: 10.1109/JSEN.2018.2880739.

[5] S. Sharma, V. Bhatia, and A. K. Mishra, "Wireless consumer electronic devices: The effects of impulsive radio-frequency interference," *IEEE Consum. Electron. Mag.*, vol. 8, no. 4, pp. 56–61, Jul. 2019, doi: 10.1109/mce.2019.2905538.

[6] P. Vallabh and R. Malekian, "Fall detection monitoring systems: A comprehensive review," *J. Ambient Intell. Humanized Comput.*, vol. 9, no. 6, pp. 1809–1833, Nov. 2018, doi: 10.1007/s12652-017-0592-3.

[7] A. Choi et al., "Deep learning-based near-fall detection algorithm for fall risk monitoring system using a single inertial measurement unit," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 30, pp. 2385–2394, 2022, doi: 10.1109/TNSRE.2022.3199068.

[8] K. Takatou and N. Shinomiya, "A cloud-based fall detection system for elderly care with passive RFID sensor tags," in *Proc. IEEE 3rd Global Conf. Life Sci. Technol. (LifeTech)*, Mar. 2021, pp. 261–265, doi: 10.1109/LifeTech52111.2021.9391901.

[9] A. Iazzi, M. Rziza, and R. O. H. Thami, "Fall detection system-based posture-recognition for indoor environments," *J. Imag.*, vol. 7, no. 3, p. 42, Feb. 2021, doi: 10.3390/jimaging7030042.

[10] E. Alam, A. Sufian, P. Dutta, and M. Leo, "Vision-based human fall detection systems using deep learning: A review," *Comput. Biol. Med.*, vol. 146, no. 1, pp. 1–22, Jul. 2022, doi: 10.1016/j.compbiomed.2022.105626.

[11] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: A review," *Sensors*, vol. 14, no. 7, pp. 12900–12936, Jul. 2014, doi: 10.3390/s140712900.

[12] N. A. Choudhury and B. Soni, "In-depth analysis of design & development for sensor-based human activity recognition system," *Multimedia Tools Appl.*, vol. 83, no. 29, pp. 73233–73272, Sep. 2023, doi: 10.1007/s11042-023-16423-5.

[13] N. A. Choudhury and B. Soni, "An efficient and lightweight deep learning model for human activity recognition on raw sensor data in uncontrolled environment," *IEEE Sensors J.*, vol. 23, no. 20, pp. 25579–25586, Oct. 2023, doi: 10.1109/jsen.2023.3312478.

[14] N. A. Choudhury and B. Soni, "An adaptive batch size-based-CNN-LSTM framework for human activity recognition in uncontrolled environment," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 10379–10387, Oct. 2023, doi: 10.1109/TII.2022.3229522.

[15] R. Jain and V. B. Semwal, "A novel feature extraction method for preimpact fall detection system using deep learning and wearable sensors," *IEEE Sens. J.*, vol. 22, no. 23, pp. 22943–22951, Dec. 2022, doi: 10.1109/JSEN.2022.3213814.

[16] X. Chai, B.-G. Lee, M. Pike, R. Wu, D. Chieng, and W.-Y. Chung, "Pre-impact firefighter fall detection using machine learning on the edge," *IEEE Sensors J.*, vol. 23, no. 13, pp. 14997–15009, Jul. 2023, doi: 10.1109/jsen.2023.3279858.

[17] A. Rezaei et al., "Unobtrusive human fall detection system using mmWave radar and data driven methods," *IEEE Sensors J.*, vol. 23, no. 7, pp. 7968–7976, Apr. 2023, doi: 10.1109/JSEN.2023.3245063.

[18] J. Lu and W.-B. Ye, "Design of a multistage radar-based human fall detection system," *IEEE Sensors J.*, vol. 22, no. 13, pp. 13177–13187, Jul. 2022, doi: 10.1109/JSEN.2022.3177173.

[19] W.-L. Hsu, J.-X. Liu, C.-C. Yang, and J.-S. Leu, "A fall detection system based on FMCW radar range-Doppler image and bi-LSTM deep learning," *IEEE Sensors J.*, vol. 23, no. 18, pp. 22031–22039, Nov. 2023, doi: 10.1109/JSEN.2023.3300994.

[20] F. Liu, G. Tang, Y. Li, Z. Cai, X. Zhang, and T. Zhou, "A survey on edge computing systems and tools," *Proc. IEEE*, vol. 107, no. 8, pp. 1537–1562, Aug. 2019, doi: 10.1109/JPROC.2019.2920341.

[21] L.-J. Kau and C.-S. Chen, "A smart phone-based pocket fall accident detection, positioning, and rescue system," *IEEE J. Biomed. Health Informat.*, vol. 19, no. 1, pp. 44–56, Jan. 2015, doi: 10.1109/JBHI.2014.2328593.

[22] D. Mohan et al., "Artificial intelligence and IoT in elderly fall prevention: A review," *IEEE Sensors J.*, vol. 24, no. 4, pp. 4181–4198, Feb. 2024, doi: 10.1109/jsen.2023.3344605.

[23] D. Lin et al., "A novel fall detection framework with age estimation based on cloud-fog computing architecture," *IEEE Sensors J.*, vol. 24, no. 3, pp. 3058–3071, Feb. 2024, doi: 10.1109/JSEN.2023.3334555.

[24] A. Singh, T. Margaria, and F. Demrozi, "CNN-based human activity recognition on edge computing devices," in *Proc. IEEE Int. Conf. Omni-layer Intell. Syst. (COINS)*, Jul. 2023, pp. 1–4, doi: 10.1109/coins57856.2023.10189270.

[25] W.-J. Chang, L.-B. Chen, M.-C. Chen, J.-P. Su, C.-Y. Sie, and C.-H. Yang, "Design and implementation of an intelligent assistive system for visually impaired people for aerial obstacle avoidance and fall detection," *IEEE Sensors J.*, vol. 20, no. 17, pp. 10199–10210, Sep. 2020, doi: 10.1109/JSEN.2020.2990609.

[26] J.-S. Lee and H.-H. Tseng, "Development of an enhanced threshold-based fall detection system using smartphones with built-in accelerometers," *IEEE Sensors J.*, vol. 19, no. 18, pp. 8293–8302, Sep. 2019, doi: 10.1109/JSEN.2019.2918690.

[27] P. Wang, Z. Zheng, B. Di, and L. Song, "HetMEC: Latency-optimal task assignment and resource allocation for heterogeneous multi-layer mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4942–4956, Oct. 2019, doi: 10.1109/TWC.2019.2931315.

[28] Y. Zhang, B. Di, P. Wang, J. Lin, and L. Song, "HetMEC: Heterogeneous multi-layer mobile edge computing in the 6G era," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4388–4400, Apr. 2020, doi: 10.1109/TCE.2023.3280484.

[29] J. P. Gou, B. S. Yu, S. J. Maybank, and D. C. Tao, "Knowledge distillation: A survey," *Int. J. Comput. Vis.*, vol. 129, no. 31, pp. 1789–1819, Jul. 2021, doi: 10.1007/s11263-021-01453-z.

[30] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh, "Improved knowledge distillation via teacher assistant," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 5191–5198, doi: 10.1609/aaai.v34i04.5963.

[31] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[32] C.-H. Wang, K.-Y. Huang, Y. Yao, J.-C. Chen, H.-H. Shuai, and W.-H. Cheng, "Lightweight deep learning: An overview," *IEEE Consum. Electron. Mag.*, vol. 13, no. 4, pp. 51–64, Jul. 2024, doi: 10.1109/MCE.2022.3181759. [Online]. Available: https://ieeexplore.ieee.org/document/9802879

[33] T. X. Hoa et al., "3DKD–An effective knowledge distillation-based model for human fall detection," in *Proc. RIVF Int. Conf. Comput. Commun. Technol. (RIVF)*, vol. 2, Dec. 2023, pp. 89–94, doi: 10.1109/rivf60135.2023.10471785.

[34] J. Ni, R. Sarbajna, Y. Liu, A. H. H. Ngu, and Y. Yan, "Cross-modal knowledge distillation for vision-to-sensor action recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 4448–4452, doi: 10.1109/ICASSP43922.2022.9746752.

[35] H.-A. Rashid and T. Mohsenin, "HAC-POCD: Hardware-aware compressed activity monitoring and fall detector edge POC devices," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2023, pp. 1–5, doi: 10.1109/biocas58349.2023.10389023.

[36] Z. Quan et al., "SMTDKD: A semantic-aware multimodal transformer fusion decoupled knowledge distillation method for action recognition," *IEEE Sensors J.*, vol. 24, no. 2, pp. 2289–2304, Jan. 2024, doi: 10.1109/jsen.2023.3337367.

[37] T.-H. Chi, K.-C. Liu, C.-Y. Hsieh, Y. Tsao, and C.-T. Chan, "Prefallkd: Pre-impact fall detection via CNN-ViT knowledge distillation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10094979.

[38] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[39] X. Yu, J. Jang, and S. Xiong, "A large-scale open motion dataset (KFall) and benchmark algorithms for detecting pre-impact fall of the elderly using wearable inertial sensors," *Frontiers Aging Neurosci.*, vol. 13, pp. 1–14, Jul. 2021, doi: 10.3389/fnagi.2021.692865.

[40] M. Saleh, M. Abbas, and R. B. L. Jeannès, "FallAllD: An open dataset of human falls and activities of daily living for classical and deep learning applications," *IEEE Sensors J.*, vol. 21, no. 2, pp. 1849–1858, Jan. 2021, doi: 10.1109/JSEN.2020.3018335.

[41] A. Sucerquia, J. López, and J. Vargas-Bonilla, "SisFall: A fall and movement dataset," *Sensors*, vol. 17, no. 1, p. 198, Jan. 2017, doi: 10.3390/s17010198.

[42] A. H. Saleknia and A. Ayatollahi, "Multi step knowledge distillation framework for action recognition in still images," in *Proc. 20th CSI Int. Symp. Artif. Intell. Signal Process. (AISP)*, vol. 10, Feb. 2024, pp. 1–7, doi: 10.1109/aisp61396.2024.10475221.

[43] C.-Y. Hsieh, K.-C. Liu, C.-N. Huang, W.-C. Chu, and C.-T. Chan, "Novel hierarchical fall detection algorithm using a multiphase fall model," *Sensors*, vol. 17, no. 2, p. 307, Feb. 2017, doi: 10.3390/s17020307.

[44] K.-C. Liu, C.-Y. Hsieh, H.-Y. Huang, S. J. Hsu, and C.-T. Chan, "An analysis of segmentation approaches and window sizes in wearable-based critical fall detection systems with machine learning models," *IEEE Sensors J.*, vol. 20, no. 6, pp. 3303–3313, Mar. 2020, doi: 10.1109/JSEN.2019.2955141.

[45] 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.

[46] A. Howard et al., "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324, doi: 10.1109/ICCV.2019.00140.

[47] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[48] A. Özdemir, "An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice," *Sensors*, vol. 16, no. 8, p. 1161, Jul. 2016, doi: 10.3390/s16081161.

[49] K.-C. Liu, K.-H. Hung, C.-Y. Hsieh, H.-Y. Huang, C.-T. Chan, and Y. Tsao, "Deep-learning-based signal enhancement of low-resolution accelerometer for fall detection systems," *IEEE Trans. Cogn. Develop. Syst.*, vol. 14, no. 3, pp. 1270–1281, Sep. 2022, doi: 10.1109/TCDS.2021.3116228.

[50] R. S. Torres, R. Visvanathan, S. Hoskins, A. Van den Hengel, and D. Ranasinghe, "Effectiveness of a batteryless and wireless wearable sensor system for identifying bed and chair exits in healthy older people," *Sensors*, vol. 16, no. 4, p. 546, Apr. 2016, doi: 10.3390/s16040546.

[51] I. Cleland et al., "Optimal placement of accelerometers for the detection of everyday activities," *Sensors*, vol. 13, no. 7, pp. 9183–9200, Jul. 2013, doi: 10.3390/s130709183.

[52] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surveys*, vol. 46, no. 3, pp. 1–33, Jan. 2014, doi: 10.1145/2499621.

[53] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "UMAFALL: A multisensor dataset for the research on automatic fall detection," *Proc. Comput. Sci.*, vol. 110, pp. 32–39, Jan. 2017, doi: 10.1016/j.procs.2017.06.110.

[54] L. Martínez-Villaseñor, H. Ponce, J. Brieva, E. Moya-Albor, J. Núñez-Martínez, and C. Peñafort-Asturiano, "UP-fall detection dataset: A multimodal approach," *Sensors*, vol. 19, no. 9, p. 1988, Apr. 2019, doi: 10.3390/s19091988.

[55] E. Torti et al., "Embedded real-time fall detection with deep learning on wearable devices," in *Proc. 21st Euromicro Conf. Digit. Syst. Design (DSD)*, Aug. 2018, pp. 405–412, doi: 10.1109/DSD.2018.00075.

[56] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," 2016, *arXiv:1611.06440*.
[57] Y. Chen et al., "Mobile-former: Bridging MobileNet and transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5260–5269, doi: 10.1109/CVPR52688.2022.00520.
[58] H. Henderi, "Comparison of min-max normalization and Z-score normalization in the K-nearest neighbor (kNN) algorithm to test the accuracy of types of breast cancer," *IJIIS, Int. J. Informat. Inf. Syst.*, vol. 4, no. 1, pp. 13–20, Mar. 2021, doi: 10.47738/ijiis.v4i1.73.

Wei-Lung Mao was born in Taiwan in 1972. He received the B.S. degree in electrical engineering from the National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan, in 1994, and the M.S. and Ph.D. degrees in electrical engineering from National Taiwan University (NTU), Taipei, in 1996 and 2004, respectively.

He is currently a Professor with the Department of Electrical Engineering and the Graduate School of Engineering Science and Technology, National Yunlin University of Science and Technology (NYUST), Douliu, Yunlin, Taiwan. His research interests include satellite navigation systems, intelligent and adaptive control systems, adaptive signal processing, neural networks, and precision control.

Chun-Chi Wang was born in Taiwan in 1995. He received the B.A. degree from the Department of Materials and Energy Engineering, MingDao University, Changhua, Taiwan, in 2018, and the M.S. degree from the Graduate Institute of Aeronautical and Electronic Technology, National Formosa University (NFU), Huwei, Yunlin, Taiwan, in 2021. He is currently pursuing the Ph.D. degree with the National Yunlin University of Science and Technology (NYUST), Douliu, Yunlin.

His research interests include automated image inspection, deep learning, robotic arm control systems, intelligent manufacturing production lines, and EtherCAT.

Po-Heng Chou (Member, IEEE) was born in Tainan, Taiwan. He received the B.S. degree in electronic engineering from National Formosa University (NFU), Huwei, Yunlin, Taiwan, in 2009, the M.S. degree in communications engineering from National Sun Yat-sen University (NSYSU), Kaohsiung, Taiwan, in 2011, and the Ph.D. degree from the Graduate Institute of Communication Engineering (GICE), National Taiwan University (NTU), Taipei, Taiwan, in 2020.

He is currently a Postdoctoral Fellow with the Research Center for Information Technology Innovation (CITI), Academia Sinica, Taipei. His research interests include deep learning-based signal processing, wireless communications, and wireless networks.

Dr. Chou received the Outstanding University Youth Award and the Phi Tau Phi Honorary Membership from NTU to honor his impressive academic achievement, in 2019. He received the Ph.D. Scholarships from the Chung Hwa Rotary Educational Foundation, from 2019 to 2020. Additionally, he was elected as the Distinguished Postdoctoral Scholar of CITI by Academia Sinica, from January 2022 to December 2023. He was invited to visit Virginia Tech (VT) Research Center, Arlington, VA, USA, as a Visiting Fellow, from August 2023 to February 2024. He received the Partnership Program for the Connection to the Top Labs in the World (Dragon Gate Program) from the National Science and Technology Council (NSTC) of Taiwan to perform advanced research at the VT Research Center, from 2024 to 2026.

Kai-Chun Liu (Member, IEEE) received the M.S. and Ph.D. degrees in biomedical engineering from National Yang-Ming University, Taipei, Taiwan, in 2015 and 2019, respectively.

From 2020 to 2023, he was a Postdoctoral Scholar with the Research Center for Information Technology Innovation, Academia Sinica, Taipei. He is currently a Postdoctoral Research Associate with the College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, MA, USA. His research interests include pervasive healthcare, wearable computing, machine learning, and bio-signal processing.

Yu Tsao (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1999 and 2001, respectively, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2008.

From 2009 to 2011, he was a Researcher with the National Institute of Information and Communications Technology, Kyoto, Japan, where he worked on research and product development in automatic speech recognition for multilingual speech-to-speech translation. He is currently a Research Fellow (Professor) and the Deputy Director of the Research Center for Information Technology Innovation, Academia Sinica, Taipei. He also serves as a Jointly Appointed Professor with the Department of Electrical Engineering, Chung Yuan Christian University, Taoyuan, Taiwan. His research interests include assistive oral communication technologies, audio coding, and bio-signal processing.

Dr. Tsao is currently an Associate Editor of IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING and IEEE SIGNAL PROCESSING LETTERS. He was a recipient of the Academia Sinica Career Development Award in 2017, National Innovation Awards from 2018 to 2021, the Future Tech Breakthrough Award in 2019, the Outstanding Elite Award from the Chung Hwa Rotary Educational Foundation from 2019 to 2020, the NSTC FutureTech Award in 2022, and the NSTC Outstanding Research Award in 2023. He is the corresponding author of a article that received the 2021 IEEE Signal Processing Society (SPS) Young Author Best Paper Award.