A Quantum Federated LSTM Approach for Fall Detection with Wearable IoT Devices

Senthan Prasanth, Quan Thanh Dao, *Graduate Student Member, IEEE,* Nhien Q. T. Thoong, *Graduate Student Member, IEEE*, Elif Ak, *Member, IEEE,* and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—Falls are common among older adults, but wearable fall-detection models often perform well on the dataset on which they are trained, yet lose accuracy when the user, device, or sensor location changes. This paper studies a privacy-preserving approach that learns from distributed wearable data while still adapting to each person. We propose a hybrid quantum long short-term memory (QLSTM) model and train it in a three-stage workflow: a public pretraining step to learn general motion patterns, federated training to learn from multiple users without sharing raw data, and a lightweight personalization step for each client. To support this setting, we collect a new inertial dataset with multiple body locations and time-annotated fall events, and we align its activity labels with common public benchmarks to enable consistent evaluation and facilitate federated transfer learning. Results show that the QLSTM transfers better than a classical LSTM when applied to new users and unseen sensor placements, achieving nearly 90% average accuracy after federated training and personalization. Compared with traditional machine learning approach, the proposed quantum machine learning model also uses fewer trainable parameters, which reduces the size of updates exchanged during federated training.**

***Index Terms*—Fall detection, federated learning, Internet of Things (IoT), quantum federated learning, quantum machine learning, transfer learning, wearable healthcare, wearable sensors**

I. Introduction

Over 14 million U.S. adults aged 65 and older, about 1 in 4, report falling each year [1]. Globally, falls are the second leading cause of unintentional injury deaths, causing an estimated 684,000 deaths annually, according to the World Health Organization [2]. Even worse, by 2050, fall-related emergency visits are projected to exceed 6 million, with more than 2.3 million requiring hospitalization [3].

Despite significant advancements in wearable devices, including improvements in embedded sensors, on-device processing power, connectivity, and advances in artificial intelligence (AI), fall detection remains an unsolved problem [4]–[6]. One of the main reasons is that human movement is extremely heterogeneous across age, gender, body composition, gait style, health status, and daily-life activity patterns [5], [7], [8]. Consequently, these differences change how they lose balance and the manner in which they fall. Older adults, for example, often fall more slowly or stiffly, while younger adults fall in faster, more dynamic ways [9]. Gender and body structure also affect stride length, balance strategies, and how people try to catch themselves [10]. These factors collectively create a wide variety of motion patterns that a single model struggles to cover. The problem is further complicated by the diversity of wearable Internet of Things (IoT) devices. For example, inertial measurement unit (IMU) configurations vary widely in terms of placement (e.g., wrist, waist, chest, thigh, ankle, or multiple-sensor arrays), sampling frequency, sensor quality, magnetometer reliability, gyroscope drift, accelerometer range, and orientation sensitivity [4]. Compounding this, falls are rare and dangerous, limiting opportunities for ethically collected real-world fall data [8]. Most datasets rely on controlled falls by healthy young adults, which means models learn patterns that do not always match everyday life. As a result, many current machine learning (ML) and deep learning (DL) models struggle to generalize across different users, contexts, and sensor placements [4]. They achieve high accuracy and sensitivity on their training datasets but often perform poorly or even fail when applied to new datasets [11], [12].

Prior works in wearable healthcare IoT have explored various forms of personalization and knowledge transfer, ranging from data augmentation [13] and generative models for user-specific adaptation to federated [14] and distributed approaches designed to mitigate non-iid (i.e., non-independent and identically distributed) data [15]. Personalized federated learning (FL) strategies, such as client grouping or sequential aggregation [16], aim to enhance knowledge sharing among similar users. Although these studies do not always target fall detection directly, they highlight a broader need: *wearable sensing systems must adapt to individual movement patterns and device-specific conditions without relying on large, centralized datasets*. This is particularly needed in fall detection, where highly variable human movement, diverse sensor placements, and limited real-world fall data make it difficult for any single model to perform well across users, devices, and everyday conditions [17], [18].

In parallel, interest in quantum ML (QML) has grown

Q. T. Dao, S. Prasanth, N. Q. T. Thoong, E. Ak, and T. Q. Duong are with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada, e-mail: {sprasanth, qtdao, qtnthoong, elif.ak, tduong}@mun.ca

The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941, and in part by the NSERC CREATE program (Grant number 596205-2025).

alongside the increasing digitization of health data and the widespread use of wearable IoT devices [19], [20]. Wearable IoT signals in healthcare are often high-dimensional and time-series data, which makes them a suitable target for quantum-enhanced analysis [21]. In recent years, QML has been explored for tasks such as human activity recognition (HAR) [22], electrocardiogram (ECG) arrhythmia classification [23], and detecting subtle anomalies in sensor data [21]. Beyond wearables, IoT, and HAR, researchers have also applied QML to clinical datasets for cardiovascular risk prediction, large-scale patient health records, and medical image classification and segmentation across modalities such as MRI and X-ray [24]. The common promise is that quantum algorithms could handle high-dimensional data and complex patterns more efficiently than classical ML algorithms [25]. However, current QML models face practical constraints (such as limited qubit counts, noise, shallow circuits, and non-trivial data encoding), which require careful design and benchmarking [22].

In our study, we address these challenges directly by introducing a quantum long short-term memory (QLSTM) FL framework and support it with a new dataset designed to fill several gaps in the literature. We collected new fall and activity of daily living (ADL) recordings across multiple sensors positions (i.e., six locations in total) and participant characteristics. To reduce the uniformity seen in existing datasets, we collected fall events with clearly defined contexts and fall styles. We aligned these falls and ADL activities with the three major public datasets (SisFall, FallAllD, and KFall), creating a comprehensive training resource that also supports future work on demographic differences and personalized modeling. Each event was recorded multiple times and annotated with precise fall start and end points, addressing the lack of detailed temporal labeling found in most current datasets and enabling more accurate sequence-based modeling. We then evaluated both classical and QLSTMs, demonstrating that the quantum variant captures temporal patterns more effectively under limited or noisy data, an essential property for fall detection. We also extended the QLSTM to a FL setting, conducting global and personalized experiments. Finally, we performed cross-position generalization to test how models trained on one sensor location transfer to others, comparing (i) direct adaptation without federated collaboration, (ii) federated global modeling without personalization, and (iii) federated modeling followed by transfer learning for personalized final models. Our contributions are summarized as follows:

- We introduce a QLSTM FL framework, addressing the lack of quantum-based and privacy-preserving personalized approaches in fall detection.
- We collect data from six sensor locations, context-defined fall styles, ADL activities, and precise fall start–end annotations, addressing gaps in the scripted nature of existing fall datasets, the lack of detailed temporal labeling, and limited sensor-placement diversity.
- We demonstrate that QLSTM models better capture temporal patterns under limited and noisy data, addressing a core limitation of classical sequence models. We also evaluate federated, personalized, and cross-position transfer settings, addressing the underexplored problem of how fall detection models adapt across individuals and wearable locations.
- We record our dataset for falls and ADLs matched to SisFall, FallAllD, and KFall labels, creating a resource that improves cross-dataset comparability and supports generalization studies on personalized wearable sensors, an area where current literature remains limited.

The remainder of this study is organized as follows. In Section II, we first present related works that situate our work within existing fall detection, wearable sensors, FL, and quantum machine learning research, highlighting the gaps our study targets. We then describe the datasets used in this study in Section III, beginning with our newly collected multi-position wearable dataset and followed by the three public benchmarks, along with the preprocessing steps applied to ensure a consistent experimental setup. Next, we detail the proposed model in Section IV, including the variational quantum circuit (VQC) underlying the QLSTM, its integration into the recurrent architecture, and the overall fall classification pipeline, together with the federated training and personalization strategy. Section V then outlines the simulation setup and shows results on QLSTM configuration selection, comparison with classical LSTM under pretraining and federated settings, efficiency in terms of parameters and communication, and cross-position generalization across sensor locations. Finally, we conclude with a discussion of the main findings and their implications for future work on personalized fall detection in Section VI.

II. Related Works

This section reviews the current state of fall detection systems, FL approaches for wearable sensors, and quantum machine learning applications in healthcare and synthesizes key insights that motivate our proposed framework.

A. Recent Fall Detection Approaches

Recent comprehensive surveys on wearable fall detection systems provide an extensive taxonomy of fall detection methods based on their algorithmic approaches, including threshold-based, conventional machine learning, and deep learning methods [4]. Following this taxonomy, threshold-based fall detection methods exploit the sharp changes in the measured signal (typically acceleration magnitude) and body orientation that often appear during a fall. Smartphone and IMU implementations usually define one or more acceleration and posture thresholds, and then apply a short confirmation stage to reduce false alarms [26], [27]. Multi-threshold designs are also used to encode different fall phases (brief violent motion, ground impact, descent direction), and can be integrated with IoT connectivity for real-time alerting [28]. Some works also

combine thresholds with fuzzy logic to improve robustness across ADLs that look "fall-like" in inertial space [29]. Pre-impact variants focus on early detection windows. For example, height-aware strategies evaluate whether motion leaves a stable region before impact and report lead times on the order of a few hundred milliseconds in controlled datasets [30]. Overall, threshold-based methods are computationally efficient and simple to deploy on wearable devices, but they are highly sensitive to threshold selection and often produce false alarms during ADLs [4].

On the other hand, ML methods improve over raw thresholds by modeling temporal structure and using richer features. Hidden Markov model (HMM)-based approaches are often used because they handle sequential dynamics and can tolerate noise, and some studies explicitly calibrate sensor orientation to reduce errors from device misalignment in realistic wear [5], [31], [32]. When building ML-based fall detection models, several works also emphasize that segmentation choices and sampling rates matter as much as the classifier [32], [33]. Lower sampling rates can save energy, but they can also remove discriminative details needed by common ML models [32]. Similarly, window size and segmentation strategy directly affect latency, power use, and reliability in online detection [34]. In order to overcome these challenges, some approaches use a two-stage design, where a permissive threshold triggers a second-stage classifier to reduce false positives [35]. In addition, proposed user-adaptive strategies fuse inertial signals with physiological cues (e.g., heart rate) and then adapt the decision boundary per user to address sampling rates and window size challenges [36]. Overall, these ML approaches, based on classifiers such as k-nearest neighbors (KNN), support vector machines (SVM), HMM, and random forests, improve fall detection performance over threshold-based methods, but they remain limited by their inability to learn discriminative representations directly from raw or minimally processed sensor signals [4].

Conversely, DL methods reduce the need for manual feature design by learning representations from raw data. Convolutional neural network (CNN)-based pipelines, for example, center a short window around an impact-like event and then run a compact model to classify fall versus non-fall [37]. Another DL architecture, named multi-kernel LSTM self-attention network (MKLS-Net), combined multi-kernel convolution, LSTM, and self-attention for wearable fall detection and reports prominent results on SisFall compared to one-layer LSTM and CNN models [4]. For pre-impact detection and activity recognition, multi-sensor or multi-source designs learn separate representations per sensor stream and then fuse them to improve accuracy [38]. Similar to ML approaches, in DL models, several works also injected domain knowledge. For example, gait-derived spatiotemporal parameters can be fed into sequence models (e.g., LSTM) for fall risk assessment [39]. Other deep models targeted near-fall patterns to support fall risk monitoring from a single IMU [40]. However, model complexity constraints (e.g., trainable parameters) remain a central theme, so recent designs often emphasized compact architectures, embedded inference, and careful power budgeting [41], [42].

More importantly, sensor placement varies across users, device hardware differs, and movement patterns change with age, impairment, and context. As a result, generalization failure is not a corner case but a structural property of wearable fall detection. Evidence across recent studies also suggests that this limitation cannot be addressed solely by deeper models or larger training sets [4], [5], [7]. Instead, it motivates approaches that explicitly adapt to user and device variability without requiring centralized access to sensitive raw data, naturally leading to FL and transfer learning on wearable sensors.

These architectural advances demonstrate substantial progress, yet they do not fully address the fundamental challenge of cross-user and cross-device generalization, which motivates federated and transfer learning approaches.

B. Transfer Learning and Federated Learning on Wearable Sensors for Fall Detection

Beyond the training and benchmarking approaches discussed above, some studies focus on the domain gap. For example, the UP-Fall dataset offers a large multimodal collection (i.e., wearable, ambient, and vision signals) that supports transfer learning experiments for personalization and elderly population studies, where data collection is more challenging [43]. Similarly, another study generated synthetic IMU data from reconstructed motion to support personalization and to overcome data collection difficulties [44]. However, a comprehensive study on cross-dataset training and evaluation also showed that increasing the domain or data fusion techniques (e.g., by adding more scripted fall data or augmented data) does not automatically translate into robust real-world performance [15]. This motivates methods that adapt to the target user or various devices and sensors by providing privacy for personal and sensitive data.

Accordingly, in fall detection, a representative example was Fed-ELM, which first adapted to a user using a small set of misclassified local samples and then performed federated aggregation to share knowledge across users while preserving privacy [45]. Beyond fall detection, FedCS tackled the same issue (e.g., non-iid clients and the need for personalization and privacy) by introducing confidence-based similarity and client grouping strategies for personalized FL in autonomous IoT [16]. Two additional studies also extended this FL approach toward the kinds of heterogeneity seen in real health wearables. For example, FedHealth is positioned as a federated transfer learning framework [46]. It used FL for aggregation and transfer learning to build personalized models, evaluated on wearable activity recognition and a Parkinson's-related application [46]. The second study, named MMFed, moved to multimodal FL on heterogeneous devices, training local autoencoders per modality and aggregating toward a global representation under a semi-synchronous scheme [47]. Finally, a

survey framed these trends more broadly, emphasizing privacy, interoperability, data management, and security challenges that FL is expected to address in distributed healthcare and IoT systems [14].

C. Quantum Machine Learning in Healthcare

In parallel, QML is an emerging field that integrates quantum computing principles (e.g., superposition and entanglement) with machine learning algorithms [25]. Several works used cardiovascular disease as a testbed for QML classifiers and hybrid pipelines. For example, an approach called QuCardio proposed quantum models for multi-class cardiovascular disease detection on an ECG image dataset with four categories (normal, abnormal heartbeat, myocardial infarction, and history of myocardial infarction) [23]. It combined quantum SVM variants and a quantum deep-learning component and reported strong classification metrics relative to its baselines [23]. In a complementary direction, another study presented a quantum-enhanced framework for heart disease prediction using the Kaggle heart disease dataset and evaluated multiple quantum learners within the same workflow, including quantum SVM and a quantum bagging ensemble [48]. These studies illustrate how QML is being explored for practical clinical prediction tasks where data are structured, and labels are available [23], [48].

Some studies focused on continuous sensing and temporal structures with QML in healthcare. A study proposed a hybrid QLSTM design for estimating physical activity energy expenditure in older adults using a public wearable dataset [20]. Their model integrated a variational quantum circuit into an LSTM-style predictor and studied circuit/gate and embedding choices, reporting improvements over common classical baselines for this regression task [20]. Another study addressed EMG-based eye-movement detection from extraocular muscle signals. They used a quantum CNN classifier and positioned the approach for assistive and oculomotor-related healthcare applications. Together, these papers emphasize that the hybrid quantum–classical approach is used as a learnable feature transformer for noisy health signals [49].

Beyond direct clinical datasets, QML was also tested for the links between environmental exposure and healthy aging using a hybrid model (CNN + QLSTM) to predict multiple air pollutants (e.g., PM2.5 and NO2) using monitoring-station data [19]. Similarly, an mmWave radar captured motion data, and a quantum CNN was trained to recognize activities (i.e., HAR). The quantum model achieved about 92.6% accuracy, outperforming a classical deep CNN (≈89.9% accuracy) on the same task [22]. This suggests QML can extract spatio-temporal features from health sensor signals effectively.

Moreover, recent surveys on quantum FL also showed that quantum models can represent high-dimensional patterns with fewer parameters, which can reduce communication and speed up convergence in federated settings [50]–[52]. Specifically, one exemplary study combined QML with federated training for IoT diagnosis [24]. They proposed a dynamic aggregation quantum FL method that used quantum neural networks with variational circuits as local models and adapted aggregation to handle heterogeneous data conditions [24]. Although this study targeted diagnosis rather than wearable fall detection, it signals that distributed QML was already being explored in privacy-sensitive health IoT problems.

D. Research Gaps and Motivations

Several lessons emerge when these lines of work are read together. First of all, most public fall datasets were not built to study FL. They use different ADLs, sensor placements, protocols, and labels, so even basic steps (client definition, preprocessing, and fair benchmarking) become inconsistent across studies [4], [15], [43]. Secondly, the main challenge is not choosing or training the right model. The core problem is distribution shift. Users differ, age and gait change over time, sensors are worn at different body locations, hardware varies, and sampling rates are inconsistent. As a result, models often perform well on the dataset they are trained on, but their performance degrades sharply when transferred to new users, devices, or settings [4], [5], [7]. Third, privacy is not optional here. ADL traces are personal healthcare data, and large centralized collections are hard to justify, which makes privacy-preserving training a requirement [14]. Fourth, the dataset bottleneck is challenging to overcome because real falls are rare and hard to collect. Possibly, only one comprehensive dataset of real falls, the FARSEEING dataset, requires multi-year monitoring and has reported only hundreds of events, with a smaller verified subset, which explains why most benchmarks stay scripted [8]. Fifth, synthetic data, augmentation, and multimodality approaches might help to increase the detection performance, but they do not reliably solve cross-dataset or real-world generalization when the added samples do not match the target person or device [15], [43], [44]. Sixth, personalization in federated settings is still underspecified in fall detection. Many methods adapt with small local corrections or simple fine tuning, but they rarely test the edge cases, such as across people and across placements and devices over time [45]–[47]. Seventh, QML is promising for healthcare because it has already been used to learn useful features from noisy biosignals and time-series data, especially in settings where data are limited and important patterns are hard to detect [20], [25]. Yet, QML has not meaningfully reached wearable fall detection; most quantum approaches studies target other healthcare problems, leaving a clear gap for sequence-based fall modelling with quantum components [20], [22]. Finally, federated QML is especially attractive because it addresses two key challenges at the same time. It preserves privacy by keeping data on local devices, and it can reduce communication and model size because quantum models may use fewer parameters. Recent work already shows that such quantum federated setups can handle heterogeneous conditions in health IoT systems [50]–[52].

III. Datasets

Alongside the algorithms and approaches discussed in the previous section, datasets also shape what constitutes "good performance." Widely used datasets include SisFall [53], UniMiB-SHAR [54], MobiAct [55], UMAFall [56], and multimodal collections such as UP-Fall [43]. More recent datasets target broader evaluation needs, such as FallAllD [57] and pre-impact benchmarks like KFall [58]. When these datasets are reviewed together, one consistent pattern emerges: *many are built from scripted falls performed by young or middle-aged participants in controlled settings.* This setup implicitly assumes homogeneous subject dynamics, stationary sensor characteristics, and near-iid data across users and devices. However, these assumptions do not hold in deployment. Sensor placement varies across users, device hardware differs, and movement patterns shift with age, impairment, and context [59].

A. Our Dataset: FallMerge-6

To capture variability in wearable fall detection under diverse sensor placements, we collected a new dataset, termed FallMerge-6, from five participants across six sensor positions: waist, left wrist, right wrist, left ankle, right thigh, and neck. Each participant wore a wearable device equipped with a tri-axial accelerometer ($\pm 16g$), tri-axial gyroscope ($\pm 2000°/s$), and tri-axial magnetometer (± 50 Ga), sampled at 200 Hz. Fig. 1 summarizes the placement configuration and device specifications.

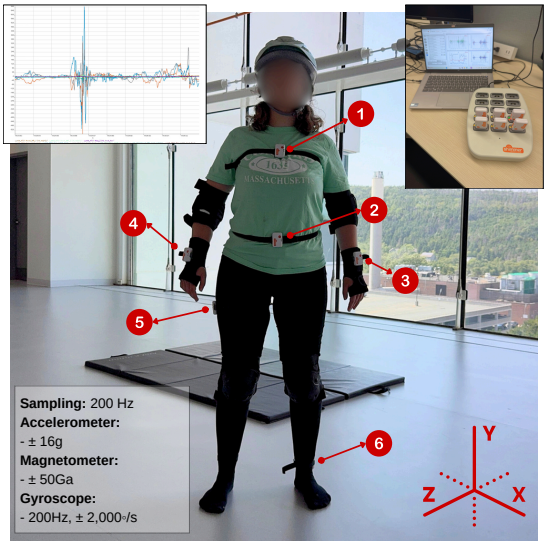

Fig. 1: Dataset collection setup, locations of six sensors, and their configurations. An example sensor reading of the gyroscope, accelerometer, and magnetometer plot is in the left corner. The sensor board and setup are on the right corner.

We recorded 34 ADLs selected to align with the activity taxonomies used in SisFall, FallAllD, and KFall, enabling consistent benchmarking and cross-dataset evaluation. Each ADL was repeated three times. To improve the validity of fall data and reduce the uniformity often observed in scripted protocols, we collected 21 fall scenarios with explicit descriptions of fall context and fall style. To enable consistent benchmarking and federated transfer across heterogeneous datasets, we aligned our 21 fall scenarios with the closest corresponding fall labels in SisFall, FallAllD, and KFall. Table I summarizes the scenario definitions and the label mapping used throughout the experiments. All trials were video recorded and labeled with precise start and end times of the fall event, providing finer temporal resolution than most existing datasets. Fig. 2 shows a representative fall trial and highlights the pre-fall, fall, and recovery phases together with the annotated fall start and fall end instants. While datasets with larger participant groups (such as SisFall) are necessary for the initial cloud pretraining phase to learn fundamental motion patterns, our dataset is specifically engineered to serve as a rigorous testbed for federated transfer learning and cross-position generalization. In personalization studies, the depth of intra-subject heterogeneity is critical. To this end, FallMerge-6 prioritizes sensor and activity diversity over a massive participant count.

B. Other Publicly Available Datasets

To support the public pretraining phase of our federated framework, we use the publicly available SisFall dataset [53]. SisFall comprises falls and ADLs from 38 participants, including 23 young adults (aged 19–30) and 15 elderly individuals (aged 60–75). Data were acquired using a waist-mounted wearable device containing two accelerometers and a gyroscope, sampled at 200 Hz. The dataset includes 19 ADL classes and 15 fall classes, totaling 2,706 ADL recordings and 1,798 fall recordings. Young participants performed both ADLs and falls, whereas elderly participants mainly performed ADLs for safety considerations; notably, one 60-year-old participant with judo experience completed the full protocol, including falls.

For consistency with FallMerge-6, and to ensure that cross-position generalization is driven primarily by sensor placement (rather than sensor modality), we use only the waist-mounted accelerometer and gyroscope channels from SisFall in our experiments (see Section V).

IV. Proposed Model

In this section, we introduce the proposed framework, detailing the data standardization process, the QLSTM network architecture, and the federated transfer learning protocol used for privacy-preserving fall detection.

A. Input Representation and Preprocessing

To ensure robustness against device heterogeneity and to construct a standardized input for the QLSTM encoder, raw inertial signals undergo a unified feature engineering pipeline.

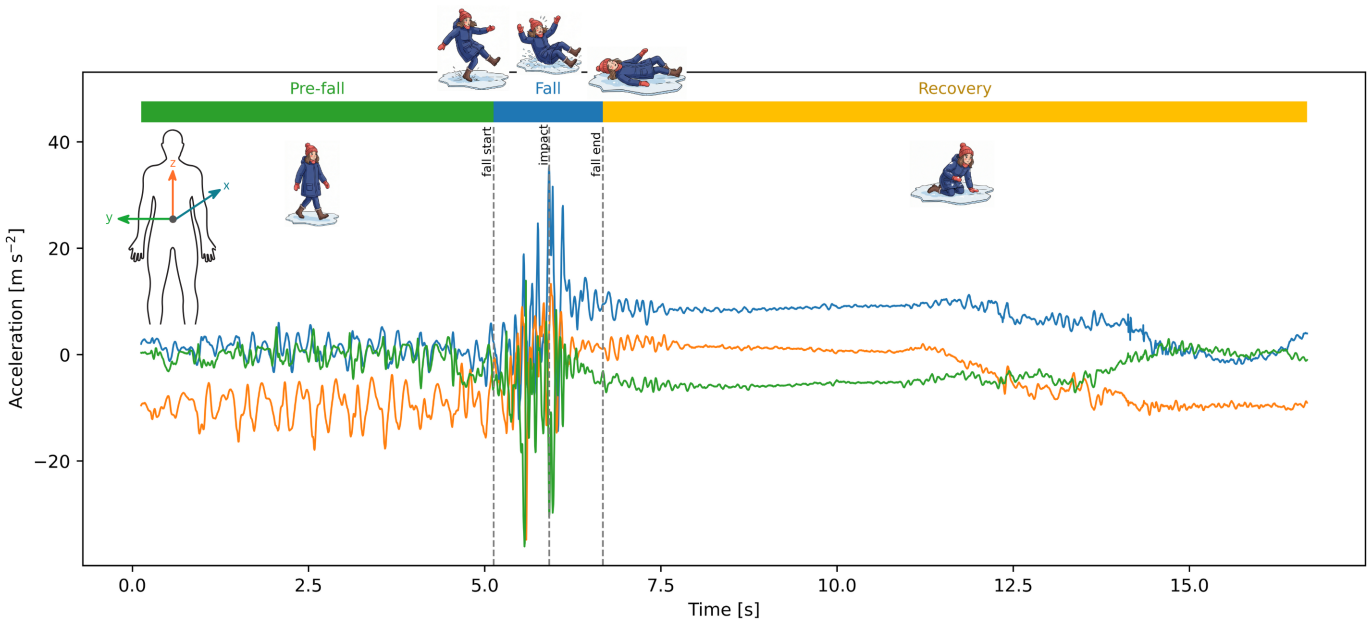

Fig. 2: Representative fall event from our FallMerge-6 dataset showing tri-axial accelerometer signals from a waist-mounted sensor. The three phases (Pre-fall, Fall, Recovery) are highlighted with temporal annotations for fall start (5.125 s), impact (≈5.8 s), and fall end (6.675 s). The inset illustrates the sensor coordinate system during upright standing.

1) Signal standardization: We focus on the six primary inertial channels: 3-axis acceleration (A_x, A_y, A_z) and 3-axis angular velocity (G_x, G_y, G_z). To address sampling rate inconsistencies across different hardware, all signals are downsampled to a uniform frequency of $f_s = 50$ Hz. Raw sensor readings are converted into physical units (g for acceleration and $^\circ/s$ for angular velocity) using device-specific scale factors. Although the device includes a magnetometer, we use only accelerometer and gyroscope channels to maintain consistency with SisFall and reduce modality mismatch.

2) Feature engineering: To enhance the discriminative power of the model for fall detection, we extract a 10-dimensional feature vector $x_t \in \mathbb{R}^{10}$ at each time step t. Let $\mathbf{a}_t = [a_{x,t}, a_{y,t}, a_{z,t}]^T$ denote the acceleration vector. The derived features include:

- **Raw channels (6):** The standardized raw values of A_x, A_y, A_z, G_x, G_y, G_z (converted to physical units and z-score normalized).
- **Acceleration magnitude (1):** Represents the overall intensity of motion, defined as $m_t = \|\mathbf{a}_t\|_2 = \sqrt{a_{x,t}^2 + a_{y,t}^2 + a_{z,t}^2}$.
- **Horizontal magnitude (1):** Captures lateral movement to distinguish falls from vertical ADLs (e.g., sitting), defined as $h_t = \sqrt{a_{x,t}^2 + a_{y,t}^2}$.
- **Jerk (1):** Measures the rate of change of acceleration, critical for detecting impact transients:

$$j_t = \frac{dm_t}{dt} \approx \frac{m_t - m_{t-1}}{\Delta t}, \tag{1}$$

where $\Delta t = 1/f_s$.

- **Tilt angle (1):** Estimates the body's inclination relative to the vertical axis, calculated as:

$$\phi_t = \arctan\left(\frac{h_t}{a_{z,t}}\right) \tag{2}$$

3) Sequence segmentation: The continuous feature stream is segmented into fixed-length sequences using a sliding window approach. We define a window size of $T_w = 100$ samples (equivalent to 2 seconds at 50 Hz) with an overlap of 20 samples. This 2-second duration was specifically selected because the critical biomechanical phases of a typical fall generally occur within this timeframe. These phases encompass the initial loss of balance, the descent, and the ground impact. A shorter window risks truncating the event and separating the initial context from the impact peak, while a longer window may dilute the critical discriminative features with irrelevant prior or subsequent ambient motion. Each window $W_i \in \mathbb{R}^{100\times 10}$ is assigned a binary label $y_i \in \{0, 1\}$ (Non-Fall/Fall) based on the ground truth annotations provided in the respective datasets.

B. Overview of the Framework

The proposed framework aims to achieve accurate, personalized fall detection through federated transfer learning while preserving privacy. For ease of reference, all key mathematical notations used throughout this work are summarized in Table II. We are given data from K different clients denoted by private datasets $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K\}$. Additionally, a public dataset $\mathcal{D}_{\text{pub}}$ is available on the cloud server. Our objective

TABLE I: Classification of fall scenarios in our dataset and corresponding label mapping to public benchmarks.

Code	Starting Activity	Reason to Fall	Direction	Context / Scenario	SisFall	FallAllD	KFall
F01	Walking	Stumbling	Forward	Trip over raised sidewalk edge	F04	101, 102	F11
F02	Walking	Stumbling	Backward	Foot catches on object, forcing body backward	N/A	N/A	N/A
F03	Walking	Stumbling	Lateral	Stepping off curb; uneven gravel causing tip	N/A	N/A	N/A
F04	Walking	Slipping	Forward	Slip on wet indoor floor or wet tile	F01	103, 104	F13
F05	Walking	Slipping	Backward	Slip on icy pavement or sliding bathroom mat	F02	107, 108	F15
F06	Walking	Slipping	Lateral	Foot skids sideways on oil or loose sand	F03	105, 106, 109, 110	F14
F07	Walking	Fainting	Forward	Orthostatic hypotension; post-meal hypoglycaemia	F06	114	N/A
F08	Walking	Fainting	Backward	Lightheaded due to heat, dehydration, or hot shower	F06	111, 112	N/A
F09	Walking	Fainting	Lateral	Transient ischaemic attack; sudden vestibular loss	F06	113	F10
F10	Jogging	Stumbling	Forward	Trip on uneven trail root; sudden direction change	F05	121, 122, 123, 124	F12
F11	Jogging	Stumbling	Lateral	Quick side-step to dodge; ankle inversion	N/A	125, 126	N/A
F12	Standing	Fainting	Forward	Sudden dizziness while standing still	F06	132	N/A
F13	Standing	Fainting	Backward	Inner-ear vertigo spell	F06	133	N/A
F14	Standing	Fainting	Lateral	Seizure episode	F06	134	N/A
F15	Sitting Up	Fainting	Forward	Fainting while trying to get up or falling asleep	F08, F13	129	F04, F06
F16	Sitting Up	Fainting	Backward	Rising too fast; post-prandial hypotension	F14	130	F08
F17	Sitting Up	Fainting	Lateral	Transferring between bed and wheelchair	F09, F15	131	F05, F07
F18	Sitting Down	Fainting	Forward	Fainting while trying to sit down	F10	129	F01
F19	Sitting Down	Fainting	Backward	Losing consciousness mid-descent; chair scoots away	F11	130	F02
F20	Sitting Down	Fainting	Lateral	Vestibular vertigo causing sideways lean during descent	F12	131	F03
F21	Walk/Stand	Fainting	Vertical	Sliding down a wall slowly; hands used to dampen fall	F07	135	F09

Note: N/A indicates no corresponding label exists in the benchmark dataset.

is to collaborate across all clients to train a global federated model $\boldsymbol{\theta}^{\mathrm{fed}}$, where any client k does not expose its private data $\mathcal{D}_k$ to the server or other clients. Furthermore, a single global model often fails to generalize to heterogeneous sensor placements (e.g., wrist vs. ankle). Therefore, we also aim to derive a personalized model $\boldsymbol{\theta}_k^{\mathrm{TL}}$ for each client k by adapting $\boldsymbol{\theta}^{\mathrm{fed}}$ such that the personalized accuracy $\mathcal{A}_k$ is maximized on the local test set:

$$\boldsymbol{\theta}_k^{\mathrm{TL}} = \arg\max_{\boldsymbol{\theta}} \mathcal{A}(\mathcal{D}_k^{\mathrm{test}}), \tag{3}$$

subject to the constraint that $\mathcal{D}_k^{\mathrm{train}}$ remains local.

The framework consists of three sequential procedures:

- **Step 1: Cloud pretraining.** A global model is initially trained on the server using the available public dataset $\mathcal{D}_{\mathrm{pub}}$. This step allows the framework to leverage existing large-scale fall datasets to learn fundamental motion patterns. By providing a well-initialized parameter set $\boldsymbol{\theta}^{\mathrm{cloud}}$, this pretraining step mitigates the "cold-start" problem, significantly accelerating convergence in subsequent federated rounds and reducing the computational burden on resource-constrained client devices during the adaptation process.
- **Step 2: Federated adaptation.** The cloud model is then distributed to all K clients. To break the data islands, we employ the Federated Averaging (FedAvg) algorithm [60]. Each client k trains the model locally on their private data $\mathcal{D}_k$ and uploads only the weight updates to the server. The server aggregates these updates to generate a refined global model $\boldsymbol{\theta}^{(r)}$. The aggregation is performed

TABLE II: Summary of notation used in this work.

Symbol	Description
K	Number of clients participating in FL.
$\mathcal{D}_{\text{pub}}$	Public dataset used for cloud pretraining.
$\mathcal{D}_k$	Local dataset stored on client k.
$\mathcal{D}_k^{\text{train}}$	Training split of client k's dataset.
$\mathcal{D}_k^{\text{test}}$	Test split of client k's dataset.
n_k	Number of samples in $\mathcal{D}_k^{\text{train}}$.
$\boldsymbol{\theta}^{\text{cloud}}$	Global model after cloud pretraining.
$\boldsymbol{\theta}^{(r)}$	Global model at federated communication round r.
$\boldsymbol{\theta}_k^{(r)}$	Client k's locally updated model in round r.
$\boldsymbol{\theta}^{\text{fed}}$	Final global model after R rounds.
$\boldsymbol{\theta}_k^{\text{TL}}$	Personalized model for client k obtained via transfer learning.
R	Number of federated communication rounds.
E_{cloud}	Number of epochs for pretraining the cloud model.
E_{local}	Number of local training epochs per client in each federated round.
E_{TL}	Number of training epochs during per-client transfer learning.

via a weighted average based on local dataset sizes $n_k = |\mathcal{D}_k^{\text{train}}|$:

$$\theta^{(r)} = \sum_{k=1}^{K} \frac{n_k}{\sum_{j=1}^{K} n_j} \theta_k^{(r)}. \tag{4}$$

This step enables the model to learn general features from diverse sensor distributions without accessing raw private data.

- **Step 3: Personalization via transfer learning.** Finally, the global federated model is personalized for each specific client. Although the global model leverages diverse data from multiple clients, its learned representation may diverge from the unique data distribution of an individual user. Therefore, we apply a transfer learning strategy in which we freeze some layers and only fine-tune the upper classification layers using the local private data $\mathcal{D}_k$. This approach allows the model to adapt effectively to the client's specific patterns. Furthermore, by training fewer parameters, we significantly save computational resources and avoid overfitting on limited local datasets. Notably, this lightweight adaptation process is also applicable to online learning, allowing the model to continuously adapt as new data becomes available.

Algorithm 1 summarizes the full procedure. While this framework is compatible with various machine learning models, we specifically employ a hybrid Quantum Long Short-Term Memory (QLSTM) network to leverage the high-dimensional processing capabilities of quantum mechanics. Details of the model will be described in the next sections.

Algorithm 1 Cloud–pretrained federated with client–specific transfer learning

Require: Public dataset $\mathcal{D}_{\text{pub}}$, private client datasets $\{\mathcal{D}_k\}_{k=1}^{K}$, cloud model training epochs E_{cloud}, number of communication rounds R, local training epochs E_{local}, transfer learning epochs E_{TL}.
Ensure: Cloud model $\boldsymbol{\theta}^{\text{cloud}}$, federated global model $\boldsymbol{\theta}^{\text{fed}}$, personalized client models $\{\boldsymbol{\theta}_k^{\text{TL}}\}$.

Stage 1: Cloud pretraining
Initialize model parameters $\boldsymbol{\theta}^{(0)}$.
for epoch $= 1$ to E_{cloud} **do**
  Update $\boldsymbol{\theta}$ on $\mathcal{D}_{\text{pub}}$.
end for
$\boldsymbol{\theta}^{\text{cloud}} \leftarrow \boldsymbol{\theta}$.
Stage 2: Federated adaptation
Initialize global weights $\boldsymbol{\theta}^{(0)} \leftarrow \boldsymbol{\theta}^{\text{cloud}}$.
for $r = 1$ to R **do**
  for $k = 1$ to K **in parallel do**
    $\boldsymbol{\theta}_k^{(r,0)} \leftarrow \boldsymbol{\theta}^{(r-1)}$.
    for epoch $= 1$ to E_{local} **do**
      Update $\boldsymbol{\theta}_k^{(r)}$ on $\mathcal{D}_k^{\text{train}}$.
    end for
    return $\boldsymbol{\theta}_k^{(r)}$ to server
  end for
  Aggregate weights to obtain $\boldsymbol{\theta}^{(r)}$ using (Eq. 4).
end for
$\boldsymbol{\theta}^{\text{fed}} \leftarrow \boldsymbol{\theta}^{(R)}$.
Stage 3: Personalization via transfer learning
for $k = 1$ to K **in parallel do**
  Initialize $\boldsymbol{\theta}_k^{\text{TL}} \leftarrow \boldsymbol{\theta}^{\text{fed}}$.
  for epoch $= 1$ to E_{TL} **do**
    Perform transfer learning on $\mathcal{D}_k^{\text{train}}$.
  end for
  Evaluate $\boldsymbol{\theta}_k^{\text{TL}}$ on $\mathcal{D}_k^{\text{test}}$.
end for
return $\boldsymbol{\theta}^{\text{cloud}}$, $\boldsymbol{\theta}^{\text{fed}}$, $\{\boldsymbol{\theta}_k^{\text{TL}}\}$.

C. Preliminaries: Quantum Computing Components

The proposed QLSTM model relies on core principles of quantum computing to implement its processing gates. The VQC constitutes the fundamental quantum processing unit, comprising three functional components: initialization, unitary transformation, and measurement.

- **Qubits and Quantum States:** Information is stored in quantum bits (qubits), the fundamental unit of quantum information. Unlike classical bits, a qubit can exist in a superposition of the basis states $|0\rangle$ and $|1\rangle$. For a circuit processing n features, we initialize n qubits in the ground state $|0\rangle$. To leverage quantum parallelism, Hadamard

gates (H) are applied to each qubit, transforming the state $|0\rangle$ into an equal superposition state $|+\rangle = (|0\rangle+|1\rangle)/\sqrt{2}$.

- **Quantum Gates and Encoding:** Operations are performed using unitary quantum gates. To map classical sensor data into the quantum Hilbert space, we utilize single-qubit rotation gates, specifically Y-rotation (R_Y) and Z-rotation (R_Z). These gates rotate the qubit state around the respective axes of the Bloch sphere by an angle θ derived from the input features. This angle encoding strategy allows continuous classical data to be represented within the quantum state. Parameterized rotation gates are also employed in the variational layers, where the rotation angles are trainable parameters optimized during the learning process.
- **Entanglement:** To capture complex dependencies between features that cannot be represented by single-qubit operations alone, multi-qubit gates are employed to create entanglement. We utilize the Control-NOT ($CNOT$) gate and the Controlled-R_Z (CR_Z) gate. These operations condition the state of a target qubit on the state of a control qubit, establishing non-classical correlations essential for building highly expressive quantum circuits.
- **Measurement:** The final step involves extracting classical information from the quantum state. This is achieved by calculating the expectation value of the Pauli-Z operator (σ_Z) for each qubit. The measurement of qubit i yields a real-valued expectation value $\langle Z_i \rangle$ in the interval $[-1, 1]$. The collection of these measurements forms an n-dimensional vector that serves as the output of the quantum circuit, ready for subsequent classical processing.

D. Quantum Long Short-Term Memory

Leveraging the quantum processing capabilities established through the VQC, we integrate quantum computing with recurrent neural networks (RNNs) to develop a temporal processing architecture optimized for time-series data analysis. The resulting QLSTM framework enhances conventional recurrent networks through quantum computation principles. Specifically, this integration directly addresses the limitations of classical architectures when processing human movement data. Time-series data derived from wearable IoT sensors is inherently noisy, highly variable, and characterized by complex, nonlinear temporal dependencies. Classical sequence models often require vast parameter spaces to capture these dynamics, which increases the risk of overfitting. By substituting classical linear transformations with VQCs, the QLSTM exploits quantum superposition and entanglement to map continuous temporal features into a high-dimensional Hilbert space. This quantum-enhanced feature mapping allows the network to capture complex sequential correlations more efficiently. Furthermore, the inherent stochasticity of quantum measurements acts as a structural regularizer, making the QLSTM highly robust against the noise and signal drift prevalent in real-world time-series data.

1) Long Short-Term Memory Cell: The LSTM architecture represents a specialized RNN designed to mitigate the vanishing gradient problem inherent in standard RNNs, particularly beneficial for sequential data processing [61]. The conventional LSTM cell incorporates four key components. First, the forget gate employs a sigmoid-activated layer to determine which information from the previous cell state warrants retention and which should be discarded. Second, the input gate regulates the admission of new information into the cell state through sigmoid activation. Third, the candidate gate generates potential new cell state values from current inputs and previous hidden states using a hyperbolic tangent layer. Fourth, the output gate applies a sigmoid-activated mechanism to filter the updated cell state to determine the composition of the hidden state output. These regulatory gates operate in coordination to maintain and update two state components: the cell state c_t and the hidden state h_t, where the former preserves long-term memory and the latter functions as working memory and cell output. The operational dynamics of the standard LSTM cell are mathematically formulated as

$$\begin{aligned} f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ g_t &= \tanh\left(W_g \cdot [h_{t-1}, x_t] + b_g\right) \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t), \end{aligned} \tag{5}$$

where f_t, i_t, g_t, and o_t represent the forget gate, input gate, candidate gate, and output gate, respectively; σ denotes the sigmoid activation function; $\tanh$ denotes the hyperbolic tangent activation function; $\odot$ represents element-wise multiplication; W and b correspond to weight matrices and bias vectors for each gate.

2) Quantum Long Short-Term Memory Cell: QLSTM cell extends the classical LSTM architecture by integrating VQCs within the gate computation framework. The complete computational pathway for each gate in our QLSTM cell involves three sequential stages. Initially, the concatenated input and previous hidden state $[h_{t-1}, x_t]$ undergoes dimensionality reduction through classical linear transformations as follows:

$$\begin{aligned} \tilde{f_t} &= W_f^{in} \cdot [h_{t-1}, x_t] + b_f^{in} \\ \tilde{i_t} &= W_i^{in} \cdot [h_{t-1}, x_t] + b_i^{in} \\ \tilde{g_t} &= W_g^{in} \cdot [h_{t-1}, x_t] + b_g^{in} \\ \tilde{o_t} &= W_o^{in} \cdot [h_{t-1}, x_t] + b_o^{in}, \end{aligned} \tag{6}$$

where W_f^{in}, W_i^{in}, W_g^{in}, and W_o^{in} are the input weight matrices and b_f^{in}, b_i^{in}, b_g^{in}, and b_o^{in} are the input bias vectors for the forget, input, candidate, and output gates, respectively. These projections reduce the dimensionality of the concatenated input to match the quantum circuits' encoding capacity, which is determined by the number of qubits.

Subsequently, the projected vectors are processed through

dedicated VQCs as

$$
\begin{aligned}
q_f &= \mathcal{M}(U_f(\tilde{f}_t)) \\
q_i &= \mathcal{M}(U_i(\tilde{i}_t)) \\
q_g &= \mathcal{M}(U_g(\tilde{g}_t)) \\
q_o &= \mathcal{M}(U_o(\tilde{o}_t)),
\end{aligned} \tag{7}
$$

where U_f, U_i, U_g, and U_o denote the VQCs for each gate; and $\mathcal{M}(\cdot)$ represents the measurement operation that extracts classical information from the quantum state via Pauli-Z expectation values. Each circuit employs n qubits and returns an n-dimensional vector of measurement results. Finally, the quantum measurement outputs undergo a second set of classical linear transformations to produce the gate activations

$$
\begin{aligned}
f_t &= \sigma(W_f^{out} \cdot q_f + b_f^{out}) \\
i_t &= \sigma(W_i^{out} \cdot q_i + b_i^{out}) \\
g_t &= \tanh(W_g^{out} \cdot q_g + b_g^{out}) \\
o_t &= \sigma(W_o^{out} \cdot q_o + b_o^{out}),
\end{aligned} \tag{8}
$$

where W_f^{out}, W_i^{out}, W_g^{out}, and W_o^{out} are the output weight matrices and b_f^{out}, b_i^{out}, b_g^{out}, and b_o^{out} are the output bias vectors for the forget, input, candidate, and output gates, respectively. These transformations map the quantum measurement results to the appropriate dimensions for the LSTM gate activations. The processed gate outputs are then combined according to the standard LSTM equations to update the cell and hidden states

$$
\begin{aligned}
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t).
\end{aligned} \tag{9}
$$

The intermediate quantum layer introduces non-linearity via quantum operations, enabling efficient capture of complex patterns. By leveraging quantum properties such as superposition and entanglement, VQCs with trainable rotation and entanglement parameters adaptively transform features. Since each LSTM gate (forget, input, candidate, and output) uses a dedicated quantum circuit, this architecture enables specialized processing per gate.

3) Fall Classifier: The quantum-enhanced fall detection system integrates a QLSTM layer within a larger neural network to analyze time-series sensor data for binary classification. The QLSTM captures temporal dependencies and outputs a hidden state summarizing motion patterns. This state is then passed through fully connected layers, beginning with a linear transformation and batch normalization to stabilize and accelerate learning. Next, the normalized output is passed through a rectified linear unit (ReLU) activation function. ReLU introduces non-linearity by zeroing out negative inputs, enabling the network to learn complex patterns. Dropout regularization is also applied, which improves generalization by randomly deactivating neurons during training, promoting more robust feature learning. Finally, a softmax layer converts the output into a probability distribution over the two classes: fall and non-fall. Cross-entropy loss is used to quantify the difference between predicted probabilities and true class labels as

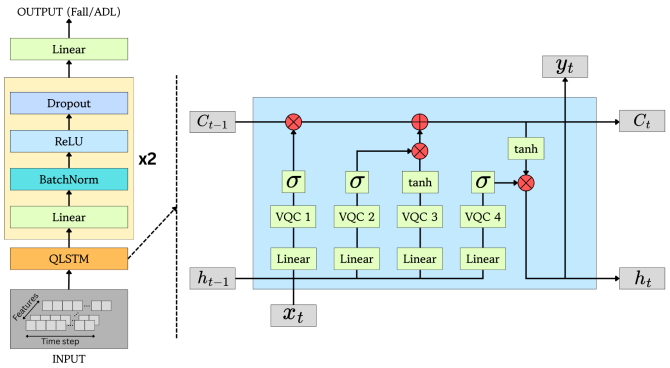

Fig. 3: Overview of the proposed QLSTM model. The left panel illustrates the complete architecture. The right panel details the internal structure of the QLSTM cell, demonstrating the integration of VQCs.

$$
\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c}^{true} \log(y_{i,c}^{pred}), \tag{10}
$$

where N represents the batch size, $C = 2$ indicates the number of classes, $y_{i,c}^{true}$ denotes the ground truth label, and $y_{i,c}^{pred}$ is the predicted probability for class c and sample i. This training objective encourages the model to assign high probability to the correct class and penalizes incorrect predictions. Fig. 3 illustrates the architecture of the proposed model and the internal structure of the QLSTM cell.

V. Experiments

This section presents the extensive experimental protocol designed to evaluate the proposed framework (Algorithm 1). We first describe the simulation setup, followed by three sets of empirical results: (i) selection of the optimal QLSTM configuration, (ii) comparison between QLSTM and classical LSTM under three adaptation schemes, and (iii) evaluation of cross-position generalization across heterogeneous sensor locations.

A. Experiment Setup

We evaluate the proposed framework using two data sources to assess performance across both public and private datasets under controlled and heterogeneous sensor placements. The public dataset $\mathcal{D}_{\text{pub}}$ is derived from the SisFall dataset using only waist-mounted accelerometer and gyroscope channels. The private datasets $\{\mathcal{D}_k\}_{k=1}^{5}$ come from our own multi-subject recordings, where each subject corresponds to one federated client. In this study, we set $K = 5$ to match the number of available participants; however, the framework naturally generalizes to any number of clients. Unless otherwise stated, all private data also utilize the waist sensor placement to ensure

controlled evaluation prior to studying cross-position generalization. Both datasets are passed through the preprocessing procedure in Section IV-A. The public dataset is split into 80% training and 20% testing, following established practices in prior studies [62], [63], whereas each private dataset is partitioned into 50% training and 50% testing. This equal split for the private client datasets is deliberately chosen to simulate a realistic scenario for wearable devices where local user data is highly limited. Such a balanced partition ensures there is sufficient data for local parameter adaptation while maintaining a substantial test set to accurately evaluate personalization performance and prevent overfitting. The testing part of $\mathcal{D}_{\text{pub}}$ is used to validate the pre-trained model's performance, while the $\mathcal{D}_k^{\text{test}}$ are used to measure performance across clients. The batch size of 128 is applied to all data.

TABLE III: Experimental setup and hyperparameters.

Parameter	**Value**
Dataset configuration	
K	5
Public data split	80% / 20%
Private data split	50% / 50%
Batch size	128
Model hyperparameters	
Hidden size	32
Optimizer	AdamW
Learning rate	5×10^{-3}
Weight decay	10^{-4}
Training protocol	
E_{cloud}	50
R	30
E_{local}	3
E_{TL}	15

We also compare the QLSTM to the LSTM counterpart. Both LSTM and QLSTM models employ a hidden size of 32 and are trained with the AdamW optimizer using a learning rate of 5×10^{-3} and weight decay 10^{-4}. During cloud pretraining, the model is trained on $\mathcal{D}_{\text{pub}}$ for 50 epochs. During federated adaptation, we perform three local epochs per round and 30 communication rounds. During transfer learning, the QLSTM/LSTM encoder is frozen, and only the classifier head is fine-tuned for 15 epochs on each client's local data. The detailed experimental setup and hyperparameter configurations are summarized in Table III. Performance is reported using accuracy on each client's test set $\mathcal{D}_k^{\text{test}}$. To isolate the contribution of each learning stage, we evaluate three schemes: (i) no federated adaptation (NoFed), (ii) federated learning without personalization (Fed), and (iii) federated learning with transfer learning personalization (Fed+TL).

1) **NoFed (Direct adaptation):** The cloud-pretrained model θ_{cloud} is evaluated on $\mathcal{D}_k^{\text{test}}$, simulating deployment of a public model without federated participation.
2) **Fed:** The global model after federated adaptation (Stage 2) is evaluated directly on D_k^{test}, isolating the benefit of collaborative learning alone.
3) **Fed + Transfer Learning:** The full pipeline in Algorithm 1, where personalization via transfer learning (client-specific fine-tuning) is applied after federated adaptation. This serves as the final personalized baseline.

B. Results

1) Selecting the optimal QLSTM cell: To determine the most suitable quantum architecture, we evaluate three QLSTM cell variants that differ in both their feature-encoding strategies and variational circuit constructions. These variants reflect increasing levels of expressivity and complexity. It is worth noting that, in all cases, the concatenated input is reduced to 10 output features before being passed to the VQCs (Eq. 6). To process these 10 output features, each variant employs 5 qubits, with each qubit responsible for encoding two classical features. The five qubit configuration utilized in our variants was explicitly chosen to balance model expressivity with practical training efficiency. While increasing the qubit count theoretically expands the quantum Hilbert space to capture more complex correlations, it exponentially increases computational costs and the risk of vanishing gradients, known as barren plateaus. Furthermore, all quantum node evaluations during training and testing were conducted using exact analytic state vector simulation. The overall structures are illustrated in Fig. 4. The three designs are summarized as follows.

- Hadamard + angle encoding (variant 1): Each qubit is initialized with a Hadamard gate (H) and encodes two classical features using fixed R_Y–R_Z rotations (as required for the 10 classical features). A shallow ZYZ-style variational layer is then applied together with an alternating CNOT–CRZ entanglement pattern. This architecture is renowned for its simplicity and hardware-friendly design.
- Simplified data re-uploading (variant 2): Data re-uploading has been shown to improve expressive power compared to standard designs where data are encoded only once before the variational circuit [64], [65]. Instead of using a single embedding stage followed by a VQC, the data re-uploading strategy defines a recurrent "re-uploading layer" composed of (i) a variational block, (ii) an entanglement pattern, and (iii) a subsequent embedding layer with trainable scaling parameters. This layer can be repeated multiple times to inject classical features into the quantum state. A final variational block is applied before measurement. In our implementation, we adopt a simplified configuration by using only one re-uploading layer. The variational block employs R_X–R_Y–R_Z rotations, while the embedding stage encodes two features per qubit through trainable, scaled R_Y and R_Z rotations. This design preserves the core advantages of data re-uploading while maintaining a lightweight and computationally manageable circuit.

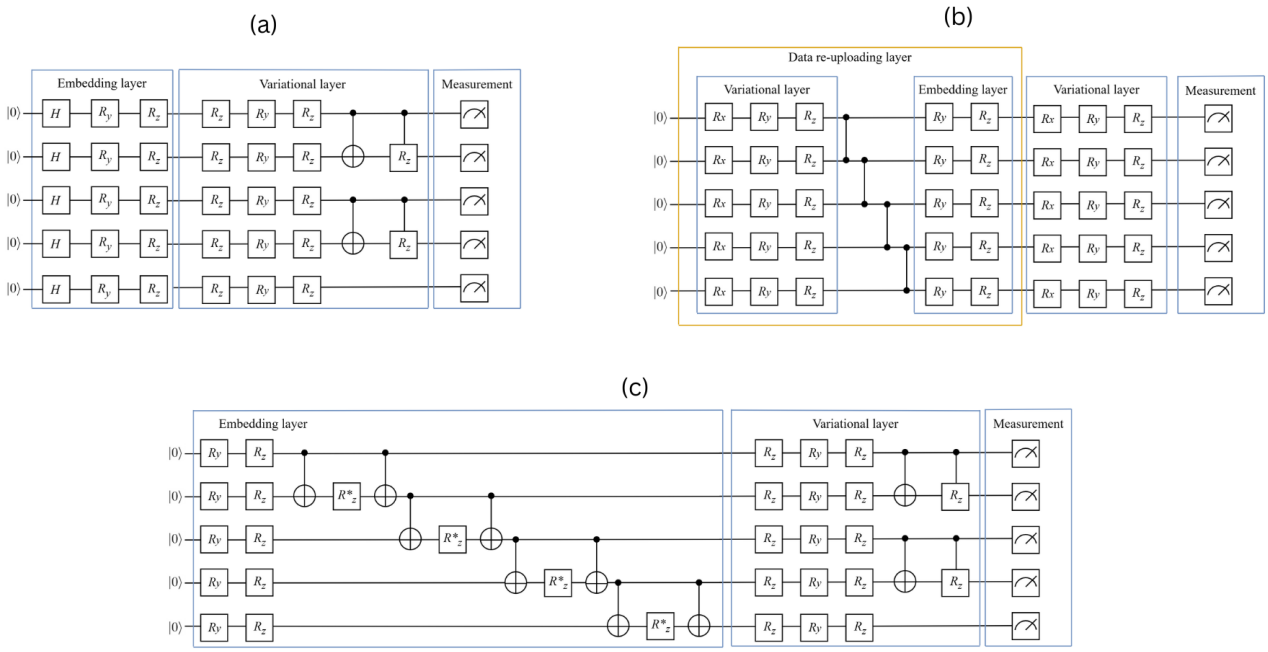

Fig. 4: Variants of quantum circuit

TABLE IV: Performance comparison of quantum variants on test data.

Model type	NoFed	Fed	Fed+TL
Variant 1	**73.78%**	86.27%	**89.89%**
Variant 2	73.31%	**87.67%**	88.41%
Variant 3	70.07%	76.08%	78.24%

- High-order encoding (Variant 3): Two features per qubit are first encoded using fixed R_Y–R_Z rotations. This is followed by a CNOT–controlled rotation–CNOT sequence, where the intermediate rotation angle is proportional to the product of adjacent features (illustrated as $R^*{}_z$), enabling higher-order feature interactions. The variational layer remains identical to variant 1, preserving a shallow ZYZ-based structure. This variant explores whether richer feature mappings combined with a deeper VQC improve representation capacity.

To quantify their internal complexity, the quantum circuits for these variants consist of 17, 30, and 17 trainable parameters, respectively. Each variant is integrated into the QLSTM cell, and the performance over all schemes is described in Table IV.

The experimental results indicate that Variant 1 consistently achieves the highest overall accuracy. Specifically, under the Fed+TL scheme, Variant 1 reaches an accuracy of 89.89%, the highest among all configurations. While Variant 2 shows slightly better performance in the Fed scheme (87.67% vs. 86.27%), Variant 1's superior performance in the most complex and high-performing scheme (Fed+TL) and strong performance across the other metrics demonstrates its robust representational capacity. Conversely, Variant 3 exhibits the lowest performance across all three metrics, suggesting that the tested high-order feature mapping did not translate into a beneficial increase in representation capacity within the QLSTM framework. Therefore, for the remainder of this paper, the variant 1 will be used, as it provides an effective balance between complexity and performance across the evaluated metrics. Moreover, Fig. 5 details the performance of Variant 1 for each individual client, revealing a consistent stepwise improvement across all five participants as the training pipeline progresses from local adaptation to federated aggregation and finally to personalization.

2) Comparison between QLSTM and classical LSTM: We next compare the proposed QLSTM encoder with a classical LSTM. The goal is to understand not only raw accuracy, but also how each model behaves across stages and clients, and what trade–offs arise in terms of model and communication efficiency.

Pretraining phase: During centralized training on $\mathcal{D}_{pub}$, the classical LSTM converges slightly faster and achieves marginally higher test accuracy than the QLSTM (91.54% vs. 85.31%). However, a deeper inspection of the optimization dynamics in Figs. 6 and 7 reveals distinct behavioral differences. First, the QLSTM loss and accuracy curves exhibit

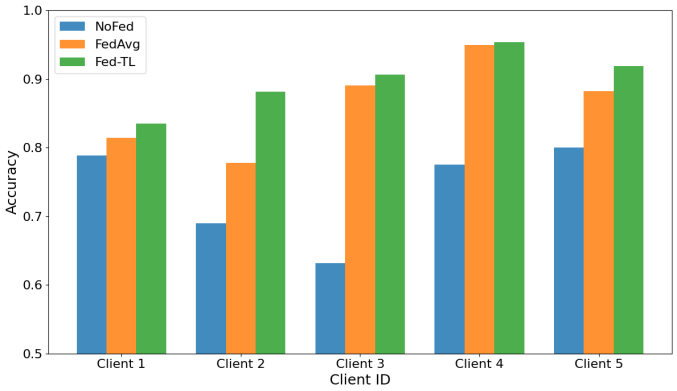

Fig. 5: Client-wise performance evolution of QLSTM variant 1 across adaptation stages.

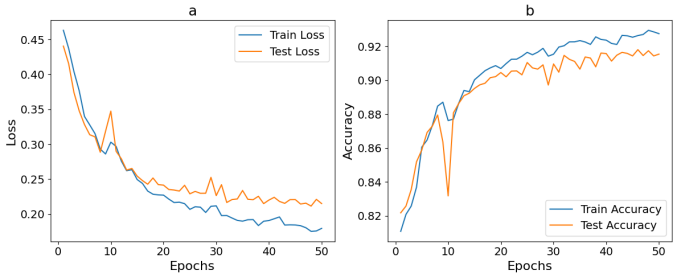

Fig. 6: Performance of LSTM model during pre-training phase. **a.** Train and test loss, **b.** Train and test accuracy.

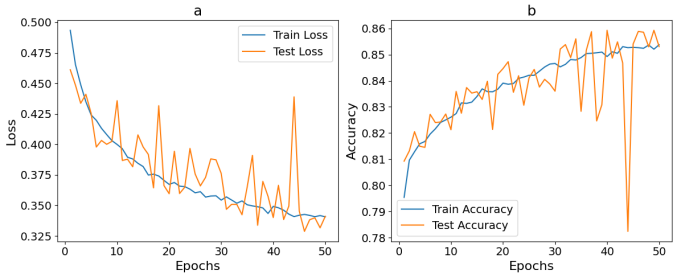

Fig. 7: Performance of QLSTM model during pre-training phase. **a.** Train and test loss, **b.** Train and test accuracy.

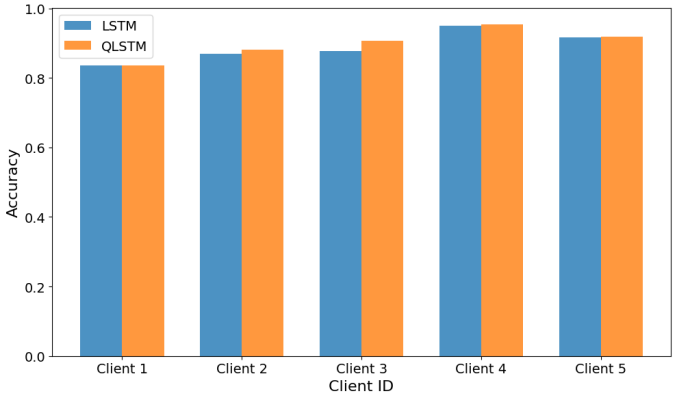

Fig. 8: Client-wise accuracy comparison: QLSTM vs LSTM.

TABLE V: Performance comparison of QLSTM vs LSTM on test data.

Model	NoFed	Fed	Fed+TL
QLSTM	**73.78%**	86.27%	**89.89%**
LSTM	49.63%	**87.4%**	88.96%

significantly higher variance compared to the smoother LSTM curves except for the significant drop around epoch 10, which likely results from the AdamW optimizer processing a noisy data batch or navigating a sharp curvature in the loss landscape). This volatility arises primarily from the probabilistic nature of quantum measurement and the complex, non-convex optimization landscape of VQCs, where gradient estimation is sensitive to stochastic noise. Second, regarding generalization, the classical LSTM displays a widening gap between training and testing performance in the later epochs, indicating the onset of overfitting. The model appears to memorize the training data, causing the testing accuracy to plateau while training loss continues to decrease. In contrast, the test loss exhibits increasing fluctuations in the later epochs because the optimizer oscillates within the highly nonconvex landscape of the VQC. However, despite these fluctuations, the QLSTM training and testing curves remain tightly coupled throughout the process. This suggests that the VQC acts as an effective structural regularizer.

Federated and personalized performance: Table V reports the average test accuracy across all clients for the three adaptation schemes. Under the NoFed setting, QLSTM clearly outperforms the classical LSTM (73.78% vs. 49.63%), demonstrating that the quantum-enhanced encoder transfers substantially better from the public distribution to private client data when no federated collaboration is available. Under the Fed setting, the performance of the two models becomes comparable (86.27% for QLSTM vs. 87.40% for LSTM), indicating that federated collaboration largely closes the performance gap observed in the NoFed scenario. In the final Fed+TL stage, QLSTM attains a slightly higher accuracy than LSTM (89.89% vs. 88.96%). Although the difference is under one percentage point, it consistently favors QLSTM, suggesting that the quantum encoder benefits from the personalized fine-tuning phase.

To better understand this behavior, Fig. 8 breaks the final performance down by client. The results show that QLSTM matches or surpasses LSTM on every client, with particularly noticeable gains on client 3. Both models achieve their highest accuracy on client 4, where the two bars nearly overlap. On the remaining clients, QLSTM consistently maintains a slight advantage, reflecting its stronger ability to adapt its representations during the personalized fine-tuning stage. This indicates that the QLSTM provides a more flexible inductive bias for capturing client–specific patterns, leading to improved or at least competitive accuracy, even when starting from a slightly weaker purely centralized pretraining performance.

TABLE VI: Complexity comparison of QLSTM and LSTM.

Model	Trainable parameters	Model size
QLSTM	**7,014**	**45KB**
LSTM	10,082	48KB

TABLE VII: Performance comparison of QLSTM and LSTM across sensor positions and adaptation schemes.

Position	Scheme	LSTM	QLSTM
Left Wrist	NoFed	51.48%	**74.12%**
	Fed	85.74%	**87.06%**
	Fed+TL	86.83%	**88.07%**
Right Wrist	NoFed	59.13%	**70.94%**
	Fed	86.05%	**87.33%**
	Fed+TL	86.95%	**88.58%**
Right Thigh	NoFed	61.02%	**76.29%**
	Fed	89.27%	**89.78%**
	Fed+TL	89.74%	**91.49%**
Neck	NoFed	54.86%	**75.73%**
	Fed	87.51%	**87.71%**
	Fed+TL	89.58%	**89.93%**
Left Ankle	NoFed	51.24%	**82.62%**
	Fed	**91.37%**	91.06%
	Fed+TL	92.11%	**92.22%**

Computational complexity comparison: Table VI reports the model complexity for both architectures. The QLSTM has 7,014 trainable parameters and a model size of 45 KB, whereas the classical LSTM uses 10,082 parameters and occupies 48 KB. This corresponds to roughly a 30% reduction in parameter count for QLSTM, despite delivering comparable or superior performance in most adaptation regimes. In FL, where model updates must be transmitted repeatedly between server and clients, this reduction translates directly into lower uplink/downlink communication cost and reduced round–trip latency. Consequently, the QLSTM offers a more communication–efficient alternative to the classical LSTM, achieving similar or better accuracy under all three adaptation schemes. Despite these advantages in communication efficiency, it is crucial to comprehensively evaluate the computational overhead required for wearable IoT environments. A distinct trade-off exists between communication costs and local processing demands. During the federated training phase, updating the quantum parameters is highly resource-intensive. It demands significantly greater local memory and processing power than a classical LSTM to execute the complex matrix multiplications underlying quantum state evolution. Similarly, the inference complexity is also elevated, as the edge device must continuously simulate the quantum node evaluations for every forward pass. Consequently, it currently imposes a heavier computational burden on the local edge device for both training and inference execution.

3) Cross-position generalization across sensor locations: To evaluate robustness under realistic deployment conditions, we assess whether the proposed QLSTM can generalize across heterogeneous sensor placements. While the cloud-pretraining stage uses only waist-mounted recordings, the private datasets include five alternative body locations: left wrist, right wrist, right thigh, neck, and left ankle. Table VII reorganizes the cross-position results into a row-wise format, where each sensor position is expanded into three evaluation schemes (NoFed, Fed, and Fed+TL), and accuracies of LSTM and QLSTM are reported side by side.

The QLSTM consistently and significantly outperforms the classical LSTM in the NoFed scheme across all five sensor positions. The largest performance gain is observed at the left ankle, where QLSTM achieves 82.62% accuracy compared to the LSTM's 51.24% (a difference of over 31 percentage points). This demonstrates that the quantum feature mapping learned during the initial cloud-pretraining stage provides a markedly richer and more generalized representation of the human activity data compared to the classical LSTM cell. This inherent generalization capability is a strong indicator of the QLSTM's suitability for real-world scenarios where fine-tuning data may be scarce.

Both the classical LSTM and the QLSTM exhibit substantial performance improvements when adapting via Fed and Fed+TL. Under these adaptive schemes, the models achieve high accuracies, mostly ranging between 85% and 93%. This confirms that the FL framework is highly effective at customizing the global model to account for position-specific data drift (covariate shift).

While the gap narrows in the adaptive schemes, the QLSTM model still demonstrates a slight, yet consistent, edge in the most complex and fine-tuned scenario, Fed+TL, in four out of five positions (left wrist, right wrist, right thigh, and neck). The right thigh position yields the highest overall accuracy for the QLSTM at 91.49% (vs. 89.74% for LSTM). This indicates that the QLSTM's hybrid structure not only provides better initial generalization but also integrates more effectively with additional fine-tuning data when transfer learning is applied, resulting in marginally superior final performance across most sensor placements. The only exception is the Fed scheme for the left ankle, where the LSTM slightly outperforms the QLSTM (91.37% vs. 91.06%).

VI. Conclusion

This work addresses a practical limitation of wearable fall detection, as models that perform well on a single dataset often degrade when applied to new users or sensor placements. We proposed a three-stage pipeline, i.e., cloud pretraining, federated adaptation, and client transfer learning, and implemented it with a QLSTM encoder. We also collected a new dataset that supports this setting, with multiple sensor locations and clear temporal fall annotations, and aligned labels to common public datasets to make comparisons easier for

federated transfer learning experiments. The findings show that FL plus client fine-tuning improves accuracy for both LSTM and QLSTM. The main difference appears when no federated adaptation is available. The QLSTM transfers better from the public pretraining distribution to private client data (73.78% vs. 49.63%). After federated adaptation, the two models become comparable, and after transfer learning, QLSTM achieves slightly higher accuracy on average (89.89% vs. 88.96%). In cross-position evaluation, QLSTM achieves large gains over LSTM in the NoFed setting for every tested location, and it remains marginally better in most Fed+TL cases. Finally, QLSTM achieves these results with fewer parameters (about 30% less), which is useful in federated settings where model updates must be transmitted repeatedly. Regarding practical feasibility, it is crucial to acknowledge the computational complexities associated with deploying quantum-based models, given current hardware limitations. While the proposed QLSTM significantly reduces the trainable parameter count to optimize federated communication, the local execution of VQCs introduces a distinct computational overhead. In the current noisy intermediate-scale quantum era, physical wearable devices cannot execute quantum gates natively. Consequently, while the representational and communication advantages of the QLSTM are evident, achieving seamless on-device deployment will require continued advancements in quantum hardware downsizing and more efficient classical simulation algorithms.

This study also identifies several directions that future work can address. Firstly, our dataset improves realism by adding context-defined fall styles and precise timing labels, but it is still collected in a controlled setting. A key next step is the evaluation on naturalistic fall recordings, especially falls from older adults, possibly using FARSEEING or a similar dataset. Second, while our experimental setup utilized five clients to match the collected dataset, the number of participants plays a critical role in federated learning dynamics. Increasing the participant count would theoretically improve the global model generalization by exposing the federated aggregation process to a wider diversity of human movement patterns. Moreover, we studied multiple body locations, but real-world heterogeneity also includes different IMU models, sampling rates, sensor ranges, and orientation noise. Future work should include cross-device and cross-sampling-rate experiments and evaluate strategies that handle these differences in the federated transfer learning setting. Furhtermore, freezing the encoder and fine-tuning a small head is simple and resource-friendly, but it may not be optimal for all clients. Future work can also explore stronger personalized FL methods, such as partial-layer personalization, client clustering, mixture-of-experts, or meta-learning-style initialization. Another useful extension is continual learning, so the model can adapt as a user's gait changes over weeks or months without catastrophic forgetting. Finally, we acknowledge the growing prominence of advanced architectures in recent literature, particularly CNN-LSTM networks and Transformer-based models. Exploring the quantum adaptations of these sophisticated architectures represents a highly promising direction for our future research.

ACKNOWLEDGMENTS

This work was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109 and in part by the NSERC CREATE program (Grant number 596205-2025).

REFERENCES

[1] R. Kakara, "Nonfatal and fatal falls among adults aged over 65 years—united states, 2020–2021," *MMWR Morb Mortal Wkly Rep.*, vol. 72, Sep. 2023.

[2] World Health Organization. (2021) Falls. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls

[3] A. Klak *et al.*, "A growing problem of falls in the aging population: A case study on poland – 2015–2050 forecast," *Eur. Geriatr. Med.*, vol. 8, no. 2, pp. 105–110, Apr. 2017.

[4] Z. Jiang, M. A. A. Al-Qaness, D. Al-Alimi, A. A. Ewees, M. Abd Elaziz, A. Dahou, and A. M. Helmi, "Fall detection systems for internet of medical things based on wearable sensors: A review," *IEEE Internet Things J.*, vol. 11, no. 21, pp. 34 797–34 810, Jul. 2024.

[5] N. T. Newaz and E. Hanada, "The methods of fall detection: A literature review," *Sensors*, vol. 23, no. 11, p. 5212, May 2023.

[6] M. Schneider, K. Seeser-Reich, A. Fiedler, and U. Frese, "Enhancing slip, trip, and fall prevention: Real-world near-fall detection with advanced machine learning technique," *Sensors*, vol. 25, no. 5, p. 1468, Feb. 2025.

[7] M. W. Creaby and M. H. Cole, "Gait characteristics and falls in parkinson's disease: A systematic review and meta-analysis," *Park. Relat. Disord.*, vol. 57, pp. 1–8, Dec. 2018.

[8] J. Klenk, L. Schwickert, L. Palmerini, S. Mellone, A. Bourke, E. A. Ihlen, N. Kerse, K. Hauer, M. Pijnappels, M. Synofzik *et al.*, "The farseeing real-world fall repository: a large-scale collaborative database to collect and share sensor signals from real-world falls," *Eur. Rev. Aging Phys. Act.*, vol. 13, no. 1, p. 8, Oct. 2016.

[9] H. Chiba, S. Ebihara, N. Tomita, H. Sasaki, and J. P. Butler, "Differential gait kinematics between fallers and non-fallers in community-dwelling elderly people," *Geriatr. Gerontol Int.*, vol. 5, no. 2, pp. 127–134, May 2005.

[10] Y. Yang, L. Chen, and F. T. Filippidis, "Accelerometer-measured physical activity, frailty, and all-cause mortality and life expectancy among middle-aged and older adults: a uk biobank longitudinal study," *BMC Med.*, vol. 23, no. 1, p. 125, Feb. 2025.

[11] R. Igual, C. Medrano, and I. Plaza, "Challenges, issues and trends in fall detection systems," *Biomed. Eng. Online*, vol. 12, no. 1, p. 66, Jul. 2013.

[12] J. A. Santoyo-Ramón, E. Casilari-Pérez, and J. M. Cano-García, "A study on the impact of the users' characteristics on the performance of wearable fall detection systems," *Sci. Rep.*, vol. 11, no. 1, p. 23011, Nov. 2021.

[13] Y.-C. Tu, C.-Y. Lin, C.-P. Liu, and C.-T. Chan, "Performance analysis of data augmentation approaches for improving wrist-based fall detection system," *Sensors*, vol. 25, no. 7, p. 2168, Mar. 2025.

[14] A. K. Bashir, N. Victor, S. Bhattacharya, T. Huynh-The, R. Chengoden, G. Yenduri, P. K. R. Maddikunta, Q.-V. Pham, T. R. Gadekallu, and M. Liyanage, "Federated learning for the healthcare metaverse: Concepts, applications, challenges, and future directions," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 21 873–21 891, Aug. 2023.

[15] C. A. Silva, E. Casilari, and R. García-Bermúdez, "Cross-dataset evaluation of wearable fall detection systems using data from real falls and long-term monitoring of daily life," *Measurement*, vol. 235, p. 114992, Aug. 2024.

[16] X. Han, Q. Zhang, Z. He, and Z. Cai, "Confidence-based similarity-aware personalized federated learning for autonomous iot," *IEEE Internet Things J.*, vol. 11, no. 7, pp. 13 070–13 081, Nov. 2023.

[17] B. Barshan and M. Ş. Turan, "A novel heuristic fall-detection algorithm based on double thresholding, fuzzy logic, and wearable motion sensor data," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 17 797–17 812, May 2023.

[18] S. Mobsite, N. Alaoui, M. Boulmalf, and M. Ghogho, "A privacy-preserving AIoT framework for fall detection and classification using hierarchical learning with multilevel feature fusion," *IEEE Internet Things J.*, vol. 11, no. 15, pp. 26 531–26 547, May 2024.

[19] F. Naz, M. Fahim, A. A. Cheema, B. D. E. McNiven, T.-V. Cao, R. Hunter, and T. Q. Duong, "Air quality and healthy ageing: Predictive modelling of pollutants using cnn quantum-lstm," *IEEE Access*, pp. 1–1, May 2025.

[20] B.-N. D. Tran, M. Fahim, B. D. E. McNiven, M. Guizani, H. Shin, and T. Q. Duong, "Quantum lstm model for estimation of energy expenditure in human aging using wearable IoT healthcare technology," *IEEE Internet Things J.*, vol. 12, no. 22, pp. 46 051–46 064, Apr. 2025.

[21] U. Ullah and B. Garcia-Zapirain, "Quantum machine learning revolution in healthcare: A systematic review of emerging perspectives and applications," *IEEE Access*, vol. 12, pp. 11 423–11 450, Jan. 2024.

[22] P. Liao, X. Wang, Y. Shan, L. An, and S. Mao, "Wireless sensing in artificial intelligence of things: A general quantum machine learning framework," *IEEE Netw.*, vol. 39, no. 3, pp. 207–214, Jan. 2025.

[23] S. Prabhu, S. Gupta, G. M. Prabhu, A. V. Dhanuka, and K. V. Bhat, "Qucardio: Application of quantum machine learning for detection of cardiovascular diseases," *IEEE Access*, vol. 11, pp. 136 122–136 135, Nov. 2023.

[24] Z. Qu, X. Zhao, L. Sun, and G. Muhammad, "DAQFL: Dynamic aggregation quantum federated learning algorithm for intelligent diagnosis in internet of medical things," *IEEE Internet Things J.*, vol. 12, no. 19, pp. 39 313–39 325, Jan. 2025.

[25] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017.

[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, May 2019.

[27] H. Jung, B. Koo, J. Kim, T. Kim, Y. Nam, and Y. Kim, "Enhanced algorithm for the detection of preimpact fall for wearable airbags," *Sensors*, vol. 20, no. 5, p. 1277, Feb. 2020.

[28] Z. Qian, Y. Lin, W. Jing, Z. Ma, H. Liu, R. Yin, Z. Li, Z. Bi, and W. Zhang, "Development of a real-time wearable fall detection system in the context of internet of things," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21 999–22 007, Jun. 2022.

[29] B. Barshan and M. S. Turan, "A novel heuristic fall-detection algorithm based on double thresholding, fuzzy logic, and wearable motion sensor data," *IEEE Internet Things J.*, vol. 10, no. 20, pp. 17 797–17 812, May 2023.

[30] F. A. S. Ferreira de Sousa, C. Escriba, E. G. Avina Bravo, V. Brossa, J.-Y. Fourniols, and C. Rossi, "Wearable pre-impact fall detection system based on 3d accelerometer and subject's height," *IEEE Sensors J.*, vol. 22, no. 2, pp. 1738–1745, Nov. 2021.

[31] S. Yu, H. Chen, and R. A. Brown, "Hidden markov model-based fall detection with motion sensor orientation calibration: A case for real-life home monitoring," *IEEE J. Biomed. Health Inform.*, vol. 22, no. 6, pp. 1847–1853, Dec. 2017.

[32] K.-C. Liu, C.-Y. Hsieh, S. J.-P. Hsu, and C.-T. Chan, "Impact of sampling rate on wearable-based fall detection systems based on machine learning models," *IEEE Sensors J.*, vol. 18, no. 23, pp. 9882–9890, Oct. 2018.

[33] 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. Dev. Syst.*, vol. 14, no. 3, pp. 1270–1281, Sep. 2021.

[34] K.-C. Liu, C.-Y. Hsieh, H.-Y. Huang, S. J.-P. 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, Nov. 2019.

[35] A. Shahzad and K. Kim, "Falldroid: An automated smart-phone-based fall detection system using multiple kernel learning," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 35–44, Jan. 2019.

[36] Y.-H. Nho, J. G. Lim, and D.-S. Kwon, "Cluster-analysis-based user-adaptive fall detection using fusion of heart rate sensor and accelerometer in a wearable device," *IEEE Access*, vol. 8, pp. 40 389–40 401, Jan. 2020.

[37] G. Wang, Q. Li, L. Wang, Y. Zhang, and Z. Liu, "CMFALL: A cascade and parallel multi-state fall detection algorithm using waist-mounted tri-axial accelerometer signals," *IEEE Trans. Consum. Electron.*, vol. 66, no. 3, pp. 261–270, Aug. 2020.

[38] L. Wang, M. Peng, and Q. Zhou, "Pre-impact fall detection based on multi-source cnn ensemble," *IEEE Sensors J.*, vol. 20, no. 10, pp. 5442–5451, Jan. 2020.

[39] C. Tunca, G. Salur, and C. Ersoy, "Deep learning for fall risk assessment with inertial sensors: Utilizing domain knowledge in spatio-temporal gait parameters," *IEEE J. Biomed. Health Inform.*, vol. 24, no. 7, pp. 1994–2005, Jul. 2020.

[40] A. Choi, T. H. Kim, O. Yuhai, S. Jeong, K. Kim, H. Kim, and J. H. Mun, "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, Sep. 2022.

[41] M. M. Kabir, J. Shin, and M. F. Mridha, "Secure your steps: A class-based ensemble framework for real-time fall detection using deep neural networks," *IEEE Access*, vol. 11, pp. 64 097–64 113, Jun. 2023.

[42] S. Campanella, A. Alnasef, L. Falaschetti, A. Belli, P. Pierleoni, and L. Palma, "A novel embedded deep learning wearable sensor for fall detection," *IEEE Sensors J.*, vol. 24, no. 9, pp. 15 219–15 229, May 2024.

[43] 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.

[44] J. Tang, B. He, J. Xu, T. Tan, Z. Wang, Y. Zhou, and S. Jiang, "Synthetic imu datasets and protocols can simplify fall detection experiments and optimize sensor configuration," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 32, pp. 1233–1245, Mar. 2024.

[45] Z. Yu, J. Liu, M. Yang, Y. Cheng, J. Hu, and X. Li, "An elderly fall detection method based on federated learning and extreme learning machine (Fed-ELM)," *IEEE Access*, vol. 10, pp. 130 816–130 824, Dec. 2022.

[46] Y. Chen, X. Qin, J. Wang, C. Yu, and W. Gao, "Fedhealth: A federated transfer learning framework for wearable healthcare," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 83–93, Apr. 2020.

[47] G. Wang, Y. Zhang, C. Ying, Q. Zhang, Z. Xiong, J. Wang, and G. Yu, "MMFed: A multimodal federated learning framework for heterogeneous devices," *IEEE Internet Things J.*, vol. 12, no. 18, pp. 36 893–36 907, Sep. 2025.

[48] S. V. Babu, P. Ramya, and J. Gracewell, "Revolutionizing heart disease prediction with quantum-enhanced machine learning," *Sci. Rep.*, vol. 14, no. 1, p. 7453, Mar. 2024.

[49] K. K. Makam, V. K. Singh, and R. B. Pachori, "Eye movement detection based on sm-ssa and quantum cnn from emg of eom signals," *IEEE Sensors Lett.*, vol. 9, no. 5, pp. 1–4, May 2025.

[50] C. Ren, R. Yan, H. Zhu, H. Yu, M. Xu, Y. Shen, Y. Xu, M. Xiao, Z. Y. Dong, M. Skoglund, D. Niyato, and L. C. Kwek, "Toward quantum federated learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 9, pp. 15 580–15 600, Sep. 2025.

[51] C. Qiao, M. Li, Y. Liu, and Z. Tian, "Transitioning from federated learning to quantum federated learning in internet of things: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 27, no. 1, pp. 509–545, Feb. 2025.

[52] A. Mathur, A. Gupta, and S. K. Das, "When federated learning meets quantum computing: Survey and research opportunities," *IEEE Commun. Surveys Tuts.*, vol. 28, pp. 1351–1380, Nov. 2025.

[53] A. Sucerquia, J. D. López, and J. F. Vargas-Bonilla, "Sisfall: A fall and movement dataset," *Sensors*, vol. 17, no. 1, p. 198, Jan. 2017.

[54] D. Micucci, M. Mobilio, and P. Napoletano, "Unimib shar: A dataset for human activity recognition using acceleration data from smartphones," *Appl. Sci.*, vol. 7, no. 10, p. 1101, Oct. 2017.

[55] G. Vavoulas, C. Chatzaki, T. Malliotakis, M. Pediaditis, and M. Tsiknakis, "The MobiAct Dataset: Recognition of Activities of Daily Living using Smartphones," in *Proc. Int. Conf. Inform. Commun. Tech. Ageing Well and e-Health*, Rome, Italy, Apr. 2016, pp. 143–151.

[56] E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "UMAFall: A Multisensor dataset for the research on automatic fall Detection," *Procedia Computer Science*, vol. 110, pp. 32–39, Jul. 2017.

[57] M. Saleh, M. Abbas, and R. L. B. 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, Aug. 2021.

[58] 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," *Front. Aging Neurosci.*, vol. 13, p. 692865, Jul. 2021.

[59] H. Ponce, L. Martínez-Villaseñor, and J. Nuñez-Martínez, "Sensor location analysis and minimal deployment for fall detection system," *IEEE Access*, vol. 8, pp. 166 678–166 691, Sep 2020.

[60] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist. (AISTATS)*, FL, USA, Apr. 2017, pp. 1273–1282.

[61] K. Greff, R. K. Srivastava., J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[62] 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. 22 943–22 951, Dec. 2022.

[63] S. Campanella, A. Alnasef, L. Falaschetti, A. Belli, P. Pierleoni, and L. Palma, "A novel embedded deep learning wearable sensor for fall detection," *IEEE Sens. J.*, vol. 24, no. 9, pp. 15 219–15 229, May 2024.

[64] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, Feb. 2020.

[65] M. Schuld, R. Sweke, and J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," *Phys. Rev. A*, vol. 103, no. 3, p. 032430, Mar. 2021.