Sensors and Materials, Vol. 38, No. 11 (2026) 6471–6494
MYU Tokyo

S & M 4691

AI-driven Signal Processing and System Integration for Automated Multi-user Behavioral Measurement in Smart Environments

Muhammad Hanzla,[1] Dina Abdulaziz AlHammadi,[2] Khaled Alnowaiser,[3] Ahmad Jalal,[1,4*] and Hui Liu[5**]

[1]Faculty of Computing and AI, Air University, E-9, Islamabad 44000, Pakistan
[2]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[3]Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[4]Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul 02841, South Korea
[5]Guodian Nanjing Automation Co., Ltd., Nanjing, China

(Received April 18, 2026; accepted July 22, 2026)

Keywords: sensor signal processing, system integration, AI applications in signal processing, automated behavioral measurement, Industry 4.0, multi-user activity recognition

With the evolution of Industry 4.0 and intelligent environments, efficient signal collection and processing are crucial to ensure system integration as automation systems are put at the heart of the process. In this study, we present a novel AI-based system to measure and authenticate behavioral signals and apply it to accelerator data obtained from a wrist-worn sensor. To counter the intricacies of nonintrusive monitoring, the proposed system integration employs variational mode decomposition (VMD) to refine the signal at multiple levels and Blackman-window segmentation to provide time consistency. The architecture uses a new AI application in signal processing through the adaptive combination of multimodal features such as gammatone cepstral and temporal moments using localized multiple kernel learning (LKML). Quantum-inspired Harris Hawk optimization is further used to optimize this process, and dynamic time warping aligns it. To attain the final classification in the automation loop, graph attention networks and temporal convolutional networks (TCNs) are merged to represent complex motion signatures. The framework is tested on the datasets Capture-24 and CAHAR with accuracies of 91.74 and 88.93%, respectively. The findings show considerable resistance to noise and time-varying effects, which contributes to the development of integrated engineering and the safe signal-based automation of smart infrastructures.

1. Introduction

Wearable devices, particularly smartwatches equipped with inertial sensors,[1,2] have transformed physical activity monitoring by enabling real-time, noninvasive motion tracking for

*Corresponding author: e-mail: ahmadjalal@mail.au.edu.pk
**Corresponding author: e-mail: hui.liu@uni-bremen.de
https://doi.org/10.18494/SAM6380

ISSN 0914-4935 © MYU K.K.

ubiquitous computing, workplace safety, and smart-city automation applications.[1] Beyond personal monitoring, these devices support remote sensing environments and facilitate the analysis of complex behavioral patterns. Despite advances in sensor signal processing, the reliable recognition of complex multi-user activities remains challenging owing to intersubject variability, sensor-placement differences, environmental noise, and asynchronous activity streams.[3] These challenges are particularly critical in Industry 4.0 environments, where robust and automated behavioral measurement across heterogeneous populations is essential.[4] Traditional methods relying on fixed segmentation, handcrafted features, and synchronized data assumptions often struggle in multiperson scenarios characterized by overlapping activities and temporal inconsistencies.

To address these limitations, we propose an end-to-end framework integrating variational mode decomposition (VMD)-based signal denoising, a locality-aware multi-kernel fusion module (LMKL) for adaptive fusion, Blackman-window segmentation, gammatone frequency cepstral (GFC) features, temporal moments, generalized cross-correlation (GCC) descriptors, Quantum-inspired Harris Hawks Optimization (QHHO) for feature optimization, dynamic time warping (DTW) temporal alignment, graph attention network (GAT)-based interuser modeling, and temporal conventional network (TCN) classification. Evaluations on the datasets Capture-24 and CAHAR demonstrate improved accuracy, robustness, and interpretability for automated multi-user activity recognition. The main contributions of the work are as follows:

- a modular, multistage architecture tailored to concurrent, multiperson automated measurement from wrist-mounted sensors,
- a novel, LMKL that adaptively learns kernel weights per temporal neighborhood to robustly fuse heterogeneous sensor representations,
- the system integration of VMD, Blackman-window segmentation, GFC/GCC/temporal descriptors, QHHO-based feature optimization, DTW alignment, and GAT-based relational embedding to process complex signal semantics, and
- comprehensive evaluation on datasets Capture-24 and CAHAR, demonstrating improved accuracy, robustness to temporal shifts and noise, and applicability to occupational monitoring and intelligent urban systems.

The remainder of this paper is organized as follows. Section 2 comprises a review of the relevant literature. In Sect. 3, we present the proposed integrated system architecture in detail. The experimental setup and results are presented in Sect. 4. In Sect. 5, we discuss the findings and their implications. Finally, in Sect. 6, we present our condisions and outline directions for future research.

2. Literature Review

Human activity recognition (HAR) using body-worn sensors is increasingly adopted in Industry 4.0 and smart environments. However, intersubject variability, sensor noise, and multi-user interactions remain challenging. Recent AI-based approaches integrate signal decomposition, feature fusion, temporal alignment, and graph modeling to enable more robust and intelligent activity recognition.

2.1 Machine-learning-based HAR approaches

Wang *et al.*[5] pioneered probabilistic multi-user HAR using coupled hidden Markov models (HMMs) and factorial conditional random fields to model temporal dependences and user interactions. B. Vidya and P. Sasikumar[6] demonstrated the effectiveness of multimodal feature fusion through discrete wavelet transform (DWT) and empirical mode decomposition (EMD)-based frameworks with support vector machine (SVM) and k-neural network (k-NN) classifiers. Gomez-Ramos *et al.*[7] and Wang and Cook[8] explored unsupervised HAR using synchronized HMMs and Gaussian-mixture filtering, respectively, for activity and occupant inference. Willetts *et al.*[9] achieved strong performance on Capture-24 using gradient boosting with engineered features, whereas Twomey *et al.*[10] employed Bayesian nonparametric clustering for unknown activity discovery. Ellis *et al.*[11] combined random forests and HMMs for interpretable activity classification, whereas Hammerla *et al.*[12] introduced deep belief networks to obtain HAR directly from raw accelerometer signals without manual feature engineering.

2.2 Deep-learning-based HAR approaches

Chen *et al.*[13] proposed MG-WHAR, a graph convolutional framework that models body location, modality, and sensor relationships for wearable HAR. Ordóñez and Roggen[14] introduced DeepConvLSTM, combining convolutional neural networks (CNNs) and long short-term memory (LSTM) for spatiotemporal feature learning. Dirgová-Luptáková *et al.*[15] demonstrated the effectiveness of Transformer-based modeling for wrist accelerometer data. Rivas-Caicedo *et al.*[16] developed a distributed neural architecture for efficient wearable HAR, whereas Zeng *et al.*[17] proposed a lightweight attention-based CNN for resource-constrained devices. Bai *et al.*[18] used multiscale dilated TCNs to improve transitional activity recognition, and Vaizman *et al.*[19] showed that integrating contextual and accelerometer data through an RNN significantly enhances activity recognition performance.

3. Materials and Methods

The proposed framework employs VMD to denoise wrist-worn accelerometer signals. LMKL fuses GFC, temporal moment, and GCC features, whereas Blackman-window segmentation preserves temporal consistency. QHHO selects discriminative features and DTW aligns multi-user activity streams. Finally, GATs model interuser relationships, and a TCN performs activity classification, as illustrated in Fig. 1.

3.1 Preprocessing via VMD

VMD is utilized as the main preprocessing step to guarantee sound downstream analysis and maintain important motion signatures. Unlike traditional denoising approaches such as low-pass filtering and wavelet thresholding, VMD can adaptively break down the raw signal from the wrist-mounted accelerometer into a finite number of compactly represented band-limited intrinsic mode functions.[20,21] The ability to isolate motion-relevant oscillatory components of

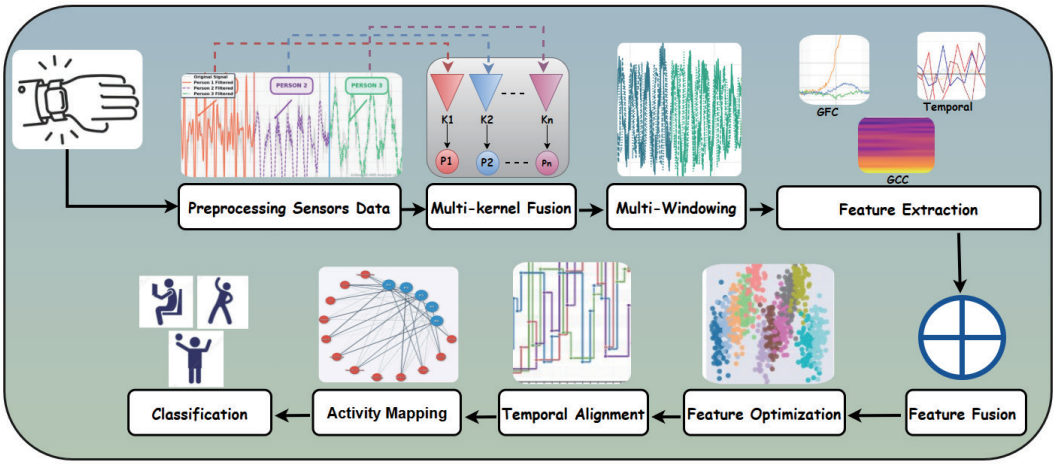


Fig. 1. (Color online) Integrated system architecture for automated multi-user signal processing, incorporating signal decomposition and adaptive feature fusion. Temporal alignment and GAT-based modeling facilitate robust integration for intelligent automation applications.

the signal without losing temporal fidelity is made possible by this capability. VMD presents the problem of decomposition as a constrained variational optimization problem, where one seeks to minimize the collective bandwidth of the extracted modes, and has the condition that the sum of the extracted modes should reconstruct the original signal. The formulation of the optimization is

$$\min_{\{u_k, w_k\}} \sum_{k=1}^{K} \left\| \left(u_k(t) \cdot e^{-j\omega k t} \right) \right\|_2^2 \quad \text{s.t.:} \quad \sum_{k=1}^{K} u_k(t) = f(t). \tag{1}$$

Here, $u_k(t)$ denotes the k-th mode, ω_k its center frequency, and $f(t)$ the input signal. The formulation ensures compact, nonredundant modes whose sum reconstructs the original signal, with center frequencies iteratively adapted to wrist-motion patterns. VMD offers superior spectral selectivity over EMD, DWT, and STFT by avoiding mode mixing and fixed basis limitations. With $K = 5$ modes, it achieved a higher signal-to-noise ratio (SNR; ≈4.2 dB over DWT and 3.1 dB over EMD on Capture-24), improving feature discriminability. The output of the preprocessing is depicted in Fig. 2.

3.2 Multi-kernel fusion

Modeling heterogeneous motion dynamics in multi-user activity recognition requires an adaptive fusion strategy capable of handling temporal inconsistencies, nonlinear motion patterns, and intersubject variability. To address this challenge, we employ LMKL, which extends conventional multiple kernel learning (MKL) by introducing input-dependent kernel weighting through a gating function.[22] Unlike standard MKL, which learns a single globally fixed combination of kernels, LMKL dynamically adjusts kernel contributions for each input

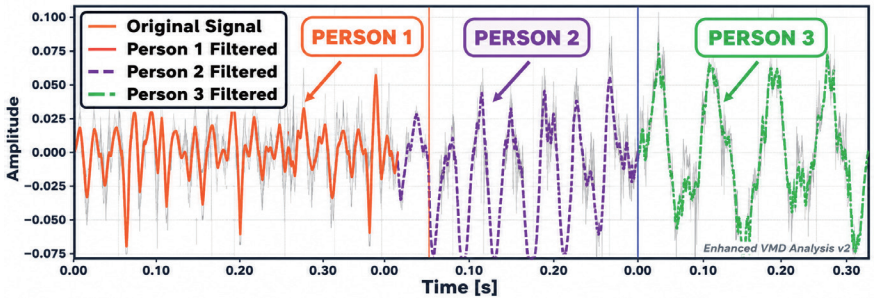


Fig. 2. (Color online) Preprocessing result: original vs VMD-denoised accelerometer data across multiple participants.

sample, enabling localized adaptation to user- and activity-specific motion characteristics. In conventional MKL, the composite kernel is formulated as

$$K(x,x') = \sum_m \beta_m k_m(x,x'). \tag{2}$$

Fixed kernel weights are ineffective for multi-user activity recognition owing to varying motion patterns. The radial basis function (RBF) kernel captures smooth motions, whereas the wavelet kernel models transient gestures. We use RBF, polynomial (degree 3), wavelet, and chi-square kernels, tuned via fivefold cross-validation. LMKL uses a two-layer softmax gating network to generate adaptive, sample-wise weights for optimal kernel fusion per feature frame:

$$f(x) = \sum_{i=1}^{N} \sum_{m=1}^{4} g_m(x;\theta)\alpha_i k_m(x,x_i), \tag{3}$$

where k_m denotes the m-th kernel, $g_m(x;\theta)$ is the gating function producing input-dependent weights, and α_i are classifier coefficients. The adaptive behavior under heterogeneous sensor distributions emerges naturally from this per-sample weighting mechanism. Each participant contributes a distinct marginal feature distribution $P_i(x)$ because of individual biomechanical differences, sensor placement variability, and activity-specific motion profiles. Since the gating network processes each feature vector independently, it learns to map these distributional variations into distinct kernel weight profiles without requiring separate user-specific models or explicit domain adaptation strategies. The joint optimization of kernel weights and classifier parameters is achieved by minimizing

$$\min_{\alpha,\theta} \frac{1}{2}\|\alpha\|^2 + C\sum_i l\left(y_i, \sum_m g_m(x_i)\sum_j \alpha_j k_m(x_i,x_j)\right). \tag{4}$$

Here, C controls the regularization–loss trade-off and $l(\cdot)$ denotes hinge loss. The index j in the inner summation ranges over the training samples, so that each sample i is compared against every other sample j through the kernel value $k_m(x_i,x_j)$. The model jointly learns gating

parameters θ and classifier coefficients α to maximize the classification margin, ensuring that feature space partitioning is driven by discriminative utility rather than input proximity. This enables per-sample kernel adaptation, capturing both persistent and transient motion patterns while reducing overfitting. Overall, LMKL provides a robust, context-aware fusion mechanism that improves downstream classification in multi-user activity recognition.

3.2.1 Kernel-specific justification and adaptive fusion strategy

The proposed LMKL framework employs four complementary kernels to model heterogeneous motion dynamics. The Gaussian RBF kernel captures nonlinear activity patterns, the polynomial kernel models higher-order feature interactions, the wavelet kernel represents transient and nonstationary motions, and the chi-square kernel handles distribution-based descriptors while remaining robust to motion intensity variations. An LMKL gating network adaptively assigns sample-specific kernel weights, enabling context-aware feature fusion for each activity instance. The output of the LMKL is depicted in Fig. 3.

3.3 Multiwindowing/segmentation

To extract coherent activity bursts among continuous sensor streams, we use an adaptive multiresolution segmentation algorithm with the Blackman–Tukey window.[23] This method, compared with fixed or single-scale windowing, uses spectral smoothing with resolution flexibility, minimizing leakage with localized transitions in motion data.[24] This works especially well in multi-user situations where feature quality is skewed by overlapping actions and uneven time intervals. On a time-series signal $s(t)$, the adaptive window $w_a(n)$ can be denoted as

$$w_a(n) = \sum_{k=-M}^{M} c_k \left[0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right) \right] e^{\frac{j2\pi kn}{N}}. \quad (5)$$

Here, c_k are adaptive weights, N is the window length, and M is the resolution bandwidth. A trade-off exists between temporal accuracy and frequency stability, where smaller N improves temporal precision but reduces frequency stability, while larger N improves frequency stability but reduces temporal resolution; M controls the spectral resolution of this balance. Blackman windows with adaptive bandwidth overlap to capture both short- and long-term patterns of activity whilst maintaining transitions including movement initiation and user handover. Its high side-lobe attenuation (approximately −73 dB) minimizes spectral leakage, resulting in boundaries of cleaner quality and more dependable features, particularly in asynchronous multi-user applications. Adaptive bandwidth also allows small gestures to be resolved and larger average activities to be sustained. On CAHAR, the boundary misclassification with Blackman replaced with a rectangular window was 6.3 times more effective. The output of windowing can be seen in Fig. 4.

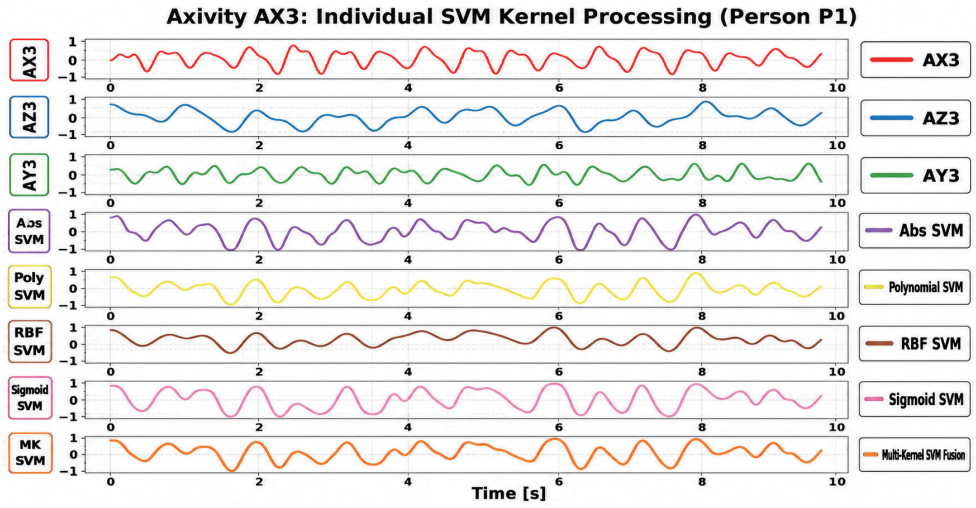


Fig. 3. (Color online) LMKL-based fusion of heterogeneous wrist-worn accelerometer features using adaptive kernel weighting.

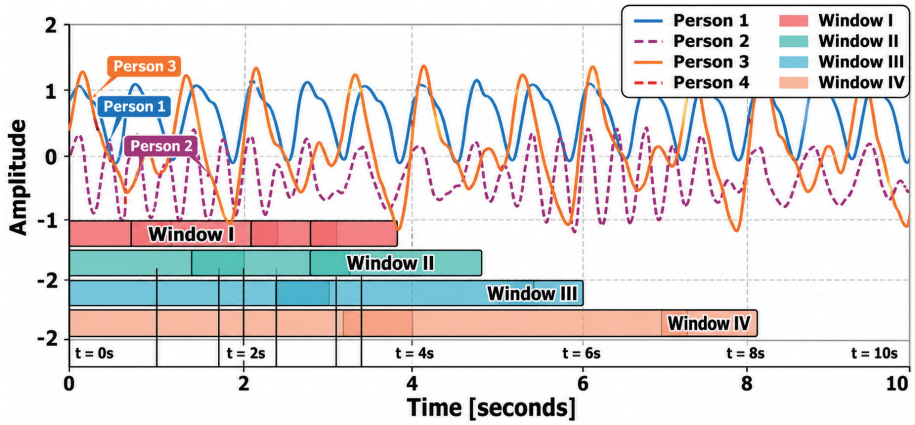


Fig. 4. (Color online) Multiwindow segmentation for time-series activity recognition analysis.

3.4 Feature extraction

Feature extraction converts preprocessed signals into a compact representation for multi-user activity modeling. We extract GFC coefficients for low-frequency rhythmic motion, temporal moments (mean, variance, skewness, and kurtosis) for time-domain dynamics, and GCC for inter-user synchronization via time-lag relationships. Together, they capture spectral, temporal, and relational motion patterns for classification.

3.4.1 GFC coefficients

GFC coefficients form the spectral backbone of our descriptor set; they provide a perceptually informed, time–frequency representation that is highly effective at capturing the transient and

band-limited motion primitives typical of wrist-mounted accelerometer signals.[25] Unlike uniform Fourier bands, a gammatone filterbank uses asymmetric, auditory-inspired bandpass filters whose impulse responses emphasize onset/offset structure and local spectral envelope properties that are particularly valuable for discriminating against short, overlapping gestures and other rapid motor events in multi-user recordings. Concretely, each gammatone filter $g_k(t)$ (centered at frequency $f_{c,k}$) is defined in the continuous-time domain by the canonical gammatone impulse response:

$$g_k(t) = at^{n-1}e^{-2\pi bkt}\cos\left(2\pi f_{c,k}t + \phi\right), t \geq 0. \tag{6}$$

Here, n is the filter order controlling the temporal envelope, b_k is the bandwidth (linked to the equivalent rectangular bandwidth (ERB) scale), a is a normalization constant, and ϕ is the phase. Stable discrete-time filter realizations are used to convolve $x(t)$, producing band-passed responses $y_k(t) = x^*g_k(t)$. Windowed band energies are then computed per frame n as

$$E_k(n) = \sum_t \left|y_k(t)\right|^2 w(n-t), \tag{7}$$

where $w(\cdot)$ is the analysis window (Blackman in our pipeline) and the squared magnitude may be derived from the analytic signal (Hilbert envelope) for robust amplitude estimation. To produce compact cepstral features, we apply a logarithm to the filterbank energies followed by a Type-II discrete cosine transform (DCT-II), yielding the GFC coefficients:

$$c_m(n) = \sum_{k=1}^{K} \log\left(E_k(n) + \varepsilon\right)\cos\left(\frac{\pi m(k-0.5)}{K}\right). \tag{8}$$

Here, K is the number of gammatone channels and M the cepstral order. First- and second-order derivatives capture temporal dynamics. Compared with fast Fourier transform (FFT) and mel-frequency cepstral coefficients (MFCCs), the gammatone filterbank is more sensitive to low-frequency motion, whereas ERB spacing and DCT provide a compact, noise-robust representation for activity modeling. The output of GFC is depicted in Fig. 5.

3.4.2 Temporal moments feature extraction

Temporal moments are statistical descriptors that capture the distributional and rotational dynamics of wrist-worn accelerometer signals in segmented activity windows. Unlike frequency-based features, they emphasize temporal evolution rather than spectral energy, providing discriminative representation across users.[26] By encoding shape, spread, asymmetry, and higher-order variations, they effectively distinguish activities with similar spectra but different motion patterns.[27] Formally, for a preprocessed signal $x(t)$ of length T, the n-th order temporal moment is defined in Eq. (9), with its normalized higher-order form in Eq. (10).

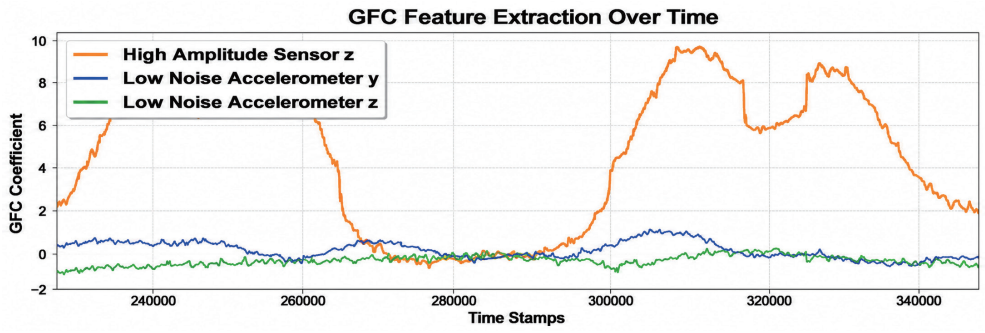


Fig. 5. (Color online) Feature extraction via GFC algorithm.

$$M_n = \frac{1}{T}\sum_{t=1}^{T}\left(t-\mu_t\right)^n x(t) \tag{9}$$

$$\mu_n^* = \frac{\sum_{t=1}^{T}\left(t-\mu_t\right)^n x(t)}{\left(\sum_{t=1}^{T}\left(t-\mu_t\right)^n x(t)\right)^{n/2}} \tag{10}$$

Temporal moments provide a compact representation of motion dynamics. The first and second moments capture activity timing and dispersion, whereas skewness and kurtosis describe motion asymmetry and impulsiveness. Combined with GFC features, they improve robustness and discrimination across interuser variability and fine-grained activities. The output of the temporal-moment is depicted in Fig. 6.

3.4.3 GCC

The proposed framework employs GCC to capture temporal synchronization and interuser alignment in activity streams. Unlike conventional correlation methods, GCC measures signal similarity in the frequency domain through relative time delays,[28] making it effective for modeling phase offsets and asynchronous activities commonly observed in multi-user environments. The GCC response is defined as

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} \Psi(f)\cdot X(f)\cdot Y^*(f)\cdot e^{j2\pi f\tau}\, df. \tag{11}$$

Here, $R_{xy}(\tau)$ is the cross-correlation at lag τ, $X(f)$ and $Y(f)$ are the Fourier transforms of the signals, and $Y^*(f)$ is the complex conjugate of $Y(f)$. The spectral weighting function $\Psi(f)$ controls the trade-off between temporal resolution and noise robustness. To improve time-delay estimation, two weighting strategies are used in GCC. The phase transform (PHAT) is given as

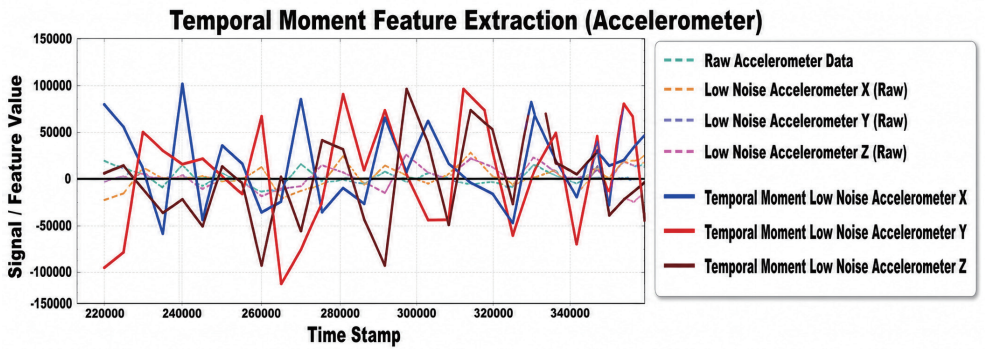


Fig. 6. (Color online) Temporal-moment-based feature extraction from segmented wrist accelerometer signals, illustrating mean, variance, skewness, and kurtosis characteristics across activities.

$$\Psi_{PHAT}(f) = \frac{1}{\left|X(f) \cdot Y^{*}(f)\right|}. \tag{12}$$

This normalization preserves only the phase relationship between signals and produces a sharp correlation peak, enabling highly precise delay estimation under relatively clean sensing conditions. In contrast, the smoothed coherence transform (SCOT) weighting is defined as

$$\Psi_{SCOT}(f) = \frac{1}{\sqrt{G_{xx}(f) \cdot G_{yy}(f)}}. \tag{13}$$

Here, $G_{xx}(f)$ and $G_{yy}(f)$ are the power spectral densities. Coherence weighting reduces incoherent noise while preserving shared dynamics, improving robustness. PHAT or SCOT is adaptively selected on the basis of SNR, and GCC is applied pairwise to estimate the optimal synchronization lag, defined as

$$\tau^{*} = \arg\max_{\tau} R_{xy}(\tau). \tag{14}$$

Here, τ^{*} is the peak cross-correlation delay. These pairwise offsets capture interuser synchronization and serve as both GCC features in LMKL fusion and alignment cues for temporal syncing. This improves feature alignment (e.g., GFC and temporal moments), enhancing discrimination between coordinated and uncoordinated movements in real-world settings. The result of GCC is depicted in Fig. 7.

3.5 Feature fusion and optimization

To preserve complementary motion cues and ensure robustness in real-world scenarios, we adopt a decision-level fusion strategy rather than feature-level fusion. In this framework, each

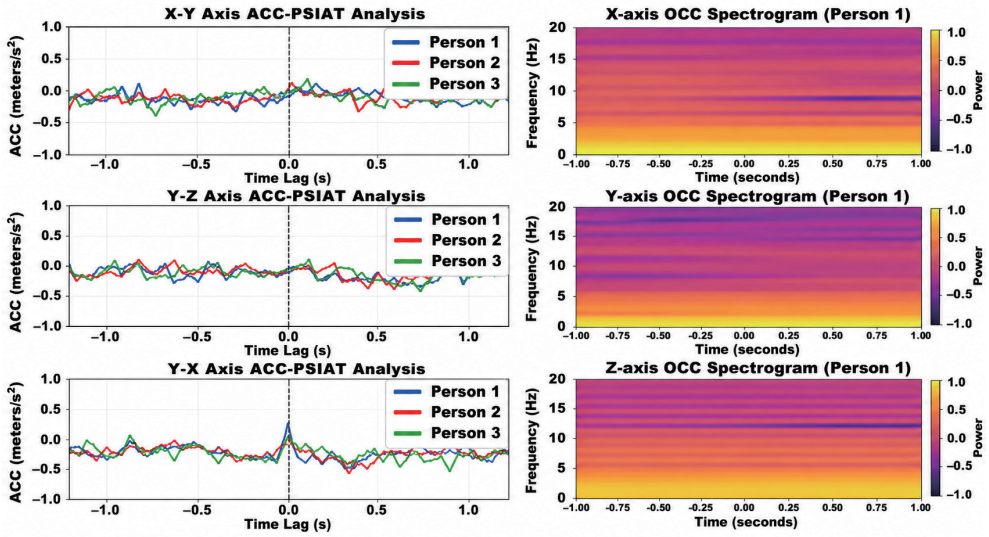


Fig. 7. (Color online) Feature extraction via GCC algorithm.

branch independently learns and produces a prediction based on its respective feature representation: (1) GFC coefficients, which capture spectral envelope characteristics and transient motion patterns; (2) temporal moments, which model distributional statistics and dynamic temporal variations; and (3) GCC features, which encode interstream synchronization information. The final decision is obtained by fusing the outputs of these three branches at the decision level, ensuring that each modality contributes independently while maintaining complementary temporal and cross-stream information. Every branch has its own classifier and gives out a decision D_i. The final prediction is obtained through weighted majority voting, in which each branch is given a reliability weight α_i that is estimated on a held-out validation fold (e.g., by calibrated posterior confidence, per-class F1, or entropy-based reliability). The joint decision is

$$D_f = \arg\max_k \sum_{i=1}^{n} \alpha_i \cdot \delta\left(D_i = k\right), \tag{11}$$

where $\delta(D_i = k)$ is an indicator function, which is 1 when the ith classifier predicts class k; otherwise, 0. We use QHHO to decrease feature redundancy and increase discriminative capacity. This metaheuristic uses the cooperative hunting of Harris Hawks and quantum-inspired position updates to efficiently search the feature space. The feature vectors are coded as hawk positions and the fitness of each is determined by both the classification confidence and the interclass separability. The updated position of the i-th hawk at the end of iteration t is

$$X_i^{t+1} = X_{best}^t - Q \cdot \left| X_i^t - 2r \cdot X_{best}^t \right|. \tag{12}$$

Here, X_{best}^{t} represents the best feature subset, whereas Q and r control exploration. QHHO selects the most discriminative features, improving generalization and reducing computational cost.[30] The optimized features are then passed to the classifier, as illustrated in Fig. 8.

3.5.1 QHHO for discriminative feature selection

The fused feature vector from GFC, temporal moments, and GCC is high-dimensional, making feature selection a combinatorial, nondifferentiable problem. QHHO is used owing to its strong population-based search and ability to avoid local optima via quantum-inspired tunneling, enabling the effective exploration of correlated feature spaces in Capture-24 and CAHAR. It selects compact yet discriminative feature subsets, improving robustness and class separability.

3.6 Temporal alignment

After feature optimization, temporal alignment is crucial for consistent multi-user activity recognition, as variations in onset, speed, and duration arise from sensor delays and user differences, causing misalignment that can degrade accuracy.[31] To address this, DTW is used to nonlinearly align sequences by stretching or compressing the time axis to find optimal matches regardless of length differences. This ensures temporal coherence and preserves synchronized activity dynamics across users:

$$DTW\left(A,B\right)=\min_{\pi}\left\{\sum_{(i,j)\in\pi}\| a_i-b_j\|^2\right\}. \tag{13}$$

Here, π indicates the warping path aligning sequences A and B under monotonicity, continuity, and boundary constraints. DTW aligns sequences of varying lengths by local time stretching/compression, reducing intraclass variance and improving multi-user generalization. In the

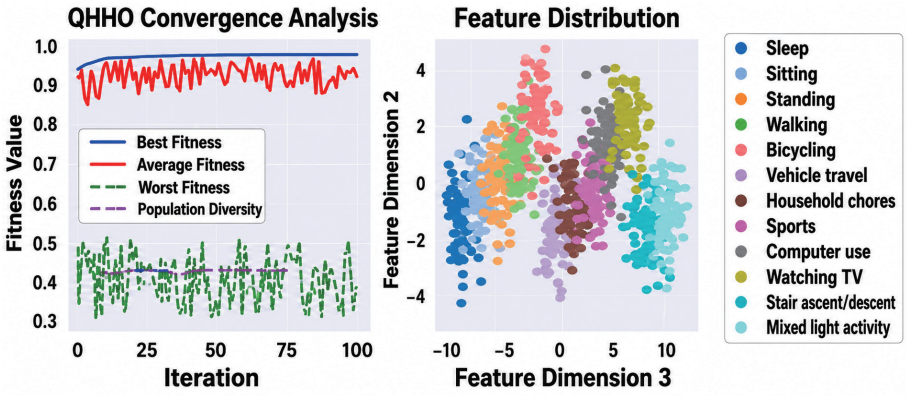


Fig. 8. (Color online) Optimizing fused features using a QHHO method.

pipeline, it is applied after QHHO feature selection, and the resulting alignment distances are used as edge weights in GAT to give stronger effect to better-aligned participants in relational modeling. The result of temporal alignment is illustrated in Fig. 9.

3.7 Activity mapping via GAT

GAT is used to model interuser dependencies after temporal alignment to address collective behavior as they tend to have both temporal and semantic relationships when interacting in synchronized activities.[32] All participants are modeled as nodes, with features of power spectral density (PSD), quaternion-based orientation, and GCC descriptors, and edges are established through the similarity of DTW, which encodes temporal correspondences. In contrast to GCNs, GAT uses an attention mechanism to weigh the relevant neighbors selectively and ignore the less informative ones[33] for more discriminative multi-user interaction modeling. For each node i, the attention coefficient is computed with respect to its neighbor j as

$$e_{ij} = LeakyReLU\left(a^T \left[Wh_i \,\|\, Wh_j\right]\right), \tag{14}$$

where Wh_i and Wh_j are the input features of nodes i and j, respectively, W is a learnable weight matrix, a is the attention vector, and ‖ denotes vector concatenation. The raw attention coefficients e_{ij} are then normalized across all neighbors of i using the softmax function to ensure comparability:

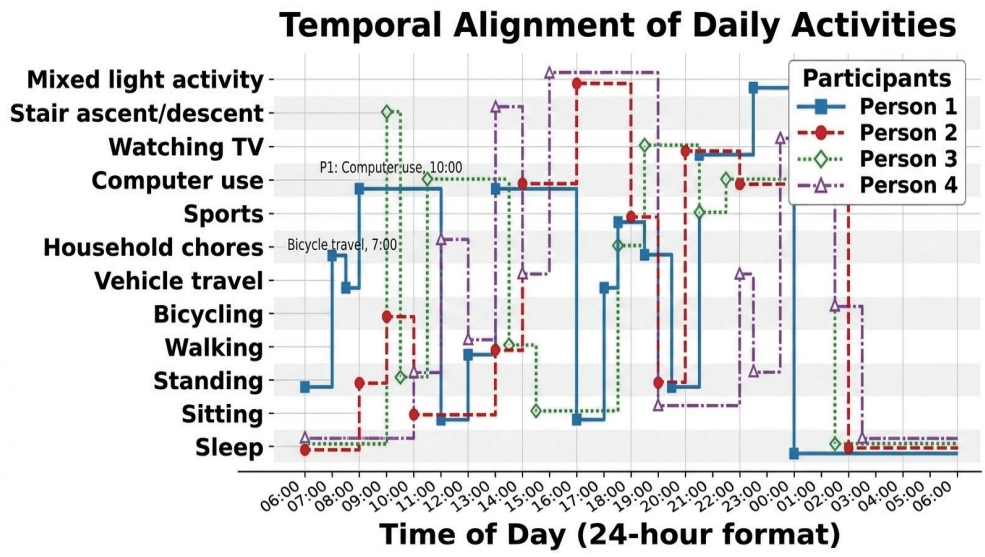


Fig. 9. (Color online) Visualization of activity transitions across participants, highlighting individual behavioral patterns and variations in routine.

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{k \in N(i)} \exp\left(e_{ij}\right)}. \tag{15}$$

The normalized coefficient α_{ij} reflects the relative importance of node j to node i, enabling the model to adaptively prioritize participants whose motion signals exhibit stronger temporal or semantic correlations. The updated node embedding is then expressed as

$$h_i^{'} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} W h_j \right). \tag{16}$$

Here, $\sigma(\cdot)$ denotes a nonlinear activation function. Multihead attention enables each node to learn its own motion dynamics while incorporating contextual information from neighboring participants, producing robust embeddings (Fig. 10). The graph $G(V, E)$ is constructed using DTW-aligned participant features, while attention mechanisms refine edge relationships to jointly model temporal synchronization and inter-user interactions.

3.8 Activity classification via GAT

The last step involves the classification of joint embeddings of the graph neural network (GNN) using TCN. In contrast to fully connected or recurrent models, TCN effectively models long-range temporal dependencies and maintains causal order.[34] Interuser relation GNN encodings are trained with stacked nonlinear activations of 1-D dilated causal convolutions, each with residual connections, which allow an exponentially growing receptive field without reuse.[35] This enables the short- and long-term dynamics to be effectively modeled, such as

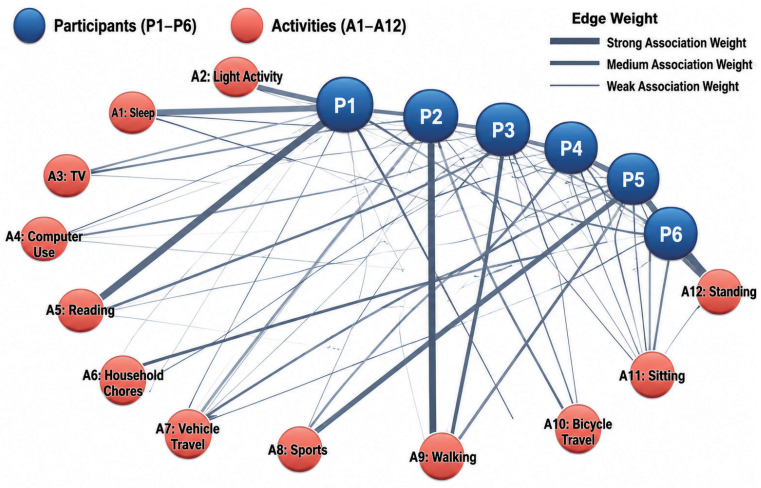


Fig. 10. (Color online) Visualization of multiparticipant activity interactions. P1–P5 nodes represent individual participants and their engagement with each activity.

$$h^{(l)} = \sigma\left(W^{(l)} *_d \cdot h^{(l-1)} + b^{(l)} + h^{(l-1)}\right). \tag{17}$$

Here, $h^{(l-1)}$ and $h^{(l)}$ denote the input and output feature maps, respectively, $W^{(l)}$ and $b^{(l)}$ represent the convolution parameters, and $*_d$ is the dilated convolution operation. Residual connections preserve gradient flow, whereas global average pooling and a Softmax layer generate the final class probabilities:

$$\hat{y} = Softmax\left(W_f .GAP\left(h^L\right) + b_f\right). \tag{18}$$

Here, (h^L) denotes the final hidden representation from the last TCN block, $GAP(\cdot)$ is the global average pooling operator, W_f and b_f are the parameters of the classification layer, and $\hat{y}$ is the predicted probability distribution over activity classes. Figure 11 shows the architecture of the proposed TCN for activity recognition.

3.8.1 Motivation for TCN for sequential activity recognition

TCN is the ultimate classification model because it is efficient and has high ability to predict long sequential activity data. It allows complete parallel processing and does not suffer vanishing gradients with long sequences (e.g., post-DTW windows as long as 256 frames). Dilated causal convolutions increase the receptive field without the addition of parameters, and they learn both long- and short-term dependencies and maintain the temporal order. The correct term is "Residual connections and batch normalization. stabilize training and maintain multilevel features. The global temporal aggregation and Softmax result in a concise and strong representation, whereas the class-weighted loss and macro-averaged metrics deal with class imbalance to evenly evaluate classification performance across all activity classes.

The parameters in Table 1 show balanced accuracy, efficiency, and generalization. A sequence length of 256 captures complete activity cycles, whereas 128 features preserve

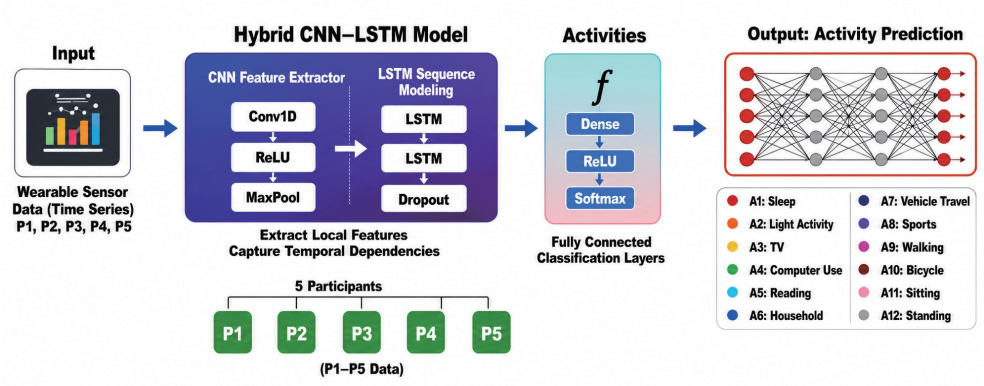


Fig. 11. (Color online) TCN architecture for multiperson activity recognition.

Algorithm 1
Proposed multi-user activity recognition framework.

Input: Sraw={A1,A2…,A151}
Output: C={Class1,Class2…,ClassN}
Method:

1. Preprocessing
Apply VMD to each accelerometer stream to extract intrinsic motion modes:

$$S_{vmd} = VMD\left(S_{raw}, K = 5\right)$$

2. Multi-kernel Fusion
Fuse sensor streams using LMKL with chi-square, wavelet, polynomial, and Gaussian:

$$F_{fused} = \sum_{k=1}^{4} a_k .LKML - Kernel_k \left(S_{sg}\right)$$

(Weights α_k optimized via Sequential Minimal Optimization)

3. Windowing
Segment the fused signal using Blackman windowing to maintain temporal stability:

$$W = BlackmanWindow\left(F_{fused}, L, \Delta\right)$$

4. Feature Extraction
Compute GFC:

$$F_{GFC} = GFC\left(W\right)$$

Extract Temporal Moments:

$$F_{Temp} = Temporal\left(W\right)$$

Generalized Cross-correlation:

$$F_{GCC} = GCC\left(W\right)$$

5. Fused features optimized and reduced dimensionality using QHHO

$$F_{opt} = QHHO\left(F_{GFC}, F_{Temp}, F_{GCC}\right)$$

6. Temporal Alignment
Align multi-user signals using DTW to synchronize activity timelines:

$$F_{aligned} = DTW\left(F_{opt}\right)$$

7. Activity Mapping
Model group semantics via GAT:

$$E_{GNN} = GraphConv\left(F_{aligned}, A_{DTW}\right)$$

8. Classification
Feed the learned embeddings into a TCN to predict activity classes:

$$C = TCN\left(E_{GAT}, \theta_{TCN}\right)$$

Table 1
TCN parameters.

Parameter	Value
Input dimension	128 (joint embeddings from GAT)
Number of TCN blocks	6
Kernel size	3
Dilation factors	[1, 2, 4]
Hidden channels	[128, 256, 256]
Residual connections	Enabled
Dropout rate	0.3
Batch normalization	Applied after each block
Fully connected layer	256 units + ReLU
Output layer	Softmax (activity classes)
Optimizer	Adam, learning rate = 10-31e-3
Loss function	Cross-entropy loss

discriminative information. Multiscale kernels within six inception blocks capture temporal patterns at different scales, and ensemble learning with dropout and adaptive learning rates improves robustness while maintaining efficient inference.

4. Experimentation and Results

The framework was implemented in Python 3.9 using PyTorch 1.12 for TCN and GAT modules, SciPy 1.10 for signal decomposition and windowing, and scikit-learn 1.0 for LMKL fusion. Feature extraction (GFC, TC, and GCC) was developed with NumPy and custom routines, whereas QHHO was implemented from scratch for feature optimization. Experiments were conducted on an Intel i7-12700H CPU with 32 GB of RAM and an NVIDIA RTX 3060 GPU.

4.1 Datasets

For this study, we utilized two datasets: Capture-24 and CAHAR. The details of each dataset are provided below.

4.1.1 Capture-24

Capture-24 [36] is a large-scale dataset collected from a wrist-worn accelerometer under real-world, free-living conditions. It includes data from 151 participants, each monitored continuously for approximately 24 h. A three-axis accelerometer sampled at 100 Hz was used to record motion data during daily activities. Activity annotations were derived using wearable cameras and sleep diaries for accurate labeling. The dataset supports fine- and coarse-grained human activity recognitions across natural, unscripted behavior.

4.1.2 CAHAR

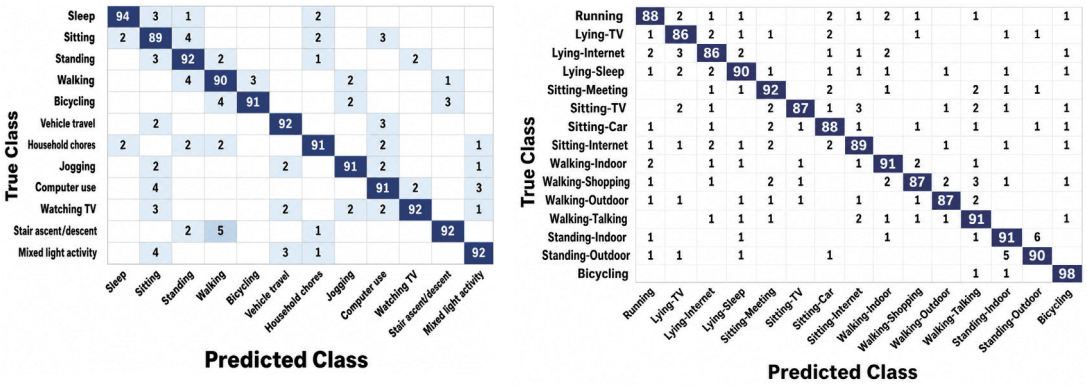
The experiments use CAHAR, a benchmark for context-aware human activity recognition,[37] containing synchronized recordings of 15 daily and composite activities performed by multiple participants under controlled and seminatural conditions. It includes data from wearable inertial and environmental sensors to capture motion dynamics, context, and interactions. Each instance is annotated with temporal boundaries, activity labels, and participant IDs. With overlapping activities, spontaneous interactions, and varied sensor placements, CAHAR provides a realistic testbed for evaluating multi-user HAR in terms of accuracy, scalability, and generalization.

4.2. Model evaluation and experimental results

Leave one subject out (LOSO) cross-validation is performed on Capture-24 and CAHAR to evaluate subject-independent generalization. Performance is measured using Accuracy, Precision, Recall, and F1-score, with results presented as confusion matrices (Fig. 12) and receiver operating characteristic (ROC) curves (Fig. 13), and in Tables 2 and 3. Computational complexity (Table 4), ablation analysis (Table 5), and comparisons with state-of-the-art methods (Table 6) further validate the effectiveness and generalization capability of the proposed framework.

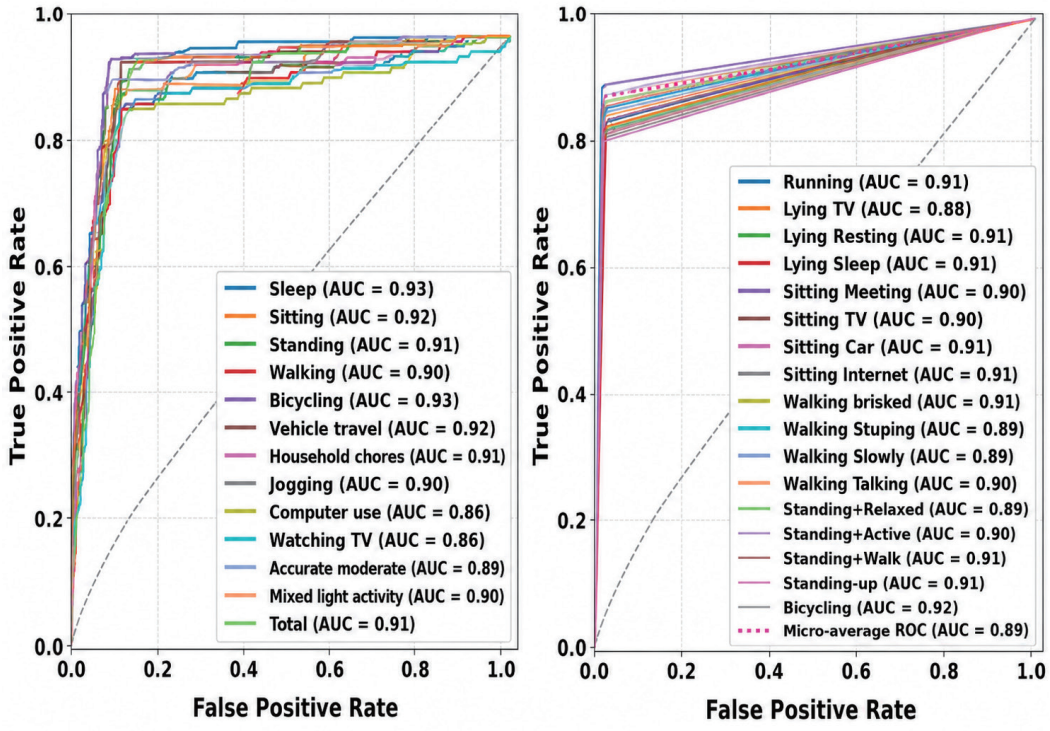
4.2.1 LOSO cross-validation protocol and subject-independent evaluation framework

LOSO cross-validation evaluates subject-independent generalization by training on all but one subject and testing on the unseen subject (151-folds for Capture-24 and subject-wise-folds for CAHAR). This provides a realistic measure of robustness against user variability, sensor placement changes, and signal drift. DTW alignment further improves temporal consistency, with its removal reducing Capture-24 accuracy from 91.74 to 82.59%.



(a) (b)

Fig. 12. (Color online) Confusion matrix results for (a) Capture-24 and (b) CAHAR.



(a) (b)

Fig. 13. (Color online) ROC curves showing the classification performance of the proposed framework on (a) Capture-24 and (b) CAHAR.

Table 2
Precision, Recall, and F1-score evaluated using Capture-24.

Classes	Precision	Recall	F1-Score
Sleep	0.9792	0.9495	0.9641
Sitting	0.8091	0.8900	0.8476
Standing	0.9679	0.9200	0.8932
Walking	0.8491	0.9000	0.8738
Bicycling	0.9479	0.9100	0.9286
Vehicle travel	0.9500	0.9500	0.9500
Household chores	0.9286	0.9100	0.9192
Sports	0.9579	0.9100	0.9333
Computer use	0.9192	0.9100	0.9146
Watching TV	0.9388	0.9200	0.9293
Stairs ascent/descent	0.9485	0.9200	0.9340
Light activity	0.9388	0.9200	0.9293

4.3 Ablation study

To assess each module's contribution, we perform a controlled one-at-a-time ablation study, removing or replacing a single component while keeping all other settings, hyperparameters, and data splits fixed. Both datasets are evaluated under the same LOSO-consistent fivefold

Table 3
Precision, Recall, and F1-Score evaluated using CAHAR.

Classes	Precision	Recall	F1-Score
Running	0.907	0.880	0.893
Lying-TV	0.926	0.894	0.903
Lying-Internet	0.878	0.860	0.869
Lying-Sleep	0.928	0.900	0.914
Sitting-Meeting	0.895	0.850	0.872
Sitting-TV	0.888	0.870	0.879
Sitting-Car	0.880	0.880	0.880
Sitting-Internet	0.830	0.930	0.877
Walking-Indoor	0.904	0.940	0.922
Walking-Shopping	0.879	0.870	0.874
Walking-Outdoor	0.902	0.920	0.911
Walking-Talking	0.861	0.930	0.894
Standing-Indoor	0.919	0.910	0.915
Standing-Outdoor	0.860	0.860	0.860
Bicycling	0.906	0.870	0.888

Table 4
Results of computational complexity analysis.

Methods	Computational complexity	Execution time (s)	Memory usage (MB)
Preprocessing (VMD)	$O(N \cdot K \cdot \log N)$	0.23	140
Multi-kernel fusion	$O(M \cdot d^2)$	1.75	410
Signal segmentation	$O(N^2)$	0.68	210
Feature extraction (GFCCs, temporal moments, GCC)	$O(N \cdot F)$	1.54	363
Feature optimization (Quantum-inspired HHO)	$O(P \cdot I \cdot d)$	1.52	580
Temporal alignment (DTW)	$O(N^2)$	0.91	315
Activity mapping (GAT)	$O(E \cdot d^2)$	0.87	502
Classification (TCN)	$O(L \cdot k \cdot N)$	1.93	490

subject-independent protocol (LMKL: 4–5 kernels, QHHO: 100 iterations, and TCN: 5 layers). We report mean ± std accuracy and the absolute drop (Δ) relative to the full model; macro-F1 follows the same trend and is omitted for brevity. Baseline values are those from linear kernel (LMKL), rectangular window segmentation, raw statistical features, and a standard CNN for TCN.

Results (Table 5) show consistent performance degradation for all ablations. VMD removal reduces accuracy by 7.12 and 7.48% for Capture-24 and CAHAR, LMKL replacement by 5.84 and 6.22%, and rectangular windowing by 4.59 and 4.61%, respectively. The largest drops occur when removing DTW (9.15 and 9.09%) and the feature ensemble (8.26 and 8.74%), followed by GAT (5.33 and 5.36%), QHHO (3.47 and 3.87%), and TCN (2.71 and 2.69%). Both datasets exhibit the same importance ranking, confirming that all modules are consistently beneficial and jointly necessary for optimal performance.

Table 6 presents the performance characteristics of the proposed framework and state-of-the-art methods evaluated on the Capture-24 and CAHAR. On Capture-24, existing approaches such

Table 5
Results of module-wise ablation study on Capture-24 and CAHAR. In each configuration, a single component of the proposed pipeline is removed or replaced while all other settings are held fixed. Δ denotes the absolute accuracy drop relative to the full model.

Configuration	Module removed/replaced	Capture-24 Acc. (%)	Δ (%)	CAHAR Acc. (%)	Δ (%)
Full model (proposed)	Complete pipeline: VMD + LMKL fusion + Blackman segmentation + feature ensemble (GFC, temporal moments, GCC) + QHHO optimization + DTW alignment + GAT relational modeling + TCN classification	91.74 ± 0.4	—	88.93 ± 0.5	—
w/o VMD preprocessing	VMD removed; raw accelerometer signals fed directly to fusion	84.62 ± 0.7	−7.12	81.45 ± 0.8	−7.48
w/o LMKL fusion	LMKL replaced with a single fixed linear kernel (standard SVM baseline)	85.90 ± 0.6	−5.84	82.71 ± 0.7	−6.22
w/o Blackman segmentation	Adaptive Blackman–Tukey window replaced with a basic rectangular sliding window	87.15 ± 0.5	−4.59	84.32 ± 0.6	−4.61
w/o feature ensemble	Multimodal descriptors removed; only raw time-domain statistics (mean, std) used	83.48 ± 0.8	−8.26	80.19 ± 0.9	−8.74
w/o QHHO optimization	QHHO removed; full unoptimized feature set used (no feature selection)	88.27 ± 0.5	−3.47	85.06 ± 0.6	−3.87
w/o DTW alignment	DTW removed; no temporal synchronization across user sequences	82.59 ± 0.9	−9.15	79.84 ± 1.0	−9.09
w/o GAT relational modeling	GAT removed; no graph-based dependency capture, flat per-user embeddings used	86.41 ± 0.6	−5.33	83.57 ± 0.7	−5.36
TCN → CNN (classifier swap)	TCN replaced with a standard CNN for sequential prediction	89.03 ± 0.4	−2.71	86.24 ± 0.5	−2.69

Table 6
Classification comparison with other state-of-the-art models and on Capture-24 and CAHAR.

Author(s)	Dataset	Method	Performance
Kim *et al.*[36]	Capture-24	CNN + HMM	0.80 (F1)
Asim *et al.*[37]	CAHAR	CAHAR framework (random forest)	86.10%
Mukherjee *et al.*[38]	—	SmartARM	80.0%
Vaizman *et al.*[39]	CAHAR	Logistic regression	87%
Arrotta *et al.*[40]	—	CNN–LSTM	82.5%
Bock *et al.*[41]	—	LSTM	74.0%
Hölzemann *et al.*[42]	—	Baseline deep architectures (CNN/RNN)	84.5%
Wang *et al.*[43]	—	Factorial CRF (FCRF) for multi-user HAR	87.93%
Ramos *et al.*[44]	—	LSTM (personalized RNN) networks	88.29%
Haider *et al.*[45]	—	Super-bagging method	84.19%
Yuan *et al.*[46]	Capture-24	Self-supervised ResNet-18	0.73 (F1)
Acquah *et al.*[47]	Capture-24	ActiNet (HARNet + HMM)	0.82 (F1)
Proposed	Capture-24	GAT + TCN	91.74%/0.92 (F1)
Proposed	CAHAR	GAT + TCN	88.93%/0.89 (F1)

as CNN+HMM, Self-supervised ResNet-18, and ActiNet achieve F1-scores of 0.80, 0.73, and 0.82, respectively. In comparison, the proposed GAT+TCN framework attains the highest performance with an accuracy of 91.74% and an F1-score of 0.92. Similarly, on CAHAR, logistic regression and the CAHAR framework based on random forest show accuracies of 87.0 and 86.10%, respectively, whereas the proposed method achieves an improved accuracy of 88.93% with an F1-score of 0.89. These results demonstrate the effectiveness and robustness of the proposed graph-based temporal learning architecture for activity recognition across different real-world datasets.

5. Discussion

The proposed architecture provides a systematic multistage framework[48,49] for automated behavioral measurement through integrated signal acquisition and processing. It captures both temporal and spatial dynamics while optimizing feature learning[50] via a metaheuristic approach, improving robustness, accuracy, and efficiency. Its layered design enables the interpretable, context-aware modeling of heterogeneous sensor signals. However, it remains sensitive to sensor placement noise and has high memory demands in the optimization stage, limiting edge deployment in Industry 4.0 systems. Overall, the results highlight its potential for scalable monitoring in industrial automation, safety, and smart infrastructure, while motivating future lightweight solutions.

6. Conclusions

In this paper, we proposed a powerful system integration approach to automated behavioral measurements by utilizing wrist-worn accelerometer data and combining VMD to gather and refine the signal, LMKL to fuse signals, and Blackman window to segment the signal. Multimodal descriptors were used to enrich the feature space, which is optimized using AI-based metaheuristics, namely, QHHO. DTW provided the time warping, whereas the sensor-based relational modeling was attained through GATs and TCN. Extensive testing on Capture-24 and CAHAR showed a high degree of generalization, with accuracies of 91.74 and 88.9, respectively. In future work, we will explore Transformer-based encoders for subtle activity transitions, a lightweight graph backbone such as GraphSAGE, and model compression via pruning and quantization to enable real-time edge AI deployment, scaling this integrated engineering method to large-scale, unconstrained Industry 4.0 and smart-environment settings for intelligent control and automation.

Acknowledgments

This work was supported through Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R508), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

References

1 M. M. Afsar, S. Saqib, M. Aladfaj, M. H. Alatiyyah, K. Alnowaiser, H. Aljuaid, A. Jalal, and J. Park: IEEE Access **11** (2023) 12460. https://doi.org/10.1109/ACCESS.2023.3239692
2 N. A. Almujally, D. Khan, N. Al Mudawi, M. Alonazi, H. F. Alhasson, A. Jalal, and H. Liu: Front. Bioeng. Biotechnol. **13** (2025) 1437877. https://doi.org/10.3389/fbioe.2025.1437877
3 A. I. Aijaz and J. Ahmad: Proc. Int. Multitopic Conf. (INMIC) (2025).
4 M. Hanzla and A. Jalal: Proc. 4th Int. Conf. Communication, Computing and Digital Systems (C-CODE) (2025) pp. 1–6.
5 L. Wang, T. Gu, X. Tao, H. Chen, and J. Lu: Activity Recognition in Pervasive Intelligent Environments (Atlantis Press, 2011) pp. 55–79. https://doi.org/10.2991/978-94-91216-05-3_3
6 B. Vidya and P. Sasikumar: Sens. Actuators. A **341** (2022) 113557. https://doi.org/10.1016/j.sna.2022.113557
7 R. Gomez-Ramos, J. D. Domingo, E. Zalama, and J. Gómez-García-Bermejo: Sensors **21** (2021) 5270. https://doi.org/10.3390/s21165270
8 T. Wang and D. J. Cook: IEEE Trans. Emerging Top. Comput. **10** (2022) 1130. https://doi.org/10.1109/TETC.2021.3072980
9 M. Willetts, S. Hollowell, L. Aslett, C. Holmes, and A. Doherty: Sci. Rep. **8** (2018) 7961. https://doi.org/10.1038/s41598-018-26174-1
10 N. Twomey, T. Diethe, X. Fafoutis, A. Elsts, R. McConville, P. Flach, and I. Craddock: Informatics **5** (2018) 27. https://doi.org/10.3390/informatics5020027
11 K. Ellis, J. Kerr, S. Godbole, G. Lanckriet, D. Wing, and S. Marshall: Physiological Meas. **35** (2014) 219. https://doi.org/10.1088/0967-3334/35/11/2191
12 N. Y. Hammerla, S. Halloran, and T. Plötz: Proc. IJCAI (2016) pp. 1533–1540
13 L. Chen, Y. Luo, L. Peng, R. Hu, Y. Zhang, and S. Miao: Appl. Intell. **53** (2023) 28169. https://doi.org/10.1007/s10489-023-04997-4
14 F. Ordóñez and D. Roggen: Sensors **16** (2016) 115. https://doi.org/10.3390/s16010115
15 E. Dirgová-Luptáková, M. Šimšík, M. Kocur, and M. Vozňák: Sensors **22** (2022) 1911. https://doi.org/10.3390/s22051911
16 J. L. Rivas-Caicedo, K. Niño-Tejada, L. Saldaña-Aristizabal, and J. F. Patarroyo-Montenegro: Electronics **14** (2025) 3203. https://doi.org/10.3390/electronics14163203
17 M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang: Proc. MobiCASE (2014) pp. 197–205. https://doi.org/10.4108/icst.mobicase.2014.257786
18 L. Bai, C. Yeung, C. Efstratiou, and M. Chikomo: Proc. ACM UbiComp (2020) pp. 537–542. https://doi.org/10.1145/3410530.3414367
19 Y. Vaizman, K. Ellis, and G. Lanckriet: IEEE Pervasive Comput. **16** (2017) 62. https://doi.org/10.1109/MPRV.2017.3971131
20 T. T. Tran, V. T. Pham, and D. T. Bui: J. Sci. Technol. **131** (2018) 100.
21 N. Ur Rehman and H. Aftab: IEEE Trans. Signal Process. **67** (2019) 6039. https://doi.org/10.1109/TSP.2019.2949350
22 Z. Zhang: Proc. IEEE ICASSP (2019) pp. 3357–3361.
23 B. Jeong, J. Lee, H. Kim, S. Gwak, Y. K. Kim, S. Y. Yoo, D. Lee, and J.-S. Choi: Front. Neurosci. **16** (2022) 856510. https://doi.org/10.3389/fnins.2022.856510
24 A. Ahmad, F. S. Schlindwein, and G. A. Ng: J. Biomed. Sci. Eng. **3** (2010) 843.
25 B. H. Prasetio, L. O. Hazmar, D. Syauqy, and E. R. Widasari: Rev. Mex. Ing. Biomed. **45** (2024) 6.
26 S. Sadek, A. Al-Hamadi, M. Elmezain, B. Michaelis, and U. Sayed: Proc. IEEE ISSPIT (2010) pp. 79–84. https://doi.org/10.1109/ISSPIT.2010.5711729
27 S. Pancholi and A. M. Joshi: IEEE Sens. Lett. **3** (2019) 6000804. https://doi.org/10.1109/LSENS.2019.2906386
28 X. Wang and J. Li: Energies **18** (2025) 2628.
29 L. Osadciw and K. Veeramachaneni: Encyclopedia of Biometrics (Springer, 2015) pp. 747–751.
30 Y. Zhang, R. Liu, X. Wang, H. Chen, and C. Li: Eng. Comput. **37** (2021) 3741. https://doi.org/10.1007/s00366-021-01423-3
31 D. S.-M. Liu and Y.-Y. Gao: Signal Process. Image Commun. **138** (2025) 117355. https://doi.org/10.1016/j.image.2025.117355
32 Q. Shi, L. Zhuo, H. Tao, and J. Yang: Int. J. Appl. Earth Obs. Geoinf. **126** (2024) 103610.
33 L. Lu, Y. Lu, R. Yu, H. Di, L. Zhang, and S. Wang: IEEE Trans. Multimedia **22** (2020) 524. https://doi.org/10.1109/TMM.2019.2930344
34 M. A. Al-qaness, A. Dahou, N. T. Trouba, M. Abd Elaziz, and A. M. Helmi: Future Gener. Comput. Syst. **160** (2024) 375. https://doi.org/10.1016/j.future.2024.06.016

35 S. R. Sekaran, P. Y. Han, and O. S. Yin: Expert Syst. Appl. **227** (2023) 120132.
36 S. Kim, J. Nathaniel, Z. Hou, T. Zheng, and P. Gentine: Sci. Data **11** (2024) 1135. https://doi.org/10.1038/s41597-024-04275-z
37 Y. Asim, M. A. Azam, M. Ehatisham-ul-Haq, U. Naeem, and A. Khalid: IEEE Sens. J. **20** (2020) 4361. https://doi.org/10.1109/JSEN.2020.2964278
38 A. Mukherjee, S. Misra, P. Mangrulkar, M. Rajarajan, and Y. Rahulamathavan: Proc. IEEE ANTS (2017) pp. 1–6
39 Y. Vaizman, K. Ellis, and G. Lanckriet: IEEE Pervasive Comput. **16** (2017) 62. https://doi.org/10.1109/MPRV.2017.3971135
40 L. Arrotta, C. Bettini, and G. Civitarese: Proc. Mobiquitous (2021) pp. 52–64.
41 M. Bock, A. Hölzemann, M. Moeller, and K. Van Laerhoven: Proc. IEEE ISWC (2021) pp. 7–12. https://arxiv.org/abs/2110.06663
42 A. Hölzemann, J. L. Romero, M. Bock, K. V. Laerhoven, and Q. Lv: Sensors **23** (2023) 5879. https://doi.org/10.3390/s23135879
43 L. Wang, T. Gu, X. Tao, H. Chen, and J. Lu: Activity Recognition in Pervasive Intelligent Environments **4** (2011) 59. https://doi.org/10.2991/978-94-91216-05-3_3
44 R. G. Ramos, J. D. Domingo, E. Zalama, J. Gómez-García-Bermejo, and J. López: Sensors **22** (2022) 8109. https://doi.org/10.3390/s22218109
45 F. Haider, F. A. Salim, D. B. W. Postma, R. van Delden, D. Reidsma, B.-J. van Beijnum, and S. Luz: Multimodal Technol. Interact. **4** (2020) 33. https://doi.org/10.3390/mti4020033
46 H. Yuan, S. Chan, A. P. Creagh, C. Tong, A. Acquah, D. A. Clifton, and A. Doherty: npj Digit. Med. **7** (2024) 91. https://doi.org/10.1038/s41746-024-01062-3
47 A. Acquah, S. Chan, and A. Doherty: arXiv preprint (2025) arXiv:2510.01712.
48 I. Zahra, Y. Wu, H. F. Alhasson, S. S. Alharbi, H. Aljuaid, A. Jalal, and H. Liu: Front. Neurorobot. **19** (2025) https://doi.org/10.3389/fnbot.2025.1631998
49 A. Alazeb, I. A. Abro, N. A. Mudawi, H. F. Alhasson, N. S. Alshassari, M. Alshehri, and F. Alhayan: PeerJ Comput. Sci. **11** (2025) e3300. https://doi.org/10.7717/peerj-cs.3300
50 I. Zahra, Y. Wu, H. Alsolai, B. Alabdullah, F. Alhayan, A. Jalal, and H. Liu: Kuwait J. Sci. **53** (2026) 100582. https://doi.org/10.1016/j.kjs.2026.100582