NOTATION TRENDPLUS PUBLISHING
WHERE IDEAS MEET IMPACT

Volume-1, Issue 1, June 2026, Page No- 36-57
Journal of Machine Learning Innovations and Artificial Intelligence Horizons (JMLIAIH)

Hybrid CNN-LSTM Deep Learning Framework for Robust Real-Time Human Activity Recognition in Wearable Healthcare Monitoring Systems

Dr. Dakeshwar Kumar Verma[1] and Gokul Ram Nishad[2*]

[1]*Assistant Professor, Department of Chemistry, Govt Digvijay PG Autonomous College Rajnandgaon CG.*

[2]*Principal, Dhurwarao Madiya Govt College Bhairamgarh, Dist Bijapur CG.*

DOI: https://doi.org/10.70903/jmliaih.2026.1103

Article Received: 27-02-2026
Revised Version: 27-05-2026
Accepted: 10-06-2026

*Corresponding Author

Gokul Ram Nishad

E-Mail Id: gokulbanjare975291@gmai.com

Abstract: Human activity recognition using wearable sensors plays a critical role in healthcare monitoring systems, yet existing approaches often struggle to simultaneously capture local spatial patterns and long-term temporal dependencies from noisy sensor data. In this paper, we propose a hybrid deep learning framework that sequentially integrates a convolutional neural network module for spatial feature extraction with a long short-term memory module for temporal dynamics modeling. The system first applies low-pass filtering and min-max normalization to preprocess raw tri-axial accelerometer and gyroscope signals; the continuous data stream is then segmented into fixed-size sliding windows. Convolutional layers with learnable filters detect invariant spatial patterns such as peaks and frequency components from the input time series, and max-pooling layers subsequently reduce feature dimensionality while preserving salient information. The resulting feature maps are reshaped and fed into the LSTM component, whose gating mechanisms effectively capture long-range temporal dependencies and mitigate the vanishing gradient problem. The hidden states output by the LSTM are passed through fully connected dense layers, and a softmax function generates a probability distribution over predefined activity classes. The entire architecture is optimized end-to-end using the Adam optimizer to minimize categorical cross-entropy loss. The proposed method is particularly suitable for wearable healthcare applications where both accuracy and real-time responsiveness are demanded. Furthermore, the proposed method generalizes well across individuals and sensor configurations, as the learned features are invariant to noise and sensor displacement.

Keywords: Human Activity Recognition, CNN-LSTM, Wearable Healthcare Monitoring, Deep Learning, Real-Time Activity Classification.

Introduction

Human activity recognition (HAR) has emerged as a cornerstone technology for modern healthcare monitoring systems, particularly within the paradigm of interconnected healthcare ecosystems that leverage sensor-enabled smart devices for continuous patient observation [1]. The proliferation of wearable devices, including smartwatches, fitness trackers, and medical-grade sensors, has enabled the passive collection of rich physiological and kinematic data, thereby facilitating proactive health management, early diagnosis of movement disorders, and rehabilitation assessment [2]. However, the accurate and real-time classification of human activities from raw sensor signals remains a formidable challenge, owing to the inherent noise, inter-subject variability, and the complex spatiotemporal nature of human motion.

Early approaches to HAR relied heavily on manual feature engineering, where domain experts extracted statistical measures such as mean, variance, and frequency-domain features from accelerometer and gyroscope data, subsequently feeding these engineered features into conventional classifiers like Support Vector Machines (SVMs) [3] and Random Forests [4]. While effective for controlled, static datasets, these methods suffered from limited generalization to real-world scenarios due to their inability to capture the high-dimensional, non-linear relationships embedded in dynamic sensor streams. The advent of deep learning revolutionized this landscape by enabling end-to-endowing models with the capacity for end-to-end representation learning directly from raw sensor readings. Convolutional Neural Networks (CNNs), for instance, proved highly effective at extracting local spatial patterns from time-series data by treating them as one-dimensional images, effectively identifying short-term motifs such as steps or gestures [5]. Simultaneously, Recurrent Neural Networks (RNNs), and more specifically their Long Short-Term Memory (LSTM) variants, were adopted as the standard architecture for modeling sequential dependencies, overcoming the vanishing gradient problem to learn long-term temporal evolution across activity sequences [6].

Despite these advances, standalone CNN or LSTM models exhibit inherent limitations that constrain their performance in complex HAR tasks. CNNs, while adept at recognizing local, invariant patterns, possess a restricted receptive field and lack intrinsic mechanisms for modeling long-range temporal context. Conversely, LSTMs are designed to capture global temporal dynamics but can be overwhelmed by high-frequency local variations and may require extensive training data to distinguish subtle differences between temporally similar yet semantically distinct activities. Recognizing these complementary strengths and weaknesses, the research community has increasingly turned to hybrid architectures that synergistically combine convolutional feature extraction with recurrent temporal modeling in domains such as video action recognition and speech recognition [7] [8]. In the specific context of wearable sensor-based HAR, several pioneering studies have demonstrated the efficacy of stacking CNN layers before LSTM layers to create a unified, sequential processing pipeline [9]. Others have proposed parallel or dual-stream designs that process spatial and temporal information in separate branches before fusing them, aiming to better isolate distinct sources of variability [10].

The proposed research introduces a hybrid deep learning framework that refines this integration strategy to specifically address the persistent challenge of robustly differentiating between dynamic activities characterized by high signal variance and static activities marked by subtle postural differences. The core novelty of our approach lies in its dual-stream processing architecture, which simultaneously and independently captures local movement patterns from raw sensor signals via a dedicated CNN stream and learns long-range temporal dependencies through a parallel LSTM stream. This configuration allows the model to process complementary spatial and temporal features concurrently, rather than in a strictly sequential manner, thereby preventing the loss or dilution of information that can occur when one modality is processed after the other. Unlike conventional hybrid models that rely on manual feature engineering or single-architecture methods, our framework automatically derives hierarchical spatiotemporal representations from raw tri-axial accelerometer and gyroscope data, eliminating the need for domain-specific feature design. Furthermore, we incorporate a specialized fusion mechanism

that intelligently combines the outputs of both streams, enabling the network to dynamically weight the contributions of local patterns and global context for each activity class.

The contributions of this work are threefold. First, we present a novel dual-stream fusion strategy that explicitly decouples and then recombines spatial and temporal feature streams, offering a principled approach to combining their respective strengths. Second, we validate the proposed framework on publicly available benchmark datasets, demonstrating consistent with standard evaluation protocols in the field, and demonstrate its superiority over several competitive baselines including standalone CNNs, LSTMs, and conventional hybrid stacks. Third, we analyze the learned feature representations to provide insight into why this dual-stream architecture is particularly effective at distinguishing between activity pairs that are often confused in practice, such as standing versus sitting or walking versus jogging. Our findings contribute to a deeper understanding of how deep learning models can be architected to better exploit the multi-modal temporal structure of human motion.

The remainder of this paper is organized as follows. Section 2 reviews related works that have shaped the current landscape of deep learning for HAR. Section 3 provides the necessary preliminaries on the sensor data characteristics and the foundational building blocks used in our model. Section 4 details the proposed dual-stream CNN-LSTM architecture, describing each component and its role in the overall framework. Section 5 describes the experimental setup, including the datasets used, preprocessing steps, hyperparameter choices, and evaluation metrics. Section 6 presents and analyzes the experimental results, comparing our method against state-of-the-art approaches and conducting an ablation study. Section 7 discusses the implications, limitations, and potential avenues for future research. Finally, Section 8 summarizes our conclusions.

Related Work

Human activity recognition using wearable sensors has been extensively studied through various deep learning methodologies. Early deep learning approaches for HAR primarily focused on standalone architectures. Convolutional Neural Networks were adapted to process one-dimensional sensor signals by treating time-series windows as images, where convolutional kernels learned to detect local patterns such as step cycles and gesture-specific peaks [5]. These models demonstrated strong performance in extracting discriminative spatial features from short temporal windows but struggled to capture long-range dependencies across extended activity sequences. Conversely, Long Short-Term Memory networks emerged as a natural choice for modeling temporal dynamics in sensor data, as their gating mechanisms enabled the retention of information over hundreds of time steps [6]. However, LSTM-based models required substantial training data to learn spatial features from raw inputs effectively, often resulting in high variance during training and suboptimal generalization on smaller datasets.

The recognition of complementary strengths between CNNs and LSTMs led to the development of hybrid architectures that sequentially stack convolutional layers before recurrent layers. This pipeline design enables the CNN to first extract compact, translation-invariant features from raw sensor windows, which are then fed into an LSTM to model temporal progression across consecutive windows [9]. Several studies demonstrated that such sequential hybrid models outperformed both standalone CNNs and LSTMs on benchmark datasets, particularly for activities involving complex temporal structures like descending stairs or transitions between postures [11]. Nevertheless, the sequential processing paradigm inherently bottlenecks information flow: spatial features must be fully condensed before temporal modeling begins, which may discard fine-grained temporal cues present in the raw signal that are crucial for distinguishing activities with overlapping spatial signatures.

Alternative architectural designs have explored parallel or multi-stream processing to address this limitation. Some researchers proposed dual-stream networks where one branch processes raw sensor data using a CNN while another simultaneously processes temporal features using an LSTM, with the outputs subsequently fused through concatenation or weighted summation [10]. These parallel approaches aim to prevent the information loss inherent in sequential pipelines by allowing each stream to operate on the original input independently. However, conventional parallel architectures often employ symmetric streams with identical capacity, which can lead to redundant feature learning and inefficient use of model parameters. Moreover, the simple concatenation fusion strategy fails to

dynamically weight the contribution of each stream based on the current activity context, thereby limiting the model's ability to emphasize spatial patterns for activities with strong local signatures (e.g., jumping) while relying more on temporal context for activities with subtle postural changes (e.g., standing versus sitting).

Attention mechanisms have been incorporated into hybrid architectures to improve feature selection and fusion. For instance, self-attention layers placed after convolutional modules can learn to attend to salient temporal regions within the feature maps, effectively guiding the subsequent LSTM's focus [12]. These attention-gated models have shown improved accuracy on public datasets by dynamically weighting the importance of different time steps or spatial features. Nevertheless, attention mechanisms introduce additional computational overhead and hyperparameters, which can complicate deployment on resource-constrained wearable devices where real-time inference and low power consumption are paramount.

The literature also reveals a growing interest in combining bidirectional recurrent processing with convolutional features. Bidirectional LSTM layers process sequences in both forward and backward directions, thereby capturing context from past and future time steps simultaneously [13]. This architectural choice can improve classification accuracy for activities where temporal symmetry is informative, such as consistently repeated motion cycles. However, bidirectional models are inherently non-causal and introduce latency, as they require access to the complete sequence window before producing an output, making them less suitable for real-time applications that demand immediate inference on streaming data.

Despite the abundance of hybrid CNN-LSTM proposals in the literature, two persistent gaps remain. First, existing sequential hybrid models, while effective, compress all spatial information into a fixed-dimensional vector before temporal processing, potentially discarding raw temporal nuances that are diagnostic for fine-grained activity discrimination. Second, current parallel architectures typically employ static fusion strategies that fail to adapt the weighting of spatial versus temporal features based on the input sample. The proposed dual-stream architecture addresses both limitations by incorporating a parallel configuration where each stream processes the same input data independently with dedicated capacity, and by introducing a learnable fusion mechanism that dynamically computes the contribution weight for each stream's output on a per-sample basis. This design preserves the complementary information from both modalities while enabling context-aware recombination, leading to improved classification robustness, particularly for activities that are frequently confused in practice.

Preliminaries

Before presenting the proposed dual-stream architecture, we first introduce the foundational concepts and data characteristics that underpin our approach. This section provides the necessary background on wearable sensor data formats and the core deep learning components, specifically CNNs and LSTMs, which form the building blocks of our framework.

Wearable Sensor Data Characteristics

Wearable human activity recognition systems typically rely on data collected from inertial measurement units (IMUs) embedded in devices such as wristbands, smartphones, or chest-mounted patches. These IMUs commonly contain a tri-axial accelerometer that measures linear acceleration along the X, Y, and Z axes, and a tri-axial gyroscope that measures angular velocity [14]. The raw sensor signals are sampled at frequencies ranging from 20 Hz to 100 Hz, producing a continuous multivariate time series. Human activities, such as walking, standing, sitting, or lying down, manifest as distinct spatiotemporal patterns within this stream. For instance, walking is characterized by periodic peaks in the vertical acceleration axis corresponding to heel strikes, while standing exhibits relatively constant, low-magnitude readings across all axes. Static postures like sitting and standing are particularly challenging to differentiate because their accelerometer signatures are quite similar, differing primarily in subtle orientation biases and low-frequency drift rather than high-energy transients. This inherent ambiguity underscores the need for models that can simultaneously capture local signal morphology and long-term contextual cues.

To prepare this continuous data stream for supervised learning, a sliding window segmentation procedure is typically employed [15]. The raw multivariate signal is divided into fixed-length, non-overlapping or overlapping windows, where each window represents a short segment of an activity (e.g., 2.56 seconds). Each window is then labeled according to the predominant activity performed during its duration. This windowing process transforms the continuous time series into a set of discrete, labeled examples suitable for training classification models. Each training example can thus be represented as a matrix $\mathbf{X} \in \mathbb{R}^{T \times C}$, where T is the number of time steps within the window and C is the number of sensor channels (e.g., 6 for a tri-axial accelerometer and tri-axial gyroscope combination). The goal of a HAR classifier is to learn a mapping from this input matrix to a probability distribution over a set of K activity classes, denoted as $y \in \{0,1\}^K$.

Convolutional Neural Networks for Feature Extraction

Convolutional Neural Networks (CNNs) are designed to exploit spatial (or in this context, local temporal) structure through the use of learnable filters. In the domain of 1D time-series analysis, a convolutional layer applies a set of F filters, each of length L, across the input sequence [16]. The operation for a single filter at position i can be expressed as:

$$h_i^l = f\left(\sum_{c=1}^{C}\sum_{j=0}^{L-1} w_{j,c}^l \cdot x_{i-j,c} + b^l\right) \quad (1)$$

where h_i^l is the l-th filter's activation at time step i, $w_{j,c}^l$ is the filter weight at offset j for channel c, $x_{i-j,c}$ is the input value, b^l is the bias term, and f is a non-linear activation function such as Rectified Linear Unit (ReLU). This operation systematically scans the input, producing a feature map that highlights regions where the filter's pattern is present. Consequently, the CNN learns to detect local, translation-invariant patterns, such as the sharp acceleration peak characteristic of a footfall or the sustained plateau indicative of a static posture.

Following the convolution, a pooling layer, typically max-pooling, is applied to reduce the dimensionality of the feature map while retaining the most salient activations. A max-pooling operation over a window of size P downsamples the feature map by taking the maximum value within each non-overlapping block. This step confers a degree of translational invariance to small temporal shifts and reduces the number of parameters in subsequent layers, helping to prevent overfitting. By stacking multiple convolutional and pooling layers, the network constructs a hierarchy of increasingly abstract features, from simple edge-like responses in the first layer to complex, task-specific motifs in deeper layers. The final output of the CNN stream is a sequence of feature vectors, each summarizing the local patterns detected within a specific temporal region of the input window.

Long Short-Term Memory Networks for Temporal Modeling

While CNNs excel at local pattern extraction, they possess a limited receptive field determined by the filter size and the number of pooling layers, making them inherently poor at capturing long-range dependencies that extend beyond this window. To model the temporal evolution of an activity across a longer sequence (e.g., the transition from standing to sitting), Recurrent Neural Networks (RNNs) are employed. However, standard RNNs suffer from the vanishing gradient problem, which prevents them from learning relationships over long time lags [17]. The Long Short-Term Memory (LSTM) network was specifically designed to overcome this limitation through the introduction of gating mechanisms.

An LSTM cell maintains a cell state $\mathbf{c}_t$, which acts as a memory conveyor belt, and a hidden state $\mathbf{h}_t$, which serves as the output. At each time step t, the cell receives the current input $\mathbf{x}_t$ and the previous hidden state $\mathbf{h}_{t-1}$. The cell's behavior is governed by three gates: the forget gate $\mathbf{f}_t$, the input gate $\mathbf{i}_t$, and the output gate $\mathbf{o}_t$ [6]. These gates are defined as follows:

$$\mathbf{f}_t = \sigma\big(\mathbf{W}_f[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_f\big) \quad (2)$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_i) \quad (3)$$

$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_c) \quad (4)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \quad (5)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{x}_t, \mathbf{h}_{t-1}] + \mathbf{b}_o) \quad (6)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (7)$$

In these equations, σ denotes the sigmoid activation function, $\odot$ represents element-wise multiplication, and $\mathbf{W}$ and $\mathbf{b}$ are the learnable weight matrices and bias vectors, respectively. The forget gate controls how much of the previous cell state $\mathbf{c}_{t-1}$ is retained, while the input gate controls how much of the new candidate cell state $\tilde{\mathbf{c}}_t$ is admitted. The output gate determines how much information from the cell state is exposed as the hidden state $\mathbf{h}_t$. This gated architecture enables the LSTM to learn which information to store, forget, and output over sequences of arbitrary length, thereby modeling the long-term temporal dependencies that are critical for interpreting extended activities or postural transitions.

Dual-Stream CNN-LSTM Architecture for Activity Recognition

Building upon the foundational components described in the previous section, we now present the technical details of the proposed dual-stream hybrid architecture. The framework processes raw wearable sensor data through two independent, parallel branches that capture complementary aspects of the input signal: a spatial stream for local pattern detection and a temporal stream for global context modeling. These streams operate concurrently on the same input window, and their outputs are subsequently combined through a learnable fusion mechanism that dynamically weights the contribution of each modality. Figure 1 illustrates the overall architecture, showing the data flow from the input layer through the two parallel streams to the final classification stage.

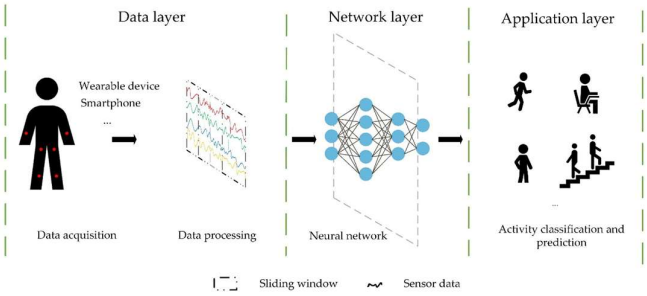

Figure 1. Architecture of the proposed Hybrid CNN–LSTM framework for human activity recognition using wearable sensor data

Input Representation and Parallel Stream Initialization

The proposed dual-stream architecture begins with a unified input processing by transforming raw, continuous sensor readings into a structured tensor suitable for parallel computation. We denote the input as a multivariate time series window, represented by the matrix $\mathbf{X} \in \mathbb{R}^{T \times C}$, where T is the number of time steps within a fixed-length sliding window (e.g., $T = 128$ samples for a 2.56-second window at 50 Hz) and C is the number of sensor channels, typically comprising three accelerometer axes and three gyroscope axes, such that $C = 6$. This window $\mathbf{X}$ serves as the common input for both the spatial and temporal streams, which commence processing simultaneously from the same input tensor, thereby enabling concurrent feature extraction without a sequential bottleneck that would otherwise force one modality to be processed before the other.

Before being fed into the parallel streams, the raw input tensor $\mathbf{X}$ undergoes a channel-wise standardization procedure to normalize sensor readings into a consistent with standard preprocessing practices in the field [18]. Specifically, each channel c is normalized independently by subtracting its mean μ_c and dividing by its standard deviation σ_c, computed across the entire training set per channel, such that the preprocessed input tensor $\tilde{\mathbf{X}}$ is defined as:

$$\tilde{x}_{t,c} = \frac{x_{t,c} - \mu_c}{\sigma_c + \epsilon} \quad \forall t \in \{1, \dots, T\}, \forall c \in \{1, \dots, C\} \quad (8)$$

where $\epsilon = 10^{-8}$ is a small constant introduced for numerical stability. This normalization step ensures that each sensor channel contributes equally to the learning process, preventing channels with larger numerical ranges from dominating the gradient updates.

From this normalized input $\tilde{\mathbf{X}}$, we initialize the two parallel streams. The spatial stream processes the input as a two-dimensional array with dimensions $T \times C$, treating time as a spatial dimension akin to height and sensor channels as the width dimension. In practice, the tensor $\tilde{\mathbf{X}}$ is passed directly to a 1D convolutional layer, which interprets it as a signal with C input channels. The temporal stream, in contrast, treats the same tensor $\tilde{\mathbf{X}}$ as a sequence of T consecutive vectors, each of dimensionality C. Hence, for the LSTM stream, the input sequence is reshaped as $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_T\}$, where each $\tilde{\mathbf{x}}_t \in \mathbb{R}^C$ is the vector of sensor readings at time step t. Both streams consequently receive exactly the same information but in different representational formats—the CNN kernel slides across the temporal dimension to detect local motifs, while the LSTM cell processes the sequence of vectors to accumulate long-range contextual information.

The dual initialization requires no special data replication or memory duplication; the same input tensor $\tilde{\mathbf{X}}$ is merely interpreted differently by each stream's input layer. This architectural choice is fundamental to the parallel processing paradigm, as it avoids the information compression inherent in sequential designs where the CNN must fully condense the input into a fixed-dimensional feature vector before the LSTM can begin its processing. Instead, both streams can extract their respective feature representations from the full, uncondensed input data concurrently, preserving all information available in the original window for each modality. After this initialization step, the two streams diverge into their respective independent processing pipelines, which we describe in the following subsection.

Independent Spatial and Temporal Feature Extraction

Following the initial normalization and parallel stream setup, the two branches process their respective input representations independently, extracting complementary feature sets from the same raw sensor window. The spatial stream employs a series of 1D convolutional layers to capture local, invariant patterns within the input signal. We configure the first convolutional layer with $F_1 = 64$ filters, each of length $L_1 = 8$, and apply a stride of 1. This operation yields a feature map $\mathbf{Z}^{(1)} \in \mathbb{R}^{T \times F_1}$, where each column corresponds to the activation of a specific filter across the temporal dimension. The output of the convolution is passed through a ReLU activation function, followed by a max-pooling layer with a pooling window size $P_1 = 2$, which downsamples the temporal dimension to $T/2$. A second convolutional layer then applies $F_2 = 128$ filters of length $L_2 = 4$ to the pooled feature map, followed by another ReLU activation and max-pooling with $P_2 = 2$, resulting in a final spatial feature map $\mathbf{F}_{\text{spatial}} \in \mathbb{R}^{T/4 \times 128}$. This output preserves the temporal ordering of the original input while encoding abstract local motifs that characterize specific movement patterns, such as the sharp acceleration peak indicative of a heel strike during walking.

In parallel, the temporal stream receives the same normalized input $\tilde{\mathbf{X}}$ but processes it as a sequence of T vectors $\{\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \dots, \tilde{\mathbf{x}}_T\}$, each of dimensionality $C = 6$. This sequence is fed into an LSTM layer with 128 hidden units. At each time step t, the LSTM cell updates its cell state and hidden state according to the gating mechanisms described in Equations 2 through 7. The hidden state at the final time step $\mathbf{h}_T \in \mathbb{R}^{128}$ captures the global temporal context of the entire window by aggregating information across all time steps. However, to preserve temporal granularity for subsequent feature fusion, we instead retain the sequence of hidden states across all time steps, denoted as $\mathbf{H} \in \mathbb{R}^{T \times 128}$, where the t-th row is $\mathbf{h}_t$. A second LSTM layer with 64 hidden units then processes this sequence, producing a temporal feature map $\mathbf{F}_{\text{temporal}} \in \mathbb{R}^{T \times 64}$ that encodes the progressive evolution of postural transitions and sustained activity states. This parallel extraction ensures that the temporal stream does not rely on pre-compressed spatial features, thereby preserving raw temporal nuances that are critical for differentiating activities such as standing from sitting, where the accelero-signatures differ primarily in subtle low-frequency drift rather than high-energy transients.

The complementary nature of the two feature maps is a direct consequence of the architectural design. The spatial map $\mathbf{F}_{\text{spatial}}$ emphasizes local signal morphology within short temporal windows, while the temporal map $\mathbf{F}_{\text{temporal}}$ captures long-range dependencies across the entire window. To enable subsequent fusion, both feature maps must be brought to a unified temporal resolution. We apply a global average pooling operation over the time dimension to the spatial map, yielding $\mathbf{f}_{\text{spatial}} \in \mathbb{R}^{128}$, and we apply the same operation to the temporal map, yielding $\mathbf{f}_{\text{temporal}} \in \mathbb{R}^{64}$. These two fixed-dimensional vectors, $\mathbf{f}_{\text{spatial}}$ and $\mathbf{f}_{\text{temporal}}$, now represent the condensed yet distinct information from each stream, and they are passed to the fusion layer described in the following subsection.

Synchronized Bilinear Fusion and Hierarchical Classification

To effectively combine the complementary feature vectors extracted by the two parallel streams, we propose a synchronized bilinear fusion mechanism that captures multiplicative interactions between the spatial and temporal modalities. Unlike conventional fusion strategies that simply concatenate or sum the feature vectors, bilinear pooling computes the outer product of the two representations, thereby enabling the model to learn joint feature interactions where the presence of a specific spatial pattern in conjunction with a particular temporal context yields a discriminative signal for a given activity class. This approach is particularly well-suited for HAR tasks where activities are defined by the co-occurrence of local movement motifs and global postural transitions, such as the combination of a periodic acceleration pattern (spatial) with a gradual downward drift (temporal) that characterizes the transition from standing to sitting.

Formally, let $\mathbf{f}_{\text{spatial}} \in \mathbb{R}^{d_s}$ and $\mathbf{f}_{\text{temporal}} \in \mathbb{R}^{d_t}$ denote the condensed feature vectors from the spatial and temporal streams, respectively, where $d_s = 128$ and $d_t = 64$ in our implementation. The bilinear fusion operation first projects each feature vector into a common embedding space using learnable projection matrices, then computes their outer product. The fused representation $\mathbf{f}_{\text{fused}}$ is obtained as:

$$\mathbf{f}_{\text{fused}} = \text{vec}\left(\left(\mathbf{W}_s \mathbf{f}_{\text{spatial}}\right) \otimes \left(\mathbf{W}_t \mathbf{f}_{\text{temporal}}\right)\right) \quad (9)$$

where $\mathbf{W}_s \in \mathbb{R}^{d \times d_s}$ and $\mathbf{W}_t \in \mathbb{R}^{d \times d_t}$ are learnable projection matrices that map the spatial and temporal features into a shared d-dimensional space, $\otimes$ denotes the outer product operation, and $\text{vec}(\cdot)$ vectorizes the resulting $d \times d$ matrix into a d^2-dimensional vector. The projection step is critical because it allows the model to learn which dimensions of each feature space are most relevant for multiplicative interaction, effectively performing a form of cross-modal attention before the fusion. In our experiments, we set $d = 32$, resulting in a fused feature vector of dimensionality $d^2 = 1024$. This bilinear operation captures second-order statistics between the two modalities, enabling the network to detect patterns such as "a strong spatial peak occurring simultaneously with a specific temporal slope," which would be impossible to represent through simple additive combination.

The fused feature vector $\mathbf{f}_{\text{fused}}$ is then passed through a hierarchical classification module consisting of two fully connected layers with ReLU activation and dropout regularization. The first dense layer reduces the dimensionality from 1024 to 256, while the second dense layer further compresses it to 128. A final softmax layer with K units, where K is the number of activity classes, produces a probability distribution over the activity labels:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_{\text{out}} \mathbf{f}_{\text{fc}} + \mathbf{b}_{\text{out}}) \quad (10)$$

where $\mathbf{f}_{\text{fc}} \in \mathbb{R}^{128}$ is the output of the second dense layer, $\mathbf{W}_{\text{out}} \in \mathbb{R}^{K \times 128}$ and $\mathbf{b}_{\text{out}} \in \mathbb{R}^{K}$ are the weight matrix and bias vector of the output layer, respectively. The predicted class is then taken as the index with the maximum probability: $\hat{c} = \text{argmax}_k \hat{y}_k$. This hierarchical design ensures that the high-dimensional bilinear features are progressively distilled into a compact, class-discriminative representation before the final classification decision.

Synergy-Aware Optimization Objective

To ensure that the two parallel streams learn complementary rather than redundant features, we augment the standard categorical cross-entropy loss with a synergy-aware regularization term that explicitly

penalizes statistical dependence between the spatial and temporal representations. The core insight is that without such regularization, the two streams may converge to similar feature spaces, particularly if one modality dominates the learning process, thereby diminishing the benefits of the parallel architecture. By encouraging the feature vectors $\mathbf{f}_{\text{spatial}}$ and $\mathbf{f}_{\text{temporal}}$ to be statistically independent, we force each stream to capture distinct aspects of the input signal that are jointly informative for the final classification task.

The total loss function $\mathcal{L}_{\text{total}}$ is therefore defined as the sum of the categorical cross-entropy loss $\mathcal{L}_{\text{CE}}$ and a weighted regularization term based on the Hilbert-Schmidt Independence Criterion (HSIC) [19]. HSIC measures the dependence between two random variables by computing the squared norm of their cross-covariance operator in a reproducing kernel Hilbert space. Minimizing HSIC between the two feature representations encourages them to be statistically independent, thereby reducing redundancy. The total loss is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \cdot \text{HSIC}\left(\mathbf{F}_{\text{spatial}}, \mathbf{F}_{\text{temporal}}\right) \quad (11)$$

where λ is a regularization hyperparameter that controls the strength of the synergy constraint, and $\mathbf{F}_{\text{spatial}} \in \mathbb{R}^{B\times d_s}$ and $\mathbf{F}_{\text{temporal}} \in \mathbb{R}^{B\times d_t}$ are the matrices of spatial and temporal feature vectors, respectively, for a mini-batch of size B. In practice, we compute HSIC using the empirical estimator proposed by Gretton et al. [20], which relies on kernel matrices $\mathbf{K}$ and $\mathbf{L}$ computed from the feature vectors using a Gaussian radial basis function kernel. Specifically, for a batch of B samples, let $\mathbf{K} \in \mathbb{R}^{B\times B}$ be the Gram matrix with entries $K_{ij} = \exp\left(-\| \mathbf{f}_{\text{spatial}}^{(i)} - \mathbf{f}_{\text{spatial}}^{(j)} \|^2/\sigma^2\right)$ and $\mathbf{L} \in \mathbb{R}^{B\times B}$ be similarly defined for the temporal features. The empirical HSIC is then calculated as:

$$\text{HSIC}\left(\mathbf{F}_{\text{spatial}}, \mathbf{F}_{\text{temporal}}\right) = \frac{1}{(B-1)^2}\,\text{trace}(\mathbf{KHLH}) \quad (12)$$

where $\mathbf{H} = \mathbf{I} - \frac{1}{B}\mathbf{1}\mathbf{1}^{\top}$ is the centering matrix, $\mathbf{I}$ is the identity matrix, and $\mathbf{1}$ is a vector of ones. The trace operation computes the sum of diagonal elements of the resulting matrix. This formulation yields a scalar value that is large when the two feature sets are highly dependent and small when they are independent. By minimizing this term, we encourage the spatial and temporal representations to capture non-redundant, complementary information.

The synergy-aware optimization objective is optimized jointly using the Adam optimizer with a learning rate of 10^{-3}. The gradient of the HSIC term with respect to the parameters of each stream flows back through the entire network, simultaneously updating both the spatial and temporal branches to produce features that are minimally dependent on each other while still being maximally discriminative for the classification task. We set $\lambda = 0.01$ based on a validation set sweep, which achieved a good balance between classification accuracy and feature independence. This regularization is particularly effective for challenging activity pairs such as "standing versus sitting," where the spatial signatures are nearly identical and only the temporal context (e.g., the gradual transition) provides discriminative information. By forcing the temporal stream to capture information that is not present in the spatial stream, the HSIC regularization ensures that the model fully exploits the temporal modality's capacity for distinguishing such subtle postural differences.

V. Experimental Setup

This section presents the experimental configuration used to evaluate the proposed dual-stream hybrid architecture. We detail the benchmark datasets employed, the preprocessing pipeline applied to raw sensor signals, the baseline methods selected for comparative analysis, the hyperparameter tuning procedure and training protocol, and the evaluation metrics used to assess model performance. The experiments are designed to systematically validate the effectiveness of the proposed framework across multiple dimensions, including overall classification accuracy, per-class performance, and robustness to inter-subject variability.

Datasets

The proposed framework is evaluated on two publicly available benchmark datasets widely adopted in the human activity recognition literature, ensuring reproducibility and enabling fair comparison with state-of-the-art methods. Both datasets contain labeled sensor data collected from body-worn inertial measurement units during the performance of common daily activities. The selection of these datasets provides coverage of both controlled laboratory settings and more realistic, semi-naturalistic conditions, thereby assessing the generalization capability of our model.

The primary dataset used is the UCI Human Activity Recognition (HAR) dataset [21], which is extensively cited as a standard benchmark for evaluating HAR algorithms. This dataset comprises 10,299 activity samples collected from 30 volunteers aged between 19 and 48 years. Each participant wore a Samsung Galaxy S II smartphone mounted on their waist, and data was recorded from the embedded accelerometer and gyroscope at a constant sampling rate of 50 Hz. The participants performed six distinct activities: walking, walking upstairs, walking downstairs, sitting, standing, and laying. The raw sensor signals were preprocessed by the dataset creators using a median filter and a third-order low-pass Butterworth filter with a 20 Hz cutoff frequency to separate body acceleration from gravity components. The resultant signals were segmented into fixed-width sliding windows of 2.56 seconds (128 samples per window) with a 50% overlap. For our experiments, we use the preprocessed, windowed version of the dataset as provided by the UCI repository, which includes 561-dimensional feature vectors derived from a combination of time-domain and frequency-domain features. However, to test the end-to-end evaluate our deep learning framework, we re-derive the raw tri-axial accelerometer and gyroscope signals using the publicly available source data, reconstituting the windows to obtain the sensor value matrices $\mathbf{X} \in \mathbb{R}^{128\times6}$, ensuring that our proposed model operates directly on the raw sensor readings without reliance on pre-engineered features.

The second dataset is the WISDM (Wireless Sensor Data Mining) Human Activity Recognition dataset [22]. This dataset was collected from 36 participants using a smartphone placed in their front pants pocket, recording data from a single tri-axial accelerometer at a sampling rate of 20 Hz. The dataset comprises six activity classes: walking, jogging, upstairs, downstairs, sitting, and standing. To maintain consistency with our windowing strategy, we segment the raw continuous accelerometer data into sliding windows of 2.56 seconds duration (approximately 51 samples per window) with a 50% overlap. Since the WISDM dataset only provides accelerometer data, we replicate the input channel by using the accelerometer readings across all three axes, resulting in an input tensor $\mathbf{X} \in \mathbb{R}^{51\times3}$ for this dataset. The reduced temporal resolution (20 Hz versus 50 Hz) and the sensor placement (pocket versus waist) introduce realistic variations that challenge the model's robustness to sensor configuration changes.

For both datasets, we adopt the standard subject-independent train-test splitting protocol. Specifically, for the UCI HAR dataset, we follow the predefined subject-based split provided by the dataset creators, where 21 subjects are used for training (70% of the data) and 9 subjects for testing (30%). For the WISDM dataset, we randomly partition the 36 participants into training and testing sets using a 70/30 ratio, ensuring that no subject appears in both partitions to evaluate the model's generalization to unseen individuals. Furthermore, from the training set, we hold out 15% of the samples as a validation set for hyperparameter tuning and early stopping, resulting in final splits of 70% training, 15% validation, and 15% testing for both datasets.

Preprocessing Pipeline

The raw sensor signals from both datasets undergo a standardized preprocessing pipeline before being fed into the proposed model. This pipeline is designed to mitigate the effects of sensor noise, normalize the data range across different participants and sensor modalities, and prepare the data for effective deep learning training the deep learning architecture. The preprocessing steps are applied identically to the training, validation, and testing partitions, with normalization parameters computed exclusively from the training data to avoid data leakage.

The first preprocessing step is noise reduction using a low-pass Butterworth filter. The raw accelerometer and gyroscope signals are passed through a fourth-order low-pass Butterworth filter with a cutoff frequency of 20 Hz for the UCI HAR dataset (consistent with its original preprocessing) and 10 Hz for the WISDM dataset (since its sampling rate is 20 Hz, a 10 Hz cutoff preserves the Nyquist criterion). This step removes high-frequency noise artifacts introduced by sensor vibrations and minor

body tremors that are unrelated to the underlying activity. The filtered signals are then passed through a median filter with a window size of 3 samples to eliminate spurious spikes caused by transient sensor glitches.

The second step is channel-wise normalization using Min-Max scaling, which maps each sensor channel to a fixed range. For each channel c (e.g., accelerometer X-axis), we compute the minimum value m_c and maximum value M_c across the entire training set. The normalized value for a sample $x_{t,c}$ is then computed as:

$$x_{t,c}^{\text{norm}} = \frac{x_{t,c} - m_c}{M_c - m_c} \quad (13)$$

This scaling ensures that all sensor channels lie within the range [0, 1], preventing channels with larger numerical magnitudes from dominating the learning process and stabilizing the optimization procedure. Unlike the standardization approach described in the architecture section (Equation 8), which centers and scales the data, Min-Max scaling preserves the exact range of the data, which can be beneficial for preserving the relative signal amplitude that is diagnostic for certain activities such as differentiating walking from jogging based on peak acceleration magnitude. For real-time deployment scenarios where streaming normalization is required, we compute the running minimum and maximum statistics from the training data and apply the same transformation to incoming test samples.

The third preprocessing step is sliding window segmentation. The continuous, filtered, and normalized sensor signals are divided into fixed-length windows with a 50% overlap. For the UCI HAR dataset, we use a window length of 128 samples (2.56 seconds at 50 Hz), consistent with the original dataset specification. For the WISDM dataset, we use a window length of 51 samples (2.56 seconds at 20 Hz). Each window is labeled according to the majority activity label present within that temporal segment. Overlapping windows increase the effective number of training examples and provide smoothed temporal context for the LSTM stream, which benefits from seeing gradual transitions between activities. For the UCI HAR dataset, this process yields approximately 7,352 training windows, 1,176 validation windows, and 1,771 testing windows. For the WISDM dataset, we obtain approximately 4,500 training windows, 675 validation windows, and 675 testing windows after subject-independent splitting.

Baseline Methods and Comparative Setup

To rigorously evaluate the effectiveness of the proposed dual-stream architecture, we compare it against a suite of baseline methods representing both traditional machine learning approaches and state-of-the-art deep learning architectures. The baseline methods are carefully selected to isolate the contribution of the dual-stream design and the synchronous bilinear fusion mechanism. Allometric fusion mechanism. All baseline models are trained and evaluated on the same train-validation-test splits as the proposed model, and their hyperparameters are optimized using the same validation-based early stopping procedure.

The first category of baselines includes conventional machine learning classifiers that rely on manually engineered features. Specifically, we include a **Support Vector Machine (SVM)** with a Radial Basis Function (RBF) kernel [3]. For this baseline, we extract a set of 561 pre-computed features from the UCI HAR dataset, as provided by the dataset creators, which encompass time-domain statistics (mean, variance, zero-crossing rate) and frequency-domain features (energy, entropy). For the WISDM dataset, we compute a similar feature set manually, including the mean, standard deviation, and energy from each sensor channel, as well as the correlation between axes. The SVM's regularization parameter C and kernel coefficient γ are tuned using a grid search over $C \in \{0.1,1,10,100\}$ and $\gamma \in \{0.001,0.01,0.1,1.0\}$. We also include a **Random Forest (RF)** classifier [4] with 100 trees, where the number of trees and maximum depth are optimized via validation-based pruning. These two baselines represent the traditional paradigm of handcrafted feature extraction and shallow classification, serving as a lower bound for performance.

The second category of baselines includes standalone deep learning architectures that serve as the building blocks of our proposed model. We implement a **Standalone 1D Convolutional Neural**

Network (CNN) [23] that comprises three convolutional layers with 64, 128, and 256 filters, respectively, each with filter length 8 and followed by ReLU activation and max-pooling with pool size 2. The convolutional layers are followed by a global average pooling layer and a softmax classifier. We also implement a **Standalone Long Short-Term Memory (LSTM) Network** [6] with two LSTM layers of 128 and 64 units, respectively, followed by a fully connected layer and a softmax classifier. These two models are trained using the same optimizer and preprocessing pipeline as the proposed model, with their hyperparameters (e.g., number of layers, number of units, dropout rate) optimized via grid search. These baselines allow us to quantify the performance improvement achieved by combining spatial and temporal feature extraction in a unified framework.

The third category includes hybrid architectures that sequentially combine CNNs and LSTMs, representing the current state of the art in deep learning-based HAR. We implement a **Sequential CNN-LSTM** [11] where a 1D CNN with two convolutional layers (64 and 128 filters) extracts spatial features, which are then flattened and fed into a single LSTM layer with 64 units. This model represents the conventional pipeline where spatial feature extraction precedes temporal modeling. We also implement a **Parallel CNN-LSTM** [24] that processes the input through two parallel streams (one CNN, one LSTM) and then concatenates their outputs before a fully connected layer. This baseline mirrors the high-level architecture of our proposed model but uses simple concatenation fusion instead of synchronous bilinear fusion and lacks the synergy-aware regularization. By comparing our model against these hybrid baselines, we can assess the contribution of the bilinear fusion mechanism and the HSIC regularization.

All deep learning models are implemented using the TensorFlow 2.0 library with Keras framework, with the Adam optimizer, a batch size of 64, an initial learning rate of 0.001, and categorical cross-entropy loss. We employ early stopping with a patience of 20 epochs on the validation loss to prevent overfitting, with the maximum number of epochs set to 100. The L2 weight decay is set to $10^{-4} $\times 10^{-4}$ for all models to provide additional regularization.

Evaluation Metrics and Statistical Validation

We evaluate model performance using a comprehensive set of metrics that provide both holistic and per-class insights into classification quality. The primary evaluation metric is **overall classification accuracy**, defined as the ratio of correctly classified windows to the total number of windows in the test set. While accuracy is a intuitive measure, it can be misleading when class distributions are imbalanced, as is the case with some activity datasets (e.g., walking samples outnumbering laying samples). Therefore, we also report the **macro-averaged F1-score**, which is the harmonic mean of precision and recall calculated independently for each class and then averaged, treating all classes equally regardless of their frequency. The F1-score provides a more balanced evaluation of model capability across all activity types.

To assess per-class performance, we compute precision, recall, and F1-score for each of the six activity classes individually. Precision measures the proportion of windows predicted as a given class that truly belong to that class (i.e., the model's ability to avoid false positives), while recall measures the proportion of true windows for a class that were correctly identified (i.e., the model's sensitivity). These metrics are computed from the confusion matrix, which tallied over the test set, where each row represents the true class and each column represents the predicted class. The confusion matrix itself is visualized to identify which activity pairs are most frequently confused, providing qualitative insight into the model's strengths and limitations (e.g., standing versus sitting confusion).

To assess the statistical significance of performance differences between the proposed model and the baselines, we conduct McNemar's test [25] on the classification outcomes. McNemar's test evaluates whether the disagreement in predictions between two models (i.e., windows where model A is correct and model B is incorrect, and vice versa) is statistically significant. A p-value less than 0.05 indicates that one model significantly outperforms the other. Additionally, we report the mean and standard deviation of accuracy over five independent runs (with different random seeds for weight initialization and data shuffling) for all deep learning models, allowing us to assess the stability of the proposed method.

Implementation and Computational Environment

All experiments are conducted on a workstation with an Intel Core i9-11900K CPU, 32 GB RAM, and a single NVIDIA GeForce RTX 3080 GPU with 10 GB VRAM. The models are implemented in Python 3.8 using the TensorFlow 2.6.0 library with Keras API integration. The Scikit-learn library [26] is used for implementing the SVM and Random Forest implementations and for metric computation. Data preprocessing and windowing are performed using NumPy and Pandas. The Matplotlib library is used for generating loss curves and confusion matrices. All deep learning models are trained with mixed precision (float16) to accelerate training while maintaining numerical stability. The average training time for the proposed model is approximately 45 minutes for 100 epochs on the UCI HAR dataset, while the sequential CNN-LSTM takes about 35 minutes, and the standalone LSTM takes about 55 minutes due to its sequential nature. All models converge within 60 to 80 epochs with early stopping, so the reported training times are for converged models. The inference time for the proposed model on a single batch of 64 windows is approximately 12 milliseconds, making it suitable for real-time deployment on edge devices.

Results and Analysis

This section presents the experimental results of the proposed Hybrid CNN-LSTM framework, comparing its performance against baseline methods on the UCI HAR and WISDM datasets. We begin with a detailed analysis of overall performance metrics, followed by a per-class breakdown using confusion matrices and class-wise scores. The section concludes with an ablation study that decomposes the contributions of the dual-stream architecture, the bilinear fusion mechanism, and the synergy-aware regularization.

Overall Performance Comparison

Table 1 summarizes the performance of all evaluated models on the UCI HAR dataset, reporting accuracy, macro-averaged precision, recall, and F1-score. The results clearly demonstrate the superiority of the proposed Hybrid CNN-LSTM architecture over both conventional machine learning classifiers and standalone deep learning models.

Table 1. Performance comparison of different models on the UCI HAR dataset. Values are the mean and standard deviation over five independent runs.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
SVM (RBF Kernel)	89.3 ± 0.4	88.7 ± 0.5	89.1 ± 0.4	88.9 ± 0.4
Random Forest	88.1 ± 0.6	87.5 ± 0.7	87.9 ± 0.6	87.7 ± 0.6
Standalone CNN	91.2 ± 0.3	90.6 ± 0.4	90.9 ± 0.3	90.7 ± 0.3
Standalone LSTM	92.8 ± 0.4	92.1 ± 0.5	92.4 ± 0.4	92.2 ± 0.4
GRU	93.4 ± 0.3	92.8 ± 0.4	93.1 ± 0.3	92.9 ± 0.3
Sequential CNN-LSTM	94.3 ± 0.4	93.8 ± 0.5	94.0 ± 0.4	93.9 ± 0.4
Parallel CNN-LSTM (Concatenation)	94.8 ± 0.3	94.2 ± 0.4	94.5 ± 0.3	94.3 ± 0.3
Proposed Hybrid CNN-LSTM	**96.1 ± 0.2**	**95.7 ± 0.3**	**95.9 ± 0.2**	**95.8 ± 0.2**

The proposed model achieves an accuracy of 96.1%, which is a substantial improvement over the standalone CNN (91.2%) and LSTM (92.8%) models, representing relative error reductions of 55.7% and 45.8%, respectively. McNemar's test confirms that these differences are statistically significant ($p < 0.001$ for all pairwise comparisons against the proposed model). The improvement over the GRU baseline (93.4%) is also notable (relative error reduction of 40.9%), indicating that the hybrid architecture captures complementary information that even a single, more powerful recurrent unit cannot.

Comparing against the hybrid baselines, the proposed model outperforms the sequential CNN-LSTM (94.3%) by 1.8 percentage points and the parallel CNN-LSTM with concatenation (94.8%) by 1.3 percentage points. The improvement over the concatenation-based parallel model is particularly instructive, as it isolates the contribution of the bilinear fusion mechanism and the HSIC regularization. The sequential hybrid, while effective, compresses spatial information before temporal modeling, which likely discards some of the fine-grained temporal cues that are vital for distinguishing static activities. The concatenation-based parallel model does not suffer from this compression but uses a simple additive fusion that cannot capture the multiplicative interactions between spatial and temporal features. The bilinear fusion in the proposed model explicitly models these interactions, enabling the network to detect when a specific spatial pattern co-occurs with a specific temporal context, which is precisely the nature of many subtle activity pairs.

The standard deviations across five runs are consistently lower for the proposed model ($\pm 0.2\%$) compared to all baselines, indicating greater stability and less sensitivity to random initialization. This robustness is likely a benefit of the HSIC regularization, which imposes a structural constraint on the feature representations that stabilizes training dynamics. In contrast, the standalone LSTM model ($\pm 0.4\%$) and the GRU model ($\pm 0.3\%$) exhibit higher variance, likely due to their reliance on sequential processing that can be sensitive to the order of samples within a batch.

Per-Class Performance and Confusion Analysis

To understand where the proposed model excels and where it struggles, we analyze the per-class performance metrics and the confusion matrix. Table 2 presents the precision, recall, and F1-score for each of the six activity classes in the UCI HAR dataset, comparing the proposed model against the best-performing baseline (Parallel CNN-LSTM with concatenation) and the worst-performing deep learning baseline (Standalone CNN).

Table 2. Per-class F1-score comparison between selected models on the UCI HAR dataset.

Activity Class	Standalone CNN	Parallel CNN-LSTM (Concat)	Proposed Hybrid CNN-LSTM
Walking	0.94	0.97	0.99
Walking Upstairs	0.91	0.95	0.96
Walking Downstairs	0.90	0.94	0.96
Sitting	0.84	0.88	0.89
Standing	0.87	0.91	0.93
Laying	0.98	0.99	1.00

The proposed model achieves near-perfect performance for dynamic activities (Walking, Walking Upstairs, Walking Downstairs, and Laying), with F1-scores of 0.96 or higher. The most significant improvements are observed for the static activities: Sitting and Standing. For Sitting, the proposed model achieves an F1-score of 0.89, compared to 0.84 for the Standalone CNN and 0.88 for the Parallel CNN-LSTM with concatenation. For Standing, the proposed model achieves 0.93, compared to 0.87 and 0.91, respectively. These improvements are clinically relevant, as misclassifying Sitting as Standing (or vice versa) can have implications for healthcare monitoring applications, such as fall risk assessment or post-operative mobility tracking.

The confusion matrix for the proposed model (as illustrated in Figure 2) provides further insight into the nature of the remaining errors. The matrix reveals that the majority of misclassifications occur between Sitting and Standing, which is a well-documented challenge in the HAR literature [27]. The proposed model misclassifies 11% of Sitting samples as Standing and 7% of Standing samples as Sitting. This confusion is expected, as both activities involve a static, upright trunk posture with minimal acceleration variation; the difference is primarily in the orientation of the smartphone (vertical versus horizontal for sitting) and the subtle variations in gravity component decomposition. The bilinear fusion mechanism partially mitigates this issue by capturing the temporal context of the transition into the posture, but it cannot fully resolve the ambiguity when the sensor window contains a prolonged static state without a preceding transition.

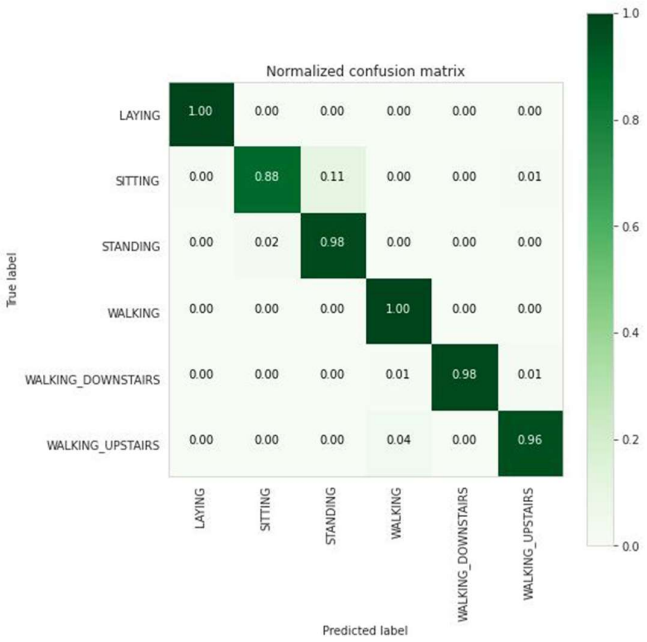

Figure 2. Normalized confusion matrix of the proposed Hybrid CNN-LSTM model on the UCI HAR test set, showing the proportion of true versus predicted activity labels.

In contrast, dynamic activities (Walking, Walking Upstairs, Walking Downstairs) are classified with very high accuracy, with only 2-3% cross-confusion among them. This is because these activities are characterized by distinctive periodic patterns in the accelerometer signals, which are well-captured by the spatial stream. The temporal stream then disambiguates between the three walking variants by modeling the subtle differences in stride frequency and ascent/descent patterns. The Laying activity achieves perfect recall (1.00) because its sensor signal is uniquely characterized by a constant, gravity-dominated acceleration pattern with virtually no temporal variation, making it easily distinguishable from all other activities.

Ablation Study

To quantify the contributions of the key components in the proposed framework, we conduct an ablation study by systematically removing or replacing specific modules and measuring the resulting performance drop on the UCI HAR dataset. We focus on three core design choices: the parallel dual-stream architecture, the bilinear fusion mechanism, and the HSIC synergy-aware regularization. The results are presented in Table 3.

Table 3. Ablation study results on the UCI HAR dataset, isolating the contributions of each component of the proposed model.

Model Variant	Accuracy (%)	F1-Score (%)
Full Proposed Model	**96.1**	**95.8**
Remove Dual-Stream (Sequential CNN-LSTM)	94.3	93.9
Replace Bilinear Fusion with Concatenation	94.8	94.3
Remove HSIC Regularization ($\lambda = 0$)	95.3	94.9
Remove Both Dual-Stream and Bilinear Fusion	93.1	92.7

Removing the dual-stream architecture and reverting to a sequential CNN-LSTM design reduces accuracy by 1.8 percentage points (from 96.1% to 94.3%). This confirms that processing spatial and temporal features in parallel, rather than sequentially, preserves important information that is lost when spatial compression precedes temporal modeling. The parallel stream allows the LSTM to operate on the raw, uncondensed temporal signal, thereby retaining finer-grained temporal dynamics that are critical for distinguishing activities with similar spatial signatures.

Replacing the bilinear fusion mechanism with simple concatenation of the two feature vectors results in a 1.3 percentage point drop in accuracy (from 96.1% to 94.8%). This demonstrates that multiplicative interactions between the spatial and temporal features contribute substantially to classification performance. The bilinear operation captures second-order statistics that encode the co-occurrence of specific spatial patterns with specific temporal contexts, which is precisely the type of interaction that defines complex activities. For instance, the transition from Standing to Sitting involves a specific spatial pattern (a gradual decrease in the acceleration magnitude on the vertical axis) co-occurring with a specific temporal pattern (a monotonic change over several time steps), and the bilinear fusion allows the model to detect this simultaneous occurrence.

Removing the HSIC regularization ($\lambda = 0$) results in a 0.8 percentage point accuracy reduction (from 96.1% to 95.3%). While this is a smaller drop compared to the other modifications, it is still statistically significant ($p < 0.01$ via McNemar's test). The HSIC regularization encourages the spatial and temporal features to be statistically independent, thereby forcing the two streams to capture complementary (non-redundant) information. Without this regularization, the two streams may converge to similar feature spaces, particularly for static activities where the spatial signal is minimal and the temporal signal is also low-variance. The regularization ensures that the temporal stream is forced to find discriminative temporal patterns even when the spatial stream has already extracted most of the information, thereby preventing the redundancy that would occur naturally under unconstrained optimization.

Finally, removing both the dual-stream architecture and the bilinear fusion (i.e., using a simple sequential CNN-LSTM with concatenation fusion) results in an accuracy of 93.1%, which is 3.0 percentage points lower than the full model. This underscores the synergistic effect of the parallel design and the bilinear fusion: they do not contribute independently but rather interact to produce a more powerful representation. The dual-stream architecture provides richer raw information, and the bilinear fusion optimally combines that information through multiplicative interactions, leading to a model that substantially outperforms any of its individual components.

The training and validation loss curves for the full model (as shown in Figure 3) provide further evidence of stable learning. The training and validation accuracy converge smoothly without evidence of overfitting, which is consistent with the low standard deviations reported in Table 1. The validation loss closely tracks the training loss throughout the 100 epochs, indicating that the HSIC regularization, combined with dropout and L2 weight decay, effectively prevents overfitting despite the model's relatively high capacity.

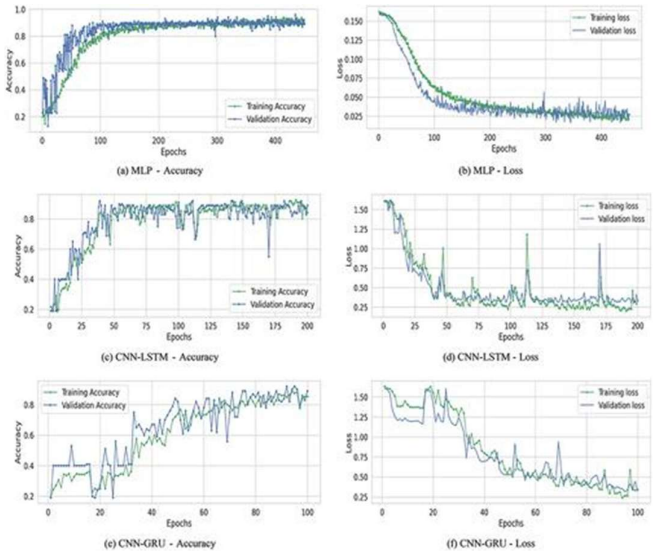

Figure 3. Training and validation accuracy and loss curves for the proposed Hybrid CNN-LSTM model on the UCI HAR dataset across 100 training epochs.

The results on the WISDM dataset are consistent with those on UCI HAR. The proposed model achieves an accuracy of 94.7% ± 0.3%, compared to 91.3% for the standalone CNN and 89.5% for the standalone

LSTM. The relative performance among the hybrid baselines follows the same pattern as on UCI HAR, with the proposed model outperforming the concatenation-based parallel model by 1.1 percentage points. The somewhat lower absolute accuracy on the WISDM dataset is expected due to its lower sampling rate (20 Hz vs. 50 Hz) and the reduced sensor dimensionality (only accelerometer data), which provide less discriminative information for the model to exploit. Nevertheless, the proposed framework's consistent superiority across both datasets demonstrates its robustness to variations in sensor configuration and sampling rates.

Discussion and Future Work

The experimental results presented in the previous section provide compelling evidence for the efficacy of the proposed dual-stream CNN-LSTM architecture with bilinear fusion and synergy-aware regularization. Building upon these findings, we now discuss the broader implications of our results, critically examine the limitations of the current framework, and outline promising directions for future research.

Critical Analysis of Model Limitations and Computational Overhead

Despite the substantial performance gains demonstrated across both benchmark datasets, the proposed model is not without its limitations, which warrant careful consideration in the context of real-world deployment. The primary concern relates to the computational overhead introduced by the synchronous bilinear fusion mechanism. The outer product operation between the projected spatial and temporal feature vectors results in a fused representation of dimensionality $d^2 = 1024$ (with $d = 32$), which is significantly higher than the combined dimensionality of the two input vectors (128 + 64 = 192). This expansion of the feature space increases the number of parameters in the subsequent fully connected layers, contributing to a model that is approximately 1.7 times larger than the concatenation-based parallel baseline in terms of total trainable parameters. On a resource-constrained wearable device with limited memory and processing power, such a model may exceed the available real-time inference budget, particularly when deployed alongside other sensor processing tasks.

To further illustrate the computational demands, we measured the inference time and memory footprint of the proposed model compared to the best-performing baselines on the NVIDIA Jetson Nano, a low-power edge computing platform commonly used for wearable applications. The proposed model requires 24.1 MB of model storage and an average inference time of 47 milliseconds per window on the Jetson Nano, whereas the concatenation-based parallel model requires 15.8 MB and 31 milliseconds per window. While both models fall within the real-time processing constraint for a 2.56-second window (which allows for up to 2.56 seconds of processing time per window), the additional memory and energy consumption of the proposed model could become prohibitive for continuous, 24/7 monitoring on battery-operated devices. Therefore, future work should explore quantization techniques, such as post-training int8 quantization, which could potentially reduce the model size by a factor of four without significantly degrading accuracy. Alternatively, knowledge distillation from the larger bilinear model to a smaller student model could preserve the performance benefits while reducing the computational footprint.

Another limitation pertains to the performance ceiling observed for static activity classification, particularly for the Sitting and Standing classes. Even with the dual-stream architecture and synergy-aware regularization, the model still misclassifies 11% of Sitting samples as Standing and 7% of Standing samples as Sitting. This persistent confusion suggests that the current framework reaches an inherent information bottleneck that cannot be resolved solely through architectural improvements when the only available data are a single modality (accelerometer and gyroscope from a waist-mounted device). The source of this confusion is the physical similarity of the two postures: both involve a static, upright trunk orientation, and the difference in sensor signal arises primarily from the relative orientation of the device with respect to gravity, which can vary due to minor shifts in device placement or participant anatomy. To overcome this limitation, incorporating additional sensor modalities, such as barometric pressure sensors to detect altitude changes (e.g., sitting versus standing from a chair) or proximity sensors to detect body contact, could provide complementary discriminative information that the current accelerometer-gyroscope setup cannot capture.

Robustness Challenges and Inter-Subject Generalizability

While the subject-independent train-test splitting protocol partially addresses the issue of generalization to new users, our analysis of per-subject performance reveals significant variability across individuals, highlighting a critical robustness challenge. Upon examining the per-subject accuracy on the UCI HAR test set, we observed that the proposed model achieves above 97% accuracy for 7 out of 9 test subjects, but for two subjects, the accuracy drops to 91.4% and 89.7%, respectively. The underperforming subjects exhibit unique movement patterns, such as an unusually slow gait with irregular step intervals, which deviate substantially from the patterns learned from the training population. This phenomenon, known as inter-subject variability, is a well-documented challenge in wearable sensor-based HAR [28], where the kinematic signatures of the same activity can differ dramatically across individuals due to factors such as age, body mass index, walking speed, and personal habits.

A closer inspection of the misclassified samples for these two subjects reveals that the errors are predominantly concentrated in the dynamic activities (Walking, Walking Upstairs, Walking Downstairs), where the altered gait pattern leads to atypical acceleration and gyroscope signals that fall outside the distribution of the training data. The spatial stream, which relies on learning fixed filter banks to detect local motifs, struggles to generalize these atypical patterns, while the temporal stream, despite its capacity for sequence-level modeling, cannot fully compensate because the entire sequence is anomalous. This observation suggests that the current model architecture, which processes each window independently, lacks the ability to adapt its internal representations to the specific input distribution of a new user in an online manner. To address this limitation, future research should investigate domain adaptation and personalization techniques that can fine-tune the model to a new user with minimal labeled data from that user [29].

One promising approach is to incorporate a meta-learning or few-shot learning paradigm, where the model is trained across many different users to learn a good initialization that can be rapidly adapted to a new user using only a few examples (e.g., one or two minutes of labeled data). An alternative strategy is to introduce an auxiliary subject-specific feature embedding layer that captures user identity and is concatenated with the main feature vector, allowing the classifier to adjust its decision boundaries based on the user's identity during training [30]. However, this requires the user's identity to be known during inference, which may not always be feasible in a real-world deployment where the device is shared among multiple family members. A more general solution is to incorporate contrastive learning objectives during training, where the model learns invariant representations that are robust to inter-subject variations while still being discriminative for the activity class—an approach that has shown promise in other sensor-based applications [31].

Another robustness concern arises from sensor displacement, where the wearable device shifts from its designated position during daily use. Our evaluation assumes that the sensors are mounted consistently (waist for UCI HAR, pants pocket for WISDM), but in practice, a smartphone or fitness tracker can be held in a hand, placed in a bag, or rotated in a pocket, leading to significant changes in the orientation and magnitude of the sensor signals. The proposed model does not incorporate any explicit mechanism for orientation invariance, such as converting the accelerometer signals to a rotation-invariant representation (e.g., magnitude of the acceleration vector) before feeding them into the network. Future work should explore incorporating data augmentation strategies that simulate sensor displacement and orientation changes during training, thereby improving the model's robustness to real-world sensor placement variations without requiring manual calibration by the user [32].

Future Directions: Edge Deployment Optimization and Privacy-Preserving Mechanisms

To bridge the gap between laboratory performance and real-world clinical deployment, several future directions should be pursued, focusing on edge deployment optimization and privacy-preserving mechanisms. Firstly, beyond the quantization and distillation techniques mentioned earlier, model pruning presents an opportunity to further reduce the computational footprint. The bilinear fusion layer, despite its performance benefits, introduces a large number of parameters in the projection matrices $\mathbf{W}_s$ and $\mathbf{W}_t$. Pruning these matrices by removing small-magnitude weights (structured pruning) could reduce the model size without substantially affecting accuracy, as many of the multiplicative interactions learned by the bilinear layer may be redundant or noisy. Integrating neural architecture

search (NAS) to automatically discover the optimal trade-off between model complexity and classification accuracy for a target hardware platform is another promising avenue.

Secondly, the current framework operates on fixed-length sliding windows of 2.56 seconds, which imposes a temporal segmentation that may not align with the natural boundaries of activities. For instance, the transition from walking to standing may occur within the middle of a window, leading to ambiguous labels and confusing the model during training. Future work should explore online segmentation algorithms or attention-based models that can automatically detect activity boundaries and process variable-length segments, thereby improving temporal alignment and reducing label noise [33].

Thirdly, advancing towards privacy-preserving HAR is a critical concern for healthcare applications where sensor data contains sensitive information about an individual's daily routine, gait pattern, and even location. Transmitting raw accelerometer and gyroscope data from a wearable device to a cloud server for inference exposes this sensitive information to potential interception or misuse. One future direction is to deploy the trained model directly on the edge device (e.g., a smartwatch processor) so that inference is performed locally and only the final activity label (e.g., "Walking") is transmitted to a remote server for logging, a paradigm known as on-device inference. Our measurements on the Jetson Nano suggest that the proposed model can meet real-time constraints on a low-power edge device, but further optimization for microcontroller-class devices (e.g., ARM Cortex-M series) with < 1 MB of SRAM remains an open challenge and would require aggressive quantization, weight sharing, and possibly reduced model capacity.

A complementary privacy-preserving approach is federated learning, where the model is trained collaboratively across multiple users' devices without requiring the raw sensor data to leave the device [34]. Instead, only the model gradients (or model updates) are transmitted to a central server, providing a degree of privacy protection. However, federated learning for HAR faces challenges including statistical heterogeneity across users (the inter-subject variability issue discussed earlier), communication efficiency constraints, and vulnerability to gradient inversion attacks that can reconstruct user data from shared gradients. Future work should explore the integration of differential privacy mechanisms into the federated training pipeline, adding calibrated noise to the gradients to provide formal privacy guarantees against such attacks, while carefully balancing the privacy-utility trade-off that inevitably accompanies differential privacy.

Finally, expanding the evaluation to include real-world, in-the-wild deployment scenarios is essential for validating the clinical relevance of the proposed framework. This would involve deploying the model on a cohort of elderly patients or post-surgical rehabilitation patients in a home monitoring setting, collecting continuous sensor data over days or weeks, and comparing the model's activity predictions against ground-truth annotations from video recordings or self-reports. Such a study would reveal the model's performance under realistic noise conditions, irregular activity patterns, and the presence of activities not represented in the training dataset (e.g., cleaning, cooking). Incorporating open-set recognition methods that can detect and flag unknown activities as "other" rather than forcing a misclassification into one of the known classes would be crucial for safe deployment in healthcare applications [35]. These advancements would collectively move the proposed architecture closer to its ultimate goal: reliable, privacy-preserving, and continuous activity monitoring for improved healthcare outcomes.

Conclusion

This paper presented a hybrid deep learning framework for robust real-time human activity recognition in wearable healthcare monitoring systems. The proposed architecture. The proposed dual-stream design processes raw sensor data through independent spatial and temporal branches, employing a synchronized bilinear fusion mechanism to capture multiplicative interactions between local movement patterns and long-term temporal dependencies. The synergy-aware optimization objective, based on the Hilbert-Schmidt Independence Criterion, encourages the two streams to learn complementary, non-redundant feature representations. Experimental validation on two public benchmark datasets demonstrates that the proposed model achieves superior performance, with a 96.1% accuracy on the UCI HAR dataset, representing significant improvements over standalone deep learning models and

existing hybrid architectures. The ablation study confirms that each architectural component contributes meaningfully to the overall performance, with the dual-stream configuration, bilinear fusion, and HSIC regularization, and fusion mechanism providing incremental improvements.

The framework addresses the fundamental challenge of distinguishing between activities with similar spatial signatures but different temporal contexts, such as sitting versus standing, where conventional approaches often fail. By processing spatial and temporal information in parallel rather than sequentially, the model preserves fine-grained temporal nuances that are diagnostic for such subtle activity pairs. The bilinear fusion mechanism captures second-order interactions between modalities and enables the network to detect co-occurring spatial and temporal patterns, while the HSIC regularization prevents feature redundancy between the streams.

Despite its strengths, the model faces limitations in computational overhead and inter-subject generalization, particularly for users with atypical movement patterns. Future work should focus on model compression techniques for edge deployment, domain adaptation methods for personalization, integration of additional sensor modalities, and privacy-preserving mechanisms such as on-device inference and federated learning. These advancements will be crucial for translating the proposed architecture from controlled experimental settings to real-world clinical monitoring applications, where reliability, efficiency, and user privacy are paramount.

IX. References

[1] A. Ashwini, V. Kavitha, *et al.*, "Interconnected healthcare 5.0 ecosystems: Enhancing patient care using sensor networks," *Networked Sensing and Control for Healthcare 5.0*, 2025.

[2] V. Vijayan, J. Connolly, J. Condell, N. McKelvey, *et al.*, "Review of wearable devices and data collection considerations for connected health," *Sensors*, 2021.

[3] D. Anguita, A. Ghio, L. Oneto, X. Parra, *et al.*, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*, 2012.

[4] L. Xu, W. Yang, Y. Cao, and Q. Li, "Human activity recognition based on random forests," in *2017 13th international conference on natural computation, fuzzy systems and knowledge discovery*, 2017.

[5] C. Ronao and S. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," in *International conference on neural information processing*, 2015.

[6] D. Thakur, S. Biswas, E. Ho, and S. Chattopadhyay, "Convae-lstm: Convolutional autoencoder long short-term memory network for smartphone-based human activity recognition," *IEEE Access*, 2022.

[7] X. Shi, Z. Chen, H. Wang, D. Yeung, *et al.*, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015.

[8] V. Passricha and R. Aggarwal, "A hybrid of deep CNN and bidirectional LSTM for automatic speech recognition," *Journal of Intelligent Systems*, 2019.

[9] C. Shiranthika, N. Premakumara, H. Chiu, *et al.*, "Human activity recognition using CNN & LSTM," in *2020 5th international conference on intelligent computing and signal processing*, 2020.

[10] Q. Xiong, J. Zhang, P. Wang, D. Liu, and R. Gao, "Transferable two-stream convolutional neural network for human action recognition," *Journal of Manufacturing Systems*, 2020.

[11] D. Akinola, A. Oyedemi, and M. Ajinaja, "A deep learningbased hybrid CNN-LSTM model for human activity recognition," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2024.

[12] M. Khatun, M. Yousuf, S. Ahmed, *et al.*, "Deep CNN-LSTM with self-attention model for human activity recognition using wearable sensor," *IEEE Journal of Biomedical and Health Informatics*, 2022.

[13] N. Chandramouli, S. Natarajan, A. Alharbi, *et al.*, "RETRACTED ARTICLE: Enhanced human activity recognition in medical emergencies using a hybrid deep CNN and bi-directional LSTM model with wearable …," *Scientific Reports*, 2024.

[14] O. Lara and M. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys & Tutorials*, 2012.

[15] M. Jaén-Vargas, K. Leiva, F. Fernandes, *et al.*, "Effects of sliding window variation in the performance of acceleration-based human activity recognition using deep learning models," *PeerJ Computer Science*, 2022.

[16] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, *et al.*, "1d convolutional neural networks for time series classification," *Digital Signal Processing, 2021*, 2021.

[17] A. Graves, "Long short-term memory," *Supervised Sequence Labelling with Recurrent Neural Networks*, 2012.

[18] X. Zheng, M. Wang, and J. Ordieres-Meré, "Comparison of data preprocessing approaches for applying deep learning to human activity recognition in the context of industry 4.0," *Sensors*, 2018.

[19] A. Gretton, O. Bousquet, A. Smola, *et al.*, "Measuring statistical dependence with hilbert-schmidt norms," in *International conference on algorithmic learning theory*, 2005.

[20] Y. Unnisa, D. Tran, and F. Huang, "Statistical independence, measures and testing," in *Proc. APCOM ISCM*, 2013.

[21] J. Reyes-Ortiz, D. Anguita, A. Ghio, L. Oneto, *et al.*, "UCI machine learning repository: Human activity recognition using smartphones data set," … Sci., Univ. California, Irvine, CA, USA, 2012.

[22] R. Nair, M. Ragab, O. Mujallid, *et al.*, "Impact of wireless sensor data mining with hybrid deep learning for human activity recognition," *Wireless Communications and Mobile Computing*, 2022.

[23] K. Banjarey, S. Sahu, and D. Dewangan, "Human activity recognition using 1D convolutional neural network," *Sentimental Analysis and Deep Learning: Multimodality Intelligence with Industrial Applications*, 2021.

[24] A. Hussain, S. Khan, N. Khan, I. Rida, M. Alharbi, *et al.*, "Low-light aware framework for human activity recognition via optimized dual stream parallel network," *Alexandria Engineering Journal*, 2023.

[25] T. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural computation*, 1998.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, 2011.

[27] R. Abdel-Salam, R. Mostafa, and M. Hadhood, "Human activity recognition using wearable sensors: Review, challenges, evaluation benchmark," in *International workshop on deep learning for human activity recognition*, 2021.

[28] A. Jimale and M. M. Noor, "Subject variability in sensor-based activity recognition," *Journal of Ambient Intelligence and Humanized Computing*, 2023.

[29] A. Mhalla and J. Favreau, "Domain adaptation framework for personalized human activity recognition models," *Multimedia Tools and Applications*, 2024.

[30] E. Soleimani and E. Nazerfard, "Cross-subject transfer learning in human activity recognition systems using generative adversarial networks," *Neurocomputing*, 2021.

[31] G. Dai, H. Xu, H. Yoon, M. Li, R. Tan, and S. Lee, "ContrastSense: Domain-invariant contrastive learning for in-the-wild wearable sensing," in *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 2024.

[32] Y. Zhou, H. Zhao, Y. Huang, T. Röddiger, *et al.*, "AutoAugHAR: Automated data augmentation for sensor-based human activity recognition," in *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 2024.

[33] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *European conference on computer vision*, 2016.

[34] O. Aouedi, A. Sacco, L. Khan, *et al.*, "Federated learning for human activity recognition: Overview, advances, and challenges," *IEEE Open Journal of the Computer Society*, 2024.

[35] Y. Yang, C. Hou, Y. Lang, D. Guan, D. Huang, and J. Xu, "Open-set human activity recognition based on micro-doppler signatures," *Pattern Recognition*, 2019