Applied Computing and Intelligence

https://www.aimspress.com/journal/aci

Applied Computing and Intelligence
5(2): 286–300.
DOI: 10.3934/aci.2025016
Received: 25 June 2025
Revised: 09 December 2025
Accepted: 11 December 2025
Published: 24 December 2025

Research article

Multilevel neural networks with dual-stage feature fusion for human activity recognition

Abeer FathAllah Brery[1,*], **Ascensión Gallardo-Antolín**[2], **Israel Gonzalez-Carrasco**[1] **and Mahmoud Fakhry**[3,*]

[1] Departamento de Informática, Universidad Carlos III de Madrid, Avenida de la Universidad, 30, Leganés, Madrid, 28911, Spain

[2] Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid, Avenida de la Universidad, 30, Leganés, Madrid, 28911, Spain

[3] CEIEC, Universidad Francisco de Vitoria, Ctra.M-515 Pozuelo-Majadahonda Km.1, 800, Pozuelo de Alarcón, Madrid, 28223, Spain

* **Correspondence:** Email: abeer_brery@aswu.edu.eg, mahmoud.fakhry@ufv.es.

Academic Editor: Xuesong Zhai

Abstract: Human activity recognition (HAR) refers to the process of identifying human actions and activities using data collected from sensors. Neural networks, such as convolutional neural networks (CNNs), long short-term memory (LSTM) networks, convolutional LSTM, and their hybrid combinations, have demonstrated exceptional performance in various research domains. Developing a multilevel individual or hybrid model for HAR involves strategically integrating multiple networks to capitalize on their complementary strengths. The structural arrangement of these components is a critical factor influencing the overall performance. This study explored a novel framework of a two-level network architecture with dual-stage feature fusion: late fusion, which combines the outputs from the first network level, and intermediate fusion, which integrates the features from both the first and second levels. We evaluated 15 different network architectures of CNNs, LSTMs, and convolutional LSTMs, incorporating late fusion with and without intermediate fusion, to identify the optimal configuration. Experimental evaluation on two public benchmark datasets demonstrated that architectures incorporating both late and intermediate fusion achieve higher accuracy than those relying on late fusion alone. Moreover, the optimal configuration outperformed baseline models, thereby validating its effectiveness for HAR.

Keywords: human activity recognition; HAR; CNN; LSTM; convolutional LSTM; USC-HAD dataset; UCI-HAR dataset

1. Introduction

Human activity recognition (HAR) plays a crucial role in numerous domains, including healthcare monitoring, security systems, intelligent environments, and surveillance applications, where accurate interpretation of human movements is critical [1, 2]. In recent years, deep learning models have significantly advanced the field by eliminating the need for manual feature engineering and achieving high classification performance through the automatic extraction of complex, discriminative features from raw sensor data. Despite the proliferation of various deep learning architectures for HAR, a systematic and comprehensive evaluation of their comparative performance is often lacking [3, 4]. Such assessments are crucial for understanding the strengths and limitations of each model and their generalizability, scalability, and applicability to different types of sensor data, activity categories, and deployment environments in the real world. A thorough comparative analysis can guide researchers and practitioners in selecting the most suitable models for specific HAR scenarios, ultimately contributing to the advancement and practical implementation of robust systems.

Among deep learning models, convolutional neural network (CNN) and long short-term memory (LSTM) architectures are the most widely used. Hybrid models are often designed to outperform individual models by offering benefits such as reduced computation time and the capability to leverage data from various sensor positions [5]. The ability of CNN-LSTM hybrid models to capture both spatial and temporal dependencies is typically achieved by using CNN layers for feature extraction from the input data, followed by LSTM layers for sequence modeling. The process of merging features from different layers or modalities, known as feature fusion, has been explored in recent models and is commonly implemented using operations such as addition or concatenation [6].

A comprehensive review of deep learning models used in smartphone and wearable sensor-based recognition systems was provided in [5]. These include models such as CNNs, LSTMs, and various hybrid architectures, each of which is discussed in terms of its unique characteristics, strengths, and limitations. In [7], a heterogeneous convolution approach divides the kernels in a CNN into two groups: one that recalibrates the other group. A dynamic CNN introduces dynamic kernels with attention that adapt weights [8], and deep convolution constructs an ensemble stream employing late fusion [9]. In [10], the authors introduced multiscale hierarchical CNNs that incorporate adaptive feature fusion and dynamic channel selection based on LSTM. Hybrid CNN-LSTM models, as proposed in [11, 12], are designed to capture spatio-temporal dynamics across multiple sensors. These models can identify key feature embeddings by incorporating self-attention mechanisms into their architecture.

The hierarchical deep LSTM (H-LSTM) model introduced in [13] uses the characteristics of the time-frequency domain for HAR. A multi-head CNN architecture was proposed in [14], where three parallel CNNs processed data from different sensors. The outputs were concatenated and passed through the LSTM and dense layers of the model. In [15], CNN and LSTM models were evaluated for HAR applications. The raw signals were preprocessed using a Butterworth filter, and nine features were extracted from a 128-sample window. The study in [16] proposed a CNN combined with LSTM to extract features and capture temporal dependencies from accelerometer and gyroscope data. Finally, in [17], a multilevel feature fusion strategy was introduced for multidimensional HAR. This approach employed a multi-head CNN for visual input and a CNN-LSTM combination to extract temporal features from multisensor time-series data. The architecture incorporates three CNN branches with a channel attention module to enhance the representation of the channel and spatial characteristics.

This study investigates two-level network architectures that employ a feature fusion strategy to integrate features from the same or different network levels for HAR. The experiments explore different combinations of CNN, LSTM, and convolutional LSTM (CLSTM) [18]. CLSTM differs from conventional CNN-LSTM architectures by integrating convolutional operations directly into the recurrent structure, thereby forming a unified spatio-temporal architecture. This study is the first to apply CLSTM in individual and hybrid configurations for HAR applications.

2. Datasets

The rapid growth of wearable technology has allowed the development of various HAR datasets; however, challenges in standardization, sharing, and accessibility often limit their reusability and reproducibility [19]. Choosing an appropriate dataset for a given HAR task involves considering multiple factors, such as the number of participants, the variety of activities, sensor modalities, and the recording environment. To evaluate the performance of the different models and ensure broad applicability and benchmark performance, we selected two well-established and widely used datasets: the USC-HAD dataset [20] and the UCI-HAR dataset [21]. These datasets encompass daily living activities and offer diverse activity labels and robust sensor configurations that are suitable for evaluating deep models.

2.1. USC-HAD

The University of Southern California Human Activity Dataset (USC-HAD) is a resource for research on human activity in the ubiquitous computing community [20]. This dataset includes 14 subjects and 12 daily activities, with sensor hardware attached to the right front hip of the subjects. Sensor recordings are the most basic and common human activities, including walking, running, jumping, sitting, sleeping, and using an elevator. To capture variations in activity, each subject was asked to perform 5 trials for each activity on different days at various indoor and outdoor locations. They used the so-called MotionNode to capture activity signals and build a dataset, which is a multimodal sensor that integrates a three-axis accelerometer, three-axis gyroscope, and three-axis magnetometer at a sampling rate of 100 Hz. The USC-HAD dataset provides a controlled yet diverse representation of real-world activities, making it a valuable benchmark for evaluating HAR model performance.

2.2. UCI-HAR

The University of California Irvine (UCI-HAR) collected this dataset from recordings of 30 subjects performing 6 activities while carrying a smartphone mounted with embedded inertial sensors [21]. Each participant was instructed to follow an activity protocol while wearing a Samsung Galaxy S II smartphone mounted on their waists. The six selected activities were standing, sitting, lying down, walking, walking downstairs, and walking upstairs. They collected triaxial linear acceleration and angular velocity signals using a phone accelerometer and gyroscope at a sampling rate of 50 Hz. The time signals were sampled in sliding windows with a fixed width of 2.56 s and 50% overlap. A feature vector consists of the mean, correlation, signal magnitude area, autoregression coefficients, energy of different frequency bands, frequency skewness, and the angles between vectors. A total of 561 features were extracted to describe each activity. Due to its structured design and rich feature set, the UCI-HAR

dataset has become a widely used benchmark for evaluating the effectiveness of machine learning and deep learning models in HAR research.

3. Proposed methodology

The block diagram presented in Figure 1 depicts the architecture of the proposed model, which incorporates a feature fusion approach and preprocessing frequency filter. Initially, raw sensor data undergo filtering and normalization to ensure that the model processes low-frequency signals with a zero mean and unit norm. The normalized accelerometer signals, which consisted of three channels, were fed into the first neural network. In parallel, the three-channel normalized gyroscope signals follow a separate but identical processing path to feed another network. The outputs from these two networks are then merged through concatenation to generate either a one- or two-dimensional multichannel feature map. This fusion strategy enables the model to learn high-level representations by integrating complementary information from both accelerometer and gyroscope data.

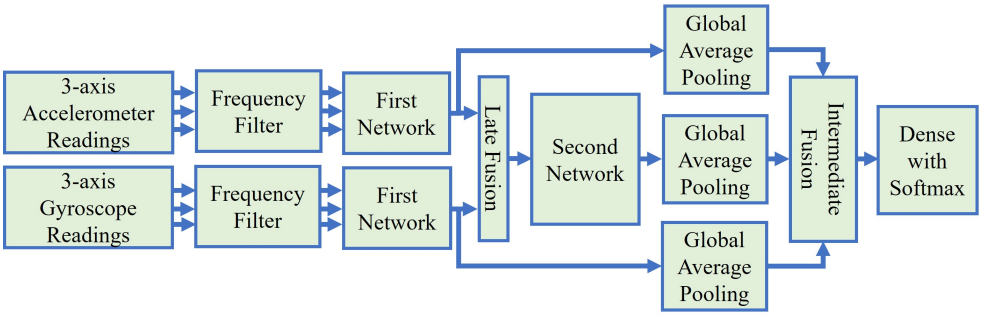

Figure 1. Block diagram of the proposed network architecture.

Subsequently, the feature maps are processed by a second neural network, which introduces an additional abstraction layer into the learned features. To further refine the extracted features, global average pooling is applied to the outputs of both the first and second networks. This operation produces compact yet informative representations by summarizing spatial information. The resulting pooled features are then combined in a concatenation layer to form a unified one-dimensional feature vector that integrates information from both network stages. Finally, the feature vector is processed by a dense layer, followed by a classification softmax layer, to enable effective and nuanced activity recognition.

To optimize the architecture selection, we systematically evaluated multiple configurations for both the first and second networks, selected from the CNN, LSTM, and convolutional LSTM architectures, each implemented in one- and two-dimensional forms. In addition, we investigated the impact of internal feature fusion strategies by comparing models that employed single- or dual-stage fusion mechanisms. This comprehensive evaluation was designed to identify the most effective neural network configuration for optimizing feature integration and improving the ability of the model to accurately recognize HAR based on raw sensor data.

Deep neural networks have emerged as powerful models for learning representations from complex data. Each architecture exhibits unique characteristics tailored to specific tasks, from foundational feedforward networks to more sophisticated CNNs and RNNs, such as LSTM. CNNs excel at extracting spatial hierarchies from images, whereas LSTMs capture the temporal dependencies in

sequential data. In addition, novel architectures have pushed the boundaries of representation learning to structured and relational data. The design of each architecture, including the layer types, activation functions, and connectivity patterns, profoundly affects its expressivity and computational efficiency.

3.1. Convolutional neural networks (CNNs)

Despite challenges such as limited data on group activities, high computational resource demands, data privacy concerns, and edge computing limitations, CNN-based models remain suitable for accurate and efficient HAR system applications [22]. CNNs can learn highly abstracted object features and are suitable for image analysis and recognition [23]. However, the CNN model also has a layer that can learn the features of sequential data with multiple variables. A typical CNN model comprises a convolutional layer followed by a smoothing rectified linear unit (ReLU), pooling, and batch normalization layers. The convolutional layer is the main component of the CNN, which operates on the principle of sliding windows to reduce computational complexity. In this layer, a kernel filter is used to extract features from the input data. For a given 2D single-channel input matrix Y and 2D convolutional filter H, the output of the convolution operation at position (i, j) is

$$Z(i, j) = \sum_{m=0}^{H_f-1} \sum_{n=0}^{W_f-1} Y(i + m, j + n)H(m, n) + b, \tag{1}$$

where $Y(i, j)$ denotes the input feature map, $H(m, n)$ is the convolutional filter, and b is the bias. For an input Y of size $H_{\text{in}} \times W_{\text{in}} \times C_{\text{in}}$ and a filter H of size $H_{\text{f}} \times W_{\text{f}} \times C_{\text{f}}$ with stride $S_H \times S_W$ and padding P, the output Z is of the size given by

$$H_{\text{out}} = \frac{H_{\text{in}} - H_{\text{f}} + 2P}{S_H} + 1, W_{\text{out}} = \frac{W_{\text{in}} - W_{\text{f}} + 2P}{S_W} + 1, \text{ and } C_{\text{out}} = C_{\text{f}}. \tag{2}$$

The next layer is the pooling layer, which is designed to reduce the size of the feature map and extract dominant features for efficient model training. Several types of pooling operations exist, including max-pooling and average pooling. The batch normalization layers apply a transformation that maintains the output mean close to 0 and a standard deviation close to 1.

3.2. Long short-term memory (LSTM) network

LSTM is a special type of RNN that was developed to overcome the weakness of RNN, which cannot learn long-term dependence [24]. LSTM consists of memory blocks called cells, which have two states: cell and hidden states. Cells in LSTM are used to make decisions by storing or ignoring information regarding the forget, input, and output gates of the model. The LSTM operates in three stages: In the first stage, the network works with the forget gate to determine the information that must be ignored or stored in the cell states. The calculation starts by considering the input at the current time step X_t and the previous value of the hidden state H_{t-1} using the sigmoid function σ, such as

$$f_t = \sigma(W_f[X_t, H_{t-1}] + b_f), \tag{3}$$

where W_f and b_f denote the weight and bias of the forget gate, respectively. In the second phase, the network converts the old cell state C_{t-1} into the new cell state C_t. This process selects new

information in the long-term memory (cell state). To obtain the new cell state value, the calculation process considers the reference values from the forget, input, and cell update gates, as follows:

$$i_t = \sigma(W_i[X_t, H_{t-1}] + b_i), \tag{4}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_c[X_t, H_{t-1}] + b_c), \tag{5}$$

where i_t, W_i, and b_i denote the output, weight, and bias of the input gate, respectively. W_c and b_c denote the weights and biases of the cell states, respectively. The symbol $\circ$ denotes a point product operation. Once the cell status update is completed, the final step is to determine the value of the hidden state H_t. This process aims for the hidden state to act as a memory, containing information about previous data, and to be used for making predictions. To determine the value of the hidden state, the calculation must have the reference value of the new cell state C_t and the output gate o_t in terms of the weights W_o and the bias b_o of the output gate, such as

$$o_t = \sigma(W_o[X_t, H_{t-1}] + b_o), \tag{6}$$

$$H_t = o_t \circ \tanh(C_t). \tag{7}$$

3.3. Convolutional LSTM (CLSTM)

Building upon the recurrent framework of LSTM, convolutional LSTM (CLSTM) distinguishes itself by employing convolution operations instead of internal matrix multiplications [18]. This architectural shift enables the CLSTM to process data while preserving its spatial dimensions (as illustrated in Figure 2), avoiding the reduction to a flat feature vector that is characteristic of the standard LSTM. This spatial retention is particularly advantageous for tasks involving grid-like data, such as images and video frames. Formally,

$$f_t = \sigma(W_f * [X_t, H_{t-1}] + b_f), \tag{8}$$
$$i_t = \sigma(W_i * [X_t, H_{t-1}] + b_i), \tag{9}$$
$$C_t = f_t \circ C_{t-1} + i_t \circ \tanh(W_c * [X_t, H_{t-1}] + b_c), \tag{10}$$
$$o_t = \sigma(W_o * [X_t, H_{t-1}] + b_o), \tag{11}$$
$$H_t = o_t \circ \tanh(C_t), \tag{12}$$

where * denotes the convolution operator. Integrating convolutional operations within the memory cell and gate computations in a neural network is a significant advancement in its ability to autonomously capture spatial hierarchies and patterns in the input data. This architectural enhancement enables the network to learn local patterns and correlations within the data, thereby fostering a nuanced understanding of spatial context. Convolutional operations within a memory cell facilitate the extraction of spatial features at different scales, thereby allowing the model to discern both fine- and coarse-grained spatial hierarchies. Additionally, the use of convolutional structures in gate computations helps the model selectively focus on relevant spatial information, promoting more effective and context-aware learning.

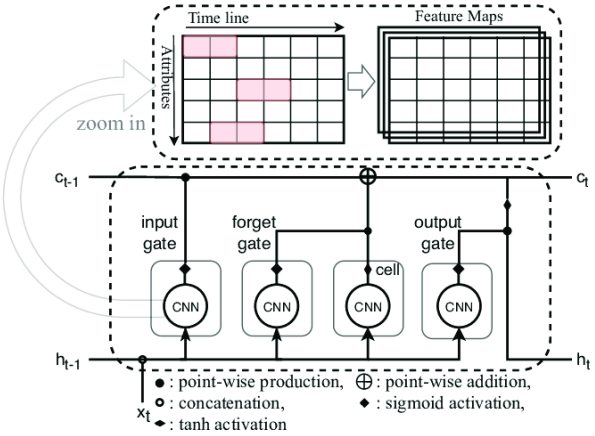

Figure 2. Block diagram of the convolutional LSTM network.

3.4. Global average pooling

Global average pooling (GAP) is a pooling operation used in convolutional neural networks (CNNs) that reduces the spatial dimensions of feature maps to a single value per channel. Unlike traditional pooling methods, such as max or average pooling, which partially reduce dimensionality, global average pooling collapses each feature map into a single number by taking the average of all elements in the feature map. Suppose we have a feature map tensor of shape (H_{in}, W_{in}, C), where H_{in} is the height of the input feature map, W_{in} is the width of the input feature map, and C is the number of channels. For each channel $c \in \{1, 2, ..., C\}$, GAP computes the average value across the spatial dimensions $H_{in} \times W_{in}$. The output for each channel is given by

$$y_c = \frac{1}{H_{in} \cdot W_{in}} \sum_{i=1}^{H_{in}} \sum_{j=1}^{W_{in}} x(i, j, c), \tag{13}$$

where $x(i, j, c)$ is the value at the position (i, j) in channel c, and y_c is the resulting average scalar for channel c. After applying GAP, the spatial dimensions are reduced to 1×1, and the output tensor has the shape $(1, 1, C)$, which can be interpreted as a vector of size C. This operation typically aggregates spatial information to reduce the number of model parameters and prevent overfitting.

4. Experimental setup

The primary objective of this experimental analysis is to conduct a comprehensive ablation study to identify the optimal configurations for the initial and secondary networks. This evaluation is performed under late feature fusion settings, with and without the incorporation of intermediate feature fusion. The investigation involves exhaustively evaluating various architectural combinations, including CNN, LSTM, and CLSTM, while effectively ablating two key design choices: core architectural components and fusion strategies. Concurrently, training hyperparameters must be carefully tuned, as both architectural and parametric choices substantially influence classification performance. To provide a robust evaluation, the performance is assessed using metrics such as accuracy, enabling a comprehensive understanding of how different configurations behave under various conditions and

datasets. Finally, the optimal model derived from this ablation study is benchmarked against state-of-the-art methods to validate its effectiveness.

4.1. Network implementation

Table 1 lists the key parameters governing the configuration of each network, providing essential insights into the critical choices required for effective optimization. These parameters directly influence the training dynamics and final model performance. Specifically, the table lists the optimization algorithm (ADAM), which controls how weights are updated during learning; the loss function (categorical cross-entropy), which is essential for quantifying the error between the predicted and true labels in classification tasks; and the batch size, which determines the number of samples processed before each internal model update. Understanding these settings is fundamental for reproducing and interpreting network behavior.

Table 1. Network implementation.

Parameter	**Value**
Raw sensor readings for USC-HAD	2 1024×3
Raw sensor readings for UCI-HAR	2 128×3
Input feature vector for UCI-HAR	1 561×1
1D CNN & 1D CLSTM:	
no. filters	128
Filter length	16
Filter stride	8
Activation	RelU
2D CNN & 2D CLSTM:	
no. filters	128
Filter length	2×8
Filter stride	2×4
Activation	RelU
LSTM:	
no. units	128
Training:	
Optimizer	ADAM
Loss	categorical cross-entropy
Batch size	no. training examples/32
no. epochs	500

4.2. Evaluation metrics

Two types of errors can arise: false negatives (FNs), when activities belonging to a specific class are incorrectly classified as belonging to other classes, and false positives (FPs), when activities from other classes are incorrectly identified as belonging to a specific class. True positives (TPs) are activities correctly identified as belonging to the intended class, whereas true negatives (TNs) are activities correctly classified as not belonging to that class. These values are crucial for calculating various

classification metrics such as the accuracy [25], i.e.,

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}\%. \tag{14}$$

5. Experiments and results

We used a 5-fold data split, a widely adopted approach that balances computational efficiency with reliable performance estimates. The USC-HAD dataset was partitioned into five subsets, and the model was trained and evaluated five times, each time using a different subset as the test set and the remaining four as the training sets. With 14 subjects and 12 activities, this splitting produced 672 (80%) data recordings for model training and 168 (20%) for testing. Each data recording comprised three vectors representing the measurements from the three-axis accelerometer and three vectors from the three-axis gyroscopes. The vector length is standardized to 1024 samples, which is achieved by truncating long vectors or replicating samples for shorter vectors. Figures 3 and 4 depict the average accuracy achieved by different network combinations evaluated with and without the intermediate feature fusion strategy.

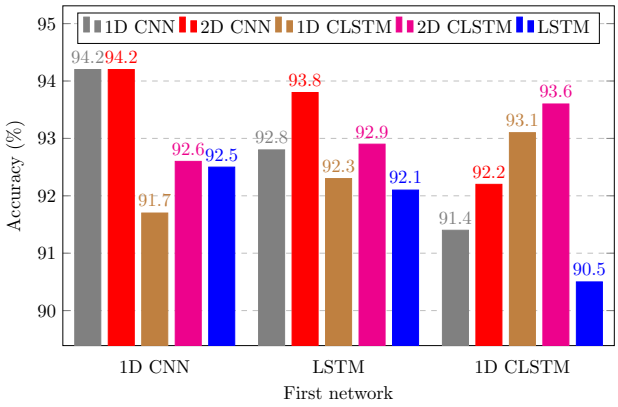

Figure 3. Accuracy of different architectures on the raw sensor readings of the dataset USC-HAD with late feature fusion.

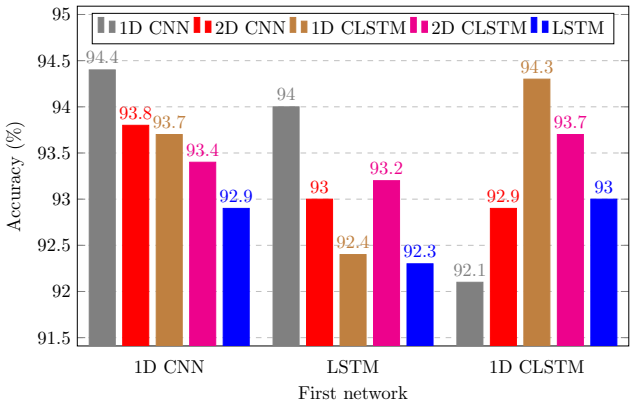

Figure 4. Accuracy of different architectures on the raw sensor readings of the dataset USC-HAD with late and intermediate feature fusion.

We conducted experiments using either raw sensor readings or commonly used feature vectors from the UCI-HAR dataset to characterize individual activities. The dataset, which contained six distinct activity classes, was partitioned into training and testing sets, resulting in 7352 data samples (either raw sensor readings or feature vectors) used for training the model and 2947 samples reserved for testing. Figures 5 and 6 present the test accuracy obtained using raw sensor readings across various combinations of network architectures evaluated using the two fusion strategies. They provide a visual comparison of how the integration of information at different processing stages affects overall classification performance. Table 2 summarizes the detailed accuracy results for each network combination on commonly used feature vectors, allowing for a more granular analysis of the configurations that yield the best recognition rates for the UCI-HAR dataset.

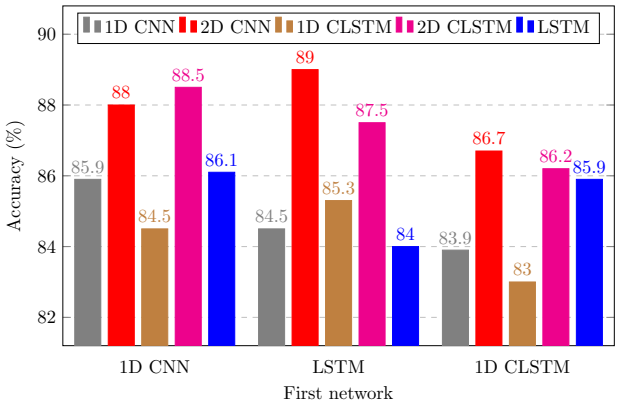

Figure 5. Accuracy of different architectures on the raw sensor readings of the dataset UCI-HAR with late feature fusion.

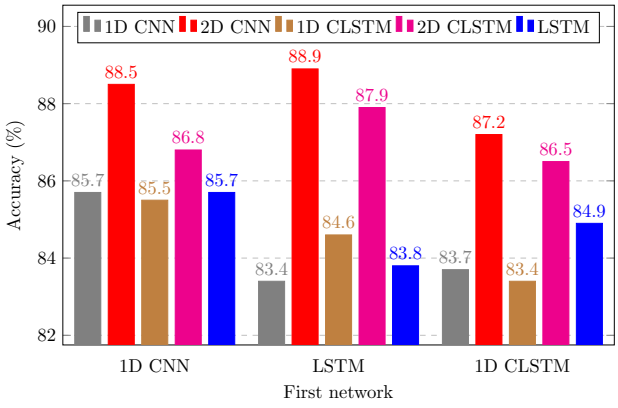

Figure 6. Accuracy of different architectures on the raw sensor readings of the dataset UCI-HAR with late and intermediate feature fusion.

Table 2. Accuracy (%) on features of the dataset UCI-HAR.

First Net	Second Net	No intermediate	With intermediate
1D CNN	1D CNN	96.30	96.75
1D CNN	LSTM	94.52	94.65
1D CNN	1D CLSTM	95.76	95.81
LSTM	LSTM	89.40	87.36
LSTM	1D CNN	92.45	92.00
LSTM	1D CLSTM	93.80	92.90
1D CLSTM	1D CLSTM	95.75	95.54
1D CLSTM	1D CNN	95.33	95.15
1D CLSTM	LSTM	93.61	92.91

5.1. Discussion

Figures 3 and 4 show the accuracy of various architectures on the raw sensor readings of the USC-HAD dataset, highlighting the effectiveness of two-stage individual or hybrid networks with or without intermediate feature fusion. In general, fusion improves the accuracy for most network stage pairings; however, some combinations show only marginal improvements or even slight decreases in accuracy when fusion is applied. The highest accuracy of 94.40% is attained when both the first and second network stages are composed of 1D CNNs with intermediate fusion. The second- and third-highest accuracies are achieved by architectures using individual 1D CLSTMs or a hybrid combination of LSTM and 1D CNN, both with fusion. Comparable accuracy is observed for models based on individual 1D CNNs, as well as hybrid combinations of 1D and 2D CNNs or LSTM with 2D CNN with no fusion.

Notably, architectures based on 1D operations (1D CNN, 1D CLSTM) consistently outperformed their 2D counterparts. This indicates that for the inertial measurement unit (IMU) data used in this study, which is fundamentally a one-dimensional temporal signal, 1D convolutions are more effective at extracting discriminative features. Although 2D architectures can learn intersensor correlations, they introduce additional complexity without a commensurate performance gain. This trade-off underscores the advantages of 1D models, which provide superior efficiency and accuracy for the target task.

Figures 5 and 6 show the accuracy of various models on the raw sensor readings of the UCI-HAR dataset, comparing the models with and without intermediate fusion. The results indicate that network architectures without fusion achieve accuracies ranging from 83% to 89%, with the highest performance observed for a hybrid combination of LSTM and 2D CNN. Models that incorporate intermediate fusion generally exhibit enhanced accuracy compared to their non-fused counterparts, underscoring the advantages of combining complementary features extracted by different networks. The top-performing models achieve accuracies of 88.90% and 88.50%, realized through the integration of LSTM with 2D CNN and 1D CNN with 2D CNN, respectively, using intermediate fusion.

Table 2 presents the accuracies of the different network architectures on the feature vectors of the UCI-HAR dataset. The highest overall accuracy of 96.75% is achieved when using a 1D CNN, followed by another 1D CNN with the intermediate layer enabled. In general, configurations involving 1D CNNs tend to outperform those involving LSTM or 1D CLSTM as the second stage, suggesting that the convolutional layers are more effective in capturing spatial features in this context.

Additionally, the use of intermediate fusion slightly improves the accuracy across most configurations, except in cases involving LSTM as the second network, where it often leads to a reduced improvement. These results highlight the importance of network architecture and feature fusion in achieving optimal accuracy.

5.2. Comparison with state-of-the-art methods

To benchmark the performance of the proposed dual-stage fusion architecture, we compared it with several established and recent state-of-the-art (SOTA) methods from the literature on the two benchmark datasets. This comparison is crucial for validating the effectiveness and competitiveness of the proposed approach. Table 3 presents the classification accuracies of the different deep models.

The table includes performance results from a heterogeneous CNN (CNN+HC) that uses grouped kernels for recalibration [7], a hierarchical LSTM (H-LSTM) designed for time-frequency characteristics [13], and standard CNN-LSTM models that combine spatial and temporal feature extractors [11, 12, 15]. Furthermore, we compare against a more complex multi-head CNN-LSTM variant [14], which uses parallel CNNs for different sensors before temporal modeling, representing a strong and sophisticated baseline.

On the USC-HAD dataset, our proposed approach (a dual-stage 1D CNN with intermediate fusion) achieves the highest reported accuracy of 94.40%, outperforming all existing methods, including the complex CNN+HC and various CNN-LSTM implementations. Similarly, for the UCI-HAR dataset, our model reaches an impressive accuracy of 96.75%, surpassing not only H-LSTM and standard CNN-LSTM architectures but also the more advanced multi-head CNN-LSTM variant.

Notably, many of the compared methods only report results on one of the two datasets, whereas our model demonstrates consistent and superior performance across both. This underscores its strong generalization capability and robustness, validating the effectiveness of the proposed dual-stage feature fusion architecture against a range of SOTA benchmarks.

Table 3. Accuracy (%) comparison.

Method	**USC-HAD**	**UCI-HAR**
CNN+HC [7]	90.67	-
H-LSTM [13]	-	91.65
CNN-LSTM	[12] 90.88	[15] 92.83
	[11] 90.91	[14] 95.76
Proposed	**94.40**	**96.75**

6. Conclusions and future work

This study systematically explored a human activity recognition (HAR) system that employs late and intermediate feature fusion within individual and hybrid deep neural network models. Through a comprehensive ablation study, we evaluated the impact of architectural components (CNN, LSTM, CLSTM) and fusion strategies across 15 different network configurations. By integrating multimodal data from multiple sensors and combining features across different network levels, the proposed methodology investigates the structures of various architectures. The experimental results on the two benchmark HAR datasets demonstrated the superiority of convolutional modeling using CNNs and

convolutional LSTMs over linear modeling using LSTMs. Moreover, our comprehensive evaluation revealed that 1D architectural components (1D CNN and 1D CLSTM) were consistently more effective than 2D components for processing the inherent temporal structure of IMU sensor data, achieving the highest accuracy with greater parameter efficiency. Furthermore, the ablation study conclusively demonstrated that fusing features at an intermediate stage consistently enhanced the classification accuracy over using late fusion alone. Finally, benchmarking against state-of-the-art methods confirmed that our optimal model (dual 1D CNNs with intermediate fusion) achieved superior performance on both datasets, highlighting its effectiveness and generalizability.

This study established a strong baseline using concatenation for its simplicity and effectiveness in our architectural investigation. Future research will explore more sophisticated, adaptive feature fusion mechanisms, such as attention-based fusion, to dynamically weight the contributions from different sensors and network levels, potentially further boosting performance and robustness. Furthermore, although the proposed dual-stage architecture demonstrates high accuracy, its computational complexity presents a challenge for deployment on low-power edge computing devices. Therefore, future work will focus on optimizing this framework using techniques such as model pruning, quantization, and neural architecture search to reduce its memory footprint and latency, facilitating its application in real-time HAR systems.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. S. G. Dhekane, T. Ploetz, Transfer learning in human activity recognition: a survey, arXiv: 2401.10185. https://doi.org/10.48550/arXiv.2401.10185
2. V. Soni, S. Jaiswal, V. B. Semwal, B. Roy, D. K. Choubey, D. K. Mallick, An enhanced deep learning approach for smartphone-based human activity recognition in ioht, In: *Machine learning, image processing, network security and data sciences: select proceedings of 3rd international conference on MIND 2021*, Singapore: Springer, 2023, 505–516. https://doi.org/10.1007/978-981-19-5868-7_37
3. S. Saini, A. Juneja, A. Shrivastava, Human activity recognition using deep learning: past, present and future, *Proceedings of 1st International Conference on Intelligent Computing and Research Trends (ICRT)*, 2023, 1–6. https://doi.org/10.1109/ICRT57042.2023.10146621
4. S. Mekruksavanich, A. Jitpattanakul, The deep learning-based human activity recognition using smart wearable sensors: a tutorial, *ReBICTE*, **8** (2022), 1. https://doi.org/10.22667/ReBiCTE.2022.02.28.001

5. E. Ramanujam, T. Perumal, S. Padmavathi, Human activity recognition with smartphone and wearable sensors using deep learning techniques: a review, *IEEE Sens. J.*, **21** (2021), 13029–13040. https://doi.org/10.1109/JSEN.2021.3069927
6. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, 770–778. https://doi.org/10.1109/CVPR.2016.90
7. C. Han, L. Zhang, Y. Tang, W. Huang, F. Min, J. He, Human activity recognition using wearable sensors by heterogeneous convolutional neural networks, *Expert Syst. Appl.*, **198** (2022), 116764. https://doi.org/10.1016/j.eswa.2022.116764
8. Y. Li, J. Wu, W. Li, A. Fang, W. Dong, Temporal-spatial dynamic convolutional neural network for human activity recognition using wearable sensors, *IEEE Trans. Instrum. Meas.*, **72** (2023), 2516912. https://doi.org/10.1109/TIM.2023.3279908
9. J. Sena, J. Barreto, C. Caetano, G. Cramer, W. R. Schwartz, Human activity recognition based on smartphone and wearable sensors using multiscale dcnn ensemble, *Neurocomputing*, **444** (2021), 226–243. https://doi.org/10.1016/j.neucom.2020.04.151
10. Q. Huang, W. Xie, C. Li, Y. Wang, Y. Liu, Human action recognition based on hierarchical multi-scale adaptive conv-long short-term memory network, *Appl. Sci.*, **13** (2023), 10560. https://doi.org/10.3390/app131910560
11. M. Sethi, M. Yadav, M. Singh, P. G. Shambharkar, Attnhar: human activity recognition using data collected from wearable sensors, *Proceedings of 6th International Conference on Information Systems and Computer Networks (ISCON)*, 2023, 1–6. https://doi.org/10.1109/ISCON57294.2023.10112183
12. S. P. Singh, M. K. Sharma, A. Lay-Ekuakille, D. Gangwar, S. Gupta, Deep convlstm with self-attention for human activity decoding using wearable sensors, *IEEE Sens. J.*, **21** (2021), 8575–8582. https://doi.org/10.1109/JSEN.2020.3045135
13. L. Wang, R. Liu, Human activity recognition based on wearable sensor using hierarchical deep lstm networks, *Circuits Syst. Signal Process.*, **39** (2020), 837–856. https://doi.org/10.1007/s00034-019-01116-y
14. W. Ahmad, M. Kazmi, H. Ali, Human activity recognition using multi-head cnn followed by lstm, *Proceedings of 15th International Conference on Emerging Technologies (ICET)*, 2019, 1–6. https://doi.org/10.1109/ICET48972.2019.8994412
15. R. Kolkar, V. Geetha, Human activity recognition in smart home using deep learning techniques, *Proceedings of 13th International conference on information & communication technology and system (ICTS)*, 2021, 230–234. https://doi.org/10.1109/ICTS52701.2021.9609044
16. J. X. Goh, K. M. Lim, C. P. Lee, 1d convolutional neural network with long short-term memory for human activity recognition, *Proceedings of IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, 2021, 1–6. https://doi.org/10.1109/IICAIET51634.2021.9573979
17. M. M. Islam, S. Nooruddin, F. Karray, G. Muhammad, Multi-level feature fusion for multimodal human activity recognition in internet of healthcare things, *Inform. Fusion*, **94** (2023), 17–31. https://doi.org/10.1016/j.inffus.2023.01.015

18. X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, W. C. Woo, Convolutional lstm network: a machine learning approach for precipitation nowcasting, *Proceedings of 29th Annual Conference on Neural Information Processing Systems*, 2015, 802–810.
19. G. Alam, I. McChesney, P. Nicholl, J. Rafferty, Open data sets in human activity recognition research-issues and challenges: a review, *IEEE Sens. J.*, **23** (2023), 26952–26980. https://doi.org/10.1109/JSEN.2023.3317645
20. M. Zhang, A. A. Sawchuk, Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors, *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, 1036–1043. https://doi.org/10.1145/2370216.2370438
21. D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones, *Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligenceand Machine Learning*, 2013, 437–442.
22. M. M. Islam, S. Nooruddin, F. Karray, G. Muhammad, Human activity recognition using tools of convolutional neural networks: a state of the art review, data sets, challenges, and future prospects, *Comput. Biol. Med.*, **149** (2022), 106060. https://doi.org/10.1016/j.compbiomed.2022.106060
23. A. Ghosh, A. Sufian, F. Sultana, A. Chakrabarti, D. De, Fundamental concepts of convolutional neural network, In: *Recent trends and advances in artificial intelligence and internet of things*, Cham: Springer, 2019, 519–567. https://doi.org/10.1007/978-3-030-32644-9_36
24. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.*, **9** (1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
25. J. Opitz, A closer look at classification evaluation metrics and a critical reflection of common evaluation practice, *Transactions of the Association for Computational Linguistics*, **12** (2024), 820–836. https://doi.org/10.1162/tacl_a_00675

AIMS Press