Journal of Biomechanics 202 (2026) 113290

Contents lists available at ScienceDirect

Journal of Biomechanics

journal homepage: www.elsevier.com/locate/jbiomech

Automated dual-stream deep network design for activity recognition

Seyed Mojtaba Mohasel [a], Alireza Afzal Aghaei [b], John W. Sheppard [c], Corey A. Pew [a,*]

[a] Department of Mechanical and Industrial Engineering, Montana State University, Bozeman, Montana, USA
[b] Independent Researcher, Isfahan, Iran
[c] Gianforte School of Computing, Montana State University, Bozeman, Montana, USA

ARTICLE INFO

Keywords:
AutoML
Class imbalance
NAS
Genetic algorithm
CNN
Time and frequency domain features

ABSTRACT

This article presents an automated deep learning (AutoDL) framework for multivariate time-series sensor data tailored to the needs of biomechanists. Four public activity recognition datasets featuring different Inertial Measurement Unit sensor placements and class imbalance ratios were selected. Time- and frequency-domain features were automatically extracted, class imbalance was addressed, and classification was performed using a dual-stream convolutional neural network, with a customized genetic algorithm for optimization. Two hypotheses were evaluated: (1) whether the ablation of frequency- or time-domain features results in a significant reduction in model performance, and (2) whether the proposed AutoDL achieves higher F1-score performance than an existing AutoDL system or a model designed under human supervision. A pairwise Wilcoxon test revealed a significant reduction in model F1-score performance when time-domain or frequency-domain features were removed in three of the four datasets ($p < 0.05$). In addition, the proposed AutoDL framework achieved higher F1-scores than both an existing AutoDL system and human-designed models in three of the four datasets ($p < 0.05$). The final models were deployed on a Raspberry Pi 3. The findings underscore the significance of automation and frequency-domain representations in capturing the rhythmic biomechanical characteristics of cyclic movements. The release of the open-source code is expected to accelerate research in biomechanics and facilitate the development of health monitoring tools (https://github.com/MojtabaMohasel/AutoDL_DualStream).

1. Introduction

In the biomechanics community, machine learning (ML) algorithms coupled with wearable sensors such as inertial measurement units (IMUs) (Ahmad et al., 2013) and surface electromyography (sEMG) (Yadav and Veer, 2023) are commonly used to identify human movements for prosthesis control (Krasoulis et al., 2019), exoskeleton control (Pesenti et al., 2023), fall detection (Mohasel et al., 2025b), and mobility monitoring (Spangler et al., 2024). Translation of raw sensor data into accurate movement classifications with the desired performance, and deployment on edge devices requires an ML pipeline (Fig. 1) (Probst et al., 2019; Weerts et al., 2020). Optimizing each stage of the ML pipeline requires expertise in machine learning, as multiple options are available based on input data type and classification requirements, each with various hyperparameters. The best choices at each stage are interdependent and unique to each dataset and use case (Van Rijn and Hutter, 2018), requiring a skill set beyond the average biomechanist, making new research in the field difficult.

Deep learning (DL) has gained considerable attention in activity recognition (Ismail et al., 2023). DL simplifies the stages of the ML pipeline by providing an end-to-end framework that facilitates feature extraction, construction, and selection (Stages 4 to 6 in Fig. 1). However, manually building deep models and determining model layers and operation types through trial and error is tedious and inefficient. Therefore, practitioners often attempt to use the same model architectures from published papers or try a limited set of layer combinations (Oguiza, 2022), which can result in suboptimal performance and limited real-world applicability for a unique dataset (Halilaj et al., 2018).

Neural architecture search (NAS) alleviates the burden of manual design by automatically exploring the architecture space and

* Corresponding author.
Email addresses: seyedmojtaba.mohasel@student.montana.edu (S.M. Mohasel), alirezaafzalaghaei@gmail.com (A. Afzal Aghaei), john.sheppard@montana.edu (J.W. Sheppard), corey.pew@montana.edu (C.A. Pew).

https://doi.org/10.1016/j.jbiomech.2026.113290
Accepted 6 April 2026
Available online 12 April 2026

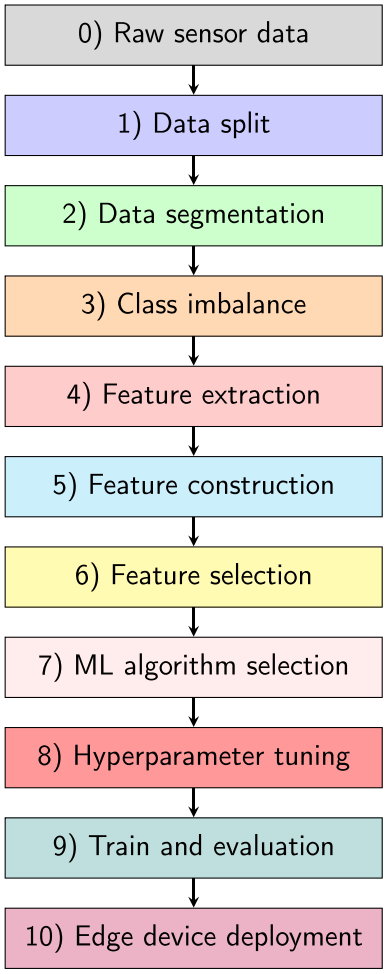

Fig. 1. Overview of the stages in an ML pipeline for sensor-based activity recognition.

corresponding hyperparameters (automating stages 4–8 in Fig. 1) (Salehin et al., 2024). Yet, the effectiveness of NAS ultimately depends on the quality and diversity of the input representations. Time-domain representations have been extensively explored in the context of automated frameworks (Ismail et al., 2023; Gupta, 2021; Van Kuppevelt et al., 2020). Frequency-domain representations, however, provide complementary information about the periodicity of human movement patterns (Sani et al., 2017; Revin et al., 2023). The lack of systematic evaluation of the combined contribution of time- and frequency-domain features to model performance in automated frameworks motivated the present study.

This study develops an open-source Python library for an automated DL pipeline. The primary contributions are (i) a software framework for biomechanists that performs window optimization with integrated class-imbalance handling and (ii) an empirical demonstration of the utility of time-frequency hyperparameters under AutoDL. To evaluate the effectiveness of the proposed automated framework, measured by the macro-average F1-score in a classification task, two hypotheses are presented as follows:

Hypothesis 1: Models developed using both time- and frequency-domain features achieve higher performance compared to models that use only time-domain or frequency-domain features.

Hypothesis 2: A DL model designed by the automated framework for activity recognition outperforms both (i) models generated by an existing AutoDL library for time-series data and (ii) a manually designed architecture with human supervision.

2. Methods

The proposed automated data processing pipeline consists of several sequential stages, which are described in detail in the following subsections. A genetic algorithm (GA) is implemented to optimize the entire DL pipeline. The GA was used for optimization due to its ability to efficiently navigate large, complex, and non-differentiable search spaces characteristic of pipeline optimization problems (Wu et al., 2022; Revin et al., 2023).

2.1. Sensor data

Four public datasets containing human movement data, AReM (4) (Palumbo et al., 2013), WISDM (7.37) (Kwapisz et al., 2011), PAMAP2 (4.84) (Reiss and Stricker, 2012), and Opportunity (6.63) (Chavarriaga et al., 2013), with different imbalance ratios (presented in parentheses), calculated as the number of majority samples divided by the number of minority samples, were utilized to develop the automated framework. These datasets feature multiple sensor configurations (Fig. 2) and include various types of activities of daily living (Tables 1–A.4 in Appendix A).

2.2. Data split

Data splitting refers to dividing data into training, validation, and test sets. To ensure generalizability, the proposed framework requires that all data from an individual be contained entirely in one of the sets (training, validation, or test) (Halilaj et al., 2018). In this study, participant data was assigned to ensure that approximately 70% of participants were used for training, 10–20% for validation, and 10–20% for testing, following the recommended data distribution for activity recognition using DL (Singh et al., 2025) (Tables 1–A.4 in Appendix A).

2.3. Sliding window

Sliding window refers to the process of dividing continuous sensor data into smaller, fixed-length windows prior to feature extraction. The choice of window size and overlap affects feature usefulness. Since human activities occur over different time spans, seven window sizes (0.25, 0.5, 1, 3, 5, 7, and 10 s) were considered to capture variation in activity duration (Banos et al., 2014). For each window size, three overlap percentages (25%, 50%, and 75%) were evaluated. Window and overlap sizes are hyperparameters encoded in a chromosome of the GA.

2.4. Neural architecture search and hyperparameter optimization

The automated framework performs feature engineering using a dual-stream convolutional neural network (CNN) (Jiang et al., 2023).

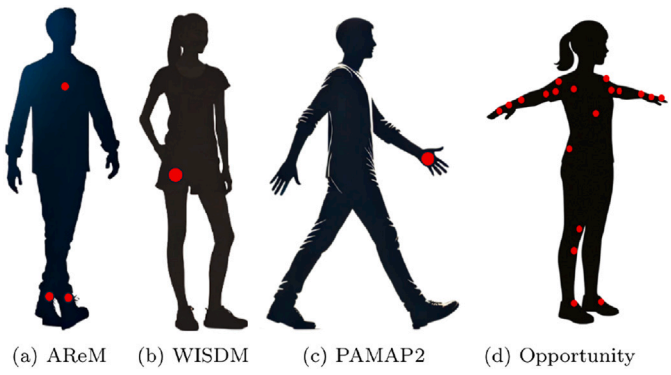

Fig. 2. Sensor placement in different activity recognition public datasets. (a) AReM (Wireless sensor network (IRIS nodes with AT86RF230 radio transceivers)), (b) WISDM (accelerometer and gyroscope sensor data from the smartphone), (c) PAMAP2 (IMU), (d) Opportunity (IMU and 3-axis accelerometers).

CNNs are widely used in human activity recognition due to their ability to capture local temporal patterns in sensor data (Lee et al., 2017; Ismail et al., 2023). The parallel feature extraction in each stream enables edge devices equipped with multi-core processors to parallelize convolutional operations across cores to meet real-time inference requirements for activity recognition.

One stream uses windowed data and applies a one dimensional (1D) CNN to extract time-domain features. Another stream applies a Short-Time Fourier Transform (STFT) to the windowed data to create image data containing frequency information (Appendix B). The output of the STFT is a spectrogram (Fig. 3) that contains time–frequency information corresponding to the specific window during which an action occurred.

Finally, the time and frequency-domain features are concatenated and passed to a fully connected neural network for classification. Hyperparameters related to the dual-stream network in neural architecture search can be organized into several subsections as follows:

2.4.1. Convolution

The convolution aims to extract increasingly abstract feature representations by passing the input data through a stack of multiple convolutional operators. A convolution operation is performed by sliding a convolutional filter over the input to exploit local relationships between adjacent data points. The output of the convolutional layer is a feature map, which is the input for the subsequent convolutional layer (Fig. 3).

The convolutional section includes seven hyperparameters: $\mathfrak{C}_{\mathfrak{B}}$ (batch normalization) and $\mathfrak{I}_{\mathfrak{C}}$ (convolutional layer presence) are binary flags. $\mathfrak{C}_{\mathfrak{F}}$ specifies the number of filters (1–500), and $\mathfrak{C}_{\mathfrak{K}}$ denotes kernel size (1–8), where 2D CNN uses a $k \times k$ filter. $\mathfrak{C}_{\mathfrak{A}}$ selects the activation function (`ReLU`, `Tanh`, and `Sigmoid`), $\mathfrak{C}_{\mathfrak{P}}$ sets the padding mode, and $\mathfrak{C}_{\mathfrak{S}}$ defines the stride (1 or 2) (Mohasel et al., 2025b).

2.4.2. Pooling

The pooling operation performs dimensionality reduction. The pooling section includes two hyperparameters: $\mathfrak{I}_{\mathfrak{P}}$ (binary flag for applying pooling) and $\mathfrak{P}_{\mathfrak{T}}$ (pooling type: `Average` or `Max`) (Géron, 2022).

2.4.3. Flatten

The flatten transforms the feature maps from the convolutional streams into a one-dimensional vector, preparing them for classification. The flatten section includes $\mathfrak{I}_{\mathfrak{G}}$ (apply flatten), $\mathfrak{I}_{\mathfrak{D}}$ (apply dropout), and $\mathfrak{D}$ (dropout rate in $[0.4, 0.5]$) as hyperparameters (Géron, 2022).

2.4.4. Classifier

The classifier section consists of fully connected dense layers that perform the final classification based on the concatenated features from both streams. The classifier includes six hyperparameters: $\mathfrak{I}_{\mathfrak{L}}$ (use dense layer), $\mathfrak{I}_{\mathfrak{B}}$ (batch normalization), $\mathfrak{L}_{\mathfrak{U}}$ (neurons per layer, 3–1024), $\mathfrak{L}_{\mathfrak{A}}$ (activation), $\mathfrak{I}_{\mathfrak{D}}$ (apply dropout), and $\mathfrak{D}$ (dropout rate in $[0.4, 0.5]$) (Géron, 2022). A maximum of three dense layers is allowed to prevent over-parameterization.

2.4.5. Train

The training section defines the optimization parameters for the model. The training section includes $\mathcal{B}$ (batch size: $\{64, 128, 256, 512, 1024\}$), η (learning rate in $[10^{-6}, 10^{-2}]$), and λ (regularization term in $[10^{-6}, 10^{-2}]$).

2.5. Loss

The loss section defines the objective function used to update the model's weights and biases. Since human activities are inherently imbalanced (Glaister et al., 2007; Mohasel and Koosha, 2026), this section first assigns different weights to each activity ($\{C_i\}_{i=1}^{M}$). It then employs weighted cross-entropy or focal loss (Appendix C), selected by the GA ($\mathcal{L}$) based on performance, rather than relying on the standard

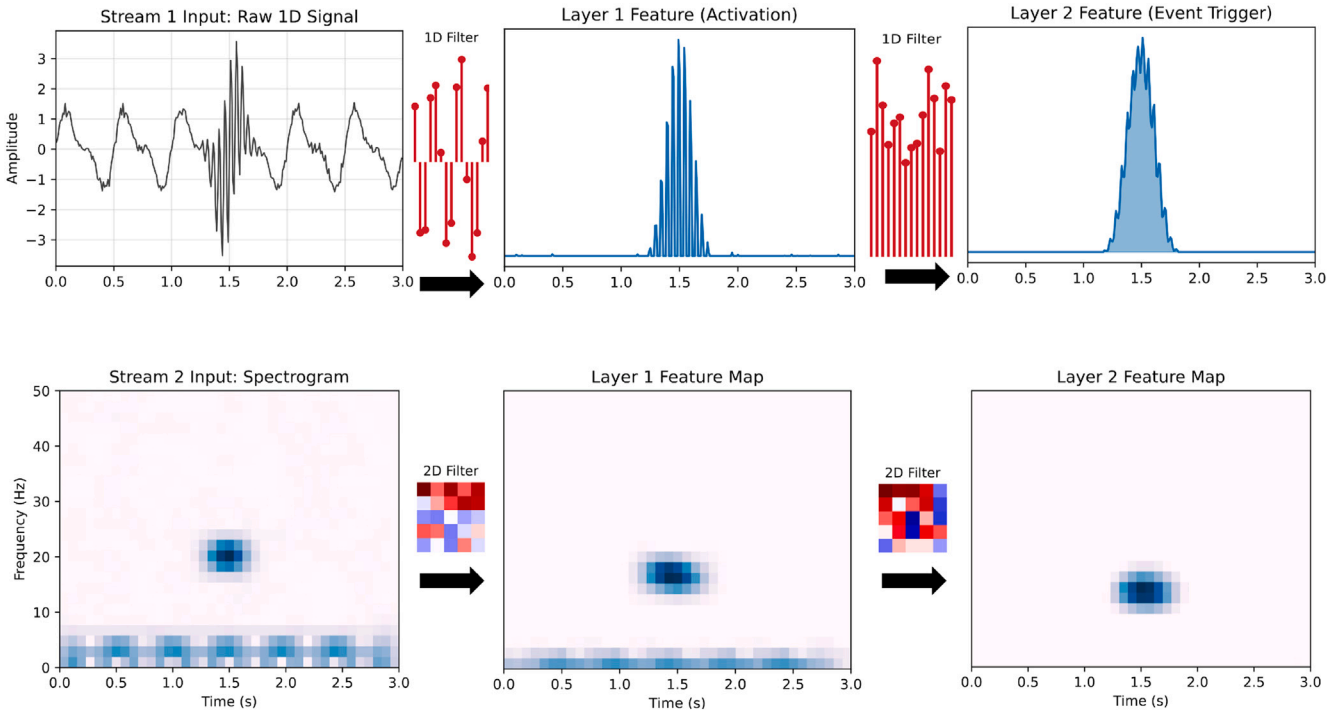

Fig. 3. The top row represents time-domain feature extraction, and the bottom row represents frequency-domain feature extraction. The spectrogram (bottom-left image) was generated by applying an STFT to a 3-second example window of synthetic accelerometer data for a walking trial during which a stumble occurred. The filters in each stream learn to extract features relevant to stumble detection. Shown are representative feature maps from layer 1 and layer 2, illustrating the progressive abstraction and localization of salient motion-related features.

cross-entropy loss to address class imbalance. An initial investigation using SMOTE (Chawla et al., 2002) and undersampling confirmed that an algorithmic-level approach was ideal for handling class imbalance.

2.6. Genetic algorithm hyperparameters

The initial population includes 50 randomly generated neural networks. Parent selection employs tournament selection (size = 4), with an 80% single-point crossover applied to each chromosome section (e.g., windowing, convolution, loss). Mutation (20%) alters one random section. Thirty generations were allocated for model optimization to ensure timely completion of results across all datasets. The provided hyperparameters are an empirically validated default set that offers a good trade-off between performance and efficiency and enabled us to get results for four datasets in a timely manner. The users can adjust the hyperparameters of the GA, while the automated framework also provides an option to optimize these hyperparameters automatically.

2.7. Benchmark

McFly (Van Kuppevelt et al., 2020), an AutoDL framework, was chosen because it was evaluated on a human activity recognition dataset (PAMAP2), utilizes state-of-the-art model architectures (InceptionTime, ResNet, ConvLSTM, CNN), and has publicly available code.

DeepConvLSTM (Ordóñez and Roggen, 2016), an architecture developed under human supervision, was selected because it was evaluated on two activity recognition datasets (Opportunity and Skoda), and has code publicly available in a GitHub repository.

2.8. Computational resources

Model architectures for AutoDL systems (proposed AutoDL and McFly) were optimized using the data split in Appendix A. Both AutoDL systems were allocated the same number of objective function evaluations (1500) for a fair comparison. Training was conducted on Montana State University's Tempest high-performance computing cluster, which is equipped with an AMD EPYC 7H12 processor, 128 GB of RAM, and an NVIDIA A100 GPU.

2.9. Statistical analysis

The data split described in Section 2.2 was used to determine the optimal architecture produced by each AutoDL framework. For statistical tests, leave-one-participant-out cross-validation was performed for generating n macro average F-1-scores. The top model architecture for the AutoDL systems was trained on $n-1$ participants and tested on the held-out n^{th} participant. The participants were then rotated. Pairwise Wilcoxon tests were subsequently conducted between the two models in each hypothesis (Appendices D and E), with the significance level (α) set to 0.05 (Rainio et al., 2024).

To elucidate the underlying feature representations in hypothesis one, the distributed Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton, 2008) method was employed. t-SNE is a dimensionality reduction technique that projects high-dimensional data onto a two-dimensional manifold while preserving local neighborhood structures. The resulting visualizations facilitate the assessment of class separability and the effectiveness of the extracted features.

2.10. Edge device deployment

Neural architecture search was conducted offline on the cluster, and only the resulting optimized model was exported and deployed to the edge device for real-time inference. The inference time is defined as the duration required by the edge device to compute the STFT to generate a spectrogram and to process both the IMU data and the corresponding spectrogram through the network, resulting in the return of a predicted label. The average and standard deviation of inference time were reported for the STFT and the network separately over 1000 window predictions.

The Raspberry Pi 3 was selected as a low-cost, credit-card-sized single-board computer featuring a 1.2 GHz 64-bit quad-core ARM Cortex-A53 processor, 1 GB of RAM, and built-in Wi-Fi and Bluetooth. Its compact size, low power consumption (2.5W typical), and GPIO interface make it a practical edge-computing platform for real-time activity recognition applications. Models were deployed on the Raspberry Pi 3 using the TFLite library.

To evaluate deployment stability and resource efficiency, a longitudinal stress test was conducted on the Raspberry Pi 3 by continuously executing model inference on input data for twelve hours, while recording variations in RAM usage, inference latency, and CPU utilization.

3. Result

Hypothesis 1: Models trained on the combined (time and frequency) feature set achieved higher macro average F1-scores than those models using only time-domain features on the AReM, WISDM, and Opportunity datasets ($p < 0.05$) (Fig. 4a).

Similarly, models with the combined feature set achieved significantly higher performance than those using only frequency-domain features on the AReM and Opportunity datasets ($p < 0.05$) (Fig. 4b).

Models trained on frequency-domain features yielded significantly higher performance than those using time-domain features on the WISDM dataset, whereas models developed using time-domain features outperformed those based on frequency-domain features on the Opportunity dataset ($p < 0.05$) (Fig. 4c). For the AReM dataset,

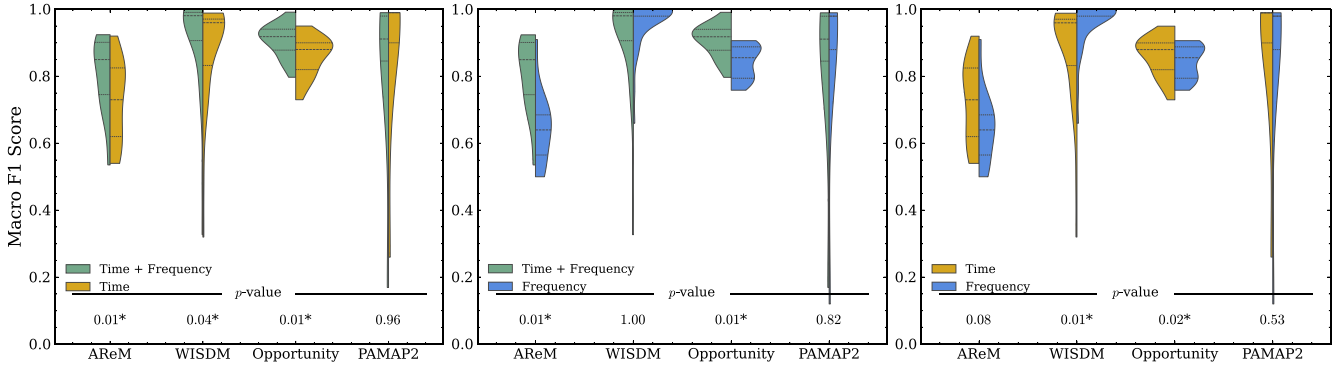

Fig. 4. (a: left) Time + Frequency vs Time, (b: center) Time + Frequency vs Frequency, and (c: right) time vs Frequency. Comparison of macro-averaged F1-scores across four public activity recognition datasets for models developed using time- and frequency-domain features, time-domain features only, and frequency-domain features only. Horizontal bars indicate the median and interquartile range. P-values were computed using a Wilcoxon test for each participant. An asterisk (*) indicates a significant difference at $\alpha = 0.05$.

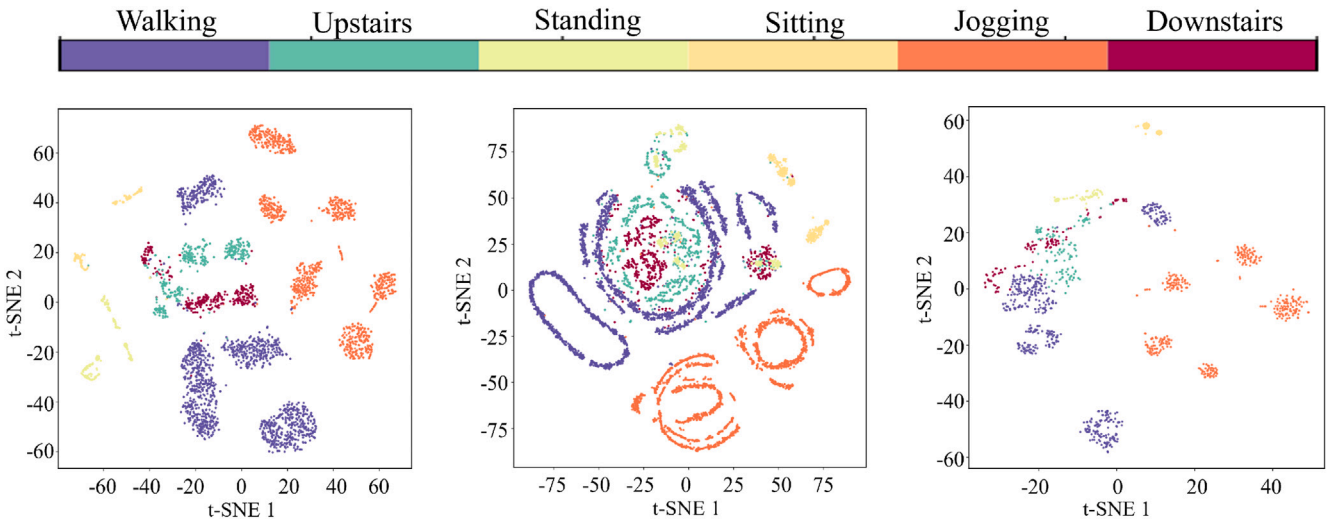

Fig. 5. (a: left) time + frequency features, (b: center) time-domain features, and (c: right) frequency-domain features. t-SNE visualization of the learned features on the WISDM dataset. Improved feature representation in the t-SNE space is indicated when feature clusters belonging to the same activity class become more compact and exhibit greater separation from clusters of other classes.

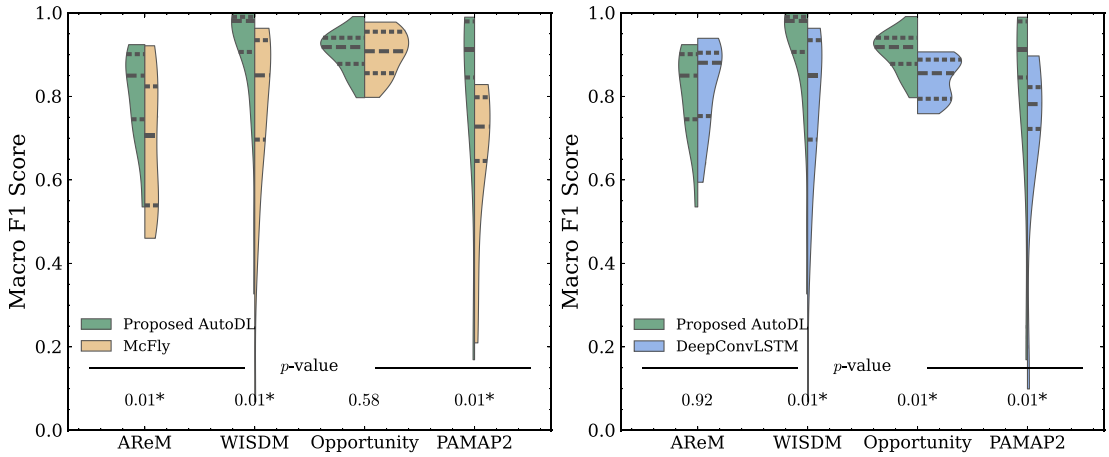

Fig. 6. Comparison of macro-averaged F1-scores between the proposed AutoDL framework and two baselines across four public activity recognition datasets. Statistically significant differences based on pairwise Wilcoxon signed-rank tests are indicated by asterisks (*, $p < 0.05$). (a: left) Comparison on proposed AutoDL and McFly. McFly selected a deep convolutional LSTM for AReM, a convolutional neural network for WISDM, InceptionTime for Opportunity, and a deep convolutional LSTM for PAMAP2 as the top-performing architectures. (b: right) Comparison on proposed AutoDL and a human designed architecture (DeepConvLSTM).

time-domain features led to higher model performance; however, the difference between time- and frequency-domain features was not statistically significant at the 0.05 level ($p = 0.08$) (Fig. 4c). No significant differences were observed for the PAMAP2 dataset ($p = 0.53$) (Fig. 4c).

The t-SNE visualization of the WISDM dataset (Fig. 5) indicates that the combined time–frequency features yield slightly more compact and separable data clusters compared to the frequency-domain features alone. Nevertheless, this apparent visual improvement did not translate into a statistically significant increase in F1-score when combined features were compared to frequency domain features ($p = 1.00$; Fig. 4b).

Hypothesis 2: The proposed AutoDL significantly outperformed McFly on the AReM, WISDM, and PAMAP2 datasets ($p = 0.01$), while no significant difference was observed on the Opportunity dataset ($p = 0.58$) (Fig. 6a). When compared to the DeepConvLSTM, the proposed AutoDL achieved significantly higher F1-scores on the WISDM, Opportunity, and PAMAP2 datasets ($p = 0.01$) and showed comparable performance on AReM ($p = 0.92$) (Fig. 6b).

High-performance computing resources were utilized to expedite the optimization of two AutoDL systems (requiring repeated training for rigorous statistical analysis). However, the estimated optimization time for the AReM dataset is approximately 1 hour, whereas for the remaining datasets it is approximately 12 ± 3 hours on a PC (e.g., an Intel Core i7 processor, 32GB RAM, and an NVIDIA GeForce RTX 3080 GPU), demonstrating the suitability of the framework for biomechanics labs.

The inference time on the Raspberry Pi 3 differed across datasets. For the STFT computation, the average inference times were: AReM, 25 ± 3 ms; WISDM, 10 ± 3 ms; Opportunity, 550 ± 50 ms; and PAMAP2, 170 ± 40 ms.

For the dual-stream network forward pass, the average inference times were: AReM, 10 ± 2 ms; WISDM, 7 ± 2 ms; Opportunity, 60 ± 10 ms; and PAMAP2, 30 ± 5 ms per window.

The longitudinal stress test demonstrated system stability, defined by the absence of memory leaks (RAM usage constant at 253 MB) and consistent inference latency (10 ± 3 ms per window) over the duration of the test. The average CPU utilization stabilized at 15.7%,

indicating sufficient computational headroom for background processes and minimal power consumption, supporting viability for battery-powered operation.

4. Discussion

This article presents a novel and flexible framework that takes wearable sensor data as input, automatically performs data windowing, feature engineering, and handles class imbalance, generating a deployable, high-performance model architecture tailored for biomechanical applications.

Hypothesis 1 findings confirm the contribution of time- and frequency-domain features to improve model performance. Activity recognition studies have predominantly relied on time-domain features (Wang et al., 2019; Gu et al., 2021), as off-the-shelf DL models such as recurrent neural networks and temporal CNNs are inherently designed for temporal data processing. However, many human activities exhibit cyclic characteristics, such as walking, running, or stair ascent and descent, which are more effectively captured in the frequency domain. Spectrograms help visualize complex, cyclical motions. For instance, walking cadence appears as a repeating vertical stripe pattern, which a CNN can identify using shift-invariant filters enhancing the discrimination between activity classes (Sani et al., 2017).

To assess the contribution of features in each domain, an analysis was performed by excluding feature representations from each domain. Removing frequency-domain features resulted in an average 3% reduction in F1-score, while removing time-domain features led to an average 3.4% reduction across datasets, confirming that both temporal and spectral representations contribute valuable information to classification performance. These findings are consistent with prior work employing IMU sensors and a dual-stream CNN for classification (Jiang et al., 2023). Unlike prior work that relied on predefined architectures for each processing stream, our proposed framework automatically optimized the network depth, width, and input resolution to adapt the model to different types of input sensor data.

The varying impact of frequency-domain features stems from biomechanic fundamentals of human walking. The WISDM dataset is dominated by cyclical gait activities, including walking and jogging, where power spectral density reveals distinct peaks corresponding to stride cadence and harmonics. Conversely, the Opportunity dataset consists largely of aperiodic postures such as sitting and standing. These activities lack a stable fundamental frequency and instead rely on signal magnitude to define body segment orientation relative to gravity. Consequently, time-domain features proved superior for Opportunity (Fig. 4c), whereas frequency-domain representations were critical in WISDM.

Furthermore, the anatomical placement of sensors (Fig. 2) influenced the efficacy of frequency-domain features. Acceleration profiles and their spectral characteristics vary significantly depending on the anatomical direction and segment location (Sejdić et al., 2013). In WISDM, the sensor is located on the thigh (Fig. 2b), a segment that undergoes continuous, pendulum-like oscillation during gait, generating a strong harmonic structure that the frequency-stream CNN can exploit. (A quantitative biomechanical validation and an interpretability analysis based on Gradient-weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al., 2017) for the WISDM dataset are provided in Appendix F). In contrast, the Opportunity dataset relies heavily on upper-extremity sensors (Fig. 2d) to capture activities of daily living. Biomechanically, upper-limb functional tasks exhibit lower-amplitude periodicity compared to lower-limb locomotion (Kavanagh and Menz, 2008), rendering frequency-domain features less discriminative than time-domain profiles, which better capture the onset, duration, and intensity of discrete motor tasks.

Hypothesis 2 results highlight the effectiveness of the proposed framework compared with an existing AutoDL. McFly, generates architectures through random search, where the number of architectures is specified by the user. However, McFly lacks an exploitation mechanism to refine promising architectures once they are discovered. Furthermore, it does not include automated window segmentation, requiring users to manually repeat the process for varying window sizes. In contrast, our proposed framework uses a GA for optimization. Although the GA demands greater computational resources than random search, it enhances both exploration and exploitation by adaptively evolving model architectures over generations.

The median F1-scores demonstrate the superiority of the proposed model across all datasets: on AReM, it achieved 0.86 compared to McFly's 0.70; on WISDM, 0.98 versus 0.86; on Opportunity, 0.91 versus 0.90; and on PAMAP2, 0.91 versus 0.72. On average, the proposed model's F1-score was 10% higher than that of McFly across all datasets.

McFly demonstrated the applicability of its framework to activity recognition datasets; however, employing model architectures such as InceptionTime and ResNet is difficult to justify in biomechanics, where study designs often include only 2–20 participants (Vagenas et al., 2018). The only dataset for which the alternative hypothesis was rejected was Opportunity ($p = 0.58$, Fig. 6a). The large number of sensors used (Fig. 2d) created a high-dimensional feature space that favored the architectures generated by McFly. Nevertheless, such sensor configurations are uncommon in real-world applications, where devices must meet comfort demands of the user (Shull and Damian, 2015).

As part of the second hypothesis, the proposed framework achieved a higher F1-score compared to DeepConvLSTM (Fig. 6b). This finding underscores the significance of automation in surpassing human-informed decisions on window and overlap sizes as well as architecture design based on the nature of the performed actions.

While DL models can automatically learn discriminative features from raw sensor data, their optimal performance typically depends on recruiting a large number of participants to discover discriminative features (Goodfellow et al., 2016), which is often impractical in biomechanics studies. Leveraging domain knowledge from expert biomechanists offers an effective way to mitigate this limitation. For instance, it was observed that features extracted from stair ascent and descent exhibited partial overlap (Fig. 5a). However, biomechanics indicates that stair ascent involves significantly greater hip and knee angles and moments, whereas stair descent is characterized by larger ankle dorsiflexion and plantarflexion angles (Protopapadaki et al., 2007). Such kinematic and kinetic relationships can be included as a function to create a handcrafted feature if the required sensors on specific body segments are available. The concatenation of customizable features with automatically extracted features can substantially improve model performance (Dong et al., 2018).

The total inference time for three of the datasets was below 200 ms, which is faster than the 700 ms requirement reported for pre-impact fall detection (Hu and Qu, 2016) and the 300 ms inference threshold noted for prosthetic control (Pew and Klute, 2017; Mohasel et al., 2025a). Notably, the generated DL models account for only a minor fraction of the total inference time, and the STFT emerged as the primary bottleneck. Optimizing model architectures for smaller window sizes can substantially reduce the time required for STFT computation when both performance and inference speed are important to the user.

The proposed design follows best practices by streamlining the organization of subjects' data into separate training, validation, and test sets (Halilaj et al., 2018), and by automatically identifying the optimal window and overlap sizes for capturing individual movement patterns. Furthermore, the framework is equipped to handle class imbalance, a persistent challenge in biomechanics that is often overlooked (Halilaj et al., 2018). The GA determines the optimal weight of each class, and a weighted cross-entropy or focal loss function addresses the class imbalance issue. Overall, the improved outcomes and the automated

nature indicate the potential of our framework as a valuable tool for biomechanists.

4.1. Limitations:

- **Models:** Model development was limited to dual-stream CNN architectures for hypothesis analysis. The high computational demand of neural architecture search in both the proposed framework and McFly constrained the exploration of other types of DL models within the framework. The final version of our framework will support model creation in the time-domain stream using long short term memory (LSTM), gated recurrent units, and ConvLSTM, which are among the most commonly used models in human activity recognition.

- **Hyperparameter Range:** Broad hyperparameter ranges were deliberately selected to facilitate adoption by users without ML expertise across various datasets. However, the framework permits customization of these ranges (e.g., reducing network depth to 2–3 convolutional blocks per stream, reducing filter counts to the range of 16 to 128, and reducing the GA population size to 20 with 10 generations) for users wishing to constrain the search space for smaller biomechanical datasets or reduced computational resources.

- **Generalizability:** The framework was validated on four public datasets with fixed sensor placements and activity types. Future work should evaluate the robustness of the developed models in real-world scenarios, such as those involving sensor displacement, occlusion, or atypical movement patterns.

5. Conclusion

This article introduces a framework that automates the stages of model development in ML. Automation applies to data splitting, data segmentation, neural architecture search, and hyperparameter optimization in both the time and frequency domains, as well as addressing class imbalance to enable the model to self-adapt to various types of input sensor data. The joint contribution of time- and frequency-domain features was validated in three of the four activity recognition datasets. The exported models outperformed those generated by McFly and DeepConvLSTM in three out of four datasets. The proposed framework provides a practical pathway for biomechanists with limited ML background to develop DL models more efficiently.

CRediT authorship contribution statement

Seyed Mojtaba Mohasel: Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Alireza Afzal Aghaei:** Writing – review & editing, Writing – original draft, Software, Formal analysis, Data curation. **John W. Sheppard:** Writing – review & editing, Supervision. **Corey A. Pew:** Writing – review & editing, Supervision, Methodology, Conceptualization.

Ethics statement

This study used publicly available datasets. All data were previously collected in accordance with relevant ethical guidelines, and no new human or animal subjects were involved in this research. Therefore, ethical approval was not required for this study.

Code availability

The implementation of the proposed method is available at https://github.com/MojtabaMohasel/AutoDL_DualStream.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

Computational efforts were performed on the Tempest High Performance Computing System, operated and supported by University Information Technology Research Cyberinfrastructure at Montana State University.

Appendix A

This section presents the subject-wise distribution of training, validation, and test splits across all datasets used in the study.

Table 1
Subject distribution across training, validation, and test sets for the AReM dataset.

Class	Train		Validation		Test	
	Subjects	Percentage	Subjects	Percentage	Subjects	Percentage
Standing	2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	18.2%	3	14.3%	1,4	14.3%
Bending2	2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	4.5%	3	14.3%	1,4	14.3%
Cycling	2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	18.2%	3	14.3%	1,4	14.3%
Lying	2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	18.2%	3	14.3%	1,4	14.3%
Sitting	2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	18.2%	3	14.3%	1,4	14.3%
Walking	2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	18.2%	3	14.3%	1,4	14.3%
Bending1	2, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15	4.5%	3	14.3%	1,4	14.3%

Table 2
Subject distribution across training, validation, and test sets for the WISDM dataset.

Class	Train		Validation		Test	
	Subjects	Percentage	Subjects	Percentage	Subjects	Percentage
Downstairs	1, 3, 4, 5, 6, 7, 11, 12, 14, 16, 18, 19, 20, 22, 23, 24, 26, 28, 29, 30, 32, 33, 34, 36	8.9%	13, 15, 17, 21, 27	10.1%	8, 10, 31	9.2%
Jogging	1, 2, 3, 4, 5, 6, 7, 11, 12, 14, 18, 19, 20, 22, 23, 24, 25, 26, 29, 32, 33, 34, 35, 36	31.7%	13, 15, 17, 21, 27	29.1%	8, 10, 31	30.3%
Sitting	3, 4, 5, 6, 7, 12, 16, 18, 19, 20, 24, 29, 30, 32, 33, 34, 35, 36	6.2%	13, 21, 27	2.9%	8, 31	4.0%
Standing	3, 5, 6, 7, 12, 16, 18, 19, 20, 24, 28, 29, 30, 32, 33, 34, 35, 36	4.3%	13, 21, 27	3.6%	8, 10, 31	6.3%
Upstairs	1, 3, 4, 5, 6, 7, 11, 12, 14, 16, 18, 19, 20, 22, 23, 24, 26, 28, 29, 30, 32, 33, 34, 36	11.0%	13, 15, 17, 21, 27	12.0%	8, 10, 31	11.2%
Walking	1, 2, 3, 4, 5, 6, 7, 11, 12, 14, 16, 18, 19, 20,22,23, 24, 25, 26, 28, 29, 30, 32, 33, 34, 35, 36	37.8%	9, 13, 15, 17, 21, 27	42.3%	8, 10, 31	39.1%

Table A.3
Subject distribution across training, validation, and test sets for the PAMAP2 dataset.

Class	Train		Validation		Test	
	Subjects	Percentage	Subjects	Percentage	Subjects	Percentage
Lying	1, 2, 3, 4, 6, 8	10.0%	5	8.7%	7	10.7%
Sitting	1, 2, 3, 4, 6, 8	10.2%	5	9.9%	7	5.1%
Standing	1, 2, 3, 4, 6, 8	9.9%	5	8.1%	7	10.8%
Walking	1, 2, 3, 4, 6, 8	12.1%	5	11.8%	7	14.1%
Running	1, 2, 4, 6, 8	4.9%	5	9.0%	7	1.5%
Cycling	1, 2, 4, 6, 8	8.2%	5	9.0%	7	9.5%
Nordic Walking	1, 2, 4, 6, 8	9.3%	5	9.6%	7	12.0%
Ascending Stairs	1, 2, 3, 4, 6, 8	6.0%	5	5.2%	7	7.4%
Descending Stairs	1, 2, 3, 4, 6, 8	5.6%	5	4.7%	7	4.9%
Vacuum Cleaning	1, 2, 3, 4, 6, 8	9.0%	5	9.0%	7	9.0%
Ironing	1, 2, 3, 4, 6, 8	12.3%	5	12.1%	7	12.3%
Rope Jumping	1, 2, 6, 8	2.5%	5	2.8%	9	2.7%

Table A.4
Subject distribution across training, validation, and test sets for the Opportunity dataset.

Class	Train		Validation		Test	
	Subjects	Percentage	Subjects	Percentage	Subjects	Percentage
Null	1D, 1A1, 1A2, 1A3, 1A4, 1A5, 2D, 2A1, 2A2, 2A3, 3D	17.0%	3A1, 3A2	14.4%	2A4, 2A5, 3A3, 3A4, 3A5	20.6%
Stand	1D, 1A1, 1A2, 1A3, 1A4, 1A5, 2D, 2A1, 2A2, 2A3, 3D	41.9%	3A1, 3A2	39.8%	2A4, 2A5, 3A3, 3A4, 3A5	31.4%
Walk	1D, 1A1, 1A2, 1A3, 1A4, 1A5, 2D, 2A1, 2A2, 2A3, 3D	23.2%	3A1, 3A2	24.4%	2A4, 2A5, 3A3, 3A4, 3A5	22.9%
Sit	1D, 1A1, 1A2, 1A3, 1A4, 1A5, 2D, 2A1, 2A2, 2A3, 3D	16.0%	3A1, 3A2	15.4%	2A4, 2A5, 3A3, 3A4, 3A5	20.4%
Lie	1D, 1A1, 1A2, 1A3, 1A4, 1A5, 2D, 2A1, 2A2, 2A3, 3D	1.9%	3A1, 3A2	6.0%	2A4, 2A5, 3A3, 3A4, 3A5	4.7%

Appendix B

This section details the mathematical formulation of the Short-Time Fourier Transform (STFT) used for generating time–frequency representations of the sensor data.

The STFT is applied to one-dimensional time-series window data as follows:

$$Y[\omega, m, c] = \sum_{k=0}^{N-1} w[k] X[mH + k, c] \exp\left(-j \frac{2\pi\omega k}{N}\right), \tag{B.1}$$

where m is the index of the sliding window, ω is the frequency bin index, where $0 \le \omega < \left\lfloor \frac{N}{2} \right\rfloor + 1$, c is the channel index, $w[k]$ is the Hann window function applied to each segment of the signal, which is defined as:

$$w[k] = \frac{1}{2} \times \left(1 - \cos\left(\frac{2\pi k}{N-1}\right)\right), \tag{B.2}$$

in which $X[m \times H + k, c]$ represents the signal segment indexed by m, windowed by k, and corresponding to channel c, H is the hop size, determining the step size between successive windows. The exponential term $\exp\left(-j \frac{2\pi \cdot \omega k}{N}\right)$ applies the Fourier Transform to each windowed segment. Hence, $Y \in \mathbb{C}^{\left(\left\lfloor \frac{N}{2} \right\rfloor + 1\right) \times M \times c}$ given $X \in \mathbb{R}^{W \times c}$ where W denotes the window size.

Appendix C

This section describes the loss functions employed in the framework, including weighted cross-entropy and focal loss for handling class imbalance.

The weighted cross-entropy loss is formulated as follows:

$$CE = -\sum_{i}^{c} w_i t_i \log(f(S)_i), \tag{C.1}$$

where w_i is the class-specific weight (higher for minority classes), t_i is the true label (1 if the sample belongs to class i, 0 otherwise), and $f(S)_i$ is the predicted probability obtained after applying the softmax function to the logits S.

Focal loss introduces a modulating factor to down-weight well-classified samples and focus learning on harder samples:

$$\mathcal{L}(\gamma) = -(1-p)^{\gamma} \log(p), \tag{C.2}$$

where p represents the predicted probability of the correct class, and γ (determined by GA) controls the degree of focusing on hard examples. When $\gamma = 0$, focal loss reduces to standard cross-entropy loss. GA identifies the appropriate loss function for the input data and determines the optimal weights for the loss function.

Appendix D

This section defines the statistical hypotheses used to evaluate the contribution of time- and frequency-domain features to model performance.

Hypothesis 1 tests whether combining time- and frequency-domain features leads to higher macro-averaged F1-scores than using only time-domain features.

- **Null Hypothesis** (H_0^1): The median macro-averaged F1-score of the combined (Time + Frequency) model is less than or equal to that of the Time-only model. Formally,

  $H_0^1 : \tilde{F1}_{\text{Time+Freq}} \le \tilde{F1}_{\text{Time}}$.

- **Alternative Hypothesis** (H_1^1): The median macro-averaged F1-score of the combined (Time + Frequency) model is greater than that of the Time-only model. Formally,

  $H_1^1 : \tilde{F1}_{\text{Time+Freq}} > \tilde{F1}_{\text{Time}}$.

Hypothesis 2 tests whether combining time- and frequency-domain features leads to higher macro-averaged F1-scores than using only frequency-domain features.

- **Null Hypothesis** (H_0^2): The median macro-averaged F1-score of the combined (Time + Frequency) model is less than or equal to that of the Frequency-only model. Formally,

  $H_0^2 : \tilde{F1}_{\text{Time+Freq}} \le \tilde{F1}_{\text{Freq}}$.

- **Alternative Hypothesis** (H_1^2): The median macro-averaged F1-score of the combined (Time + Frequency) model is greater than that of the Frequency-only model. Formally,

$$H_1^2 : \tilde{F1}_{\text{Time+Freq}} > \tilde{F1}_{\text{Freq}}.$$

Hypothesis 3 tests whether there is a significant difference in macro-averaged F1-scores between models trained with time-domain features and those trained with frequency-domain features.

- **Null Hypothesis** (H_0^3): The median macro-averaged F1-scores of the Time-only and Frequency-only models are equal. Formally,

$$H_0^3 : \tilde{F1}_{\text{Time}} = \tilde{F1}_{\text{Freq}}.$$

- **Alternative Hypothesis** (H_1^3): The median macro-averaged F1-scores of the Time-only and Frequency-only models differ. Formally,

$$H_1^3 : \tilde{F1}_{\text{Time}} \neq \tilde{F1}_{\text{Freq}}.$$

Appendix E

This section outlines the statistical hypotheses used to compare the proposed AutoDL framework with baseline models.

Hypothesis 1 tests whether the proposed AutoDL framework achieves higher macro-averaged F1-scores than an existing open-source AutoDL system (McFly) across multiple activity recognition datasets.

- **Null Hypothesis** (H_0^1): The median macro-averaged F1-score of the proposed AutoDL framework is less than or equal to that of McFly. Formally,

$$H_0^1 : \tilde{F1}_{\text{Proposed}} \leq \tilde{F1}_{\text{McFly}}.$$

- **Alternative Hypothesis** (H_1^1): The median macro-averaged F1-score of the proposed AutoDL framework is greater than that of McFly. Formally,

$$H_1^1 : \tilde{F1}_{\text{Proposed}} > \tilde{F1}_{\text{McFly}}.$$

Hypothesis 2 tests whether the proposed AutoDL framework achieves higher macro-averaged F1-scores than a manually designed deep learning architecture (DeepConvLSTM) across the same datasets.

- **Null Hypothesis** (H_0^2): The median macro-averaged F1-score of the proposed AutoDL framework is less than or equal to that of the DeepConvLSTM model. Formally,

$$H_0^2 : \tilde{F1}_{\text{Proposed}} \leq \tilde{F1}_{\text{DeepConvLSTM}}.$$

- **Alternative Hypothesis** (H_1^2): The median macro-averaged F1-score of the proposed AutoDL framework is greater than that of the DeepConvLSTM model. Formally,

$$H_1^2 : \tilde{F1}_{\text{Proposed}} > \tilde{F1}_{\text{DeepConvLSTM}}.$$

Appendix F

This section presents a biomechanical validation of the learned frequency-domain features using spectral analysis and Grad-CAM-based interpretation.

A two-stage validation was conducted to ensure the proposed AutoDL framework utilizes physically grounded features. First, a segment-specific spectral analysis was performed on the raw sensor data to characterize the physical cadences associated with each activity. Subsequently, an interpretative saliency analysis named Gradient-weighted Class Activation Mapping (Grad-CAM) was executed to verify that the deep network's frequency-domain branch specifically attends to these biomechanically relevant features for classification.

Analysis 1: Welch's method was applied to the vector magnitude of the thigh-segment acceleration to identify dominant rhythmic frequencies. As shown in Fig. F.7 distinct spectral representations were identified for various activities: 'Walking' exhibited a frequency peak at 2.01 Hz (120.6 steps/min), while 'Jogging' shifted to 2.56 Hz (153.4 steps/min). These findings indicate that different activities possess unique spectral signatures within the functional human movement range, providing a meaningful physical basis for the use of frequency-domain classification in addition to time-domain features.

Analysis 2: A saliency analysis was performed to examine whether the dual-stream network effectively prioritizes these spectral differences. Grad-CAM was applied to the 2D-CNN branch to visualize model attention in the time-frequency domain. As illustrated in Fig. F.8, the network's high-importance regions (red hotspots) align precisely with the rhythmic periodicities identified in the first analysis. Specifically, for 'Walking', the model's attention is concentrated in the 2 Hz band, whereas for 'Jogging', the attention shifts to the 2.6 Hz band and its

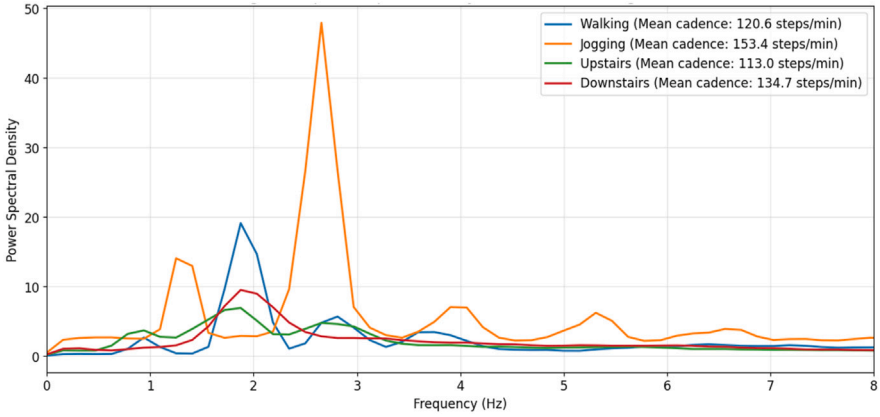

Fig. F.7. Segment-specific spectral analysis and frequency-cadence Mapping of thigh-mounted sensor data. The plot illustrates the mean power spectral density (PSD) of the Acceleration vector magnitude for four distinct activities (Walking, Jogging, Upstairs, and Downstairs). Dominant spectral peaks within the functional human gait range (1.0–3.0 Hz) represent the fundamental frequency of movement for each activity. The legend provides the quantitatively mapped mean physical cadence in steps per minute (SPM), calculated from the peak frequency (f_{peak}) as SPM = $f_{\text{peak}} \times 60$. Note the significant rightward shift and prominent higher-frequency harmonics (at ~4 Hz and ~5.3 Hz) in the jogging profile (orange) compared to the walking profile (blue), reflecting the increased step rate and higher-intensity impact transients characteristic of running.

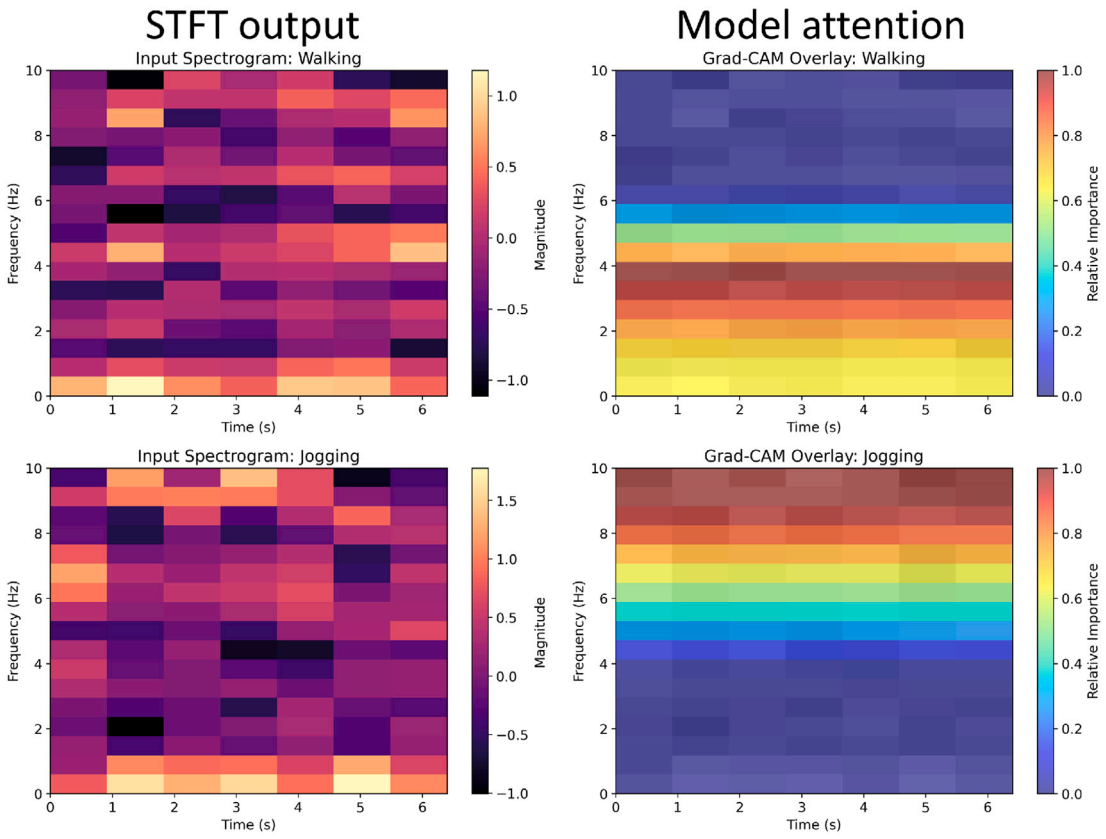

Fig. F.8. Visual interpretation of frequency-domain features using Grad-CAM. Left panels illustrate the input Spectrograms (log-magnitude) derived from the thigh-mounted sensor for walking (top) and jogging (bottom) over a 6.4-second window. Right panels show the corresponding Grad-CAM saliency overlays, where red hotspots indicate spectral regions of high relative importance for model classification. A distinct spectral shift in model attention is observed: for walking (top-right), the network prioritizes the fundamental gait frequency and lower harmonics (2.0–4.0 Hz). Conversely, for jogging (bottom-right), the attention shifts to higher-frequency components (6.0–10.0 Hz), capturing the increased cadence and higher-intensity impact transients characteristic of running. This spatial alignment between model attention and physical movement periodicities validates that the network's frequency branch successfully utilizes Biomechanically relevant features for activity discrimination.

associated harmonics. This spatial alignment confirms that the AutoDL framework has successfully learned to extract and utilize physically valid biomechanical features for activity discrimination.

Data availability

Data sharing is not applicable to this article as no new data were created in this study.

References

Ahmad, N., Ghazilla, R.A.R., Khairi, N.M., Kasi, V., 2013. Reviews on various inertial measurement unit (IMU) sensor applications. Int. J. Signal Process. Syst. 1, 256–262.

Banos, O., Galvez, J.-M., Damas, M., Pomares, H., Rojas, I., 2014. Window size impact in human activity recognition. Sensors 14, 6474–6499.

Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S.T., Tröster, G., Millán, J.D.R., Roggen, D., 2013. The opportunity challenge: a benchmark database for on-body sensor-based activity recognition. Pattern Recognit. Lett. 34, 2033–2042.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.

Dong, M., Han, J., He, Y., Jing, X., 2018. HAR-net: fusing deep representation and hand-crafted features for human activity recognition. In: International Conference on Signal and Information Processing, Networking and Computers. Springer, pp. 32–40.

Géron, A., 2022. Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media, Inc.

Glaister, B.C., Bernatz, G.C., Klute, G.K., Orendurff, M.S., 2007. Video task analysis of turning during activities of daily living. Gait Posture 25, 289–294.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press, Cambridge, MA, USA.

Gu, F., Chung, M.-H., Chignell, M., Valaee, S., Zhou, B., Liu, X., 2021. A survey on deep learning for human activity recognition. ACM Computing Surveys (CSUR) 54, 1–34.

Gupta, S., 2021. Deep learning based human activity recognition (HAR) using wearable sensor data. Int. J. Inf. Manag. Data Insights 1, 100046.

Halilaj, E., Rajagopal, A., Fiterau, M., Hicks, J.L., Hastie, T.J., Delp, S.L., 2018. Machine learning in human movement biomechanics: best practices, common pitfalls, and new opportunities. J. Biomech. 81, 1–11.

Hu, X., Qu, X., 2016. Pre-impact fall detection. Biomed. Eng. Online 15, 61.

Ismail, W.N., Alsalamah, H.A., Hassan, M.M., Mohamed, E., 2023. AUTO-HAR: an adaptive human activity recognition framework using an automated CNN architecture design. Heliyon 9.

Jiang, S., Strout, Z., He, B., Peng, D., Shull, P.B., Lo, B.P.L., 2023. Dual stream meta learning for road surface classification and riding event detection on shared bikes. IEEE Trans. Syst. Man Cybern. Syst 53 (11), 7188–7200.

Kavanagh, J.J., Menz, H.B., 2008. Accelerometry: a technique for quantifying movement patterns during walking. Gait Posture 28, 1–15.

Krasoulis, A., Vijayakumar, S., Nazarpour, K., 2019. Multi-grip classification-based prosthesis control with two EMG-IMU sensors. IEEE Trans. Neural Syst. Rehabil. Eng. 28, 508–518.

Kwapisz, J.R., Weiss, G.M., Moore, S.A., 2011. Activity recognition using cell phone accelerometers. ACM SIGKDD Explor. Newsl. 12, 74–82.

Lee, S.-M., Yoon, S.M., Cho, H., 2017. Human activity recognition from accelerometer data using convolutional neural network. In: 2017 IEEE International Conference on Big Data and Smart Computing (Bigcomp). IEEE, pp. 131–134.

Maaten van der, L., Hinton, G., 2008. Visualizing data using t-SNE. J. Mach. Learn. Res. 9, 2579–2605.

Mohasel, S., Aghaei, A.A., Pew, C., 2025a. Personalized Control for Lower Limb Prosthesis Using Kolmogorov-Arnold Networks. arXiv preprint arXiv:2505.09366.

Mohasel, S.M., Koosha, H., 2026. A robust and lightweight support vector machine for imbalanced and noisy data via benders decomposition. Neurocomputing 132629.

Mohasel, S.M., Sheppard, J., Molina, L.K., Neptune, R.R., Wurdeman, S.R., Pew, C.A., 2025b. MicroNAS: An Automated Framework for Developing a Fall Detection System. arXiv preprint arXiv:2504.07397.

Oguiza, I., 2022. Tsai-A State-of-the-Art Deep Learning Library for Time Series and Sequential Data. Github, San Francisco, CA, USA.

Ordóñez, F.J., Roggen, D., 2016. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. Sensors 16, 115.

Palumbo, F., Barsocchi, P., Gallicchio, C., Chessa, S., Micheli, A., 2013. Multisensor data fusion for activity recognition based on reservoir computing. In: Evaluating AAL Systems

Through Competitive Benchmarking: International Competitions and Final Workshop, EvAAL 2013, July and September 2013. Proceedings 3. Springer, pp. 24–35.
Pesenti, M., Invernizzi, G., Mazzella, J., Bocciolone, M., Pedrocchi, A., Gandolla, M., 2023. IMU-based human activity recognition and payload classification for low-back exoskeletons. Sci. Rep. 13, 1184.
Pew, C., Klute, G.K., 2017. Turn intent detection for control of a lower limb prosthesis. IEEE Trans. Biomed. Eng. 65, 789–796.
Probst, P., Boulesteix, A.-L., Bischl, B., 2019. Tunability: importance of hyperparameters of machine learning algorithms. J. Mach. Learn. Res. 20, 1–32.
Protopapadaki, A., Drechsler, W.I., Cramp, M.C., Coutts, F.J., Scott, O.M., 2007. Hip, knee, ankle kinematics and kinetics during stair ascent and descent in healthy young individuals. Clin. Biomech. 22, 203–210.
Rainio, O., Teuho, J., Klén, R., 2024. Evaluation metrics and statistical tests for machine learning. Sci. Rep. 14, 6086.
Reiss, A., Stricker, D., 2012. Introducing a new benchmarked dataset for activity monitoring. In: 2012 16th International Symposium on Wearable Computers. IEEE, pp. 108–109.
Revin, I., Potemkin, V.A., Balabanov, N.R., Nikitin, N.O., 2023. Automated machine learning approach for time series classification pipelines using evolutionary optimization. Knowl.-based Syst. 268, 110483.
Salehin, I., Islam, M.S., Saha, P., Noman, S.M., Tuni, A., Hasan, M.M., Baten, M.A., 2024. AutoML: a systematic review on automated machine learning with neural architecture search. J. Inf. Intell. 2, 52–81.
Sani, S., Massie, S., Wiratunga, N., Cooper, K., 2017. Learning deep and shallow features for human activity recognition. In: International Conference on Knowledge Science, Engineering and Management. Springer, pp. 469–482.
Sejdić, E., Lowry, K.A., Bellanca, J., Redfern, M.S., Brach, J.S., 2013. A comprehensive assessment of gait accelerometry signals in time, frequency and time-frequency domains. IEEE Trans. Neural Syst. Rehabil. Eng. 22, 603–612.
Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626.
Shull, P.B., Damian, D.D., 2015. Haptic wearables as sensory replacement, sensory augmentation and trainer–a review. J. Neuroeng. Rehabil. 12, 59.
Singh, D., Merdivan, E., Kropf, J., Holzinger, A., 2025. Class imbalance in multi-resident activity recognition: an evaluative study on explainability of deep learning approaches. Univers. Access Inf. Soc. 24, 1173–1191.
Spangler, J., Mitjans, M., Collimore, A., Gomes-Pires, A., Levine, D.M., Tron, R., Awad, L.N., 2024. Automation of functional mobility assessments at home using a multimodal sensor system integrating inertial measurement units and computer vision (IMU-vision). Phys. Ther. 104, pzad184.
Vagenas, G., Palaiothodorou, D., Knudson, D., 2018. Thirty-year trends of study design and statistics in applied sports and exercise biomechanics research. Int. J. Exerc. Sci. 11, 239–259.
Van Kuppevelt, D., Meijer, C., Huber, F., Van der Ploeg, A., Georgievska, S., van Hees, V.T., 2020. Mcfly: automated deep learning on time series. Softwarex 12, 100548.
Van Rijn, J.N., Hutter, F., 2018. Hyperparameter importance across datasets. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2367–2376.
Wang, J., Chen, Y., Hao, S., Peng, X., Hu, L., 2019. Deep learning for sensor-based activity recognition: a survey. Pattern Recognit. Lett. 119, 3–11.
Weerts, H.J.P., Mueller, A.C., Vanschoren, J., 2020. Importance of tuning hyperparameters of machine learning algorithms. arXiv preprint arXiv:2007.07588.
Wu, D., Guan, Q., Fan, Z., Deng, H., Wu, T., 2022. AutoML with parallel genetic algorithm for fast hyperparameters optimization in efficient IoT time series prediction. IEEE Trans. Ind. Inform. 19, 9555–9564.
Yadav, D., Veer, K., 2023. Recent trends and challenges of surface electromyography in prosthetic applications. Biomed. Eng. Lett. 13, 353–373.