Semantic Noise Filtering in Post-Stroke Kinematic Data: An Automated Deep Learning Approach

Victor Y. K. Kwon[1], Luis G. S. Rodrigues[2], Bartolomeu Zamprogno[1], Marcelo P. Guimarães[3], Leonardo C. D. Rocha[4], Eduardo Zambon[1], Vinícius F. S. Mota[1], and Diego R. C. Dias[1(✉)]

[1] Universidade Federal do Espírito Santo (UFES), Vitória, Brazil
victor.kwon@edu.ufes.br, {bartolomeu.zamprogno,eduardo.zambon, vinicius.f.mota,diego.dias}@ufes.br
[2] Universidade Estadual Paulista (UNESP), São Paulo, Brazil
guilherme.rodrigues@unesp.br
[3] Universidade Federal de São Paulo (UNIFESP), São Paulo, Brazil
marcelo.paiva@unifesp.br
[4] Universidade Federal de São João del-Rei (UFSJ), São João del-Rei, Brazil
lcrocha@ufsj.edu.br

Abstract. Post-stroke rehabilitation increasingly relies on motion capture technologies and extended reality (XR). However, datasets collected in these environments, such as those stored in the Rehabilitation Database (ReBase), are often contaminated by noise. A critical challenge in this domain is distinguishing between clinically relevant variations, such as compensatory movements, and spurious artifacts caused by gestures unrelated to the rehabilitation task. This work proposes a computationally efficient, automated filtering approach based on a multilayer perceptron (MLP). Unlike complex recurrent architectures, the MLP we propose effectively captures gait patterns with observation intervals as short as 5 s, making the solution computationally inexpensive and highly discriminative. We trained the model based on knee angle kinematics, reaching an overall accuracy of 93.18%. Regarding precision, the system achieved 100% accuracy (zero false positives), ensuring that no artifacts are erroneously preserved as valid data. Thus, results indicate that the model successfully generalizes to new patients without manual calibration. By replacing manual curation with this automated deep learning solution, our method improves the Rehabilitation Database scalability, by optimizing storage efficiency and enabling reliable monitoring of motor recovery treatments.

Keywords: ReBase · Neural Network · Data Filtering · Neurological Rehabilitation · Body Kinematics

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16768, pp. 601–617, 2027.
https://doi.org/10.1007/978-3-032-30488-9_37

1 Introduction

A stroke occurs when cerebral blood flow is interrupted, caused either by the obstruction (ischemia) or rupture (hemorrhage) of a blood vessel, resulting in neural cell death. This event frequently leads to severe brain damage, which manifests clinically as body paralysis (hemiparesis), loss of motor coordination, and functional deficits in one side of the body. To mitigate these sequelae, neurofunctional rehabilitation involves systematic, repetitive stimulation of muscle groups to promote neuroplasticity and motor relearning.

Within the context of neuromotor rehabilitation, extended reality (XR) has been employed as an effective clinical tool [2,4,10,12]. XR can be conceptualized as the combination of virtual reality (VR), augmented reality (AR), and mixed reality (MR) interface technologies. Given that XR enables the development of simulation environments and serious games, when it is used as a rehabilitation protocol the engagement these environments foster tends to reduce the monotony of conventional rehabilitation exercises.

In XR environments, a very important characteristic is the mode of interaction. Generally, a patient uses their own body as the mode of interaction, something made possible by natural user interfaces (NUIs) that eliminate the need for conventional devices such as keyboards and mice, which require fine motor movement that are often compromised in neurological patients. Therefore, NUI-based technologies enable the body itself to act as an interaction device, with tracking often performed using depth sensors (RGB-D) and inertial measurement units (IMUs), that digitize user movements into biomechanical data.

Although NUIs facilitate tracking the user body, such sensors introduce a problem inherent to this type of application: *continuous user tracking*. In motor rehabilitation scenarios, especially in home use where supervision by a healthcare professional is not always available, data acquisition can be hampered by movements unrelated to treatment, such as postural adjustments, distracting gestures, or interruptions during the session. This causes data collection to be extremely noisy, making it hard to later separate actual physiotherapy movements from undesirable ones.

In this article, we propose and implement a data filter to remove undesirable movements during physiotherapy sessions. We validate our method with ReBase [15], a database specialized in body data management. In this database, small variations, such as fatigue tremors and motor compensations, coexist with undesirable movements during sessions. Therefore, we apply our filter to the data already stored in ReBase in order to remove undesirable movements.

Our main goal is to develop a neural network architecture capable of classifying kinematic observation intervals and of accurately separating legitimate motor behavior from noisy artifacts. This approach aims to identify and remove inconsistencies unrelated to the rehabilitation task, retaining only clinical markers, such as fatigue patterns and compensatory movements inherent to the movement type. In this way, we can ensure that the monitoring of post-stroke recovery is based on anonymized data that is highly clinically representative.

The remainder of this paper is organized as follows. In Sect. 2, we present a brief review of related work, contextualizing our proposal with respect to existing classical and deep learning-based filtering approaches in the literature. In Sect. 3, we establish the theoretical framework, detailing the fundamentals of multilayer perceptron (MLP) networks and the regularization techniques used to develop our model. In Sect. 4, we describe the methodology employed in the study, covering the entire process, from the acquisition and preprocessing of kinematic data, to the definition of the neural network architecture and training strategy. In Sect. 5, we present our results, including quantitative performance metrics and a qualitative analysis of filtering in validation and generalization scenarios. Finally, in Sect. 6, we present the conclusions and possible directions for future work within the context of neuromotor rehabilitation.

2 Related Work

Recent literature indicates significant progress in the creation of data repositories dedicated to motor rehabilitation. Several studies have focused on curating multimodal databases generated from depth sensors (RGB-D) and inertial measurement units (IMUs), such as the UI-PRMD dataset [16], which catalogs physiotherapy movements using Kinect and Vicon sensors; and KIMORE [3], a solution that provides kinematic records focused on rehabilitation exercises with clinical-quality annotations. On a broader scale, initiatives such as NTU RGB+D 120 [9] have established relevant deep learning-based parameters for skeletal analysis. Similarly, in the specific context of stroke, IntelliRehabDS [11] expanded this approach by including data from real patients performing motor tasks under various conditions.

Another line of research focus on automated analysis of movement quality. Initially, approaches based on convolutional neural networks (CNNs) and long-term memory networks (LSTMs) were widely used to extract spatiotemporal features. Works such as those by Liao *et al.* [8] demonstrate the effectiveness of these architectures in classifying execution (correct vs. incorrect). More recently, the state of the art has evolved towards graphical convolutional networks (GCNs), which model the structure of the human skeleton as a topological graph. Yan *et al.* [18] introduced spatiotemporal GCNs (ST-GCNs), that allow for a more refined capture of joint dependencies, which several authors have successfully applied to detect subtle compensatory patterns in post-stroke patients.

A main limitation of the approaches just presented is the premise of *ideal segmentation*: A vast majority of these models assume that the input data stream is composed exclusively of pre-trimmed exercise attempts, containing only the action of interest. Consequently, a critical gap exits in the handling of continuous data collected in uncontrolled environments, where such prior segmentation is nonexistent.

Although classical signal processing techniques, such as Kalman filters [1], Butterworth filters [17], or B-Spline smoothing [14], are effective in attenuating high-frequency noise (such as sensor tremor or tracking instability), they are

incapable of distinguishing what we call *semantic noise*. Non-therapeutic functional movements, such as distraction gestures, rest pauses, water ingestion, or social interactions during the session, remain in the datasets, contaminating clinical analysis. Unlike traditional signal-cleaning approaches or supervised action segmentation [7], our proposal aims to perform unsupervised or semi-supervised semantic filtering by leveraging neural networks to identify and remove temporal segments that do not correspond to therapeutic activity, ensuring that the ReBase repository stores only clinically interpretable data.

Table 1 synthesizes the fundamental distinctions between the predominant filtering techniques in the literature and the approach proposed in this work. Based on this review, we observe that classical methods, such as Butterworth filter standardization [17] and Kalman filtering [1], are effective at attenuating stochastic noise. However, because they are applied purely in the frequency domain or for state estimation, these techniques lack contextual sensitivity, which tends to smooth out clinically relevant pathological tremors with spectral signatures similar to noise.

Unlike classical models, approaches focused on human activity recognition (HAR), such as modern spatial GCNs, prioritize classification performance but fail in continuous-flow scenarios where the number of irrelevant events is high. Our proposal fills this gap by introducing the concept of *semantic filtering*, in which the model not only processes the raw signal, but also identifies the movement intentionality, differentiating behaviors unrelated to the task (semantic noise) without compromising the integrity of motor recovery markers.

3 Theoretical Framework

3.1 Multilayer Perceptron (MLP)

In this work, we employ artificial neural networks (ANNs) to handle the complex task of classifying human movement patterns and distinguishing them from noise. The theoretical foundation and practical implementation of these models are based on the work of Goodfellow *et al.* [6] and Géron [5], respectively.

A MLP is a feedforward neural network composed of at least three layers: an input layer, that receives features of the raw data; one or more hidden layers, consisting of interconnected neurons (often called linear threshold units), that extract intermediate representations; and an output layer, that generates the network final prediction. Every other layer apart from the output layer typically includes a bias neuron and is fully connected to the subsequent layer. An architecture with multiple hidden layers between the input and output layers is called a deep neural network (DNN). A DNN can extract features and learn elaborated patterns from input data, something that would not otherwise be evident under simpler processing methods, such as linear analysis.

3.2 Learning and Optimization Process

Training of the MLP is formulated as a non-linear optimization problem that aims to minimize a cost function $J(\theta)$, where $\theta = \{W_l, b_l\}_{l=1}^{L}$ represents the

Table 1. Detailed comparative analysis between literature filtering/recognition approaches and our proposal in this work.

Approach	Characteristics and Limitations
Winter [17]	**Technique:** Butterworth Filter (low-pass). **Focus:** High-frequency noise reduction (sensor tremor). **Limitation:** May smooth pathological tremors and does not remove irrelevant voluntary movements.
Ahmed et al. [1]	**Technique:** Kalman Filter. **Focus:** State estimation and squared error minimization. **Limitation:** Blind to semantic context and assumes all captured movement is part of the task.
Liao et al. [8] **& Yan et al.** [18]	**Technique:** Deep Learning (skeleton-based CNNs and GCNs). **Focus:** Action recognition and movement quality. **Limitation:** Assumes pre-trimmed videos and fails to distinguish artifacts in unsupervised continuous streams.
Our Proposal	**Technique:** Multilayer Perceptron (MLP). **Focus:** Semantic filtering (block classification). **Advantage:** Learns latent gait morphology in fixed windows, removing irrelevant segments with low computational cost.

parameter set (weight matrices W_l and bias vectors b_l) of a network with L layers. Given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ of input vectors x_i and corresponding expected outputs y_i, the learning process via the backpropagation algorithm is described by the following steps:

1. **Forward Pass:** For each layer $l \in \{1, \ldots, L\}$, the pre-activation z_l and the activation a_l vectors are computed iteratively as:

$$z_l = W_l a_{l-1} + b_l, \tag{1}$$

$$a_l = g_l(z_l), \tag{2}$$

   where $a_0 = x$ (input vector) and g_l is the differentiable activation function of layer l (see below).
2. **Loss Function Evaluation:** The prediction error between network output $\hat{y}$ and expected output y is quantified by a loss function $\mathcal{L}(\hat{y}, y)$. In the context of binary classification, binary cross-entropy is frequently used, defined as:

$$\mathcal{L}(\hat{y}, y) = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right]. \tag{3}$$

3. **Backward Pass (backpropagation):** Gradient ∇_θ of the loss function $\mathcal{L}$ with respect to network parameters θ is calculated by applying the chain rule from Calculus. The error term δ_l for layer l is given by:

$$\delta_l = \left(W_{l+1}^T \delta_{l+1} \right) \odot g_l'(z_l), \tag{4}$$

where $\odot$ denotes the pointwise product and g_l' is the first derivative of activation function g_l. The required partial derivatives are then obtained by:

$$\frac{\partial \mathcal{L}}{\partial W_l} = \delta_l a_{l-1}^T, \quad \frac{\partial \mathcal{L}}{\partial b_l} = \delta_l. \tag{5}$$

4. **Parameter Update:** Weights are iteratively adjusted in the opposite direction of the gradient. For the commonly used stochastic gradient descent (SGD) method with a learning rate η, the update rule of network parameters θ for training step $t+1$, based on step t, is defined as:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta J(\theta_t), \tag{6}$$

where ∇_θ is the gradient of the loss computed in step 3. Modern variants of gradient descent algorithms, such as the Adam optimizer, adapt η based on the first and second-order moments of the gradients.

Activation Functions and Non-Linearity. The capacity of the network to approximate complex functions and model non-linear kinematics depends on the adequate use of activation functions $g(\cdot)$. The most common types of functions and their respective analytical formulations are:

– **Rectified Linear Unit (ReLU):** Predominant in hidden layers due to its computational efficiency and mitigation of the vanishing gradient problem.

$$\text{ReLU}(z) = \max(0, z) \tag{7}$$

– **Sigmoid (Logistic):** Used in the output layer to map activations into a probabilistic interval $(0, 1)$ to enable binary classification.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{8}$$

3.3 Regularization and Dropout

MLPs are susceptible to *overfitting*, which occurs when a model memorizes the training data instead of learning generalizable patterns from input. Given the noisy nature of rehabilitation data, mitigating overfitting is necessary, so a *dropout* technique was employed.

Dropout involves randomly deactivating a subset of neurons during training with a predefined probability p. Let a_l be the activation vector of layer l. With dropout, the modified activation $\tilde{a}_l$ is given by:

$$\tilde{a}_l = a_l \odot m_l, \quad m_l \sim \text{Bernoulli}(p), \tag{9}$$

where $\odot$ denotes pointwise multiplication, and $m(l)$ is a binary mask sampled from a Bernoulli distribution. By forcing the network to train with slightly different architectures at each iteration, neurons are prevented from co-adapting excessively, thereby promoting robustness. During inference, dropout is disabled, and weights are scaled by p to maintain network consistency.

4 Methodology

4.1 Data Acquisition and Preprocessing

The dataset used in this study consists of CSV files containing time-series data of knee flexion and extension angles (left and right knee). Raw data were obtained from a remote repository and subjected to an initial temporal segmentation stage. The complete implementation of the proposed MLP, along with pre-trained weights and the anonymized dataset, is available at our public repository[1].

To create the ground truth (supervised labeling), a threshold-based algorithm was implemented. The continuous data stream was segmented into non-overlapping 5-second time windows. Each window was classified in a binary manner as either a:

- **Valid Block (Walking):** Segments where the angular amplitude of either knee exceeded a pre-defined threshold (ranging between 30 and 52°C, specifically adjusted for each collection file); or
- **Invalid Block (Noise/Rest):** Segments where the angular variation remained below the established threshold.

We applied vector standardization to ensure consistent inputs to the neural network. Considering an approximate sampling rate of 30 Hz, a target size of 150 samples per knee was defined for each 5-second window. Windows with fewer samples were filled with zeros, while larger windows were truncated. Vectors for left and right knees were concatenated, yielding a 300-dimensional feature vector for each instance.

4.2 Dataset Splitting

We adopted the following cross-validation strategy: of the seven files initially processed, we used the first six to compose the training set (X_{train}, y_{train}), and the last file was used as the test set (X_{test}, y_{test}) to ensure the model was evaluated on data from a session never seen during training.

We shuffled the data before training the model in order to remove temporal dependencies. We also normalized the data using a standard scaler (zero mean and unit variance). The scaler was applied only to the training data to avoid data leakage.

4.3 Neural Network Architecture

We implemented an MLP-type neural network using the TensorFlow/Keras library to classify kinematic blocks. The proposed architecture consists of a deep MLP, structured to process input vectors of dimension 1×300, corresponding to the concatenation of the standardized time series of the knee angle. We define

[1] https://github.com/diegocolombodias/semanticnoisefiltering.git.

a dense layer configuration with decreasing dimensionality to extract latent features progressively. Thus, the processing flow begins with two consecutive hidden layers of 128 neurons each, followed by two layers of 64 neurons. Finally, the output layer is a single neuron (1×1) that performs binary classification on the validity of the kinematic segment.

Due to the noisy nature of kinematic data and the high dimensionality of the input vector, a regularization strategy is a very important step to ensure the generalization capability of the model. We implement an aggressive dropout in the initial and intermediate layers. Specifically, we applied a dropout rate ($p-1$) of 0.4 after the first three dense layers (128, 128, and 64 neurons). This forces the network to learn distributed, robust features, thereby avoiding excessive co-adaptation of neurons to specific patterns in the training set. In the penultimate layer (64 neurons), we reduced the rate to 0.2 in order to preserve the integrity of high-level features before the final classification layer. This approach prevents the model from memorizing the stochastic noise inherent in motion capture, favoring the learning of gait signal morphology.

Table 2. Summary of hyperparameters and configurations of our MLP.

Hyperparameter/Configuration	Value
Input Dimension	1×300
Number of Hidden Layers	4
Neurons per Layer	128, 128, 64, 64
Activation Function (Hidden)	ReLU
Activation Function (Output)	Sigmoid
Dropout Rates	0.4 (Layers 1–3), 0.2 (Layer 4)
Optimizer	Adam
Loss Function	Binary Cross-Entropy
Batch Size	32
Training Epochs	100
Classification Threshold	0.5

We composed the model with the Adam optimizer and binary cross-entropy loss function, and trained for 100 epochs with a batch size of 32. Table 2 presents all the hyperparameters defined for the network.

4.4 Validation and Visualization Strategy

We measured classifier performance on the test set using overall accuracy and the confusion matrix, which highlights the number of false positives (noise incorrectly classified as walking) and of false negatives (undetected walking). To verify the method applicability, we developed a visualization routine to generate comparative signal grids. The visual validation process consists of:

1. **Pair-by-pair comparison:** For each recording file, we generate two side-by-side graphs. The first displays the complete, unfiltered time series, highlighting all noise and motion artifacts. The second displays the reconstructed time series with only the blocks classified as valid by the model.
2. **Generalization test on unseen data:** In addition to the standard test data, we applied the model to three new data collection files (`knee_angles_corrigidoM1_48`, `M5_36`, and `M7_59`) that did not participate in any training or cross-validation steps.
3. **Acceptance criteria:** Validation is considered successful if the filtered graph preserves the morphological integrity of the knee flexion waves during activity periods, showing gaps (absence of data) during the periods corresponding to the noise observed in the original graph.

It is important to note that the nomenclature we adopted for the generalization datasets explicitly encodes the variability inherent in manual curation. The numerical suffixes in the filenames `M1_48`, `M5_36`, and `M7_59` correspond to the specific angular thresholds (48°, 36°, and 59°, respectively) required to generate the ground truth for each patient. This variance indicates that a static threshold approach cannot accommodate the patients diversity. By successfully filtering these signals without any input relating to these specific parameters, our proposed MLP demonstrates that it has learned the latent morphology of the gait signal, overcoming the need for individual calibration required by the manual method.

Additionally, we performed exploratory data visualization using Python scripts and the skeleton-viewer environment [13]. This tool enables visual, intuitive inspection of knee angular data, along with motion capture playback, as illustrated in Fig. 1. This visual approach allows us to identify model behaviors that textual metrics might obscure, such as excessive fragmentation of continuous blocks or the preservation of non-therapeutic movements.

5 Results and Discussion

In this section, we analyze the proposed model via neural architecture validation, quantitative evaluation of metrics, and qualitative inspection of filtering in known and unknown scenarios.

5.1 Architecture Validation and Metric Performance

Figure 2 shows the resulting confusion matrix obtained from the test set. Out of 440 analyzed blocks, the model correctly classified 410 instances, achieving a global accuracy of 93.18%. It should be highlighted that the model achieved zero false positives (100% accuracy). Within the ReBase ecosystem, this means that the neural network acts as a high-security filter, preventing spurious noise from being erroneously classified as exercise and ensuring the database integrity. The 30 false negatives observed (90.35% sensitivity) indicate a conservative model profile that prioritizes data purity over full recovery in extreme cases.

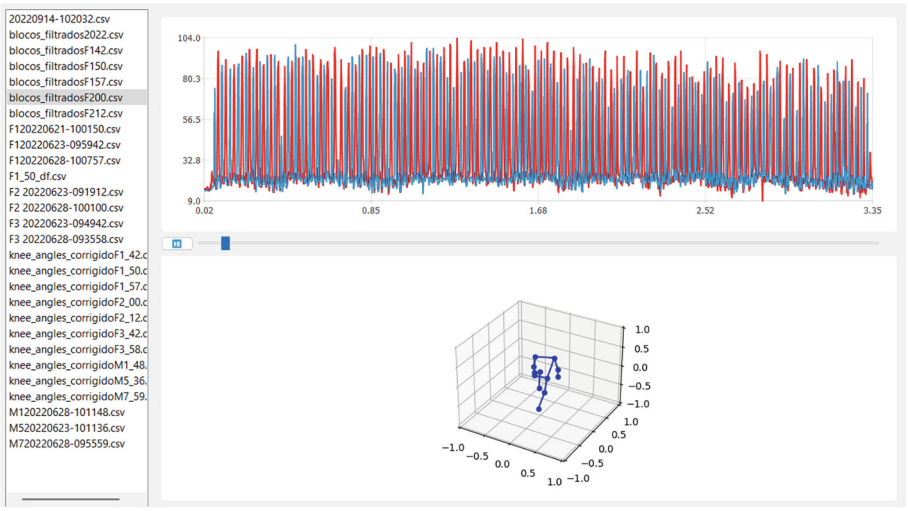

Fig. 1. The skeleton visualization environment displaying a CSV file, synchronizing knee angles with motion capture data.

Obtained results confirm the effectiveness of the proposed MLP architecture. The strategy of using four dense layers with progressive dimensionality reduction (128 → 64 neurons) and dropout regularization proved capable of modeling nonlinear gait features without overfitting.

5.2 Training Metrics and Convergence

We ran the neural network training process for 100 epochs, continuously monitoring loss and accuracy on the training and validation sets. Table 3 summarizes the final results obtained at the end of the learning cycle.

During training, we noted rapid model convergence. By the tenth epoch, the training accuracy exceeded 96%, indicating that the proposed architecture has sufficient capacity to model the complexities of the kinematic dataset. At the end of 100 epochs, the model achieved a training accuracy of 99.68% and a minimal residual loss of 0.0095, demonstrating almost complete learning of the patterns present in the training set.

Regarding generalization, the model maintained robust performance on the test set (unseen data), stabilizing at an accuracy of 93.18%. Although there is a divergence between training loss and validation loss in the final epochs, a behavior inherent to deep networks and a potential onset of overfitting, the stability of the validation accuracy (which fluctuated between 93% and 95% in the last 20 epochs) confirms that the dropout layers were effective in preserving the system overall generalization capability.

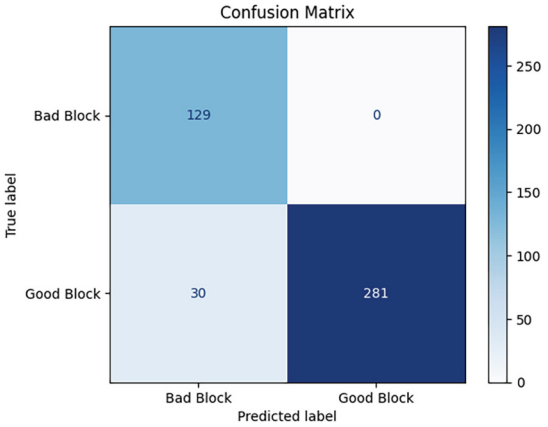

Fig. 2. Confusion matrix of the test set. The absence of false positives validates the filter safety for clinical applications.

Table 3. Final performance metrics after 100 training epochs.

Metric	Training Set	Test/Validation Set
Final Accuracy	99.68%	93.18%
Final Loss	0.0095	0.4271
Peak Accuracy	100.00%	95.00%

5.3 Qualitative Analysis: Semantic Segregation

Figures 3 and 4 show model application to the validation set. Left columns in Figs. 3 and 4 correspond to the unfiltered (raw) data, whereas right columns display the filtered results. A visual comparison between left and right indicates the network semantic segregation capability, as further discussed below.

Effectiveness of the proposed approach becomes visually evident in the pairwise comparison between the raw and processed signals. To correctly interpret these results, it is necessary to observe the morphological distinction between the signal of interest (gait) and behavioral noise.

The clearest example of this distinction occurs in Fig. 3, top row (Data #1). In the unfiltered graph, we observe a long initial period (indices from 0 to 8000) characterized by low amplitude and high frequency variations. Clinically, this represents the moment when the patient is standing still, receiving instructions, or adjusting equipment. This behavior, although captured by the sensor, has no biomechanical value. By analyzing the filtered graph, we can see that the

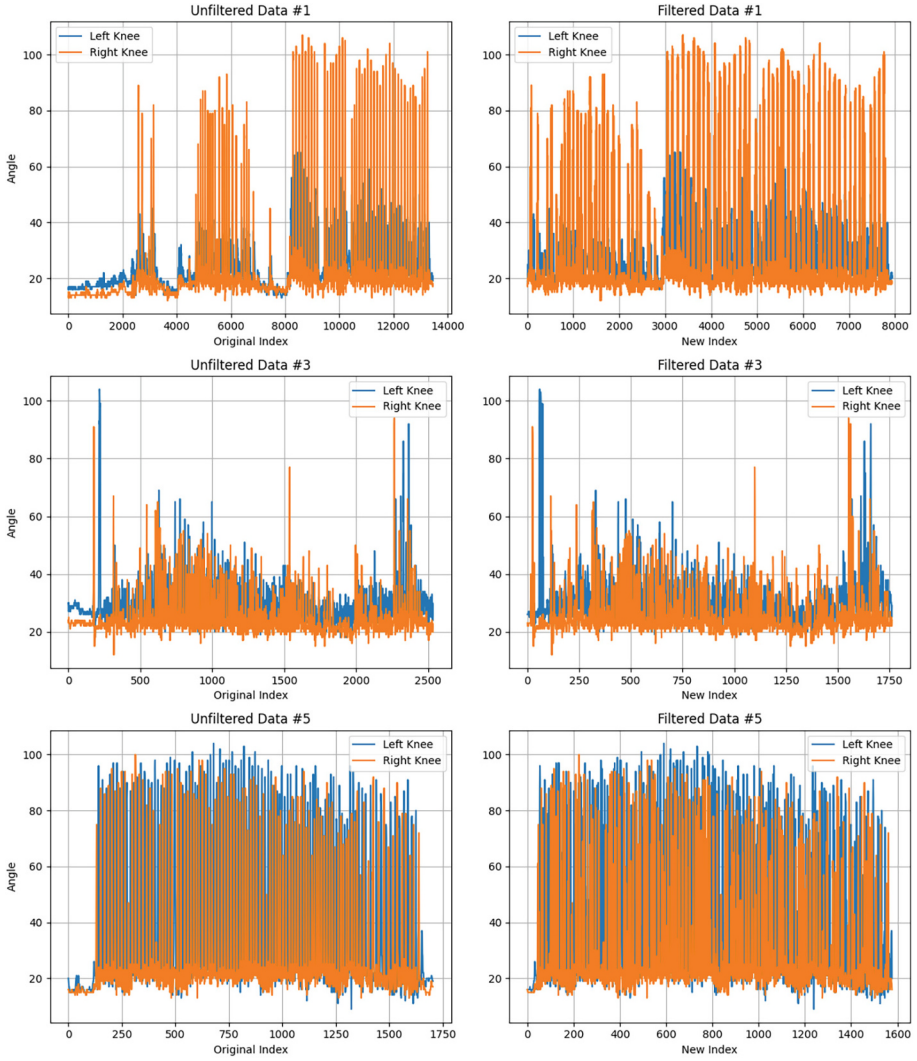

Fig. 3. Comparison between raw and filtered signals, part 1. The neural network effectively suppresses high-amplitude noise and periods of inactivity, preserving only the gait cyclic morphology.

neural network removed this entire initial segment. The resulting graph begins immediately at the first consistent gait cycle. Note that the x-axis (New Index) has been reduced, indicating that the model not only eliminated noise but also discarded unnecessary data, thus optimizing storage.

A critical point in automated filters is the risk of distorting useful signals. However, when observing high-amplitude segments (peaks representing knee flexion above 60°C) in Figs. 3 and 4, we can confirm that the waveform integrity

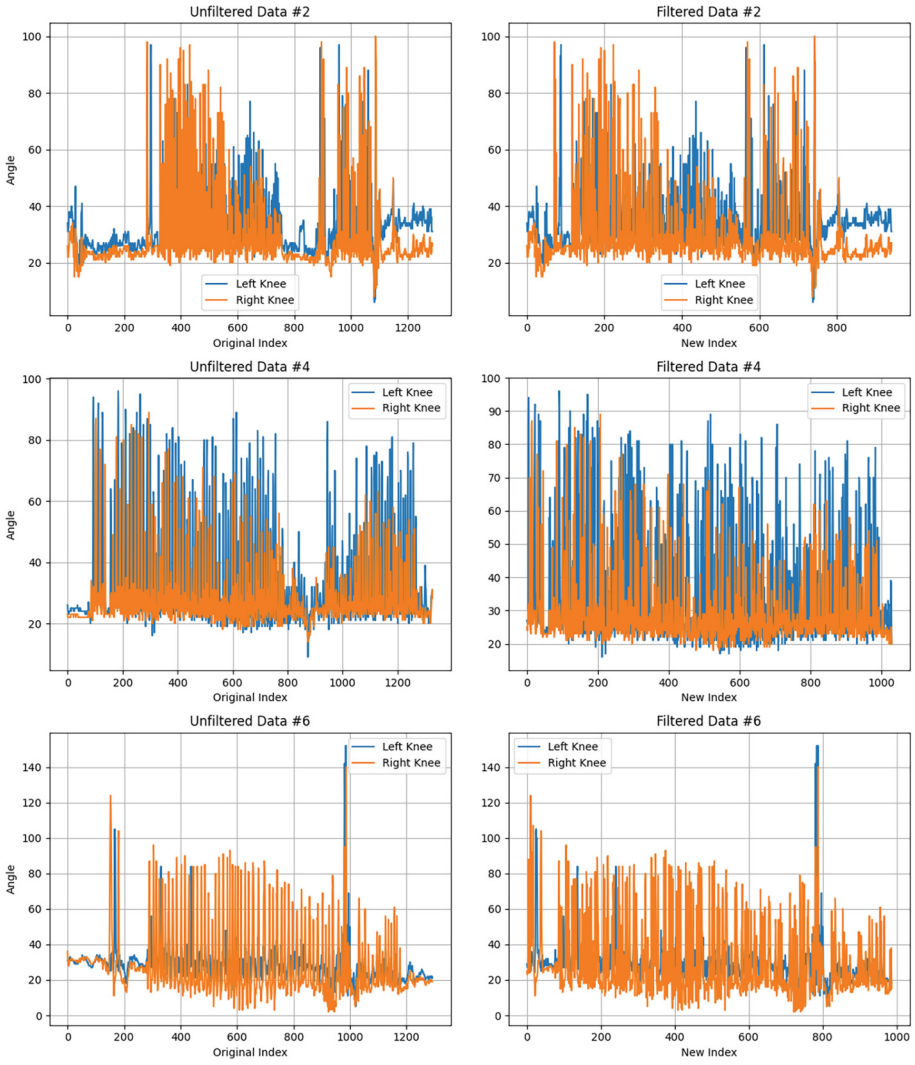

Fig. 4. Comparison between raw and filtered signals, part 2. The neural network effectively suppresses high-amplitude noise and periods of inactivity, preserving only the gait cyclic morphology.

is maintained. Thus, the neural network learned to recognize the typical sinusoidal pattern of human gait. Unlike hard-threshold filters (which could cut off wave peaks or fail at higher valleys), the MLP preserved natural dynamics of the movement, ensuring that subsequent metrics, such as range of motion (ROM), remain reliable.

5.4 Generalization Test

To validate the model robustness for new patients and ensure its applicability in real-world scenarios, we tested the network on unpublished recordings not included in the training set. The results of this qualitative evaluation, presented in Fig. 5, demonstrate the system autonomy in discerning genuine biophysical patterns from spurious artifacts, thus eliminating the need for manual threshold adjustments.

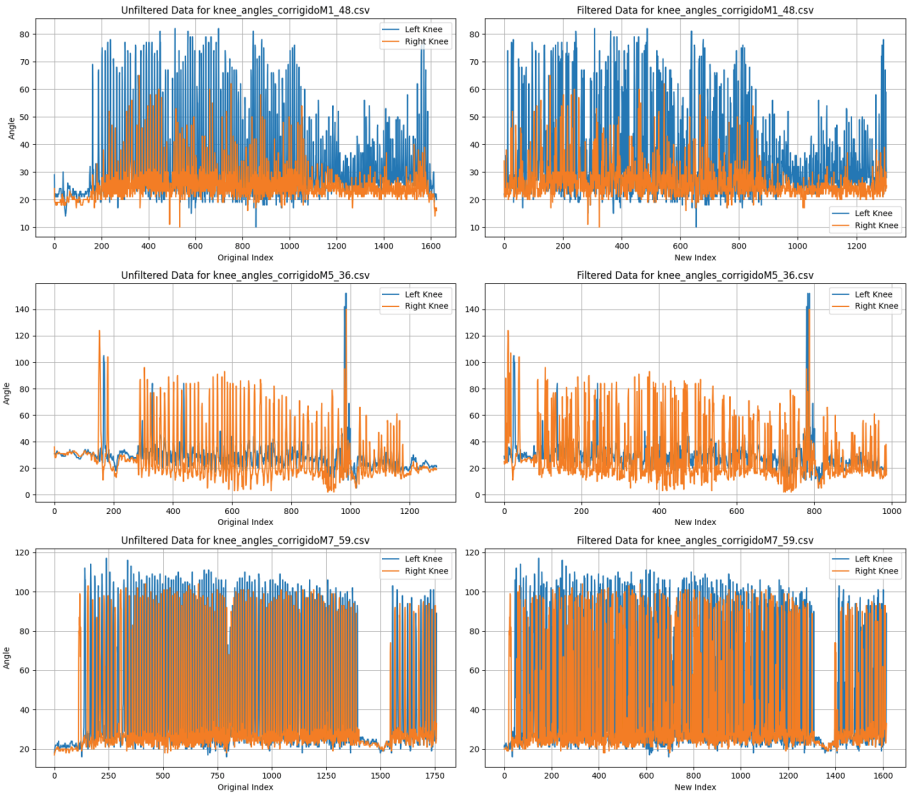

Fig. 5. Generalization test on three new recordings (`M1_48`, `M5_36`, `M7_59`). The model demonstrated autonomy in cleaning background noise and segmenting pauses without manual threshold adjustment.

Graphs from recording session `M1_48` (top part of Fig. 5) show the classifier ability to clean the signal from its initial recording moments, completely suppressing the background noise that precedes motor activity without degrading the integrity of subsequent gait cycles.

This contextual sensitivity becomes even more evident in the central recording (`M5_36`), where the network handled exercise intermittency better than classical approaches. In this scenario, a patient performs the gait, pauses to rest (as

evidenced by a drop in amplitude in the original graph), and resumes the exercise. Classic moving-average filters tend to create an artificial ramp of descent and ascent at the extremes of this pause. Our model, however, identified the absence of a gait pattern and removed it. The result is a clean signal that concatenates only moments of effective effort, allowing the therapist to assess movement consistency without navigating through data gaps.

Finally, the behavior observed in the lower recording (`M7_59`), more specifically the detection of an acquisition gap near index 1300, supports the hypothesis that the neural network internalized a latent representation of the human movement morphology. By systematically rejecting patterns that deviate from this biomechanical structure, the proposed solution demonstrates its robustness as an alternative to manual curation of the ReBase ecosystem.

In summary, the visual changes in the graphs confirm that the model went beyond simple stochastic noise reduction, acting as a semantic editor that retains only the blocks where the kinematic signature of gait is present.

6 Conclusion

We presented a practical approach for processing noisy kinematic signals from rehabilitation datasets, focusing on preserving informative segments that reflect typical cyclic gait patterns. Initially, we performed manual filtering based on visual inspection and empirically adjusted thresholds, thereby identifying consistent and relevant data blocks for biomechanical analysis.

Based on this selected dataset, we trained an MLP-type neural network to learn the characteristic patterns of valid segments. After training, the model demonstrated good performance on the automated filtering task, eliminating the need for manual intervention and replicating the behavior observed during the supervised preprocessing phase with a high accuracy rate.

Replacing manual filtering with a deep learning-based approach has proven to be not only feasible but also advantageous in terms of scalability, standardization, and time efficiency. Based on the results presented, we have evidence suggesting that neural networks will enable consistent application of the process to new recordings, reducing human effort and minimizing the risk of bias or subjective inconsistencies in data selection.

In conclusion, the results obtained reinforce the potential for integrating machine learning techniques into clinical procedures, paving the way for more autonomous and reliable systems for processing biomedical signals. As future work, we intend to integrate this model directly into the ReBase ecosystem recording pipeline, allowing for real-time data cleaning. Furthermore, we plan to expand the training set to include neuromotor pathologies and additional sensor modalities, e.g., IMUs. Ultimately, this approach underpins the development of data-driven tele-rehabilitation tools, where ensuring information quality is a prerequisite for remote clinical decision support.

Acknowledgments. The authors acknowledge partial funding from the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) - Funding Code 001; the National Council for Scientific and Technological Development - Brazil (CNPq) - Funding Codes 442344/2023–6, 444602/2024–0, and 407466/2025–8; the Espírito Santo Research and Innovation Support Foundation (FAPES) – Call No. 02/2026 (Participation in Technical-Scientific Events) – Grant No. 428/2026 and project #2024-9G1WP BPC; and the Minas Gerais Research Funding Foundation (FAPEMIG)

References

1. Ahmed, F., et al.: Kalman filter-based noise reduction framework for posture estimation using depth sensor. In: 2019 IEEE 18th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC), pp. 150–158. IEEE (2019)
2. Camargo, L.F., Miguel, G.R., Feitosa, J.D.C., Rodrigues, L.G.S., Dias, D.R.C., Brega, J.R.F.: Combining computer vision and artificial intelligence to support the rehabilitation of post-stroke patients. In: International Conference on Computational Science and Its Applications, pp. 415–432. Springer (2025)
3. Capecci, M., et al.: The KIMORE dataset: kinematic assessment of movement and clinical scores for remote monitoring of physical rehabilitation. IEEE Trans. Neural Syst. Rehabil. Eng. **27**(7), 1436–1448 (2019)
4. Dias, D.R., Alvarenga, I.C., Guimarães, M.P., Trevelin, L.C., Castellano, G., Brandão, A.F.: eStreet: virtual reality and wearable devices applied to rehabilitation. In: International Conference on Computational Science and Its Applications, pp. 775–789. Springer (2018)
5. Géron, A.: Hands-On Machine Learning with Scikit-Learn and TensorFlow, 2 edn. O'Reilly Media (2020)
6. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
7. Lea, C., et al.: Temporal convolutional networks for action segmentation and detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 156–165 (2017)
8. Liao, Y., Vakanski, A., Xian, M.: A deep learning framework for assessing physical rehabilitation exercises. IEEE Trans. Neural Syst. Rehabil. Eng. **28**(2), 468–477 (2020)
9. Liu, J., et al.: NTU RGB+ D 120: a large-scale benchmark for 3D human activity understanding. IEEE Trans. Patt. Anal. Mach. Intell. **42**(10), 2684–2701 (2019)
10. Maier, M., et al.: Effect of specific over nonspecific VR-based rehabilitation on poststroke motor recovery: a systematic meta-analysis. Neurorehabil. Neural Repair **33**(2), 112–129 (2019)
11. Miron, A., et al.: IntelliRehabDS (IRDS) - a dataset of physical rehabilitation movements. Data **6**(5), 46 (2021)
12. Ribeiro, E., Brandão, A., Guimarães, M., Rocha, L., Brega, J.R., Villaça, R., Dias, D.: A body tracking framework for neuromotor rehabilitation with multimedia application support. In: International Conference on Computational Science and Its Applications, pp. 345–360. Springer (2025)
13. Rodrigues, L.G.S., et al.: A lower limb mixed reality system for the elderly. In: Proceedings of the 24th Symposium on Virtual and Augmented Reality, p. 5. ACM, New York (2022). https://doi.org/10.1145/3604479.3604531

14. Tepper, M., Sapiro, G.: Fast L1 smoothing splines with an application to Kinect depth data. In: 2013 IEEE International Conference on Image Processing, pp. 504–508. IEEE (2013)
15. Trotta, T., et al.: ReBase: a framework for acquiring and managing body tracking data. In: Proceedings of the Symposium on Virtual and Augmented Reality, p. 9. ACM, New York (2024). https://doi.org/10.1145/3691573.3691578
16. Vakanski, A., Jun, H.p., Paul, D., Baker, R.: A data set of human body movements for physical rehabilitation exercises. Data **3**(1), 2 (2018)
17. Winter, D.A.: Biomechanics and Motor Control of Human Movement. John Wiley & Sons (2009)
18. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)