



Article

Machine Learning Regression-Driven Improved Step Length Estimator with Smartphone Accelerometry: A Comparative Performance Study

Rumpa Chakraborty ¹, Saptadipa Mazumder ² , Pradip K. Das ³ and Pampa Sadhukhan ^{1,*}

¹ School of Mobile Computing & Communication, Jadavpur University, Kolkata 700106, India; rumpachakraborty462@gmail.com

² Information Technology, Guru Nanak Institute of Technology, Khardaha 700114, India; saptadipa.mazumder@gnit.ac.in

³ Department of Computer Science & Engineering, Jadavpur University, Kolkata 700032, India; pkdas@ieee.org

* Correspondence: pampas.smcc@jadavpuruniversity.in; Tel.: +91-33-2335-6122

Abstract

Precise step length estimation (SLE) is a key necessity for not only navigation systems design but also gait health monitoring in neurological conditions. Among existing solutions, non-invasive inertial sensor-based approaches operating without dedicated infrastructure are more cost-effective. Many such methods, however, rely on bodily affixed inertial sensors rather than freely held smartphone sensors. Traditional signal processing approaches, on the other hand, offer varying accuracy across diverse gait patterns due to user parameter calibration. This study, thus, proposes a regression-based SLE framework employing eight regression algorithms: linear regression (LR), k-nearest neighbors, support vector machine, decision tree, elastic network, random forest, histogram-based gradient boosting (HGB) regressor, and artificial neural network (ANN). Their extensive and rigorous evaluations across varied window sizes, using a dataset collected in normal and fast walking modes with two device positions (hand-held and trouser-pocket) during three evaluation scenarios, demonstrate the HGB regressor's outstanding performance, achieving the lowest mean absolute error (MAE) below 1 cm across four different contexts under leave-one-out cross-validation-based evaluation and three in the seen test evaluations. Moreover, the findings report the ANN's exceptional generalization capacity over other models and the previous method IRT-SD-SLE in unseen test evaluations, with an MAE not exceeding 6.3 cm. The extensive evaluations of training and testing times reveal the highest computational efficiency for LR, moderate efficiency for the HGB regressor, and the highest training cost for the ANN, indicating a clear trade-off between MAE and computational expense. Additionally, this study includes an insightful discussion on the performance results, including the trade-offs between accuracy and efficiency.



Academic Editor: Omneya A. Attallah

Received: 20 May 2026

Revised: 16 July 2026

Accepted: 21 July 2026

Published: 27 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: step length; regression; accelerometer; gait; MAE; training; test

1. Introduction

Precise step length estimation (SLE) is of paramount importance in the development of various applications, including pedestrian dead reckoning (PDR)-based positioning and navigation systems [1,2], health monitoring for patients with neurological disorders [3,4], gait kinematics measurement [5], physical fitness tracking [6], and fall risk assessment [7], among others. Inertial sensor-based PDR methods have demonstrated significant utility,

not only in designing effective mobility aid tools for individuals with physical challenges, such as the visually impaired [8], but also in providing location-based services (LBSs) in regions lacking other signal sources [9,10]. The limited availability of sufficient subjects or patients required for clinical trials of neurological conditions such as Alzheimer's disease, Cerebral Palsy, Amyotrophic Lateral Sclerosis (ALS), and Mild Cognitive Impairment presents a potential obstacle to drug development for these neurological diseases [3]. Conversely, the measured gait parameters—such as step length and step duration—hold promise for detecting these neurological conditions and monitoring their progression, thereby serving as valuable biomarkers [11]. Low-cost non-invasive inertial sensors, as opposed to instrumented walkways or optical motion capture systems entailing dedicated infrastructure deployment, are more practical for measuring pedestrians' step length using gait pattern analysis across various motion modes in both home and laboratory settings [12,13].

Among the existing inertial sensor-based solutions for SLE, those operating on signals from inertial sensors affixed to body segments, like the foot, waist, or thigh, tend to produce more reliable results than those relying solely on data from the built-in sensors of smartphones held freely, as demonstrated by previous studies [14–17]. The device orientation and posture dynamics during user movements pose significant challenges in designing accurate and reliable SLE methods based solely on smartphone inertial data. Consequently, this study concentrates on overcoming these limitations by processing only the signals collected from freely held or placed smartphones.

To mitigate the adverse effects of the smartphone's orientation and gravitational bias on step detection (SD) and SLE accuracy, our earlier proposed method, IRT-SD-SLE [15], initially transforms raw acceleration measurements from the device's body coordinate system to the Earth coordinate system and employs a sliding window-based low-pass filter before peak–valley detection. Furthermore, our previous smartphone accelerometer-based SLE method outperformed existing techniques, with an accuracy improvement of around 15% or more for both step detection and step length estimation by utilizing a peak–valley detection scheme across different walking modes and device postures, as evidenced by experimental results in [15]. However, it still requires calibration of certain user-specific parameters to achieve optimal accuracy. Additionally, earlier findings indicated variability in the precision of IRT-SD-SLE across various walking modes and device postures. To eliminate the need for user-specific parameter calibration and curb the influence of device orientation and walking mode, this study proposes a regressor algorithm-driven SLE framework using smartphone accelerometer data. The adoption of machine learning not only removes the dependence on user-specific parameters but also enhances stability and reliability across different motion modes and device postures.

In recent years, numerous research initiatives have investigated SLE methods employing supervised learning [18], self-supervised learning [19], deep learning models like CNNs, Gated Recurrent Units (GRUs), attention mechanisms, and Res2Net modules, and so on [20–32], alongside dimensionality reduction techniques such as principal component analysis (PCA) [33], to overcome the limitations inherent in traditional signal processing, including the necessity for user parameter calibration. Many of these existing approaches, however, depend on inertial measurement unit (IMU) sensors fixed to body segments rather than the built-in inertial sensors of freely held smartphones [19,26–32].

Very few studies address the challenge of processing unconstrained smartphone sensor data affected by continuous orientation changes during user movements, which significantly diminishes SLE accuracy. Among these, Vandermeeren et al. [18] analyzed the influence of feature selection on ML-based step length estimation, demonstrating performance improvements when relevant features were selected, although their approach was

highly recursive. Another smartphone-based method proposed by Vežočník et al. [33] applied principal component analysis to derive step length from smartphone accelerometry, evaluated in different positions such as pocket, hand, upper arm, and pelvis, achieving an average absolute error of 6.44 cm. However, their experiments used sensor data from smartphones fixed to body segments rather than from unconstrained devices. Abiad et al. [34] proposed a data-driven step recognition approach using a Histogram Gradient Boosting decision tree model applied to accelerometer and gyroscope signals.

In a recent study, Shi et al. [35] introduced a method utilizing Ensemble Empirical Mode Decomposition with adaptive noise (CEEMDAN) and SVM classifiers for step detection across diverse device postures and gait patterns. In our recently proposed smartphone sensor-based SLE approach, in addition to converting sensor readings into the Earth coordinate system using a quaternion, automated adjustment to direction and surface slope is incorporated to make the proposed SLE approach slope-adaptive and perform well on curved surfaces and staircases [36]. In another recent study, Xu et al. proposed a deep learning approach using a smartphone's inertial sensors for gait event detection and evaluated its clinical significance [37]. However, such models necessitate sufficient training data, proper validation, and testing across different users, walking speeds, and smartphone positions. Liang et al., on the other hand, reported the usefulness of smartphone sensor-based gait assessment for estimating temporal gait parameters with accuracy dependent on sensor placement, number of walking steps, and reliability of gait-event detection [38].

Additionally, SLE approaches employing certain deep learning models often demand substantial computational resources and extensive training datasets, thereby rendering them impractical for real-time applications on resource-limited devices. Conversely, the effective application of feature selection methods with machine learning (ML) models, as demonstrated in [18], and PCA, as presented in [33], facilitates feature set dimensionality reduction, resulting in comparatively lower computational demands and greater suitability for deployment on portable devices. However, the SLE model proposed in [33] was evaluated and validated using inertial sensor data collected from a constrained smartphone affixed rigidly to the upper arm, hand, hip, and thigh, rather than a freely held one, which is more usual. Similarly, in [18], the researchers examined accelerometer and gyroscope data collected from a smartphone during texting mode, without considering other device positions or different motion modes (e.g., normal, fast, etc.).

Consequently, an extensive evaluation and performance analysis of the SLE method employing various supervised regression algorithms, based on sensor data collected from unconstrained smartphones across diverse motion modes and device positions, is absent in the current literature. Therefore, to address this critical research gap in smartphone sensor-based SLE models, the present study aims to develop a computationally efficient SLE model exploiting machine learning regression algorithms such as linear regression (LR), k-nearest neighbor (k-NN), support vector machine (SVM), decision tree (DTree), elastic net (Els net), random forest (RF), histogram-based gradient boosting (HGB) regressor, and artificial neural network (ANN). To demonstrate the improved performance of the proposed regression model-based SLE method compared to the previous ones, an extensive performance evaluation of the regressor models in terms of mean absolute error (MAE) achieved for SLE, indicating their prediction accuracy along with training and test times for measuring computational efficiency, is conducted using accelerometer data collected from unconstrained smartphones across varied motion modes and device positions. Additionally, this study includes an insightful discussion on the performance results, including the trade-offs between accuracy and efficiency. The contributions of the present study are delineated below in a listed form.

- **A simplistic regression model-based improved SLE method:** This study presents an improved SLE model devoid of any user-related parameters and insensitive to the device's orientation by utilizing several simplistic regression algorithms, ranging from the simplest k-NN to ensemble methods like RF and HGB regressors, as well as a multilayer perceptron (MLP)-based ANN.
- **An extensive and rigorous comparative performance study:** The study also includes an extensive and rigorous comparative performance analysis of the considered regression algorithms in terms of MAE, along with the training and inference times, across varied window sizes in diverse contexts of walking mode and device postures under three evaluation scenarios to demonstrate their generalization capacity across diverse gait patterns, walking modes and device postures in various conditions as well as to explore the trade-off between the prediction accuracy and computational efficiency of these models, which is still missing in the existing literature.
- **Broad applicability:** Sole reliance on the accelerometer sensor makes the proposed method deployable on almost every smartphone.
- **Stable and reliable accuracy:** Insensitivity of the proposed method to the variation in motion modes and device posture enables it to acquire reliable but stable accuracy across the varying motion modes and device postures.

The structure of the paper is as follows. Section 2 presents the detailed design of the proposed SLE framework. In Section 3, various regression models exploited by our proposed SLE model are evaluated extensively across diverse scenarios, followed by a thorough analysis with an insightful discussion in Section 4. Section 5 draws our conclusions and presents the scope for further research.

2. Study Design

Our proposed regression model-based SLE model is presented in this section. The workflow diagram of our proposed method is shown in Figure 1. The proposed SLE model comprises five subprocesses: data acquisition, preprocessing, feature extraction, model training with parameter tuning, and finally, step length prediction with model evaluation. Each of these subprocesses is described in a separate subsection.

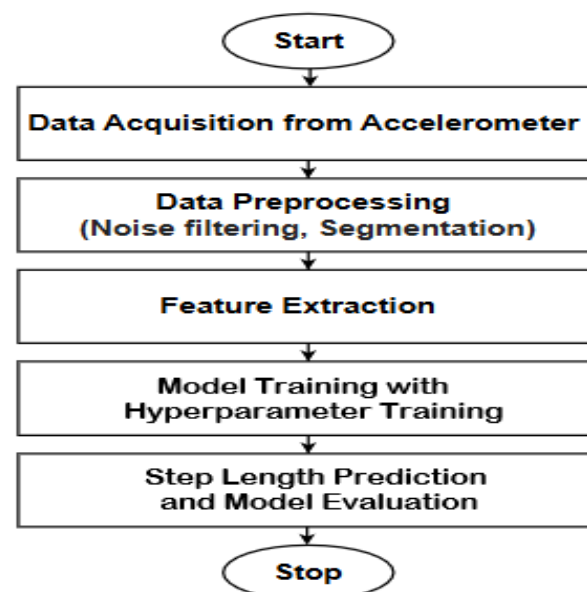


Figure 1. Flowchart of the proposed step length estimator.

2.1. Data Acquisition

The proposed method needs to acquire the signal measurements from the smartphone's built-in accelerometer. For data collection, we used an Android smartphone, the Samsung Galaxy M13 model, and conducted a walking experiment in which ten participants walked along a 30-m-long linear path at two different speeds, normal and fast, with the smartphone either in hand or in their trouser pockets, as reported in our earlier work [15]. The design details of the Android app used to collect data have already been provided there. Unlike the previous case, only the raw triaxial acceleration components recorded along the X, Y, and Z axes were used without transforming them from the sensor coordinate system to the Earth coordinate system, making them independent of device orientation. By combining two distinct motion modes (normal and fast) with two distinct device positions (handheld and pocket), four different walking scenarios result: normal-handheld (NH), normal-pocket (NP), fast-handheld (FH), and fast-pocket (FP). Alongside the sensor readings across its three axes, the number of steps and distance walked by each participant were also recorded in a separate CSV file for each of the four walking trials or scenarios (NH, NP, FH, FP). The recorded data for all participants in a specific walking trial constitute a separate dataset, yielding four different datasets named according to the walking trial in which they were collected. The training, testing, and validation of the considered regression algorithms are conducted on these four datasets.

Ten participants, including both males and females, aged 23 to 47 years, with heights ranging from 160 to 180 cm and weights from 60 to 75 kg, depicted sufficient variability in their gait patterns. The anthropometric and demographic data for each participant, including height, body weight, age, and gender, along with the number of steps walked and distance covered in four walking scenarios, are provided in Table 1 to showcase the variability in the participants' body characteristics and walking patterns. The sampling rate for accelerometer readings was set to 50 Hz, and each participant walked for 32 to 48 s to cover the 30-m distance, depending on their walking speed, yielding a total of 1600–2400 samples in every walking trial.

Table 1. Anthropometric characteristics and walking details of the participants.

ID	Age	Gender	Height (cm)	Weight (kg)	Steps Walked				Distance Walked (m)
					NH	NP	FH	FP	
1	45	Male	178.00	75	43	43	36	35	30.0
2	43	Female	150.00	53	49	48	43	42	30.0
3	44	Male	169.00	87	47	46	39	38	30.0
4	43	Male	170.68	72	42	41	33	34	30.0
5	24	Male	162.56	58	51	50	42	40	30.0
6	23	Male	167.64	80	45	46	42	41	30.0
7	25	Male	156.00	56	47	48	40	41	30.0
8	25	Female	157.00	59	47	47	41	41	30.0
9	44	Female	160.00	64	48	47	41	41	30.0
10	47	Male	176.78	81	46	45	40	38	30.0

2.2. Data Preprocessing

In this step, noise filtering is applied first to remove high-frequency components from raw sensor readings, making the signal smoother. Then the filtered acceleration signal is segmented into several equal-sized parts, each one corresponding to a single step or stride based on the predefined window size.

2.2.1. Noise Filtering

A Butterworth low-pass filter [39] allows low-frequency signals to pass through while attenuating higher-frequency noise above a specified cutoff frequency. In this experiment, the cutoff frequency is set to 10 Hz to filter out any frequency components above it. In our experiment, the `scipy.signal.butter` [40] function is used to reduce noise in accelerometer sensor data, yielding a smoothed signal more suitable for estimating step length or motion analysis. Figure 2 illustrates the raw X, Y, and Z components, as well as the raw and filtered acceleration magnitude.

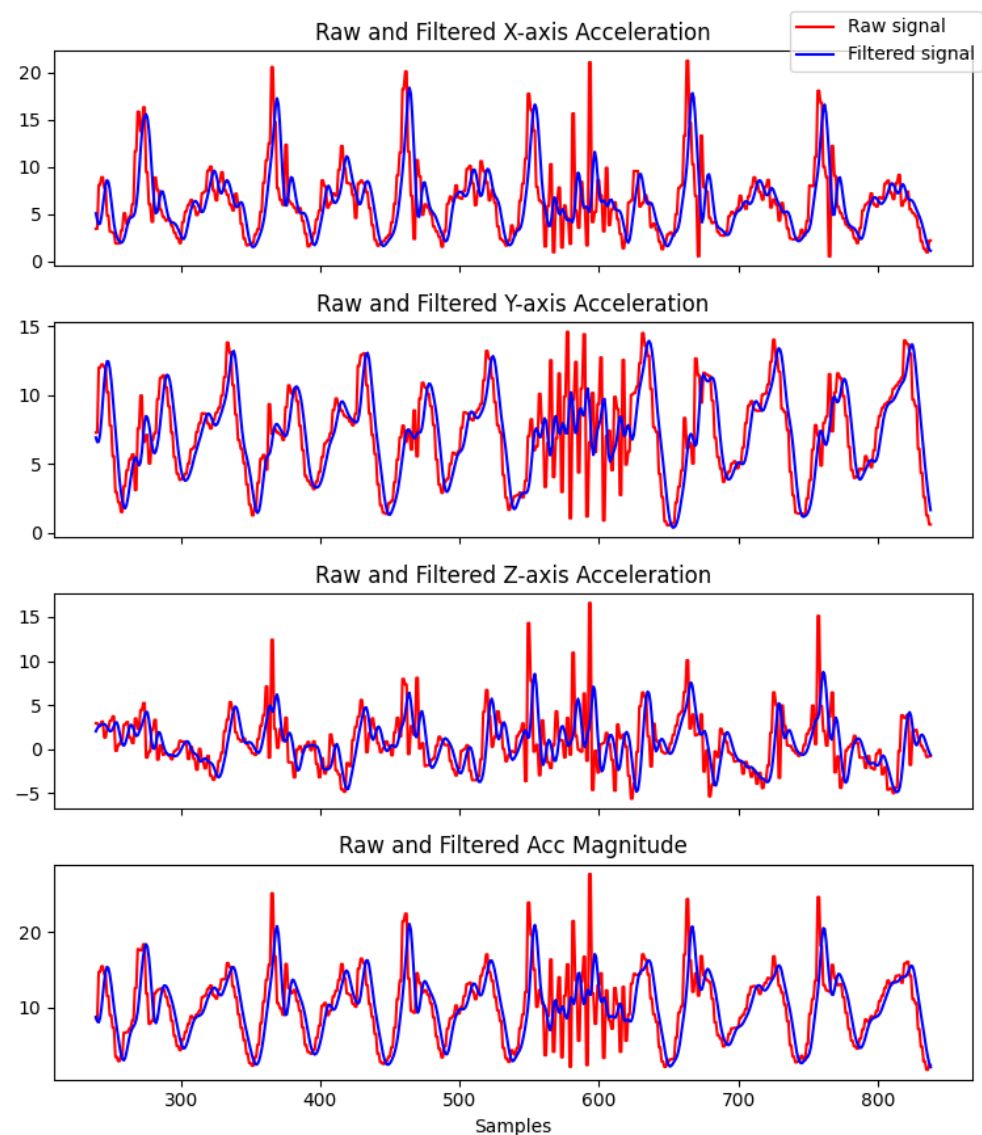


Figure 2. Raw X, Y, Z axes components and magnitude of the accelerometer with filtered acceleration magnitude.

2.2.2. Segmentation

The filtered sensor signal is divided into segments corresponding to each step or stride, based on either a predefined window size or a pair of peak–trough sequences in the sensor signal. Our proposed method partitions the filtered acceleration magnitude into several distinct segments to extract important information related to gait patterns, cadence, step frequency, etc., which are useful features for predicting the step length. Figure 3 shows the segmented raw and filtered acceleration magnitude with peak–trough sequences detected.

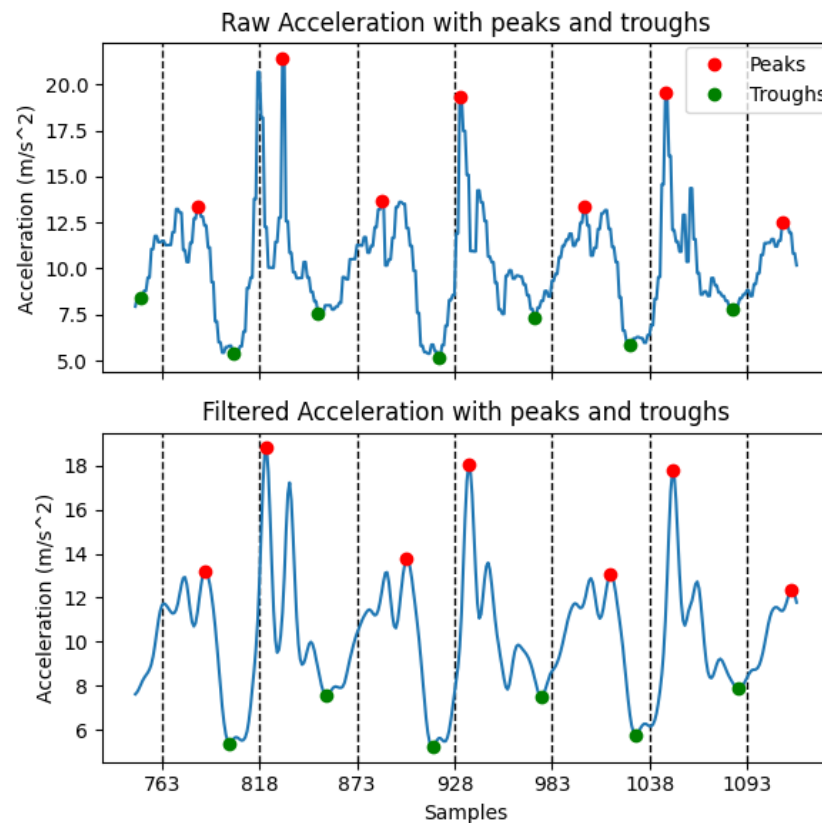


Figure 3. Segmented and filtered acceleration magnitude with detected peak–valley sequences.

2.3. Feature Extraction

A total of 49 features belonging to three different categories—time-domain, frequency-domain, and peak–trough—are extracted from the segmented and filtered sensor signals to train the regression models. These carefully crafted 49 features enable the machine learning models in our proposed SLE framework to identify the hidden characteristics of each dataset from four different walking trials used to train them and perform better than the previous method in terms of estimation error by mitigating the influences of the device orientation without involving such a transformation of the coordinate system. The distinct properties and usefulness of these three feature types are outlined below.

- (i) **Time-domain Features:** A brief description of various time-domain features considered in this study, including their mathematical formulae, is provided in tabular form in Figure 4. Time-domain features such as mean, standard deviation, minimum, maximum, median, mean absolute deviation (MAD), and interquartile range (IQR) quantify basic signal behaviors for each axis. Higher-order features, such as kurtosis and skewness, describe the distribution’s shape. Signal energy captures intensity, and the signal magnitude area (SMA) summarizes the total motion magnitude, while axis correlations provide structural insights.
- (ii) **Frequency-domain Features:** Applying the Fourier Transform to sensor data enables the extraction of frequency domain features, such as dominant frequencies, which are crucial for understanding the periodic components of the data. By converting the time-domain signal into its frequency components, the Fourier Transform reveals how different frequencies contribute to the overall signal. Dominant frequencies, identified as the peaks in the frequency spectrum, indicate the most significant oscillatory patterns within the data, such as the regular rhythm of footsteps in gait analysis. These frequency-domain features enable distinguishing different rhythmic patterns

from eliminating noise, providing insights into the cyclic nature of movements, and thus enhancing the accuracy of models used to estimate step length. The nine most important such features considered by our proposed SLE model are shown in tabular form in Figure 5.

- (iii) **Peak–Trough-related Features:** Peak values represent the maximum accelerations or forces experienced during a step, highlighting key moments of impact or exertion. Conversely, trough or valley values capture the minimum accelerations or forces, identifying points of least activity or relaxation. Together, these features provide a comprehensive view of step dynamics, facilitating accurate analysis and modelling for applications such as step length estimation and gait assessment. Six important peak–trough-related features considered in this study, including the window size, i.e., the segment length, are briefly described in tabular form in Figure 5, following the frequency-domain features.

#	Feature	Description	Formula
1	μ_{ax}	Mean of acceleration along the x-axis	$\frac{1}{N} \sum_{i=1}^N a_x(i)$
2	σ_{ax}	Standard deviation of acceleration along the x-axis	$\sqrt{\frac{1}{N} \sum_{i=1}^N (a_x(i) - \bar{a}_x)^2}$
3	$\min_{a_x}$	Minimum value of acceleration along the x-axis	$\min(a_x(i))$
4	$\max_{a_x}$	Maximum value of acceleration along the x-axis	$\max(a_x(i))$
5	MAD_{ax}	Mean absolute deviation of acceleration along the x-axis	$\text{mean}(a_x(i) - \mu_{ax})$
6	med_{ax}	Median value of acceleration along the x-axis	$\text{med}(a_x(i))$
7	IQR_{ax}	Interquartile range of acceleration along the x-axis	$a_x^{Q3} - a_x^{Q1}$
8	E_{a_x}	Signal energy of acceleration along the x-axis	$\sum_{i=1}^N (a_x(i))^2$
9	$kurtosis_{ax}$	Kurtosis (tailedness) of acceleration along the x-axis	$\frac{1}{N\sigma_{ax}^4} \sqrt{\frac{1}{N} \sum_{i=1}^N (a_x(i) - \bar{a}_x)^4}$
10	$skewness_{ax}$	Skewness (asymmetry) of acceleration along the x-axis	$\frac{1}{N\sigma_{ax}^3} \sqrt{\frac{1}{N} \sum_{i=1}^N (a_x(i) - \bar{a}_x)^3}$
11	μ_{ay}	Mean of acceleration along the y-axis	$\frac{1}{N} \sum_{i=1}^N a_y(i)$
12	σ_{ay}	Standard deviation of acceleration along the y-axis	$\sqrt{\frac{1}{N} \sum_{i=1}^N (a_y(i) - \bar{a}_y)^2}$
13	$\min_{a_y}$	Minimum value of acceleration along the y-axis	$\min(a_y(i))$
14	$\max_{a_y}$	Maximum value of acceleration along the y-axis	$\max(a_y(i))$
15	MAD_{ay}	Mean absolute deviation of acceleration along the y-axis	$\text{mean}(a_y(i) - \mu_{ay})$
16	med_{ay}	Median value of acceleration along the y-axis	$\text{med}(a_y(i))$
17	IQR_{ay}	Interquartile range of acceleration along the y-axis	$a_y^{Q3} - a_y^{Q1}$
18	E_{a_y}	Signal energy of acceleration along the y-axis	$\sum_{i=1}^N (a_y(i))^2$
19	$kurtosis_{ay}$	Kurtosis (tailedness) of acceleration along the y-axis	$\frac{1}{N\sigma_{ay}^4} \sqrt{\frac{1}{N} \sum_{i=1}^N (a_y(i) - \bar{a}_y)^4}$
20	$skewness_{ay}$	Skewness (asymmetry) of acceleration along the y-axis	$\frac{1}{N\sigma_{ay}^3} \sqrt{\frac{1}{N} \sum_{i=1}^N (a_y(i) - \bar{a}_y)^3}$
21	μ_{az}	Mean of acceleration along the z-axis	$\frac{1}{N} \sum_{i=1}^N a_z(i)$
22	σ_{az}	Standard deviation of acceleration along the z-axis	$\sqrt{\frac{1}{N} \sum_{i=1}^N (a_z(i) - \bar{a}_z)^2}$
23	$\min_{a_z}$	Minimum value of acceleration along the z-axis	$\min(a_z(i))$
24	$\max_{a_z}$	Maximum value of acceleration along the z-axis	$\max(a_z(i))$
25	MAD_{az}	Mean absolute deviation of acceleration along the z-axis	$\text{mean}(a_z(i) - \mu_{az})$
26	med_{az}	Median value of acceleration along the z-axis	$\text{med}(a_z(i))$
27	IQR_{az}	Interquartile range of acceleration along the z-axis	$a_z^{Q3} - a_z^{Q1}$
28	E_{a_z}	Signal energy of acceleration along the z-axis	$\sum_{i=1}^N (a_z(i))^2$
29	$kurtosis_{az}$	Kurtosis (tailedness) of acceleration along the z-axis	$\frac{1}{N\sigma_{az}^4} \sqrt{\frac{1}{N} \sum_{i=1}^N (a_z(i) - \bar{a}_z)^4}$
30	$skewness_{az}$	Skewness (asymmetry) of acceleration along the z-axis	$\frac{1}{N\sigma_{az}^3} \sqrt{\frac{1}{N} \sum_{i=1}^N (a_z(i) - \bar{a}_z)^3}$
31	Sma	Sum of magnitudes across x, y, and z axes	$\frac{1}{N} \sum (a_x(i) + a_y(i) + a_z(i))$
32	Corr _{xy}	Correlation between x-axis and y-axis accelerations	$\frac{\sum (a_x(i) - \bar{a}_x)(a_y(i) - \bar{a}_y)}{(N-1) \sigma_{ax} \sigma_{ay}}$
33	Corr _{xz}	Correlation between x-axis and z-axis accelerations	$\frac{\sum (a_x(i) - \bar{a}_x)(a_z(i) - \bar{a}_z)}{(N-1) \sigma_{ax} \sigma_{az}}$
34	Corr _{yz}	Correlation between y-axis and z-axis accelerations	$\frac{\sum (a_y(i) - \bar{a}_y)(a_z(i) - \bar{a}_z)}{(N-1) \sigma_{ay} \sigma_{az}}$

Figure 4. List of time-domain features considered by our proposed SLE.

	#	Feature	Description	Formula
Frequency domain features	35	$f_{max_{ax}}$	Maximum frequency component from FFT of x-axis acceleration	$\max(FFT(a_x))$
	36	$idx_{f_{max_{ax}}}$	Frequency index of $f_{max_{ax}}$	
	37	$f_{wa_{ax}}$	Weighted average frequency of FFT components on the x-axis	$\frac{\sum f(i) \cdot f(a_x(i)) }{\sum f(a_x(i)) }$
	38	$f_{max_{ay}}$	Maximum frequency component from FFT of y-axis acceleration	$\max(FFT(a_y))$
	39	$idx_{f_{max_{ay}}}$	Frequency index of $f_{max_{ay}}$	
	40	$f_{wa_{ay}}$	Weighted average frequency of FFT components on the y-axis	$\frac{\sum f(i) \cdot f(a_y(i)) }{\sum f(a_y(i)) }$
	41	$f_{max_{az}}$	Maximum frequency component from FFT of z-axis acceleration	$\max(FFT(a_z))$
	42	$idx_{f_{max_{az}}}$	Frequency index of $f_{max_{az}}$	
	43	$f_{wa_{az}}$	Weighted average frequency of FFT components on the z-axis	$\frac{\sum f(i) \cdot f(a_z(i)) }{\sum f(a_z(i)) }$
Peak-Trough related	44	num_peaks_troughs	Number of detected peaks and troughs in the signal	Count of local maxima and minima
	45	Mean peak height (μ_{ph})	Average value of the highest points (peaks) in a signal	$\mu_{ph} = \frac{\sum_{i=1}^q ph_i}{q}$
	46	Mean trough depth (μ_{td})	Average value of the lowest points (troughs) in a signal	$\mu_{td} = \frac{\sum_{i=1}^q td_i}{q}$
	47	Mean peak duration (μ_{τ}^{peak})	The mean difference in the duration of two consecutive peaks of acceleration	$\frac{1}{q} \sum_{i=1}^q \tau_{i+1}^{peak} - \tau_i^{peak}$
	48	Mean trough duration (μ_{τ}^{trough})	The mean difference in the duration of two consecutive troughs of acceleration	$\frac{1}{q} \sum_{i=1}^q \tau_{i+1}^{trough} - \tau_i^{trough}$
	49	Window_size	Size of the time window used for feature extraction	Duration or number of samples

Figure 5. List of frequency-domain and peak–trough-related features considered by our proposed SLE.

2.4. Model Training with Hyperparameter Tuning

Our proposed SLE model employs eight different regression algorithms, which include LR, k-NN, SVM, decision tree, elastic net, RF, HGB regressor, and ANN, to predict the step length in diverse motion modes and device postures. A brief outline of their regression functionality is presented first, followed by a discussion on their training process with hyperparameter tuning.

2.4.1. Machine Learning Regression Algorithms

Among the regression algorithms considered, linear regression (LR) leverages the relationship between input and output variables to predict a continuous numerical value for the latter. Such a model usually assumes a straight-line connection between the input (independent variable) and the output (dependent variable) [41]. Unlike LR, k-NN does not work based on the underlying pattern of the data. Rather, it predicts a value by finding the ‘k’ closest data points (neighbors) to a given input and averaging their associated outputs in regression [42]. The third one, i.e., support vector machine regression, when combined with a radial basis function (RBF) kernel, becomes more flexible and can model nonlinear relationships between input and output variables [43]. Unlike linear regression, which fits a straight line, RBF-SVM regression fits a smooth curve that can capture complex patterns in the data [44]. Elastic net (Els net) regression, a combination of lasso regression (which performs feature selection) and ridge regression (which shrinks coefficients to prevent overfitting), applies both L1 and L2 regularization to the linear regression model [45].

On the other hand, Decision Tree (DTree) regression predicts a continuous numerical value by splitting the data into several smaller groups based on certain rules or criteria [46]. In the decision tree generated by this regression model, each non-leaf node represents an input feature, and the actual output values constitute the leaf nodes. The random forest (RF) regressor, a type of ensemble machine learning model, combines the predictions of multiple decision trees built on different aspects of the data [47] to generate its own. Taking the average of constituent trees’ predictions as the final prediction makes the model

more accurate and less likely to overfit noise. Another ensemble technique considered by our proposed SLE model is the histogram-based gradient boosting (HGB) regressor [48]. The use of a histogram, i.e., an integer-based data structure, by such an ensemble model to create the trees enables it to speed up processing larger input samples compared to traditional gradient boosting tree regressors, which use sorted continuous values for the same purposes [49]. Finally, artificial neural network (ANN) models, including a specific variant, the multilayer perceptron (MLP), resemble the structure of the human brain in design. They can learn complex relationships between input features and output targets, even when non-linear relationships hold between them [50]. Such ML models consist of three layers—input, hidden, and output. The input features enter the network through the input layers. Then, computations are performed in the hidden layers to detect underlying patterns in the data, and the final prediction is obtained via a single neuron in the output layer [51].

2.4.2. Training with Hyperparameter Tuning

Each adopted model is trained using the training dataset constituted by the input feature vector of 49 features reported in Section 2.3.

Assume L represents the set of m ($m > 0$) step segments, where each segment (denoted as $l_i \in L$, where $0 < i \leq m$) is a sequence of multiple data points, as illustrated in Figure 3 in the previous subsection. For each step segment, the feature vector is constituted by concatenating all 49 features derived from the filtered acceleration signal components. Mathematically, the feature vector (x_i) for the i th step segment l_i is denoted by the following equation.

$$x_i = [f_1^i f_2^i \cdots f_{48}^i f_{49}^i] \quad (1)$$

where f_i^j denotes the j th feature in the list of 49 features delineated in Figures 1 and 2, and is derived from the i th step segment l_i . By collecting the feature vectors for all step segments into a matrix, where each row corresponds to a feature vector, the input feature matrix X is constructed. Mathematically, it is represented as follows.

$$X = \begin{bmatrix} f_1^1 & f_2^1 & \cdots & f_{48}^1 & f_{49}^1 \\ f_1^2 & f_2^2 & \cdots & f_{48}^2 & f_{49}^2 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ f_1^m & f_2^m & \cdots & f_{48}^m & f_{49}^m \end{bmatrix} \quad (2)$$

Presuming each step segment denotes a single step, the vector of ground truth step lengths is represented as $SL = [sl_1 sl_2 \cdots sl_k]$, which is taken as the true output for the above input feature matrix. In this case, the average step length for each participant, calculated as the total walking distance divided by the number of steps, is considered the ground truth for the actual step length.

To tune the hyperparameters of the ML models considered by our proposed SLE framework, we employ the 'GridSearchCV' function from the scikit-learn package [52]. Such a search method tests all possible parameter combinations taken from the predetermined hyperparameter space using a predefined cross-validation scheme [53]. The tuned hyperparameters for the regression models considered in this study are 'n_neighbors' and distance metric for k-NN, 'C' and 'gamma' for RBF-SVM, 'max_depth' and 'min_samples_split' for decision tree, 'n_estimators' and 'max_depth' for RF regressor, 'max_iter' and 'max_depth' for HGB regressor, and 'alpha' for ANN. We use a 5-fold cross-validation scheme to tune the above hyperparameters.

2.5. Step Length Prediction and Model Evaluation

The trained models with the tuned hyperparameters specified in the previous subsection are used to predict the step length and evaluate their performances also in terms of mean absolute error (MAE). The seen and unseen data collected across different walking modes and phone positions are used to evaluate all considered models, and their results under various scenarios are presented in the next section. To compute the MAE, the following mathematical equation is used.

$$MAE = \frac{1}{m} \sum_{i=1}^m |sl_i - \hat{sl}_i|, \quad (3)$$

where sl_i and $\hat{sl}_i$ are actual step length observed and the same predicted by the regression model corresponding to the i th step segment, with m denoting the number of step segments.

3. Results

To validate the effectiveness of our proposed SLE framework, specifically the models' performance across diverse gait patterns, we adopted a leave-one-out cross-validation (LOOCV) strategy for each participant, in which the participant constituting the test set is fully excluded from the training set. To further demonstrate the generalizability of our proposed step length estimator, we evaluate the considered regression models on the MAE for estimated step length on the unseen test data, followed by the seen test data. The calculation of MAE from the ground truth and the predicted step length is already shown in Section 2.5. Apart from the MAE-based evaluation, all considered models are also evaluated and compared in terms of training and test times to assess their computational efficiency.

The LOOCV for each of the four scenarios (NH, NP, FH, and FP) trains all considered models on all participants' datasets, except the one used as the test case, as illustrated in Figure 6, collected in that specific scenario. On the other hand, evaluating the considered regression models on all participants' datasets collected in a specific context among the four (NH, NP, FH, and FP), provided these were trained on the datasets collected in the remaining three contexts, results in an unseen test data-based evaluation, while training on the datasets collected in all four contexts for all participants makes it a seen test data-based evaluation, as depicted in Figure 7. Additionally, to investigate the effect of window size, which acts as an input feature for our proposed SLE model, the regression models are evaluated on window sizes ranging from 40 to 100. Such an investigation helps in finding the most appropriate window size for each context. The evaluation results of the regression models under three different scenarios, i.e., evaluations using the LOOCV strategy as well as the unseen and seen test data, are presented in Sections 3.1, 3.2 and 3.3, respectively.

3.1. Leave-One-Out Cross-Validation (LOOCV) Strategy-Based Evaluations

Evaluations of all considered models, using LOOCV for each participant across all four scenarios (NH, NP, FH, and FP), were conducted to assess their performance under diverse gait patterns. As stated before, this evaluation strategy for each of the four walking contexts trains the models on all participants' datasets, except for the participant used as the test case, collected in that specific scenario. Figure 6 illustrates the LOOCV evaluation strategy for our dataset, which includes 10 participants, thus showing 10 different test cases. Since the datafile associated with a specific participant selected as a test case is not included in the training dataset, which is constituted by concatenating the datafiles for the remaining participants, and recorded sensor readings with walking data for each participant under each of the four walking trials (NH, NP, FH, FP) are stored in a separate CSV file, as reported in Section 2.1, it fully ensures no potential data overlap between the training and test set occurs under each of the four walking trials in the LOOCV evaluation strategy. The

achieved MAE for all considered regression models across window sizes from 40 to 100, based on separate evaluations using the LOOCV strategy in the four scenarios—NH, NP, FH, and FP—is compared in Figures 8, 9, 10 and 11, respectively.

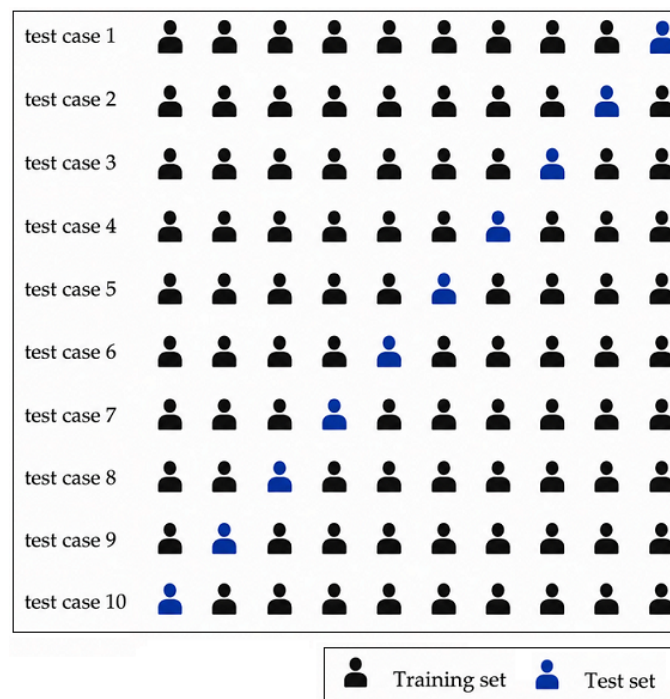


Figure 6. Leave-one-out cross-validation-based evaluation on a dataset with 10 participants.

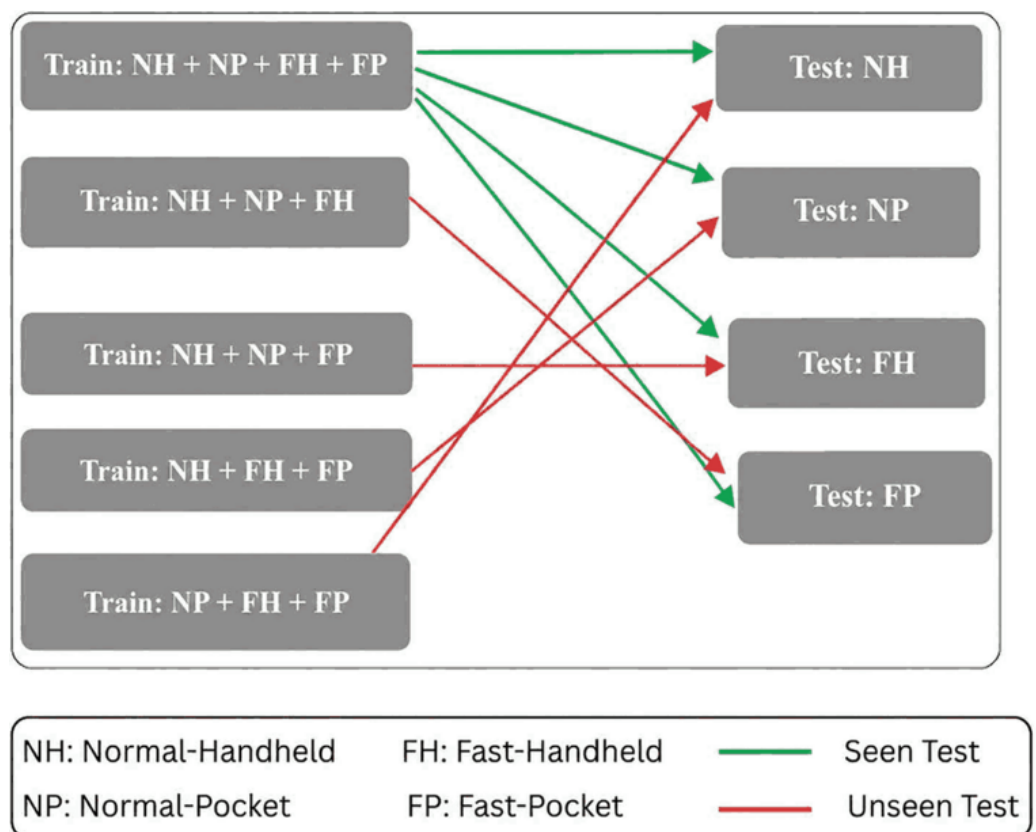


Figure 7. Evaluation scenarios using unseen and seen test data.

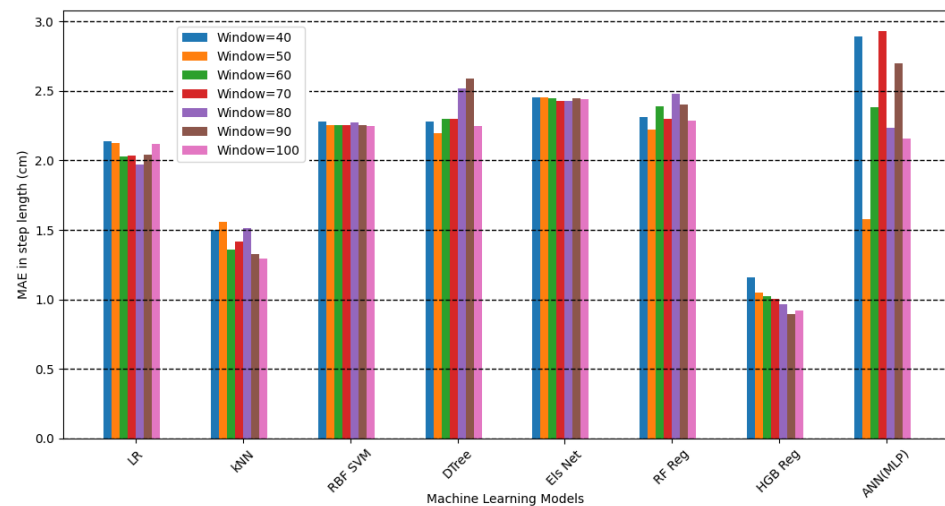


Figure 8. Comparisons of MAE in step length predicted by various regressor models at different window sizes using the LOOCV strategy in the normal-handheld scenario.

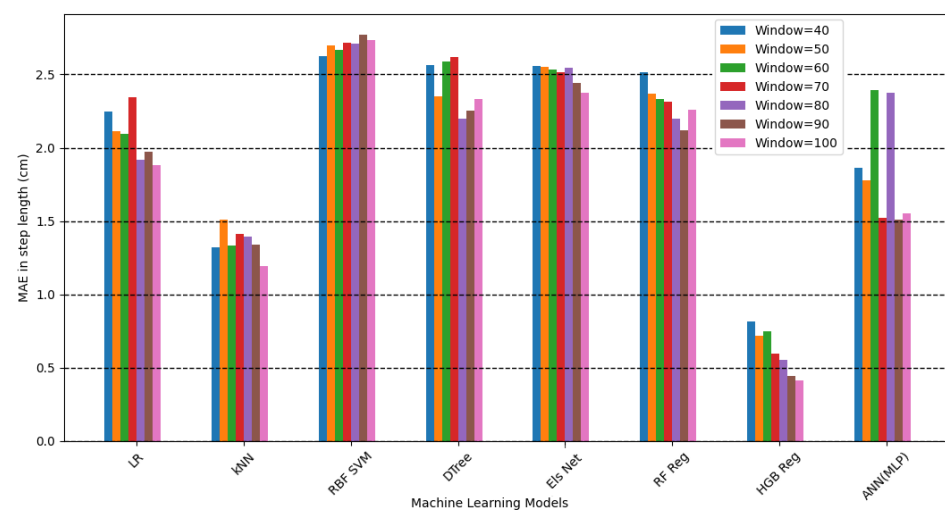


Figure 9. Comparisons of MAE in step length predicted by various regressor models at different window sizes using the LOOCV strategy in the normal-pocket scenario.

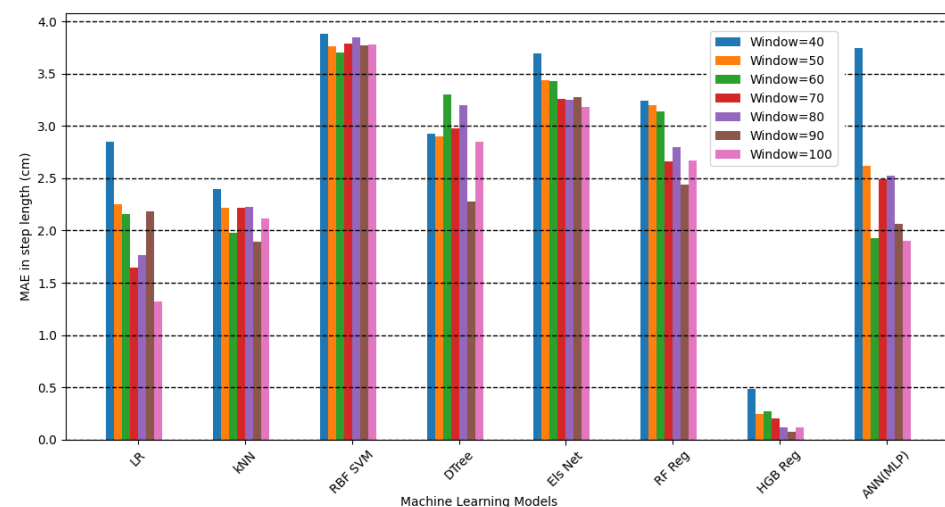


Figure 10. Comparisons of MAE in step length predicted by various regressor models at different window sizes using the LOOCV strategy in the fast-handheld scenario.

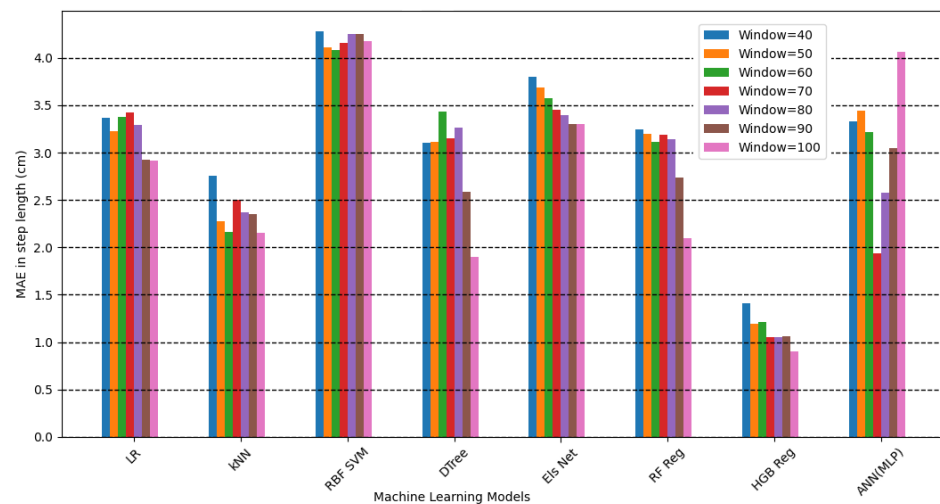


Figure 11. Comparisons of MAE in step length predicted by various regressor models at different window sizes using the LOOCV strategy in the fast-pocket scenario.

Figures 8–11 demonstrate that the HGB regressor achieves superior performance in all four scenarios, outperforming the other considered models with an MAE below 0.5 cm, 1.0 cm, and 1.5 cm across window sizes in FH, NP, and NH and FP, respectively. Among the other models, k-NN, unlike the context-wise evaluations, achieves better MAE results in the NH and NP scenarios, with slightly degraded results in the other two scenarios. On the other hand, ANN performs better for certain window sizes in each of the four scenarios, and RBF-SVM performs worst in all scenarios except NH, while LR performs better than Els net, DTree, and RF regressors, in all scenarios except FP.

The variation in training and test times across window sizes for the considered regression models under all four contexts is shown in Tables 2–5. The reported training and test times across window sizes indicate that, in most cases, they decrease with increasing window size. The results reported in all four tables also show that the ANN, among the considered models, incurs significantly high training time but relatively short test time, where the training and test times of the considered ensemble methods (RF and HGB regressors) are higher than those of the other models, except the ANN. It is further observed from these tables that the training and test times incurred by the HGB regressor across the window sizes are substantially shorter than those for the RF regressor, and k-NN, being a lazy learner, incurs a higher test time than its training time, unlike the other models.

Table 2. Comparisons of training and test time across window sizes (40–100) for different models using the LOOCV strategy in the normal-handheld scenario.

Model	Training Time (ms)							Test Time (ms)						
	40	50	60	70	80	90	100	40	50	60	70	80	90	100
LR	1.66	1.67	1.27	1.32	1.16	1.11	1.26	0.13	0.16	0.12	0.12	0.12	0.13	0.12
kNN	0.36	0.39	0.32	0.32	0.35	0.31	0.32	1.08	1.09	1.00	0.98	1.04	0.98	1.02
RBF SVM	3.21	2.23	1.77	1.58	1.19	1.12	0.98	1.88	1.53	0.90	0.81	0.65	0.49	0.41
DTree	3.07	2.21	2.00	1.79	1.68	1.55	1.46	0.18	0.15	0.15	0.15	0.15	0.14	0.14
Elastic Net	2.17	1.89	1.81	1.71	1.56	1.63	1.53	0.14	0.13	0.13	0.12	0.12	0.12	0.12
RF Reg.	239.28	209.95	198.81	177.27	168.42	165.46	154.92	5.95	5.92	5.62	5.60	5.52	6.19	5.41
HGB Reg.	162.06	163.59	166.67	133.63	122.81	114.39	102.76	3.97	3.94	4.00	3.03	2.79	3.05	2.70
ANN (MLP)	1109.13	885.07	1027.43	945.00	865.43	605.68	929.64	0.32	0.40	0.30	0.29	0.28	0.28	0.28

Table 3. Comparisons of training and test time across window sizes (40–100) for different models using the LOOCV strategy in the normal-pocket scenario.

Model	Training Time (ms)							Test Time (ms)						
	40	50	60	70	80	90	100	40	50	60	70	80	90	100
LR	1.92	1.39	1.36	1.37	1.29	1.02	1.07	0.16	0.12	0.14	0.13	0.14	0.12	0.12
kNN	0.41	0.33	0.33	0.34	0.31	0.31	0.29	1.31	1.05	0.97	1.12	1.01	0.95	0.92
RBF SVM	3.30	2.23	2.25	1.75	1.29	1.45	1.06	1.70	1.11	1.20	0.75	0.52	0.67	0.42
DTree	2.96	2.21	2.41	1.97	1.72	1.62	1.47	0.24	0.15	0.17	0.18	0.17	0.16	0.17
Elastic Net	2.82	1.92	1.89	1.96	1.64	1.78	1.54	0.18	0.13	0.12	0.15	0.13	0.13	0.13
RF Reg.	266.59	209.89	188.90	196.95	184.42	173.45	157.39	5.73	7.04	5.79	6.13	6.61	6.05	5.34
HGB Reg.	180.37	169.61	159.12	140.06	134.54	135.13	106.54	3.86	2.82	4.00	3.67	3.99	3.76	3.78
ANN (MLP)	1096.23	872.13	1022.18	1024.26	944.77	723.61	947.49	0.32	0.31	0.29	0.34	0.30	0.33	0.30

Table 4. Comparisons of training and test time across window sizes (40–100) for different models using the LOOCV strategy in the fast-handheld scenario.

Model	Training Time (ms)							Test Time (ms)						
	40	50	60	70	80	90	100	40	50	60	70	80	90	100
LR	1.85	1.45	1.38	1.19	1.33	1.17	1.14	0.14	0.13	0.13	0.12	0.13	0.12	0.12
kNN	0.36	0.33	0.30	0.30	0.32	0.33	0.32	0.96	0.97	0.95	0.93	1.01	1.05	0.92
RBF SVM	3.25	2.24	2.36	2.23	1.31	1.27	0.97	1.10	0.71	0.74	0.63	0.40	0.35	0.30
DTree	3.22	2.30	2.24	2.39	1.65	1.56	1.50	0.21	0.16	0.16	0.17	0.14	0.14	0.19
Elastic Net	2.59	2.07	1.75	2.00	1.54	1.87	1.63	0.21	0.13	0.12	0.14	0.12	0.14	0.13
RF Reg.	249.35	218.63	200.72	228.51	189.05	182.47	156.96	5.47	5.81	5.60	7.49	6.15	5.43	5.15
HGB Reg.	220.13	158.05	158.30	174.90	131.99	133.41	115.83	4.83	2.87	3.30	3.86	3.87	2.49	3.39
ANN (MLP)	1150.18	947.29	1251.98	1087.93	866.39	717.18	1131.21	0.29	0.28	0.29	0.27	0.26	0.27	0.36

Table 5. Comparisons of training and test time across window sizes (40–100) for different models using the LOOCV strategy in the fast-pocket scenario.

Model	Training Time (ms)							Test Time (ms)						
	40	50	60	70	80	90	100	40	50	60	70	80	90	100
LR	1.67	1.35	1.26	1.23	1.31	1.14	1.16	0.12	0.12	0.12	0.12	0.14	0.12	0.13
kNN	0.32	0.31	0.31	0.30	0.32	0.30	0.35	1.02	0.96	0.95	0.94	0.95	0.94	1.10
RBF SVM	3.09	2.24	2.16	1.37	1.24	1.12	1.13	0.93	0.70	0.57	0.41	0.36	0.33	0.33
DTree	2.74	2.21	2.13	1.77	1.71	1.60	1.45	0.16	0.16	0.15	0.16	0.15	0.15	0.14
Elastic Net	1.85	1.98	1.84	1.68	1.73	1.66	1.53	0.13	0.13	0.12	0.12	0.16	0.13	0.12
RF Reg.	238.20	210.98	199.94	175.39	178.17	159.59	155.84	5.71	5.38	5.25	5.21	5.78	5.25	5.31
HGB Reg.	164.21	155.07	152.91	129.51	130.37	113.30	98.55	2.91	3.40	4.78	2.76	4.16	4.07	3.02
ANN (MLP)	1077.95	871.93	1014.25	923.30	958.32	600.49	917.14	0.31	0.27	0.27	0.26	0.27	0.29	0.27

3.2. Evaluation Using Unseen Data

In this case, the regression models were trained on datasets collected from all four contexts, except the one selected for the test dataset, selecting three out of four datasets, and then evaluated on the dataset explicitly set aside for testing. Based on the specific context of the selected test dataset, four unseen test evaluation scenarios appear—evaluation using unseen NH, evaluation using unseen NP, evaluation using unseen FH, and evaluation using unseen FP—as depicted by the red arrows in Figure 7. Since the walking context of selected test data is not included in the training dataset, as depicted by the red arrows in Figure 7, this ensures no potential data overlap between the training and test sets in these evaluation scenarios, like in the previous LOOCV evaluations. The experimental results in these categories are presented below.

3.2.1. Evaluation Using Unseen Normal-Handheld (NH) Test Data

Figure 12 and Table 6 compare the achieved MAE and training and test times, respectively, for various regression models considered in this study, across window sizes from 40 to 100 in the unseen NH scenario. Unlike the LOOCV evaluation cases, ANN outperforms the other models considered here at a window size of 50, achieving the lowest MAE in this

scenario. At the same time, Els net and LR perform better than the other models across window sizes and at window sizes below 80, respectively, in this scenario. It is also noted from Figure 12 that the achieved MAE for k-NN, RBF-SVM, decision tree, and RF remains almost the same across window sizes, whereas that of the HGB regressor is slightly lower in this scenario.

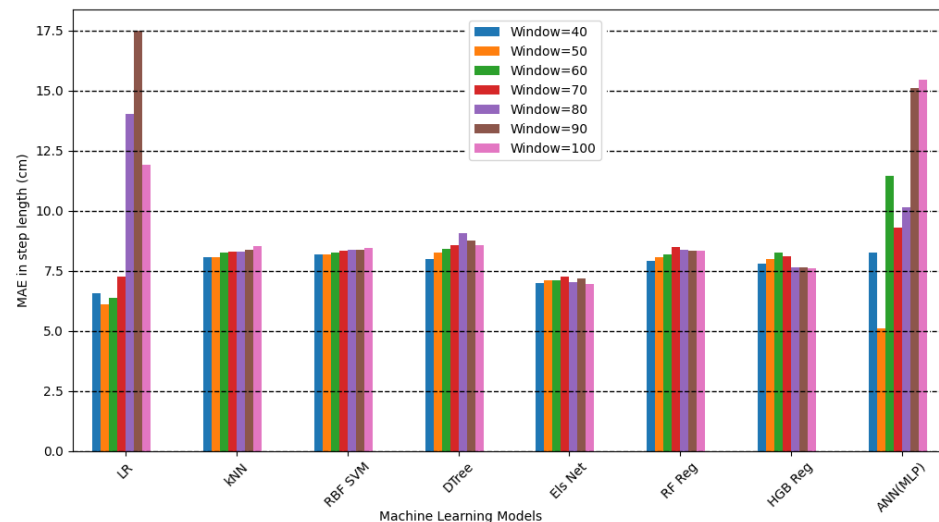


Figure 12. Comparisons of MAE in step length predicted by various regressor models at different window sizes in an unseen normal-handheld scenario.

Table 6. Comparisons of training and test time across window sizes (40–100) for different models using unseen data under the normal-handheld scenario.

Model	Training Time (ms)							Test Time (ms)						
	40	50	60	70	80	90	100	40	50	60	70	80	90	100
LR	1.62	1.35	1.21	1.06	1.01	0.99	0.97	0.15	0.14	0.12	0.13	0.12	0.12	0.12
kNN	0.32	0.32	0.31	0.31	0.31	0.33	0.30	1.03	1.17	1.03	1.00	1.00	0.99	1.02
RBF-SVM	2.43	1.62	1.25	1.11	1.00	0.93	0.87	2.23	1.22	0.85	0.69	0.57	0.47	0.42
DTree	2.21	1.82	1.67	1.53	1.40	1.34	1.29	0.17	0.15	0.15	0.14	0.14	0.14	0.15
Els Net	1.34	1.26	1.22	1.16	1.14	1.16	1.17	0.13	0.13	0.13	0.13	0.12	0.13	0.13
RF Reg.	207.31	179.89	170.72	156.62	151.35	147.73	142.77	5.96	5.78	5.84	5.53	6.03	5.85	5.81
HGB Reg.	160.50	131.88	126.74	105.38	92.87	100.96	89.61	3.45	3.23	3.48	3.73	2.54	4.45	3.47
ANN (MLP)	322.84	534.60	312.69	601.43	515.08	509.64	471.71	0.35	0.34	0.30	0.30	0.28	0.28	0.36

The comparison of training and test times for various regression models, as shown in Table 6, reveals that ANN incurs the highest training time among the models considered in this study and a relatively shorter test time across the window sizes in this specific evaluation, as in LOOCV evaluation scenarios. On the other hand, the training and test times for the HGB regressor across window sizes remain shorter than those for the RF regressor but longer than those for other models except ANN, as observed in the previous LOOCV evaluation scenarios.

3.2.2. Evaluation Using Unseen Normal Packet (NP) Test Data

The achieved MAE and training and test times for various regression models considered in this study, across window sizes from 40 to 100 in the unseen NP scenario, are compared in Figure 13 and Table 7, respectively. In this scenario, ANN outperforms the other models considered in this study for window sizes below 80, achieving the lowest MAE at a window size of 70. Among the remaining models, Els net performs better than the other models with decreasing MAE across window sizes, and LR outperforms all other considered models at a window size of 80 in this scenario. It is also noted from Figure 13 that k-NN and RBF-SVM achieve similar results, and the results for DTree and HGB regres-

sors remain almost the same across window sizes. Compared to them, the RF regressor performs slightly better in this scenario.

The training and test times incurred by various regression models, as reported in Table 6, reveal that their variation across window sizes and models in the unseen NP evaluation follows the same trend observed in the unseen NH evaluation and previous LOOCV evaluation scenarios.

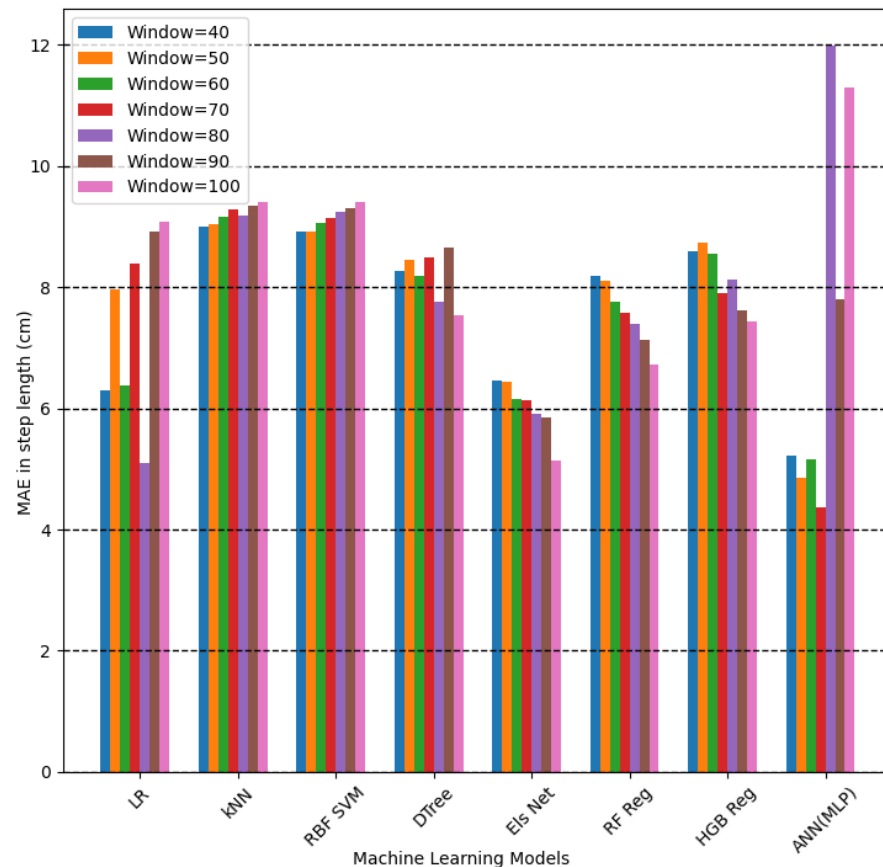


Figure 13. Comparisons of MAE in step length predicted by various regressor models at different window sizes in an unseen normal-pocket scenario.

Table 7. Comparisons of training and test time across window sizes (40–100) for different models using unseen data under the normal-pocket scenario.

Model	Training Time (ms)							Test Time (ms)						
	40	50	60	70	80	90	100	40	50	60	70	80	90	100
LR	1.43	1.32	1.37	1.07	1.05	1.14	1.01	0.13	0.13	0.13	0.14	0.11	0.12	0.12
kNN	0.32	0.32	0.34	0.32	0.30	0.35	0.31	1.11	1.09	1.01	0.99	1.04	0.99	0.99
RBF SVM	2.14	1.58	1.35	1.15	1.02	0.95	0.92	1.74	1.21	0.89	0.69	0.59	0.51	0.44
DTree	2.26	1.98	1.74	1.71	1.47	1.40	1.37	0.14	0.15	0.15	0.15	0.14	0.14	0.14
Els Net	1.82	1.77	1.72	1.64	1.44	1.62	1.41	0.13	0.13	0.12	0.12	0.12	0.12	0.12
RF Reg.	216.18	190.57	175.89	165.74	159.60	150.87	154.40	5.72	5.94	5.92	5.77	5.76	6.33	5.69
HGB Reg.	168.32	132.99	126.51	120.11	108.35	103.14	101.66	3.01	3.23	4.42	3.65	3.20	3.47	2.90
ANN (MLP)	894.12	607.31	483.91	741.68	745.35	715.45	883.44	0.36	0.35	0.28	0.29	0.28	0.29	0.30

3.2.3. Evaluation Using Unseen Fast-Handheld (FH) Test Data

The achieved MAE and training and test times for various regression models considered in this study, across window sizes from 40 to 100 in the unseen FH scenario, are compared in Figure 14 and Table 8, respectively. In this scenario, the ANN achieves the lowest MAE at a window size of 100, outperforming the other models considered in this study. However, compared to unseen NH and NP evaluations, all considered models'

performance degrades, as their achieved MAE across the window sizes exceeds 6 cm. On the other hand, the achieved MAE for elastic net across window sizes, and the same for LR at certain window sizes, remain lower than those of the other models, except ANN at window size 100.

The reported training and test times for various regression models, as shown in Table 8, reveal a similar variation trend in training and test times across window sizes and models in the unseen FH evaluation, as observed in the unseen NH and NP evaluations and all LOOCV evaluation scenarios.

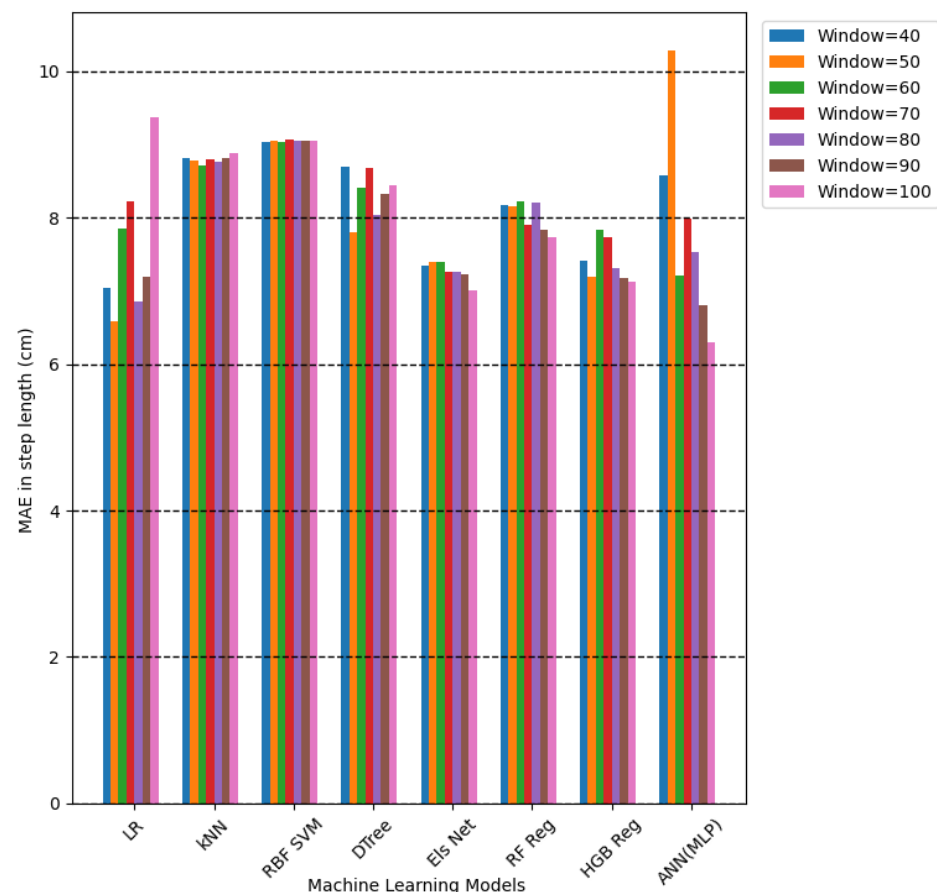


Figure 14. Comparisons of MAE in step length predicted by various regressor models at different window sizes in an unseen fast-handheld scenario.

Table 8. Comparisons of training and test time across window sizes (40–100) for different models using unseen data under the fast-handheld scenario.

Model	Training Time (ms)							Test Time (ms)						
	40	50	60	70	80	90	100	40	50	60	70	80	90	100
LR	1.70	1.59	1.47	1.37	1.34	1.18	1.06	0.13	0.16	0.12	0.12	0.12	0.12	0.11
kNN	0.36	0.32	0.32	0.31	0.34	0.34	0.31	0.95	0.95	0.88	0.91	0.97	1.00	1.01
RBF SVM	3.30	2.13	1.91	1.24	1.14	1.09	0.93	0.79	0.79	0.43	0.34	0.30	0.29	0.25
DTree	3.03	2.29	2.09	1.77	1.66	1.59	1.43	0.15	0.19	0.15	0.14	0.15	0.15	0.14
Els Net	2.09	1.92	1.53	1.54	1.55	1.50	1.45	0.12	0.17	0.13	0.12	0.12	0.13	0.12
RF Reg.	248.75	211.02	190.41	180.25	167.72	162.41	152.34	5.95	5.83	5.46	5.42	5.40	5.39	5.52
HGB Reg.	171.83	146.71	136.79	134.36	107.57	104.67	105.57	3.49	4.44	3.78	4.03	2.49	3.52	2.93
ANN (MLP)	1057.31	840.13	946.51	843.29	700.16	868.92	873.28	0.29	0.39	0.27	0.27	0.26	0.26	0.26

3.2.4. Evaluation Using Unseen Fast-Pocket (FP) Test Data

Figure 15 and Table 9 compare the achieved MAE and training and test times, respectively, for various regression models considered in this study, across window sizes from 40 to 100 in the unseen FP scenario. In this scenario, similar to other unseen evaluations,

ANN achieves the lowest MAE at a window size of 70, outperforming the other models considered here. Among the remaining models, Els net performs better than the other models with decreasing MAE across window sizes, and LR performs better at window sizes below 90 in this scenario. It is also noted from Figure 15 that k-NN and RBF-SVM achieve similar results, and the results for RF and HGB regressors remain almost the same across window sizes. Compared to them, the DTree regressor performs slightly better at window sizes above 40 in this scenario.

The training and test times for various regression models, as reported in Table 9, reveal that their variation across window sizes and models in the unseen FP evaluation follows the same trend observed in the other unseen evaluations and all the LOOCV evaluation scenarios.

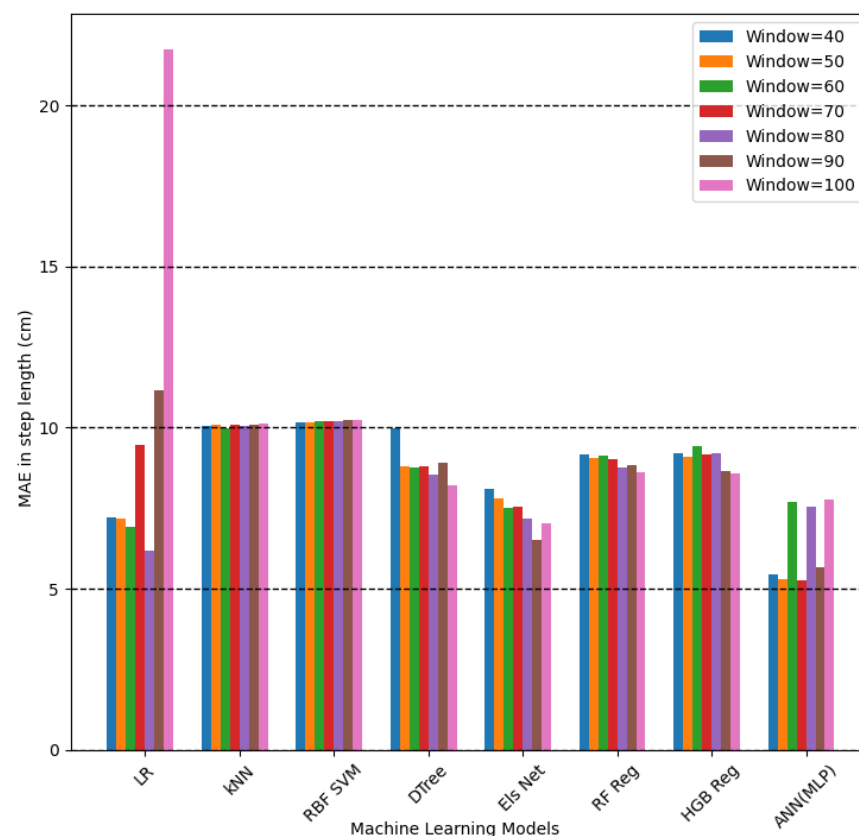


Figure 15. Comparisons of MAE in step length predicted by various regressor models at different window sizes in an unseen fast-pocket scenario.

Table 9. Comparisons of training and test time across window sizes (40–100) for different models using unseen data under the fast-pocket scenario.

Models	Training Time (ms)							Test Time (ms)						
	40	50	60	70	80	90	100	40	50	60	70	80	90	100
LR	1.64	1.67	1.43	1.46	1.28	1.17	1.25	0.12	0.17	0.12	0.12	0.13	0.12	0.12
kNN	0.36	0.37	0.33	0.36	0.34	0.36	0.34	0.96	1.02	0.87	0.83	1.01	0.98	1.00
RBF-SVM	3.07	2.60	2.11	1.80	1.26	1.14	1.08	0.77	0.56	0.42	0.37	0.30	0.28	0.25
DTree	2.64	2.78	2.59	1.93	1.71	1.56	1.54	0.14	0.17	0.14	0.15	0.15	0.14	0.14
Els Net	1.63	1.53	1.77	1.79	1.50	1.45	1.53	0.13	0.14	0.12	0.12	0.14	0.12	0.12
RF Reg.	248.75	208.79	208.95	199.32	176.68	167.31	166.38	6.03	6.09	5.54	5.74	5.57	5.40	5.77
HGB Reg.	167.03	145.99	146.75	141.72	120.36	111.16	107.24	3.15	2.87	3.46	2.39	3.16	3.75	2.68
ANN (MLP)	872.42	896.20	1059.05	863.83	654.25	638.61	978.22	0.30	0.31	0.31	0.27	0.28	0.30	0.27

3.3. Evaluation Using Seen Data

In this case, the regression models were trained on all four datasets collected in four walking contexts and then evaluated on a single dataset from these four, resulting in four evaluation scenarios: evaluation using NH, evaluation using NP, evaluation using FH, and evaluation using FP test data similar to the previous unseen evaluation cases. Since the walking context of test data is included in the training dataset, as illustrated by the green arrow in Figure 7, these evaluations are based on seen test data and potential overlap between the training and test sets. The training and test times for various regression models across window sizes from 40 to 100 in these evaluations follow the same trend observed in the previous LOOCV and unseen data-based context-wise evaluation scenarios. Thus, the evaluation results concerning the MAE only for these four categories are presented below.

3.3.1. Evaluation Using Normal-Handheld (NH) Test Data

In this scenario, the regression models trained on datasets collected in all four contexts were evaluated only on a dataset collected in normal walking mode with the device in a handheld position. As outlined above, all regression models considered in this study were compared in terms of MAE and training and test durations across a range of window sizes from 40 to 100. Figure 16 compares the MAE for estimating step length across different window sizes for the regression models considered in this study. Figure 16 depicts that the HGB regressor achieves superior performance among the regression models considered here, reducing the MAE below 0.5 cm across window sizes, with the lowest MAE at a window size of 60 in the normal-handheld scenario. In contrast, the MAE for LR, decision tree, random forest, elastic net, and ANN decreases as the window size increases in this scenario. Among them, LR, DTree, and RF gain a noticeable improvement in error reduction compared to other models.

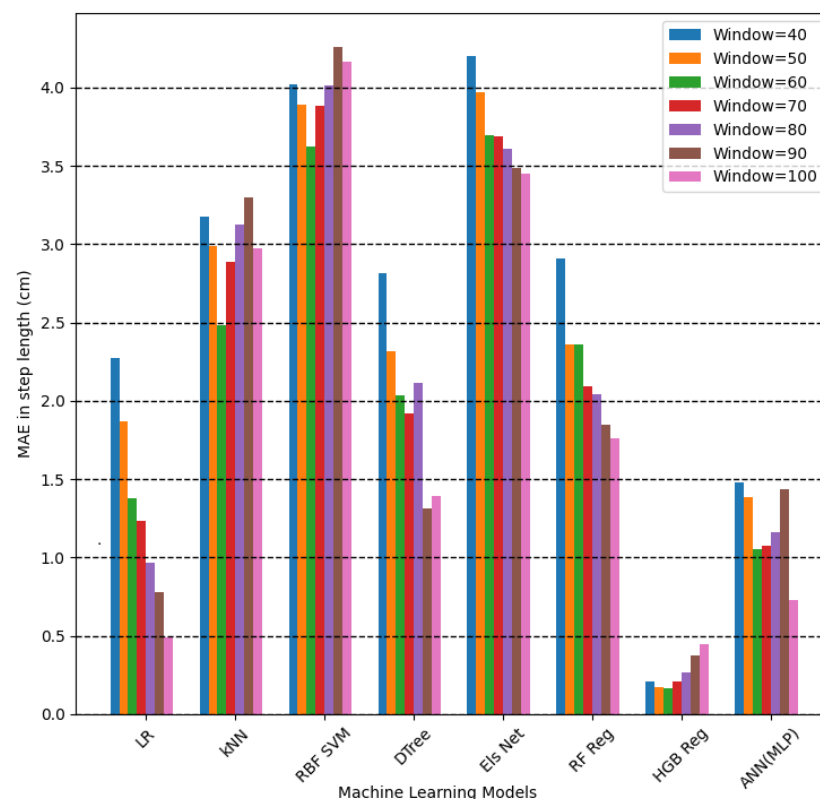


Figure 16. Comparisons of MAE in step length predicted by various regression models at different window sizes using the seen data in a normal-handheld scenario.

3.3.2. Evaluation Using Normal-Pocket (NP) Test Data

In this scenario, the regression models trained on datasets collected in all four contexts were evaluated only on a dataset collected in normal walking mode with the device placed in a trouser pocket. The MAE achieved by the various regression models across window sizes from 40 to 100 in this scenario is compared in Figure 17. Figure 17 shows that the HGB regressor and LR achieve better performance across different window sizes than the other models considered here, reducing the MAE to well below 1 cm in most cases, with LR achieving the lowest MAE at the window size of 100 in the normal-pocket scenario. It is also noted that the achieved MAE for LR, decision tree, elastic net, random forest, and ANN decreases as the window size increases. At the same time, the opposite trend is observed for the HGB regressor in this scenario, as in the previous one (i.e., NH). Moreover, a slight improvement in MAE reduction was observed for most models, specifically ANN, across different window sizes in this scenario compared to NH.

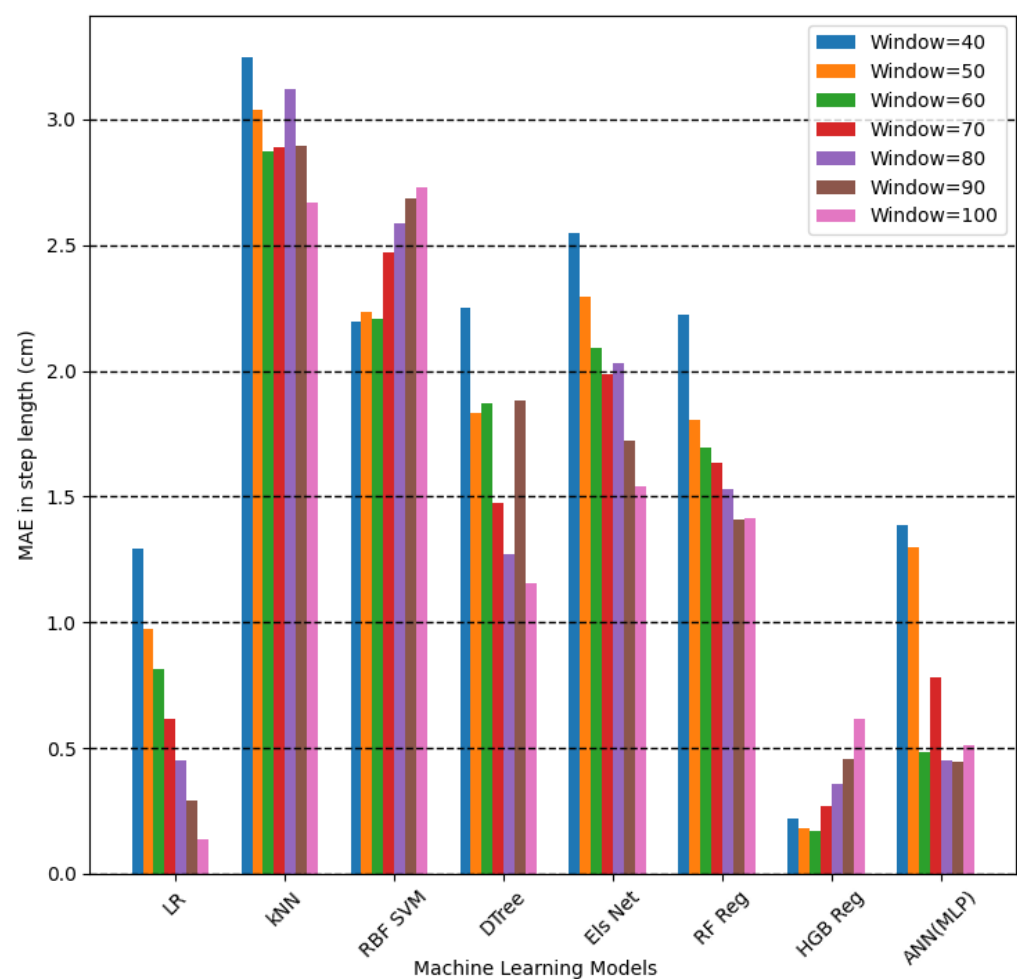


Figure 17. Comparisons of MAE in step length predicted by various regressor models at different window sizes using the seen data in a normal-pocket scenario.

3.3.3. Evaluation Using Fast-Handheld (FH) Test Data

In this scenario, the regression models trained on datasets collected in all four contexts were evaluated only on a dataset collected in fast-walking mode with the device in a handheld position. Figure 18 compares the achieved MAE and training and test times, respectively, for various regression models considered in this study, across window sizes from 40 to 100. Figure 18 shows that the HGB regressor achieves superior performance among the regression models considered here, reducing the MAE substantially below

1 cm in this scenario, similar to NH. At the same time, the RBF SVM model performs the worst, achieving an MAE of nearly 8 cm across all window sizes. In contrast, LR, decision tree, random forest, and ANN models perform well with window sizes above 50, while the achieved MAE for k-NN and Els net remain almost the same across window sizes in this scenario.

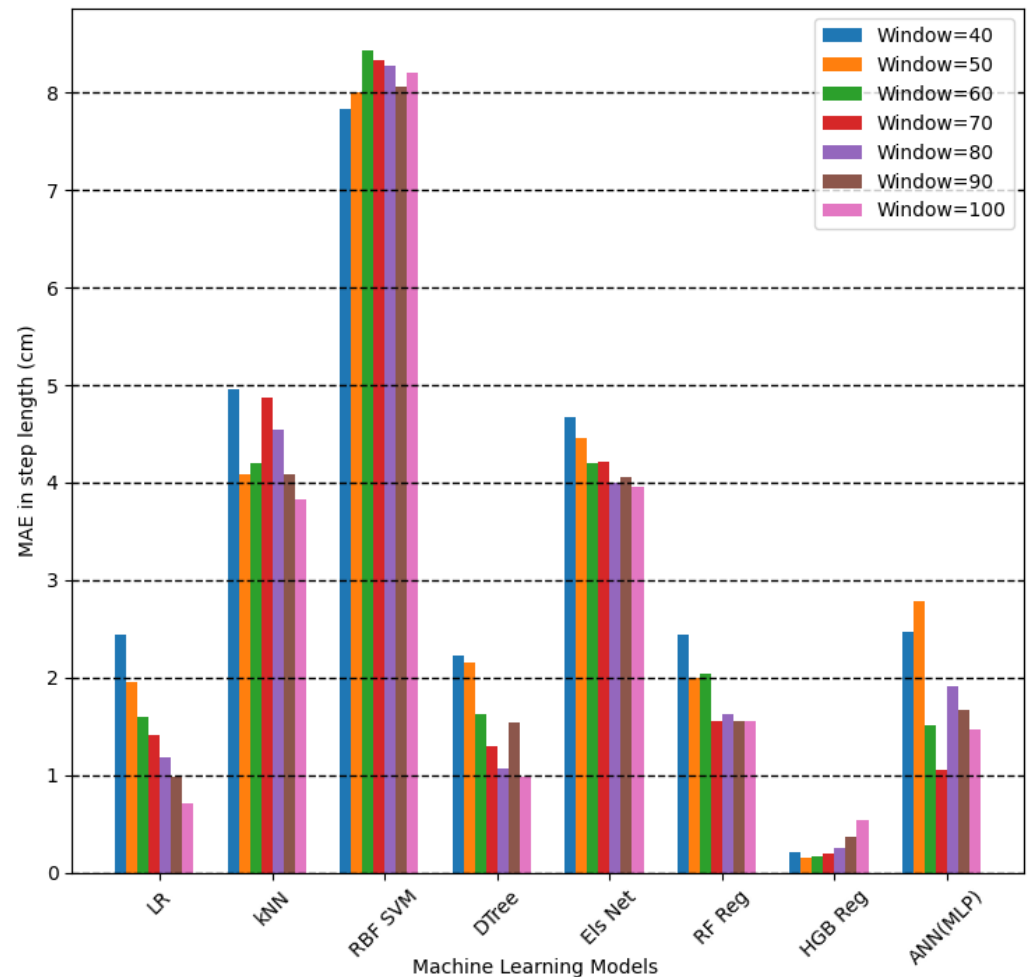


Figure 18. Comparisons of MAE in step length predicted by various regressor models at different window sizes using the seen data in a fast-handheld scenario.

3.3.4. Evaluation Using Fast-Pocket (FP) Test Data

In this scenario, the regression models trained on datasets collected in all four contexts were evaluated only on a dataset collected in fast-walking mode with the device placed in a trouser pocket. The achieved MAE for regression models considered in this study, across window sizes from 40 to 100, is compared in Figure 19. It is evident from Figure 19 that the HGB regressor performs better than the other models for window sizes below 80. At the same time, the LR's performance improves for window sizes of 80 or above, compared to the HGB regressor in this scenario. Figure 19 also shows a significant reduction in MAE for ANN compared to the other models, except for the HGB regressor and LR. At the same time, SVM gains the worst performance, with MAE above 8 cm across window sizes in this scenario.

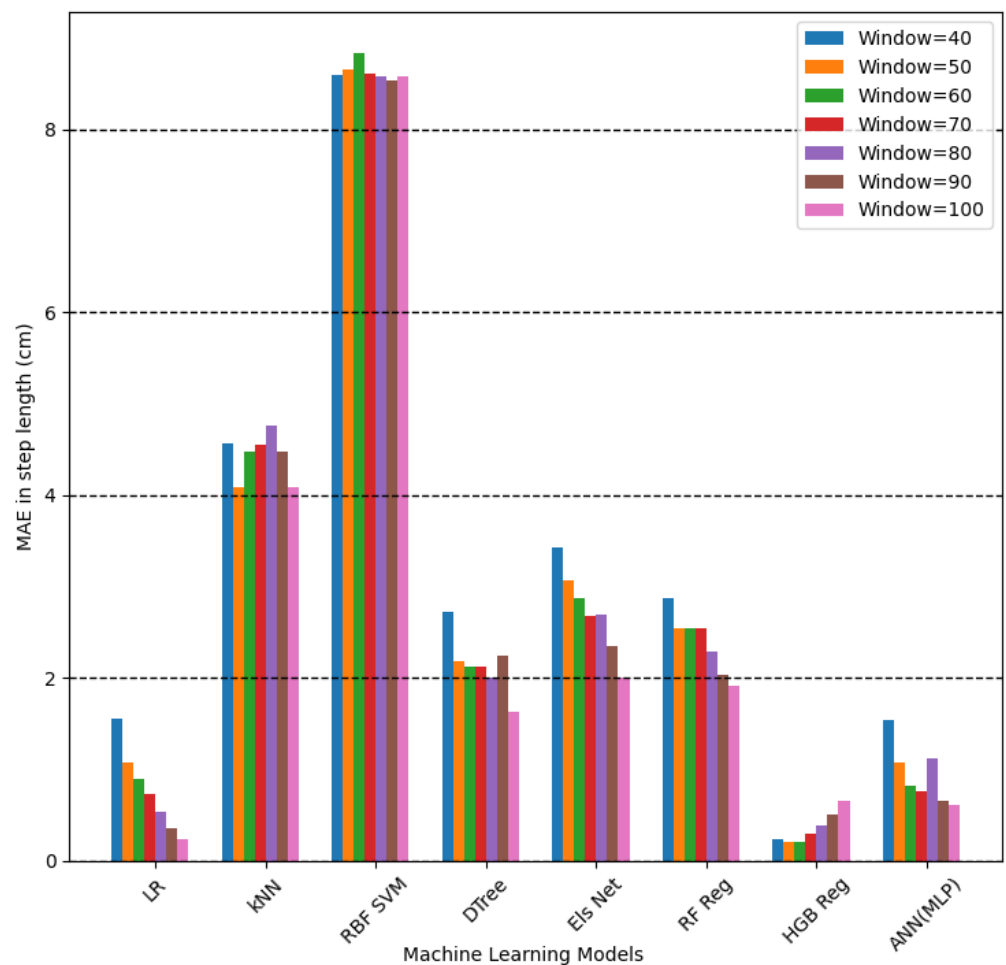


Figure 19. Comparisons of MAE in step length predicted by various regressor models at different window sizes using the seen data in a fast-pocket scenario.

3.4. Comparative Results

The best performance of the considered regression models across all four walking contexts, in terms of the lowest achieved MAE and the window size at which it was achieved, is first compared separately for three evaluation scenarios (LOOCV, unseen and seen) in Tables 10–12. Table 10 shows that four of the eight models considered here achieve the lowest MAE at a window size of 100 in NH, NP, and FH contexts, whereas six models perform best at this window size in the FP context under seen test evaluations. However, the best-performing model in the NH, FH, and FP contexts, the HGB regressor, achieves the lowest MAE at a window size of 60 and 50 in the normal and fast walking modes, respectively. Table 10 also shows that the lowest MAE in all four contexts is 0.14 cm, achieved at a window size of 100 by LR in the NP context.

Table 11 shows that five of eight models achieve the lowest MAE at a window size of 50 in the NH context, whereas four models in the NP and FH contexts, and only two in the FP context, perform best at the window size of 100. However, the best-performing model in all four contexts, ANN, achieves the lowest MAE, ranging from 4.37 to 6.30 cm at different window sizes, as shown in Table 11. Table 12 shows that four of the eight models considered here achieve the lowest MAE at a window size of 100 in NH, NP, and FH contexts, whereas six models perform best at this window size in the FP context under seen test evaluations. However, the best-performing model in the NH, FH, and FP contexts, the HGB regressor, achieves the lowest MAE at a window size of 60 and 50 in the normal

and fast walking modes, respectively. Table 12 also shows that the lowest MAE in all four contexts is 0.14 cm, achieved on a window size of 100 by LR in the NP context.

Table 10. Best achieved performance comparisons of the considered models across scenarios under LOOCV evaluations.

Model	Normal-Handheld		Normal-Pocket		Fast-Handheld		Fast-Pocket	
	MAE (cm)	WS	MAE (cm)	WS	MAE (cm)	WS	MAE (cm)	WS
LR	1.97	80	1.88	100	1.32	100	2.92	100
KNN	1.30	100	1.19	100	1.89	90	2.15	100
RBF SVM	2.25	100	2.62	40	3.70	60	4.08	60
DTree	2.20	50	2.20	80	2.28	90	1.90	100
Els Net	2.43	80	2.38	100	3.19	100	3.30	90
RF Reg	2.22	50	2.12	90	2.44	90	2.09	100
HGB Reg	0.90	90	0.41	100	0.08	90	0.90	100
ANN(MLP)	1.57	50	1.51	90	1.91	100	1.94	70

Table 11. Best achieved performance comparisons of the considered models across scenarios under unseen test evaluations.

Model	Normal-Handheld		Normal-Pocket		Fast-Handheld		Fast-Pocket	
	MAE (cm)	WS	MAE (cm)	WS	MAE (cm)	WS	MAE (cm)	WS
LR	6.10	50	5.10	80	6.59	50	6.18	80
KNN	8.06	50	8.99	40	8.72	60	9.99	60
RBF SVM	8.17	50	8.91	50	9.04	60	10.15	40
DTree	8.25	50	7.54	100	7.81	50	8.22	100
Els Net	7.01	40	5.14	100	7.00	100	6.51	90
RF Reg	7.90	40	6.72	100	7.73	100	8.60	100
HGB Reg	7.62	100	7.43	100	7.13	100	9.11	50
ANN(MLP)	5.12	50	4.37	70	6.30	100	5.25	70

Table 12. Best achieved performance comparisons of the considered models across scenarios under seen test evaluations.

Model	Normal-Handheld		Normal-Pocket		Fast-Handheld		Fast-Pocket	
	MAE (cm)	WS	MAE (cm)	WS	MAE (cm)	WS	MAE (cm)	WS
LR	0.49	100	0.14	100	0.72	100	0.24	100
KNN	2.48	60	2.67	100	3.83	100	4.09	100
RBF SVM	3.62	60	2.19	40	7.83	40	8.54	90
DTree	1.31	90	1.16	100	0.98	100	1.62	100
Els Net	3.45	100	1.54	100	3.96	100	2.01	100
RF Reg	1.76	100	1.41	90	1.55	90	1.92	100
HGB Reg	0.17	60	0.17	60	0.16	50	0.20	50
ANN(MLP)	0.73	100	0.44	90	1.05	70	0.62	100

To demonstrate the superiority of our newly proposed regression model-based SLE framework, its predictive performance is compared with our previously proposed peak-valley detection-based SLE method [15], which was also evaluated in four diverse contexts, in terms of achieved MAE. However, at this stage, the other existing machine learning-based

SLE methods have not been considered, as they are designed to operate on measurements from sensors other than the accelerometer, which is the only sensor our proposed method uses, or a smartphone affixed to a specific body segment rather than a freely held one, which is our case. For each of the four distinct contexts (NH, NP, FH, FP) under three evaluation scenarios (i.e., LOOCV, unseen, and seen), the lowest achieved MAE is selected for fair comparison. Alongside the MAE result, Table 13 also reports the regression model that achieved the lowest MAE and the specific window size at which it was achieved. For instance, in the normal-handheld condition, the HGB regressor achieves the lowest MAE among the considered regression models in the evaluations with the LOOCV strategy and seen test data, with an MAE of 0.9 and 0.17 cm at window sizes 90 and 60, respectively, as reflected in the third row of Table 13. The reported comparison of the MAE results indicates that the HGB regressor performs best in leave-one-out cross-validation and most seen-data cases, and ANN performs best in the unseen-data evaluation condition.

Table 13. Comparison of the best MAE across scenarios (evaluations with seen data and unseen data, LOOCV) with previously proposed IRT-SD-SLE [15].

Scenario	LOOCV			Evaluation with Unseen Data			Evaluation with Seen Data			IRT-SD-SLE [15]	
	Model	WS	MAE (cm)	Model	WS	MAE (cm)	Model	WS	MAE (cm)	WS	MAE (cm)
Normal-Handheld	HGB Reg.	90	0.9	ANN (MLP)	50	5.11	HGB Reg.	60	0.17	50	13.08
Normal-Pocket	HGB Reg.	100	0.41	ANN (MLP)	70	4.37	LR	100	0.14	50	11.57
Fast-Handheld	HGB Reg.	90	0.1	ANN (MLP)	100	6.30	HGB Reg.	50	0.16	50	16.09
Fast-Pocket	HGB Reg.	100	0.9	ANN (MLP)	70	5.25	HGB Reg.	50	0.20	50	13.72

The MAE results for the IRT-SD-SLE reported in Table 13, and the same for various considered models across the window sizes 40–100, shown in Figures 8–19, on the other hand, reveal that the achieved MAE of the considered regression models under LOOCV and seen test evaluation cases remain well below the same for the previous method in all four walking contexts. However, in the unseen test evaluations, the achieved MAE for LR and ANN, at window sizes 80–90 and 90–100, respectively, in the NH context, ANN at window size 80 in the NP context, and LR at window size 100 in the FP context, is larger than that of IRT-SD-SLE. All considered models outperform the previous method, IRT-SD-SLE, in terms of MAE across all three evaluation scenarios (LOOCV, unseen, and seen) in the FH context, indicating the superiority of the ML regression model over the traditional signal-processing-based peak–valley method under the most challenging fast handheld walking context.

4. Discussion

The reported results show notable variations in the MAE achieved by various models and their training and test times across scenarios, varied window sizes, and evaluation conditions. The MAE results for various models across varied window sizes in the four distinct contexts of three evaluation scenarios are analyzed first. Then, the computational efficiency, measured by training and test times, is analyzed, followed by a discussion of the trade-off between prediction accuracy and computational efficiency.

4.1. Analysis of MAE Results

The performance of the considered regression models was analyzed under three evaluation scenarios, namely, leave-one-out cross-validation (LOOCV), unseen test evaluation, and seen test evaluation. The results indicate a strong influence of the evaluation strategy, walking condition, smartphone placement, and window size on the predictive performance of the considered models.

LOOCV strategy-based evaluations reveal that the HGB regressor achieves remarkable results across varied window sizes in all four contexts, with an MAE sufficiently below 1.5 cm. Following the HGB regressor, k-NN also offers superior performance across varied window sizes in all four contexts of such evaluation scenarios. The evaluation results also indicate that all considered models perform better in normal-walking mode than in fast-walking mode, as demonstrated in seen and unseen test evaluations.

Evaluation of the considered regressor models on unseen test data across four diverse contexts, on the other hand, reveals that the ANN model outperforms the other models at specific window sizes, achieving the lowest MAE below 5.3 cm in all contexts except the FH. However, significant performance degradation at certain higher window sizes for ANN occurs in NH and NP scenarios, exceeding the achieved MAE beyond 10 cm, and the same is observed for LR in NH and FH. On the contrary, the elastic net model in this evaluation scenario achieves stable performance across the considered window sizes in all four contexts, with MAE not exceeding 8 cm, which is also exhibited by the HGB regressor in the fast-walking mode only, though its achieved MAE remains below 9 cm in the normal walking mode. Among the remaining models, the decision tree and RF regressors achieve similar stable performance across the considered window sizes in all four contexts in this evaluation scenario. Likewise, the K-NN and RBF-SVM models exhibit the same across all four contexts, with the achieved MAE further increasing in the fast-walking mode. The evaluation results of unseen test scenarios further indicate that all considered models perform better in normal-walking mode than in fast-walking mode, with the lowest MAE in the NP context, as observed in seen test evaluations due to better device stability in normal-walking mode than in fast-walking mode.

The evaluation results of various regression models on seen test datasets across four distinct contexts, on the other hand, demonstrate that the HGB regressor achieves the highest performance at smaller window sizes in all four contexts. Conversely, the LR model exhibits commendable performance at larger window sizes, specifically beyond 50 in the NP and FP scenarios. Among the other models, the ANN provides satisfactory results across the considered window sizes in all contexts except FH, achieving an MAE adequately below 2 cm. In contrast, the MAE for RBF-SVM and k-NN models deteriorates in the fast-walking mode under these evaluation conditions. The decision tree and random forest models in such evaluation scenarios achieve comparable performance, with MAE not exceeding 3 cm across the considered window sizes in all contexts. Meanwhile, the elastic net model shows diminished performance in both fast and normal handheld scenarios, with MAE exceeding 4 cm at certain window sizes. Figures 7–10 also demonstrate that all considered models, except the HGB regressor, achieve better results across varied window sizes under the NP evaluation on seen data compared to the other three (i.e., NH, FH, FP), mainly because the device remains more stable in this specific context, attaining less vibration noise in sensor readings. For the same reason, they even perform better in normal-walking mode than in fast-walking mode in such evaluations, as the results show.

Overall, the findings show that no single model performed best in all evaluation settings. HGB Regression achieved the most robust performance in LOOCV evaluation, ANN showed the best generalization performance in the unseen test evaluation, and LR achieved the best single-case result in the seen test evaluation. Among the models considered here, HGB Regression and ANN appear to be the most suitable models for smartphone-sensor-based step length estimation. In contrast, k-NN and RBF SVM showed comparatively higher MAE values, particularly in the unseen test evaluations, indicating weaker generalization ability for this task.

Another crucial point is examining the effect of window size on the MAE achieved by the considered models, since their performances not only differ across window sizes, but

also the best result is not achieved at identical window sizes across four distinct contexts in three evaluation scenarios. Under the LOOCV strategy, the best results are reported at window sizes of 90 for NH, 100 for NP, 90 for FH, and 100 for FP. In the unseen test data-based evaluations, the best results are reported at window sizes of 50 for NH, 70 for NP, 100 for FH, and 70 for FP. In the seen test data-based evaluations, the lowest MAE values are reported at window sizes of 60 for NH, 100 for NP, 50 for FH, and 50 for FP. Those results indicate that no single window size is optimal for all contexts and evaluation scenarios. Instead, the most suitable window size appears to depend on the scenario and the evaluation protocol. Therefore, the choice of window size should be treated as an important experimental parameter.

The predictive performance results reported in Tables 10–12 further indicate that window size has a significant effect on model performance. In the LOOCV and seen test data-based evaluations, larger window sizes such as 90 and 100 often produced better results, possibly because larger windows capture more stable gait-related information. However, in the unseen test evaluation, medium window sizes such as 50, 70, and 80 obtained better results for some models. Therefore, the optimal window size is not fixed and depends on the model, walking condition, smartphone placement, and evaluation scenario. Defining an optimal compromise window size for all four walking contexts may thus entail formulating an optimization problem subject to predictive performance requirements, computational efficiency, walking context, and the evaluation scenario.

Lastly, in terms of achieved MAE, the remarkable performance of the HGB regressor in all four contexts of LOOCV and three of the seen data-based evaluations, that of LR in the seen data-based NP context, and the superiority of the ANN in unseen data-based evaluations, compared to the previously proposed IRT-SD-SLE, is reported in Table 13, which justifies the effectiveness of several regressor models in predicting the step length accurately across diverse gait patterns, walking modes and device postures compared to the traditional signal processing-based peak–valley detection method, which is adopted in IRT-SD-SLE [15].

4.2. Computational Efficiency Analysis

The computational efficiency of the considered models is measured by their training and test/inference times. At first, their training times are interpreted, followed by the same on test/inference time.

4.2.1. Training Time Analysis

The training time comparison shows a clear difference in computational cost among the evaluated models. LR, k-NN, RBF-SVM, DTree, and Els net consistently show very low training times across the reported window sizes. Among these models, k-NN and LR, followed by Els net, are particularly fast to train. DTree and RBF-SVM require slightly more training time than the simplest linear models, but their computational cost remains low in absolute terms. In contrast, ensemble methods like RF and HGB Regressors have substantially longer training times, in the range of hundreds of milliseconds. This indicates a noticeably greater training cost for the ensemble-based models. Among all considered models, the ANN has the longest training time in most of the reported cases, incurring much higher training costs than classical machine learning models. Another trend visible in the training time results is that training time often decreases as the window size increases. Based on the reported values, this pattern is noticeable in all models except k-NN and ANN at certain window sizes. As the window size increases, the number of training instances decreases, which in turn keeps the training time for all classical regressor models except k-NN (i.e., LR, RBF-SVM, Els net, and DTree) and ensemble models (i.e., HGB and RF)

decreasing gradually with increasing window size. k-NN, being a lazy learner, consumes very little training time, irrespective of the number of training instances [54].

4.2.2. Test Time Analysis

The test time results for various models across varied window sizes in different evaluation scenarios reveal that LR, Els net, and DTree consistently offer very low inference times across window sizes in most scenarios. Their relatively low and stable inference times make them computationally attractive when judging from a deployment perspective. Although ANN has the longest training time, it incurs relatively low inference time. This indicates that its main computational burden is concentrated in training rather than inference. In contrast, k-NN shows the opposite behavior. Even though its training time is very small, its test time is higher than that of several other low-complexity models. During training, k-NN stores the instances, which makes its training time very low. But it compares the query instance to all training instances to select the closest neighbors during the test phase, increasing inference time with the number of training instances [54]. Thus, k-NN is less favorable when low inference time is an important requirement. The RBF-SVM shows moderate test time, with a general tendency to decrease as the window size increases, since the number of selected support vectors decreases with decreasing training dataset size. The RF regressor followed by the HGB regressor consumes the largest test time among the considered regressor models. This indicates that such ensemble models incur a high prediction cost compared to the other models.

4.3. Prediction Accuracy vs. Computational Efficiency Trade-Off Analysis

The results on MAE and training and test times for various regressor models reported in the previous section indicate a clear trade-off between prediction accuracy, measured by MAE, and computational efficiency. The key insights about this trade-off are as follows.

- HGB regressor achieves the lowest MAE (i.e., best prediction accuracy) in most of the seen test data and LOOCV-based evaluation scenarios, but its training and inference times are higher than those of classical regressor models like LR, RBF-SVM, Els net, DTree, etc.
- Among the HGB and RF regressors, the former outperforms the latter in terms of achieved MAE and computation efficiency across varied window sizes in all evaluation contexts because the former relies on a histogram, which enables it to speed up processing larger input samples compared to the latter [49].
- ANN acquires the best achievable accuracy (i.e., lowest MAE) across all four contexts in unseen test evaluations, with low prediction cost, but its training time is the highest overall.
- The linear model, like LR, not only has the highest computation efficiency since it incurs relatively lower training and test times among the classical models, but also achieves the lowest MAE in some cases, specifically in the NP context of the seen test evaluations.
- The other classical regressor models, like Els net, RBF-SVM, DTree, etc., offer much lower computational cost, although they do not always yield the lowest MAE (i.e., best accuracy).

These observations suggest that the most suitable model depends on the intended application. If lower MAE is the primary objective in evaluations that ensure generalizability across gait patterns (i.e., LOOCV-based evaluation), then the HGB regressor appears favorable within the present comparison. On the other hand, ANN will be the most favorable model among the considered ones if the lowest MAE achievement in evaluations that ensures generalizability across walking modes and device postures (i.e., unseen test

scenarios) is the primary objective. However, if low training cost and faster prediction are required, LR and Els net may be more practical choices, as both yield lower MAE results than the other models in some cases. Therefore, model suitability depends on the balance between prediction accuracy and computational efficiency.

5. Conclusions and Future Research Scope

This study introduces a machine learning regression-driven SLE framework utilizing a smartphone accelerometer to overcome limitations of traditional signal processing approaches. Eight distinct regressor models, ranging from simple linear regression and k-NN to ensemble models including RF and HGB regressors, as well as a multilayer perceptron-based artificial neural network, are effectively employed within the proposed SLE framework. Comprehensive performance evaluations of these models in terms of MAE for step length prediction across varied window sizes from 40 to 100, within four distinct contexts (NH, NP, FH, FP)—which result from the combination of two walking modes and device postures—were conducted under three evaluation scenarios: using LOOCV, unseen and seen test data. These assessments aimed to determine the models' generalization capabilities across diverse walking modes, device postures, and gait patterns. In addition to MAE, the models were evaluated based on training and testing times to analyze their computational efficiency and explore the trade-off between efficiency and prediction accuracy. A performance comparison between the regressor models and our previously proposed IRT-SD-SLE method, regarding achieved MAE across all four contexts, demonstrates that certain regressor models excelled at specific window sizes, surpassing our earlier method that relied on hand-crafted signal processing. Furthermore, comprehensive and rigorous evaluations of various regressor models across different window sizes and contexts provided several insights, as outlined below, into their performance and applicability in diverse scenarios.

- The performance of various models significantly depends on window size, necessitating further investigation to identify optimal window sizes across different contexts.
- In LOOCV evaluations, which ensure generalization across diverse gait patterns, alongside most of the seen test evaluations, the HGB regressor performed most effectively, achieving the lowest MAE below 1 cm across all contexts, but with lower computational efficiency than other models, except for RF and ANN.
- Among the considered models, the ANN demonstrated the strongest generalization capacity, with the lowest MAE across all four contexts (NH, NP, FH, FP), not exceeding 6.3 cm, accompanied by relatively low inference times in unseen test evaluations. However, its optimal performance comes at the highest training cost.
- The linear model, such as LR, offers the greatest computational efficiency, while sometimes yielding the lowest MAE in specific contexts during seen test evaluations.
- The selection of the most appropriate model in a given scenario depends on balancing prediction accuracy and computational efficiency, due to the inherent trade-off between them.
- The superior device stability and reduced vibrations observed in normal walking mode, compared to fast walking, result in lower MAE for all considered models, including the previously proposed IRT-SD-SLE.

The present study, however, has several limitations: the absence of experimentation with pathological gaits, the selection of an optimal window size for all three evaluation scenarios, the handling of models' accuracy degradation during fast walking, and a practical deployment strategy for computationally intensive ANN models, among others, which provide further scope for research, as delineated below. The experimental design in the present study involves participants with healthy gait walking at normal and fast speeds,

which limits the validity of the proposed SLE framework for pathological gaits and its applicability to gait monitoring in patients with neurological conditions. To address this limitation, we plan to prepare a synthetic dataset generated from simulated pathological gaits to evaluate and validate our proposed framework, followed by evaluation and validation on a dataset comprising real pathological gaits, subject to their availability. We intend to carry this out in the near future. To address the problem of degraded predictive performance, i.e., increasing MAE of all considered models during fast walking due to increased vibrations during such movements, as noted in the MAE results given in the present study, we plan to extend the proposed SLE framework pipeline by integrating advanced adaptive noise filtering techniques during the preprocessing phase to compensate for such dynamic motion artifacts in the near future. The optimal window size selection for all four walking contexts considered in this study may entail formulating an optimization problem subject to predictive performance and computational efficiency requirements, walking context, and the evaluation scenario, requiring further investigation in the coming days. Lastly, inspired by the experimental study on tiny model development by Mihigo et al. in [55], we also envision exploring two architectures: designing an on-device inference architecture based on model quantization/optimization, as well as adopting an edge intelligence-driven client-server architecture, and comparing them in terms of their predictive performance, resource consumption, inference delay, etc., in the near future to devise an optimal solution to a practical deployment strategy for computationally intensive ANN models.

Author Contributions: Conceptualization, P.S.; methodology, R.C., S.M., and P.S.; software, R.C. and P.S.; validation, P.S.; formal analysis, R.C. and P.S.; investigation, P.S.; resources, R.C., S.M. and P.S.; data curation, R.C., S.M. and P.S.; writing—original draft preparation, R.C., S.M., and P.S.; writing—review and editing, P.K.D. and P.S.; visualization, P.K.D. and P.S.; supervision, P.S.; project administration, P.S.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors gratefully acknowledge the facilities and support provided by the Director and all other staff members of the School of Mobile Computing and Communication, Jadavpur University, a Center of Excellence set up under the “University with potential for Excellence” Scheme of the UGC.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xu, H.; Meng, F.; Liu, H.; Shao, H.; Sun, L. An Adaptive Multisource Data Fusion Indoor Positioning Method Based on Collaborative Wi-Fi Fingerprinting and PDR Techniques. *IEEE Sens. J.* **2024**, *24*, 31481–31494. [[CrossRef](#)]
2. Li, J.; Song, Y.; Ma, Z.; Liu, Y.; Chen, C. A Review of Indoor Localization Methods Leveraging Smartphone Sensors and Spatial Context. *Sensors* **2024**, *24*, 6956. [[CrossRef](#)] [[PubMed](#)]
3. Poleur, M.; Markati, T.; Servais, L. The use of digital outcome measures in clinical trials in rare neurological diseases: A systematic literature review. *Orphanet J. Rare Dis.* **2023**, *18*, 224. [[CrossRef](#)] [[PubMed](#)] [[PubMed Central](#)]
4. Hummel, J.; Schwenk, M.; Seebacher, D.; Barzyk, P.; Liepert, J.; Stein, M. Clustering Approaches for Gait Analysis within Neurological Disorders: A Narrative Review. *Digit. Biomark.* **2024**, *8*, 93–101. [[CrossRef](#)] [[PubMed](#)] [[PubMed Central](#)]
5. Ventura, J.D.; Hodges, B.H. Carrying children, groceries, and water across varying terrain: Changes in gait and comfortable walking speed. *Gait Posture* **2023**, *103*, 140–145 [[CrossRef](#)] [[PubMed](#)]
6. Adamson, L.; Vandamme, L.; Prior, T.; Miller, S.C. Running-Related Injury Incidence: Does It Correlate with Kinematic Sub-groups of Runners? A Scoping Review. *Sports Med.* **2024**, *54*, 1163–1178. [[CrossRef](#)] [[PubMed](#)] [[PubMed Central](#)]

7. Abiad, N.A.; Houdry, E.; El Khoury, C.; Renaudin, V.; Robert, T. A method for calculating fall risk parameters from discrete stride time series regardless of sensor placement. *Gait Posture* **2024**, *111*, 182–184. [[CrossRef](#)] [[PubMed](#)]
8. Helal, A.; Moore, S.; Ramachandran, B. Drishti: An integrated navigation system for visually impaired and disabled. In *Proceedings of the 5th International Symposium on Wearable Computers, Zurich, Switzerland, 18–21 October 2001*; IEEE: Piscataway, NJ, USA, 2001; pp. 149–156.
9. Sadhukhan, P.; Das, P.K. Location-Based Services (LBS). In *The Encyclopedia of Human Geography*; Warf, B., Ed.; Springer: Cham, Switzerland, 2024.
10. Sadhukhan, P.; Mukherjee, N.; Das, P.K. Location-based services for smart living in urban areas. In *Precision Positioning with Commercial Smartphones in Urban Environments*; Springer: Cham, Switzerland, 2021; pp. 53–69.
11. European Medical Agency. Opinion on Qualification of Digital Technology-Based Methodologies to Support Approval of Medicinal Products. Available online: https://www.ema.europa.eu/en/documents/other/questions-and-answers-qualification-digital-technology-based-methodologies-support-approval-medicinal-products_en.pdf (accessed on 10 September 2024).
12. Lee, H.J.; Park, J.S.; Yang, H.W.; Shin, J.W.; Han, J.W.; Kim, K.W. A normative study of the gait features measured by a wearable inertia sensor in a healthy old population. *Gait Posture* **2023**, *103*, 32–36. [[CrossRef](#)] [[PubMed](#)]
13. Guimarães, V.; Sousa, I.; Correia, M.V. A Deep Learning Approach for Foot Trajectory Estimation in Gait Analysis Using Inertial Sensors. *Sensors* **2021**, *21*, 7517. [[CrossRef](#)] [[PubMed](#)]
14. Soni, R.; Trapasiya, S. A survey of step length estimation models based on inertial sensors for indoor navigation systems. *Int. J. Commun. Syst.* **2022**, *35*, e5053.
15. Sadhukhan, P.; Gayen, B.; Chowdhury, C.; Mukherjee, N.; Wang, X.; Das, P.K. IRT-SD-SLE: An Improved Real-Time Step Detection and Step Length Estimation Using Smartphone Accelerometer. *IEEE Sens. J.* **2023**, *23*, 30858–30868. [[CrossRef](#)]
16. Sadhukhan, P.; Gayen, B.; Chowdhury, C.; Mukherjee, N.; Wang, X.; Das, P.K. Human Gait Modeling with Step Length Estimation based on Single Foot Mounted Inertial Sensors. *SSRN* **2024**, 4830580.
17. Ghosh, T.; Paul, A.; Sadhukhan, P.; Das, P.K.; Roy, N. Improved Step Length Estimation with IMU Based Gait Analysis Using Sequential Neural Network. *arXiv* **2025**, arXiv:2501.00000.
18. Vandermeeren, S.; Bruneel, H.; Steendam, H. Feature selection for machine learning based step length estimation algorithms. *Sensors* **2020**, *20*, 778. [[CrossRef](#)] [[PubMed](#)]
19. Sui, J.-D.; Chang, T.-S. IMU Based Deep Stride Length Estimation With Self-Supervised Learning. *IEEE Sens. J.* **2021**, *21*, 7380–7387. [[CrossRef](#)]
20. Shi, L.-F.; Zhou, W.; Yan, X.; Shi, Y. Novel Step Detection Algorithm for Smartphone Indoor Localization Based on CEEMDAN-HT. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 8507509. [[CrossRef](#)]
21. Park, J.; Lee, J.H.; Park, C.G. Pedestrian Stride Length Estimation Based on Bidirectional LSTM and CNN Architecture. *IEEE Access* **2024**, *12*, 124718–124728. [[CrossRef](#)]
22. Wang, Q.; Ye, L.; Luo, H.; Men, A.; Zhao, F.; Ou, C. Pedestrian Walking Distance Estimation Based on Smartphone Mode Recognition. *Remote Sens.* **2019**, *11*, 1140. [[CrossRef](#)]
23. Gu, F.; Khoshelham, K.; Yu, C.; Shang, J. Accurate Step Length Estimation for Pedestrian Dead Reckoning Localization Using Stacked Autoencoders. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 2705–2713. [[CrossRef](#)]
24. Ping, Z.; Zhidong, M.; Pengyu, W.; Zhihong, D. Pedestrian Stride-Length Estimation Based on Bidirectional LSTM Network. In *Proceedings of the 2020 Chinese Automation Congress (CAC)*, Shanghai, China, 6–8 November 2020; pp. 3358–3363.
25. Wang, Q.; Ye, L.; Luo, H.; Men, A.; Zhao, F.; Huang, Y. Pedestrian Stride-Length Estimation Based on LSTM and Denoising Autoencoders. *Sensors* **2019**, *19*, 840. [[CrossRef](#)] [[PubMed](#)]
26. Hannink, J.; Kautz, T.; Pasluosta, C.F.; Gaßmann, K.-G.; Klucken, J.; Eskofier, B.M. Sensor-Based Gait Parameter Extraction with Deep Convolutional Neural Networks. *IEEE J. Biomed. Health Inform.* **2017**, *21*, 85–93. [[CrossRef](#)] [[PubMed](#)]
27. Elhousni, M.; Huang, X. Pedestrian Tracking with Gated Recurrent Units and Attention Mechanisms. In *Proceedings of the 2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, Seville, Spain, 10–21 October 2020; pp. 1–5.
28. Elhoushi, M.; Georgy, J.; Noureldin, A.; Korenberg, M.J. A Survey on Approaches of Motion Mode Recognition Using Sensors. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 1662–1686. [[CrossRef](#)]
29. Vezočnik, M.; Juric, M.B. Adaptive inertial sensor-based step length estimation model. *Sensors* **2022**, *22*, 9452. [[CrossRef](#)] [[PubMed](#)]
30. Chen, B.; Zhang, R.; Wang, S.; Zhang, L.; Liu, Y. Deep-Learning-Based Inertial Odometry for Pedestrian Tracking Using Attention Mechanism and Res2Net Module. *IEEE Sens. Lett.* **2022**, *6*, 6003804. [[CrossRef](#)]
31. Verbiest, J.R.; Bonnechère, B.; Saeys, W.; Van de Walle, P.; Truijen, S.; Meyns, P. Gait stride length estimation using embedded machine learning. *Sensors* **2023**, *23*, 7166. [[CrossRef](#)] [[PubMed](#)]
32. Zadka, A.; Rabin, N.; Gazit, E.; Mirelman, A.; Nieuwboer, A.; Rochester, L.; Din, S.; Pelosin, E.; Avanzino, L.; Bloem, B.; et al. A wearable sensor and machine learning estimate step length in older adults and patients with neurological disorders. *npj Digit. Med.* **2024**, *7*, 35. [[CrossRef](#)]

33. Vezočník, M.; Kamnik, R.; Juric, M.B. Inertial Sensor-Based Step Length Estimation Model by Means of Principal Component Analysis. *Sensors* **2021**, *21*, 3527. [CrossRef] [PubMed]
34. Abiad, N.A.; Kone, Y.; Renaudin, V.; Robert, T. Smartstep: A Robust STEP Detection Method Based on SMARTphone Inertial Signals Driven by Gait Learning. *IEEE Sens. J.* **2022**, *22*, 12288–12297. [CrossRef]
35. Shi, L.-F.; Jing, F.; Wu, X.-H.; Shi, Y. Efficient Step Detection and Step Length Estimation Method Based on Smartphones. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 9525611. [CrossRef]
36. Mazumder, S.; Sadhukhan, P. Orientation-Invariant Step Length Estimation for Arbitrary Indoor Trajectories. In Proceedings of the 2025 2nd International Conference on Emerging Trends in Electronic Devices and Computational Techniques (EDCT), Kolkata, India, 20–21 November 2025; pp. 1–4.
37. Xu, K.; Yu, W.; Yu, S.; Zheng, M.; Zhang, H. The Detection of Gait Events Based on Smartphones and Deep Learning. *Bioengineering* **2025**, *12*, 491. [CrossRef] [PubMed]
38. Liang, S.G.; Chung, H.Y.; Chu, K.W.; Gao, Y.H.; Lau, F.Y.; Lai, W.I.; Fong, G.C.-H.; Kwong, P.W.-H.; Lam, F.M.H. Validity and Reliability of a Smartphone-Based Gait Assessment in Measuring Temporal Gait Parameters: Challenges and Recommendations. *Biosensors* **2025**, *15*, 397. [CrossRef] [PubMed]
39. Butterworth, S. On the Theory of Filter Amplifiers. *Wirel. Eng.* **1930**, *7*, 536–541.
40. “Butterworth Filter.” *scipy.signal.butter* 1.17.0 Documentation. Available online: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.butter.html> (accessed on 5 September 2025).
41. Nelder, J.A.; Wedderburn, R.W.M. Generalized Linear Models. *J. R. Stat. Soc. Ser. A Gen.* **1972**, *135*, 370–384. [CrossRef]
42. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
43. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
44. Schölkopf, B.; Smola, A.J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*; MIT Press: Cambridge, MA, USA, 2002.
45. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. Stat. Softw.* **2010**, *33*, 1–24. [CrossRef] [PubMed]
46. Murthy, S.K. Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. *Data Min. Knowl. Discov.* **1998**, *2*, 345–389. [CrossRef]
47. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
48. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
49. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS’17)*; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 3149–3157.
50. Caudill, M. Neural network training tips and techniques. *AI Expert*, 1 January 1991; pp. 56–61.
51. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
52. “GridsearchCV.” *Scikit-Learn* 1.8.0 Documentation. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (accessed on 24 October 2025).
53. Simon, S.; Kolyada, N.; Akiki, C.; Potthast, M.; Stein, B.; Siegmund, N. Exploring Hyperparameter Usage and Tuning in Machine Learning Research. In Proceedings of the 2023 IEEE/ACM 2nd International Conference on AI Engineering—Software Engineering for AI (CAIN), Melbourne, Australia, 15–16 May 2023; pp. 68–79. [CrossRef]
54. Kotsiantis, S.B. Supervised Machine Learning: A Review of Classification Techniques. *Informatica* **2007**, *31*, 249–268.
55. Mihigo, I.N.; Zennaro, M.; Uwitonze, A.; Rwigema, J.; Rovai, M. On-Device IoT-Based Predictive Maintenance Analytics Model: Comparing TinyLSTM and TinyModel from Edge Impulse. *Sensors* **2022**, *22*, 5174. [CrossRef] [PubMed]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.