Deep Learning for Multiple Sclerosis Differentiation Using Multi-Stride Dynamics in Gait

Rachneet Kaur, Joshua Levy, Robert W. Motl, Richard Sowers, *Member, IEEE*, and Manuel E. Hernandez

Abstract*—*Objective: **Multiple sclerosis (MS) is a chronic neurological condition of the central nervous system leading to various physical, mental and psychiatric complexities. Mobility limitations are amongst the most frequent and early markers of MS. We evaluated the effectiveness of a DeepMS2G (deep learning (DL) for MS differentiation using multi-stride dynamics in gait) framework, which is a DL-based methodology to classify multi-stride sequences of persons with MS (PwMS) from healthy controls (HC), in order to generalize over newer walking tasks and subjects. *Methods:* We collected single-task *Walking* and dual-task *Walking-while-Talking* gait data using an instrumented treadmill from a balanced collection of 20 HC and 20 PwMS. We utilized domain knowledge-based spatiotemporal and kinetic gait features along with two normalization schemes, namely standard size-based and multiple regression normalization strategies. To differentiate between multi-stride sequences of HC and PwMS, we compared 16 traditional machine learning and DL algorithms. Further, we studied the interpretability of our highest-performing models; and discussed the association between the lower extremity function of participants and our model predictions. *Results:* We observed that residual neural network (ResNet) based models with regression-based normalization were the top performers across both task and subject generalization classification designs. Considering regression-based normalization, a multi-scale ResNet attained a subject classification accuracy and F$_1$-score of 1.0 when generalizing from single-task *Walking* to dual-task *Walking-while-Talking*; and a ResNet resulted in the top subject-wise accuracy and F$_1$ of 0.83 and 0.81 (resp.), when generalizing over unseen participants. *Conclusion:* We used advanced DL and dynamics across domain knowledge-based spatiotemporal and kinetic gait parameters to successfully classify MS gait across distinct walking trials and unseen participants. *Significance:* Our proposed DL algorithms might contribute to efforts to automate MS diagnoses.**

***Index Terms*—Deep learning, gait, multiple sclerosis.**

I. Introduction

Multiple sclerosis (MS) is an immune-mediated, neurodegenerative disease that affects approximately 1 million people in the United States and more than 2.5 million globally [1], [2], with a shift in peak prevalence to adults 55–64 years of age [3]. MS can be immensely heterogeneous; persons with MS (PwMS) may suffer from extremely mild to severe muscle immobility, speech and vision complications, and memory issues [4]. Gait and balance dysfunction are common symptoms in PwMS, with nearly 85% of PwMS describing gait disorders as a major complication [5] and roughly 50% of patients needing walking assistance within 15 years of MS onset [6]. Gait performance declines have been observed in PwMS, particularly as disability increases [7], [8], [9], [10]. Past studies have found reduced gait speed, shorter steps, extended stride time, wider base of support, reduced single support phase, and a prolonged double support phase in PwMS compared to controls [7], [8]. However, most gait-based methods for identifying MS have relied upon traditional statistical techniques to examine differences in spatiotemporal features and correlations with disability. Compared to statistical testing that analyze features individually, machine learning (ML) models are capable of utilizing linear and nonlinear combinations of spatiotemporal and kinetic gait features to potentially improve MS gait identification.

Given the increased access to objective gait data from wearable technologies or traditional gait labs, supervised ML methodologies have been increasingly used in human gait analysis across neurological populations, including MS [11], [12], [13]. In particular, ML methods like random forest and artificial neural networks have been used to identify gait changes in Parkinson's disease in [11], [12], whereas [13] focused on MS-related changes. With the increasing successes of deep learning (DL) across domains, recent works [14], [15] compared several ML models with the long short-term memory (LSTM) DL approach to distinguish between low and high fall risk in neurological gait. The LSTM model outperformed all traditional ML methods (classification accuracy: 0.94 (LSTM) vs. 0.88 (top

Manuscript received 4 March 2022; revised 23 September 2022 and 12 December 2022; accepted 9 January 2023. Date of publication 20 January 2023; date of current version 20 June 2023. *(Corresponding author: Rachneet Kaur.)*

Rachneet Kaur is with the Department of Industrial and Enterprise Systems Engineering, University of Illinois at Urbana Champaign, Champaign 61801 USA (e-mail: rk4@illinois.edu).

Joshua Levy is with the Department of Computer Science, University of Illinois at Urbana-Champaign, USA.

Robert W. Motl is with the Department of Kinesiology and Nutrition, University of Illinois Chicago, USA.

Richard Sowers is with the Department of Industrial and Enterprise Systems Engineering and the Department of Mathematics, University of Illinois at Urbana Champaign, USA.

Manuel E. Hernandez is with the Department of Translational and Biomedical Sciences, University of Illinois at Urbana Champaign, USA (e-mail: mhernand@illinois.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TBME.2023.3238680, provided by the authors.

Digital Object Identifier 10.1109/TBME.2023.3238680

ML model) in [14] and 0.86 (LSTM) vs. 0.73 (top ML model) in [15]), showcasing the potential of DL in human gait analysis. See [16], [17] for a more detailed comparison of ML and DL approaches in gait analysis.

This work attempts to examine MS related changes in spatiotemporal and kinetic gait features across multiple strides; and evaluate the effectiveness of **deep** learning for **MS** differentiation using **m**ulti-**s**tride dynamics in **g**ait (**DeepMS2G**). Specifically, we propose a DL-based methodology to classify multi-stride sequences of PwMS from healthy controls (HC), so as to generalize across different walking tasks and subjects. Building upon prior work examining MS classification using traditional ML frameworks on individual strides [13], we categorized PwMS using the following 2 classification designs:

a) *Task generalization* demonstrating the generalization over different tasks. Specifically, we train binary (healthy vs. MS) supervised classifiers on *Walking* (W) trials and test them on *Walking-while-Talking* (WT) trials, to examine how findings from data collected in a clinic or lab may generalize to more realistic gait tasks.

b) *Subject generalization* establishing the generality over newer subjects. Specifically, we train binary classifiers on some subjects and apply them to an independent set of withheld test subjects.

Concretely, our contributions are as follows:

- We presented a DL approach to differentiate MS related changes from controls using multi-stride dynamics in spatiotemporal and kinetic gait features. Of particular novelty is our focus on MS.
- We utilized multi-stride dynamics from 21 extracted kinematic and kinetic gait features.
- We benchmark the comparative performance of 16 diverse ML and DL models for MS differentiation across two classification frameworks, i.e. task and subject generalization and two feature scaling strategies, i.e., body size- and multiple regression-based normalization.
- We investigated the explainability of our top-performing algorithms via ablation study on gait features and feature importance. Moreover, we discussed the association between the lower extremity function of participants and our model predictions. This post hoc analysis of DL models was absent in previous analogous studies.

II. Related Works

Neurological gait disorders like MS are characterized by reduced mobility, abnormal gait mechanics, poor balance and muscle weakness, as well as cognitive and autonomic dysfunction [18], [19]. These symptoms typically lead to fatigue and physical inactivity and consequently increase the risk of development of secondary diseases. Several works on movement analysis have utilized wearable inertial measurement unit sensors [20], electromyography (EMG) [21], and motion capture systems [22] to predict neuromuscular changes in neurological gait. Past studies on gait-based methods for identifying MS have relied upon statistical significance tests such as t-test, and ANOVA (analysis of variance) to examine differences in average and variability of spatiotemporal features, and correlations with neurological impairment assessed by Kurtzke's Expanded Disability Status Scale [9], [23], [24], [25]. Compared to the statistical tests that analyze features individually, ML and DL models are capable of determining multivariate discriminants by taking into account multiple features. Further, these algorithms can also produce non-linear decision boundaries, potentially leading to superior accuracies. Recently, several studies have focused on traditional ML to classify gait patterns in PwMS [13], [26]. Additionally, authors in [15] used a long short-term memory model to distinguish between low and high fall risk in PwMS via accelerometer sensors. We utilized data driven DL for classification of multi-stride sequences of PwMS from HC utilizing domain knowledge-based spatiotemporal and kinetic gait features. Note that in comparison to [15], we studied a different classification task and used a separate cohort.

In this paper, we utilized spatiotemporal and kinetic gait parameters as input features, which contain valuable domain knowledge with the potential to improve classification performance. Further, since the effect of gait normalization is seemingly unexplored in the existing MS literature, we compared the classification ability of all models with standard size-based and multiple regression normalization schemes, first explored in [12], [27], across both the studied task- and subject generalization model designs. Moreover, we explored the explainability of our top-performing algorithms; and discussed the association between the lower extremity function of participants and our model predictions.

III. Design of Experiments: Subjects and Setup

A. Study Participants

The sample consisted of 40 subjects; 20 PwMS (age: 61.05 ± 6.87 years $[49–75$ years$]$, male/female: $5/15$) and 20 age, weight, height and gender-matched HC (age: 61.2 ± 5.87 years $[48–68$ years$]$, male/female: $5/15$) from the local community. Our inclusion criteria ensured all participants were medically stable, i.e., a score of above 18 on the telephone interview for cognitive status [28], with no recent lower limb injury; further, subjects were right-side dominant and had corrected to normal vision. All included PwMS were relapse-free for the past month, had mild to moderate disability, i.e., 4.3 ± 1.62 $[1.0–6.0]$ on the Kurtzke's Expanded Disability Status Scale (EDSS) [29], and had no other cognitive disorder that may additionally influence their body balance. Note that 2 HC and 3 PwMS were excluded for holding the handrails while walking on the treadmill and thus biasing their force readings. Also, we separately reserved a group of 30 additional HC (age: 67.6 ± 10.34 years $[50–87$ years$]$, male/female: $9/21$) to normalize our extracted gait features (see IV-B for further details). All participants signed an informed consent (Protocol No. 15674) prior to experimental trials.

B. Experimental Paradigm

We used a C-Mill, Motekforce Link instrumented treadmill for participants to walk at their self-paced speed during all experimental trials. The treadmill had a built-in force plate supporting kinetic data acquisition, specifically, allowing for vertical ground reaction forces to be collected. Each participant completed a 75 seconds trial at their self-selected pace under 2 configurations, i.e., single-task paradigm W and dual-task condition WT; participants recited alternate letters of the alphabet while walking for WT trials. All participants were instructed to equally prioritize their attention between gait motion and the cognitive exercise during WT trials. Past studies have demonstrated differentiation between the single- and dual-task

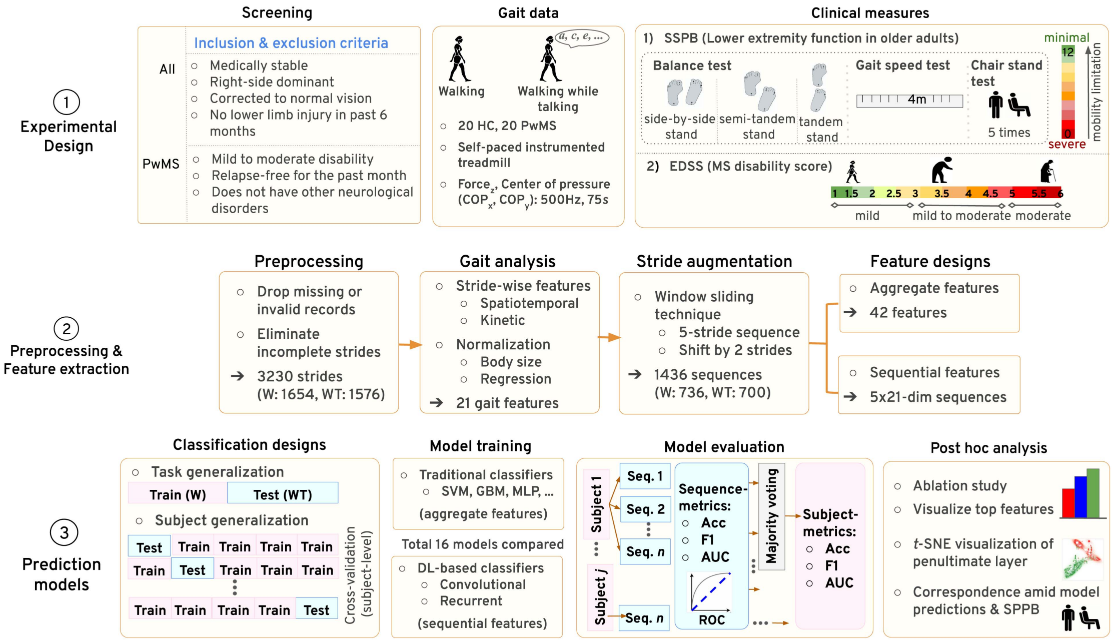

Fig. 1. Workflow pipeline. The proposed **DeepMS2G** (**deep** learning for **MS** differentiation using **m**ulti-**s**tride dynamics in **g**ait) framework.

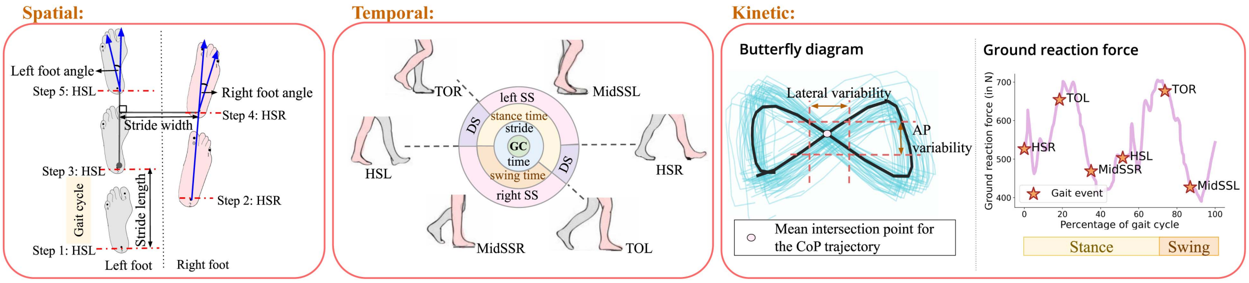

Fig. 2. Gait features. Left: Spatial features, namely stride width, length and foot progression angles, Middle: Temporal features, namely stride, stance, swing, single support (SS) and double support (DS) times, Right: Kinetic features, namely butterfly diagram-based variability and forces. GC is gait cycle, AP is anterior-posterior, and CoP is center of pressure. See [13] for detailed definitions of features.

designs, where WT (in comparison to W) illustrated more resemblance to real life daily gait in middle-aged to older adults [30]. Throughout each walk, the built-in treadmill software recorded 1) position coordinates and time stamps for each gait event, such as left and right heel strike, using a single force plate, 2) ground reaction forces, 3) treadmill speed, and 4) center of pressure position coordinates at 500 Hz.

IV. Gait Feature Extraction and Designs

Our gait data analysis pipeline is illustrated in Fig. 1.

A. Gait Terminology and Feature Extraction

A stride or gait cycle has the following phases (in order), HSR: heel strike right, TOL: toe-off left, MidSSR: midstance right, HSL: heel strike left, TOR: toe-off right, MidSSL: midstance left, and subsequent HSR beginning the next stride. We extracted 21 characteristic spatiotemporal and kinetic features across strides from the raw gait data to comprehend distinguishing patterns between HC and PwMS gait. See Fig. 2. The extracted features can be organized in following 4 categories:

- Spatial: 4 spatial gait features, i.e., stride width, stride length and the left and right foot progression angles [31], were extracted for each stride. Fig. 2 (left) diagrammatically summarizes the definition for these features on an overground view of the gait patterns. See [13] for detailed definitions of features.
- Temporal: 7 temporal features, i.e., swing time, stance time, stride time, supporting (right single, initial double and terminal double) times and cadence, were extracted for each stride. Fig. 2 (middle) illustrates these features on a sagittal plane view of a stride starting at HSL.
- Spatiotemporal: 2 spatiotemporal markers, i.e., 1) stride speed defined as the ratio of stride length and stride time, and 2) walk ratio defined as the ratio of stride length to the count of strides covered in a minute, were extracted for each stride.
- Kinetic: 8 kinetic features, i.e., 6 forces, at each of the 6 gait events, and 2 butterfly diagram-based features, were extracted for each stride. A butterfly diagram [32] defines the recurrent center of pressure trajectory for

TABLE I
BODY SIZE-BASED NORMALIZATION

Raw gait parameter	Dimensionless quantity
Length: $L \in \{$stride length, stride width$\}$	$\widetilde{L} = {}^{L}/_{h}$
Time: $T \in \{$stride time, stance time, swing time, supporting times$\}$	$\widetilde{T} = {}^{T}/\sqrt{h/g}$
Force (F_z^e): 6 forces, one at each gait event e	$\widetilde{F_z^e} = {}^{F_z^e}/_{wg}$
Cadence: C	$\widetilde{C} = {}^{C}/60\sqrt{g/h}$
Stride speed: SS	$\widetilde{SS} = {}^{SS}/\sqrt{gh}$
Angle: $\theta \in \{\theta_L, \theta_R\}$	$\widetilde{\theta} = \theta$
Walk ratio: W	$\widetilde{W} = {}^{W}/\frac{h}{60\sqrt{g/h}}$
center of pressure: $P \in \{\beta_L, \alpha_L\}$	$\widetilde{P} = {}^{P}/S_{size}$

several strides throughout a participant's walking trial. We extracted 2 characteristic gait features from the butterfly diagram, namely, 1) lateral shift in the intersection point of the center of pressure trajectory, and 2) lateral squared deviation from the average intersection point for a trial. See Fig. 2 (right).

After eliminating nonconsecutive strides and those with missing or invalid gait events, we obtained 1654 (HC/PwMS: 905/749) and 1576 (HC/PwMS: 878/698) strides from W and WT trials (resp.), across 18 HC and 17 PwMS. Deriving these multiple samples per subject's walk significantly augmented as well as introduced variations to our data.

B. Data Normalization

Similar to prior work on PwMS [13] and other neurological disorders [12], [27], we compared the following 2 normalization approaches to reduce the intrinsic bias of our extracted gait features on the demographics of the subject and thus improve the MS gait identification accuracy:

- Body size-based normalization (*Size-N*): Gait features were normalized to dimensionless quantities via division by their corresponding dimension-matched body size-based scaling factors [33]. Denoting the body weight (in kg), height (m), shoe size (m) and acceleration of gravity (9.81 m/s^2) by w, h, S_{size} and g (resp.), Table I summarizes the size-normalized or *size-N* gait features.
- Multiple regression-based normalization (*Regress-N*): We regressed the gait features of normative walking data from 30 additional HC (age: 67.6 ± 10.34 years $[50\text{–}87$ years$]$, male/female: $9/21$) on multiple demographic characteristics and used these as baselines to normalize our extracted gait features. We derived the same 21 gait features from a total of 3923 valid strides obtained from our 30 additional HC. A regression model, which minimized the Tukey biweight loss of standard Gaussian residual errors, was fitted to each gait feature. In this regression, independent variables were the demographics (weight, height, gender and age); and dependent variables were subject-wise averaged gait feature values (as defined earlier in Section IV-A). Note that we only used these 30 additional HC that were not part of the main study (Section III-A). Gait features from both trials (W, and WT) of the main study subjects were then normalized to dimensionless quantities, where their predicted values were obtained via their corresponding fit and subject demographics.

C. Stride Augmentation

Building upon past work on fall risk assessment [14], we followed the moving window method to assemble the extracted gait features from 5 time-consecutive strides, creating a 5 × 21-dimensional sequence (*data sample*) with 21 features (*time series*) over 5 temporally ordered strides (*time steps*). Subsequently, we moved our window by 2 strides to devise the next 5-stride data sample. Thus we derived numerous multi-stride samples per subject, each capturing the gait variability and dynamics across 5 heterogeneous strides. This way, we substantially augmented and introduced variations to our original subject-level data in Section III-A. This data augmentation approach might assist in the generality and training process of our complex DL models. Overall, we formed 736 (HC: 416, PwMS: 320) and 700 (HC: 399, PwMS: 301) 5-stride sequences from W and WT trials (resp.), across 35 subjects.

D. Feature Designs

Next, we used our derived 5 × 21-dimensional samples to design 1D aggregated gait features vector and 2D sequential data, suited for our traditional ML- and DL-classifiers (resp.).

- Aggregated features: We used mean and standard deviation to aggregate our 5 × 21, i.e., 2D, sequences along the time dimension and construct a 1D feature vector of length 42, which is the expected input for any classical ML model like decision tree, etc. Thus we compiled a dataset of 1436 data samples across W and WT trials with 42 average- and deviation-based features per sample.
- Sequential features: We directly used the extracted 2D-sequences (5 × 21) as the input for all our convolutional as well as recurrent DL models. This 2D data encompasses both domain-knowledge along with temporal variations in subject's gait and further, did not risk losing information during aggregation. Overall, our input for DL models was 1436 samples, each consisting of a 21-channel sequence, with spatiotemporal and kinetic gait parameters, over 5 consecutive time steps, capturing possible dynamics in the gait data.

V. CLASSIFICATION AND EVALUATION

We examined binary classification to differentiate between 5-stride sequences of HC and PwMS across task- and subject generalization frameworks. Overall, we compared 16 models (see Section V-A); in particular, 9 traditional ML algorithms, 4 convolutional and 3 recurrent DL architectures, across both classification frameworks, with corresponding model training and evaluation details in Sections V-B and V-C (resp.). For task generalization, all models were trained on 736 5-stride sequences across 35 subjects in W trials and tested to categorize 700 sequences of the same subjects in WT trials. Given a limited dataset with 35 subjects, we used 5-fold cross-validation for

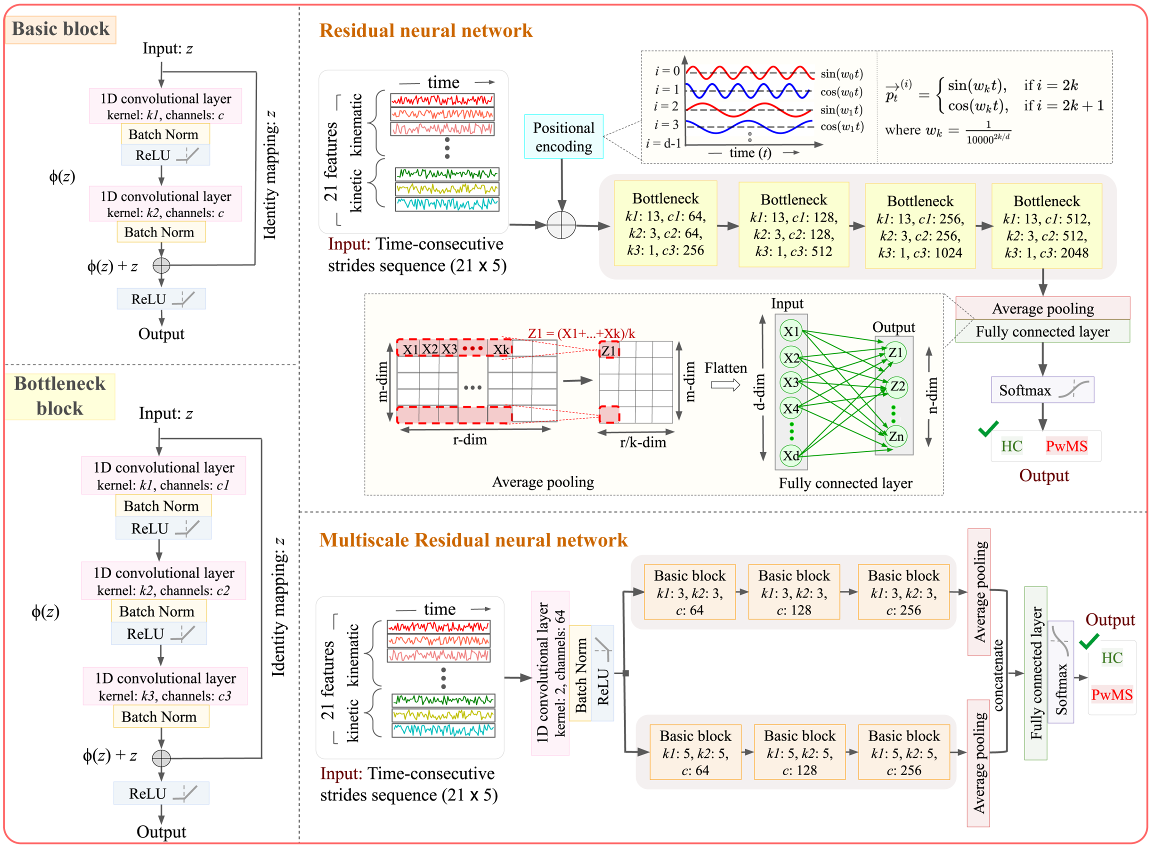

Fig. 3. Convolutional architectures. Top left: Basic block, Bottom left: Bottleneck block, Top right: ResNet, Bottom right: MSResNet. Note: ⊕ denotes element-wise addition in the basic and bottleneck residual blocks.

subject generalization design. Further, all models were compared across both *size-N* and *regress-N* normalized features. All features were Z-score normalized before inputting them to the model.

A. Classification Models

Firstly, we examined 9 traditional ML algorithms: logistic regression (LR), support vector machine with linear (LSVM) and radial basis function (RBF SVM) kernels, decision tree (DT), random forest (RF), adaptive boosting (AdaBoost), eXtreme gradient boosting (XGBoost), gradient boosting machine (GBM) and multilayer perceptron (MLP) [34]. We used the aggregated gait features as input for these classical ML algorithms. Next, we compared the following 4 convolutional DL models (see Sections V-A1 to V-A4) and 3 recurrent models (Sections V-A5 to V-A7); for these algorithms, a sequence of 5 consecutive strides was used directly as the model's input. These algorithms have been previously used for vision-based gait analysis in our past work [35].

1) 1D Convolutional Neural Network (CNN): Our CNN architecture consisted of multiple convolutional blocks where each block was composed of a 1D convolutional layer succeeded by a batch normalization layer, non-linear activation function, dropout layer [36] and a pooling layer. The convolution function hierarchically extracted low-level features from the input data in the initial few convolutional layers to more complex high-level characteristics as subsequent layers are applied in the architecture. We experimented with several activation functions to introduce non-linearity into our convolutional layer output neurons, including a rectified linear unit (ReLU). We also explored dropout layers to randomly disable neurons and their corresponding connections to avoid over-fitting during the training process. The output was then passed through multiple feed forward layers and finally, our final linear layer yielded a vector of length 2.

In contrast to recurrent DL models with an inherent sense of sequential processing for temporal data, CNNs (where the entire sequence is fed at once) may not necessarily handle strides within a multi-stride sequence relative to their positional order. Consequently, we used the sinusoidal *positional encoding* [37] to explicitly add this information to the input.

2) Residual Neural Network (ResNet): ResNets learn residual functions relative to the layer inputs and thereby, assist in the training of deeper models [38]. The fundamental units for our ResNet architecture were 2 types of residual blocks, namely, basic and bottleneck blocks. A basic (or bottleneck) block consist of 2 (or 3) 1D convolutional layers, batch normalization and ReLU non-linearity; the last layer's activation function was used following the addition of the learnt residual mapping with the input. Similar to Section V-A1, we also experimented with using a *sine-cosine* positional encoding to augment order information to our input. Fig. 3 shows a sample ResNet architecture (top right) along with the designs for basic (top left) and bottleneck (bottom left) residual blocks.

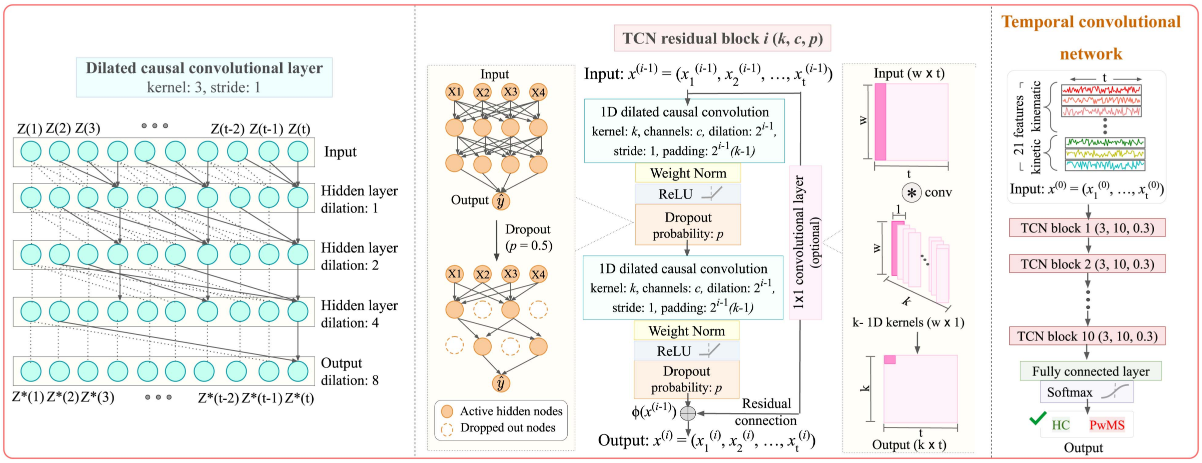
Dilated causal convolutional layer
kernel: 3, stride: 1
Z(1) Z(2) Z(3) • • • Z(t-2) Z(t-1) Z(t)
Input
Hidden layer dilation: 1
Hidden layer dilation: 2
Hidden layer dilation: 4
Output dilation: 8
Z*(1) Z*(2) Z*(3) • • • Z*(t-2) Z*(t-1) Z*(t)
Input
X1 X2 X3 X4
Output ŷ
Dropout (p = 0.5)
Active hidden nodes
Dropped out nodes
TCN residual block i (k, c, p)
Input: x^(i-1) = (x_1^(i-1), x_2^(i-1), …, x_t^(i-1))
1D dilated causal convolution kernel: k, channels: c, dilation: 2^(i-1), stride: 1, padding: 2^(i-1)(k-1)
Weight Norm
ReLU
Dropout probability: p
1x1 convolutional layer (optional)
ϕ(x^(i-1))
Residual connection
Output: x^(i) = (x_1^(i), x_2^(i), …, x_t^(i))
Input (w x t)
w
t
conv
k- 1D kernels (w x 1)
k
Output (k x t)
Temporal convolutional network
21 features
kinetic kinematic
Input: x^(0) = (x_1^(0), …, x_t^(0))
TCN block 1 (3, 10, 0.3)
TCN block 2 (3, 10, 0.3)
TCN block 10 (3, 10, 0.3)
Fully connected layer
Softmax
HC PwMS
Output

Fig. 4. TCN architecture. Left: A dilated causal convolution with dilation factors d = 1, 2, 4, 8 and convolution kernel size of 3, convolution stride of 1, where $Z(1), \ldots, Z(t)$ being the input and $Z^*(1), \ldots, Z^*(t)$ being the output; Middle: A single TCN residual block, with $x^{(i-1)}$ being the input and $x^{(i)}$ as the output of the i-th TCN block; Right: A TCN of 10 blocks, connected with a fully connected end layer with softmax activation function, to generate the classification probabilities.

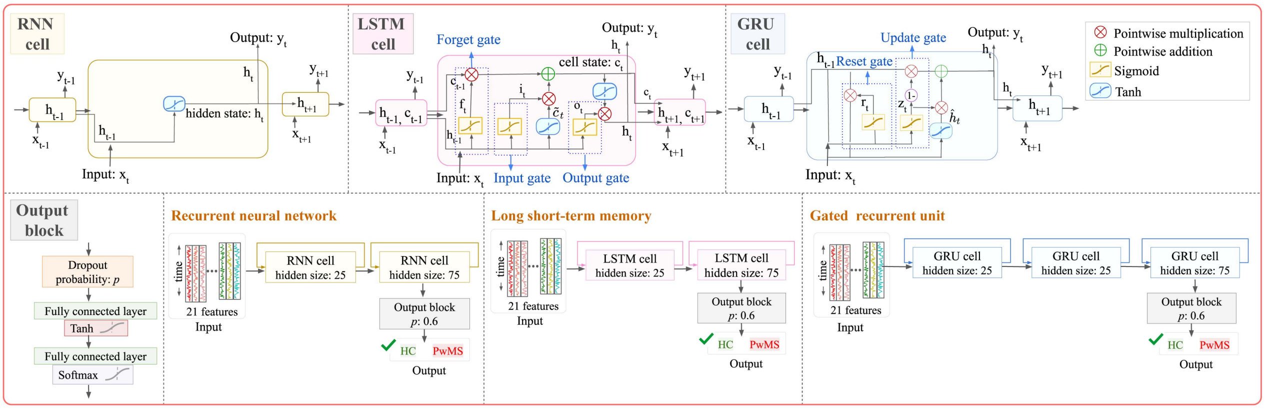

Fig. 5. Recurrent architectures. Top: A single RNN (left), LSTM (middle) and GRU (right) cell with input x_t, hidden state h_t, cell internal state c_t, and output y_t. Bottom: A cascade of layers of RNN (left), LSTM (middle) and GRU (right) cells, connected with a linear end layer, with softmax activation function, to generate the classification probabilities. LSTM cell: Forget gate f_t determines the information to discard from the cell state c_{t-1} by looking at the current input x_t and prior hidden state h_{t-1}. Input gate i_t decides the values to update and the corresponding update to cell state is given by $\tilde{c}_t$. Finally, output gate o_t decides the portions of cell state to output. GRU cell: Update gate z_t selects the information to add and discard in the hidden state, and reset gate r_t determines on how much prior information to forget based on the current input x_t and past hidden state h_{t-1}. The updated hidden state is given by h_t.

3) Multi-Scale Residual Neural Network (MSResNet): Often, utilizing a fixed single-scale convolutional kernel size to extract features from only one scale may not be optimal. Consequently, we experimented with the multi-scale kernel-based ResNet architecture [39] to derive features from multiple scales. The extracted features from the initial convolutional block were sent through 2 branches of 3 basic blocks with {64, 128, 256} filters, where convolutional kernels in the 2 branches were fixed to be 3 and 5 (resp.). Next, these CNN-extracted multi-scale features were concatenated to a single vector. This vector was fed as input to a dense network with 2 output neurons (one for each class: HC and PwMS). Fig. 3 depicts our MSResNet architecture (bottom right).

4) Temporal Convolutional Network (TCN): TCN [40] utilizes residual connections as well as dilated causal convolutions, where dilations enable the model to look quite far back in the past while making predictions and causality ensures no future data leaks to the past. Fig. 4 visually details the TCN architecture consisting of 10 (hyperparameter) TCN residual blocks on the right, with the corresponding structure of a single TCN block in the middle and dilated causal convolution framework on the left. Note that each TCN block consists of a weight normalization layer (see [41] for details).

5) Vanilla Recurrent Neural Network (RNN): RNNs intrinsically integrate the sequential order of strides as internal memory in their backbone architecture. Fig. 5 schematically details a single RNN cell at the top left with input (x_t), hidden (h_t) and output (y_t) states and a sample RNN architecture at the bottom left.

6) Long Short-Term Memory (LSTM): LSTM [42] resolves the *vanishing gradient problem* that is existent in vanilla RNNs when dealing with longer sequences, given its feedback loop structure. A single LSTM unit, as depicted in Fig. 5, utilizes a cell state and input, forget and output gates, to either include or eliminate data to the cell state. We experimented with both uni- and bi-directional LSTM layers.

7) Gated Recurrent Unit (GRU): Similar to LSTM, GRU [43] also utilizes 2 gates, namely, reset and update gates to handle the *vanishing gradient problem* in recurrent networks. Our GRU model was a stack of n (hyperparameter) uni- or bi-directional GRU layers, where each layer i output a sequence of hidden size s_i (hyperparameter) features. The features from the n-th layer at the last time step were followed with a dense network to output class probabilities (Fig. 5).

B. Model Training

To prevent information leakage, we ensured that no single subject had its multi-stride sequences split between training and validation folds. All computations were implemented on an NVIDIA GPU (12 GB Tesla P100) using PyTorch v1.7.0 DL platform in Python 3.6. In all classifiers, we set a fixed random seed for reproducible results. We processed our data in batches of 128 samples each and randomly shuffled training samples at every epoch to reduce bias. We tried several optimization algorithms, namely, stochastic gradient descent with and without momentum, root mean square propagation (RMSprop), adaptive moment estimation (Adam), and Adam with decoupled weight decay (AdamW), each with different learning rate schedules as well as weight decay [44]. In addition to weight decay and early stopping (with $patience$ (hyperparameter) epochs), using dropout between network layers also helped prevent over-fitting in our models. To manage the possible disparity in scales of the processed model features, we tried layer normalization [45] to normalize each feature to zero mean and unit variance. A thorough experimental hyperparameter search was performed on the validation set to determine optimal framework for each learning classifier.

C. Evaluation Details

For evaluating our task generalization classifiers, we used the test set metrics, namely, precision (P), recall (R), accuracy (A), F_1 score (F_1) and area under receiver operating characteristic curve (AUC); for subject generalization, we used the mean and standard deviation in cross-validation metrics. All models were evaluated at 2 categorizations, namely, 5-stride sequence- and subject-level; majority voting was used to classify subjects as HC or PwMS. Thus a correctly classified subject's walk had a majority of multi-stride sequences accurately detected as of the appropriate cohort. We denote the sequence and subject-level evaluation metrics with seq (i.e. P_{seq}, R_{seq}, A_{seq}, F_{1seq}, AUC_{seq}) and sub (i.e. P_{sub}, R_{sub}, A_{sub}, F_{1sub}, AUC_{sub}) in the sub script (resp.). Further, for all DL models, we monitored learning curves for convergence of training accuracy and cross entropy loss metrics across epochs.

VI. Experimental Results

In general, MS subjects had a broader stride width and a shortened stride length; and additionally, a reduced cadence, speed and single support time along with a prolonged double support, stance and stride times. Fig. 6 shows the averaged gait cycle waveforms for HC and PwMS in the W trial. A similar pattern was observed in trial WT as well. We notice no statistically significant differences between the waveforms of the two groups; this significant overlap between the waveforms of healthy and MS groups depicts the challenges of classifying between the healthy and MS gait. Further, the spatiotemporal differences seen in the plot are potentially driven by the changes in the step length, i.e., we see a bigger peak with a longer step or in other words, steps drive the peaks for the gait cycle waveforms. These observations are indeed aligned with the past findings regarding gait changes in PwMS. However, no single feature demonstrated a clear distinction between PwMS and HC; and thereby, a supervised learning approach is meaningful for this domain.

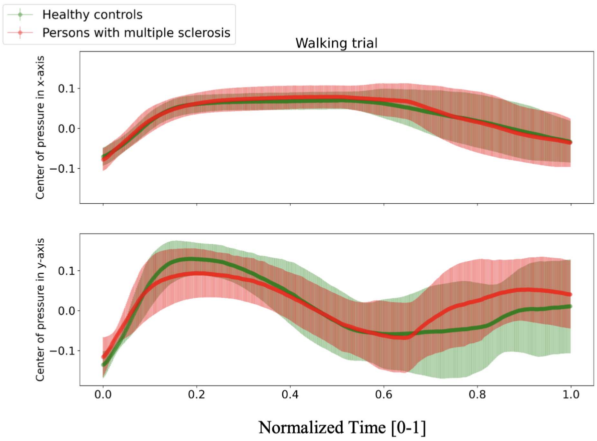

Fig. 6. Averaged gait cycle waveforms for healthy controls and persons with multiple sclerosis in the walking trial. Error bars to the averaged solid plot line represent standard deviation in the waveforms.

A. Statistical Analysis

Considering trial W, statistically significant differences between means were observed in body size-normalized terminal double support, force on TOL, left foot progression angle, and butterfly diagram-derived lateral shift and squared deviation. When normalized using the regression technique, significant differences were noted in terminal double support, lateral shift and squared deviation. Considering WT, statistically significant differences between means were observed in *size-N* and *regress-N* terminal double support, lateral shift and squared deviation.

B. Prediction Models

In order to classify sequences and subjects between HC and PwMS for task- (VI-B1) and subject generalization (VI-B2) designs, 16 diverse traditional machine and DL algorithms were compared with *size-N* and *regress-N* data. These sequences were fairly balanced across both classes in our data.

1) Task Generalization: Table II summarizes the sequence- and subject-wise evaluation metrics for the top-3 ML and DL task generalization classifiers on categorizing the test set sequences of trial WT. Majority voting evidently upgraded all the sequence-wise performance metrics within each algorithm, such as from 82.3% to 88.6% accuracy on MSResNet with *size-N* data. The results further improved across all metrics for DL algorithms with *regress-N* data in contrast to when using *size-N* data. The top-3 DL algorithms, that is, MSResNet, GRU and

TABLE II

TASK GENERALIZATION: COMPARING SEQUENCE- AND SUBJECT-WISE TEST SET PERFORMANCE ACROSS TOP-3 ML AND DL ALGORITHMS

			Sequence-wise evaluation metrics					Subject-wise evaluation metrics				
	Algorithm	Normalization	Accuracy	Precision	Recall	F_1 score	AUC	Accuracy	Precision	Recall	F_1 score	AUC
ML	DT	Size-N	0.731	0.698	0.661	0.679	0.759	0.886	0.882	0.882	0.882	0.918
		Regress-N	0.751	0.700	0.741	0.719	0.755	0.829	0.824	0.824	0.824	0.908
	XGBoost	Size-N	0.799	0.796	0.714	0.753	0.895	0.886	0.933	0.824	0.875	0.978
		Regress-N	0.823	0.842	0.724	0.779	0.916	0.886	**1.0**	0.765	0.867	0.964
	MLP	Size-N	0.779	0.735	0.757	0.746	0.825	0.914	0.938	0.882	0.909	0.930
		Regress-N	0.823	0.802	0.781	0.791	0.887	0.914	0.938	0.882	0.909	0.944
DL	MSResNet	Size-N	0.823	0.853	0.711	0.775	0.897	0.886	**1.0**	0.765	0.867	0.943
		Regress-N	**0.917**	**0.879**	**0.937**	**0.907**	**0.971**	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	RNN	Size-N	0.844	0.840	0.787	0.813	0.915	0.971	1.0	0.941	0.970	0.949
		Regress-N	0.873	0.831	0.884	0.857	0.935	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
	GRU	Size-N	0.840	0.857	0.754	0.802	0.913	0.971	**1.0**	0.941	0.970	0.949
		Regress-N	0.880	0.856	0.867	0.861	0.948	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**

*Note: ML is machine learning, DL is deep learning, AUC is area under the receiver operating curve, Size-N is body size-based normalization, and Regress-N is multiple regression-based normalization. The numbers in **bold** represent the highest model performance. See section V-A for details on algorithms.*

TABLE III

SUBJECT GENERALIZATION: COMPARING SEQUENCE- AND SUBJECT-WISE CROSS-VALIDATION PERFORMANCE ACROSS TOP-3 ML AND DL ALGORITHMS

			Sequence-wise evaluation metrics					Subject-wise evaluation metrics				
	Algorithm	Normalization	Accuracy	Precision	Recall	F_1 score	AUC	Accuracy	Precision	Recall	F_1 score	AUC
ML	LR	Size-N	0.569±0.07	0.462±0.28	0.414±0.16	0.417±0.20	0.565±0.09	0.629±0.17	0.473±0.27	0.567±0.33	0.511±0.29	0.593±0.21
		Regress-N	0.616±0.03	0.517±0.26	0.50±0.14	0.483±0.19	0.633±0.07	0.686±0.11	0.513±0.27	0.627±0.40	0.550±0.31	0.70±0.13
	LSVM	Size-N	0.559±0.06	0.461±0.28	0.416±0.15	0.413±0.19	0.542±0.10	0.60±0.17	0.473±0.27	0.533±0.34	0.491±0.30	0.593±0.18
		Regress-N	0.619±0.07	0.505±0.29	0.508±0.22	0.478±0.23	0.641±0.09	0.714±0.13	0.553±0.32	0.633±0.40	0.572±0.33	0.660±0.21
	DT	Size-N	0.510±0.05	0.419±0.29	0.425±0.27	0.382±0.23	0.458±0.09	0.543±0.06	0.50±0.32	0.517±0.34	0.461±0.25	0.563±0.22
		Regress-N	0.642±0.12	0.572±0.19	0.583±0.14	0.550±0.13	0.578±0.12	0.743±0.17	0.713±0.25	0.810±0.26	0.721±0.21	0.733±0.22
DL	ResNet	Size-N	0.673±0.05	0.545±0.25	0.383±0.24	0.428±0.24	0.528±0.21	0.743±0.11	0.547±0.15	0.850±0.20	0.644±0.14	0.693±0.14
		Regress-N	**0.728±0.05**	0.590±0.23	**0.709±0.26**	**0.630±0.22**	**0.70±0.20**	**0.829±0.11**	**0.833±0.21**	**0.810±0.19**	**0.808±0.17**	**0.813±0.16**
	TCN	Size-N	0.592±0.09	0.503±0.31	0.405±0.17	0.423±0.19	0.536±0.17	0.629±0.11	0.533±0.09	0.633±0.22	0.563±0.13	0.647±0.17
		Regress-N	0.670±0.10	**0.606±0.34**	0.464±0.23	0.493±0.24	0.612±0.15	0.743±0.11	0.613±0.21	0.767±0.23	0.656±0.17	0.640±0.10
	RNN	Size-N	0.663±0.06	0.561±0.32	0.504±0.24	0.505±0.23	0.578±0.19	0.714±0.09	0.680±0.19	0.677±0.23	0.661±0.18	0.747±0.15
		Regress-N	0.671±0.09	0.577±0.18	0.416±0.17	0.475±0.17	0.635±0.15	0.771±0.07	0.553±0.30	0.720±0.37	0.613±0.31	0.60±0.08

*Note: The numbers in **bold** represent the highest model performance.*

RNN (in order), had a sequence-wise accuracy, A_{seq}, of 91.7%, 88% and 87.3% (resp.), with the regression normalized data. Further, majority voting gave a perfect subject-level classification accuracy, A_{sub}, across the 3 top DL algorithms. In contrast, these DL models had an A_{seq} of less than 85% and the maximum A_{sub} of 97.1% when using the *size-N* data. Similarly, the highest sequence classification AUC (AUC_{seq}) was 0.97 using MSResNet, followed by 0.95 and 0.94 using GRU and RNN (resp.), with the *regress-N* data, while the maximal AUC_{seq} was 0.92 with *size-N* data using the vanilla RNN architecture. The top-3 ML models, namely, DT, XGBoost and MLP, all had an A_{sub} of less than 92% vs. a perfect A_{sub} with DL models using *regress-N* data. MSResNet with *regress-N* data was an overall top-performer for task generalization with an accuracy, F_1 and AUC of 91.7%, 0.91 and 0.97 (resp.), at a sequence-level, followed by GRU and RNN with a matching perfect subject-level classification. This top MSResNet architecture, illustrated in Fig. 3, was trained for 45 epochs (as decided by the early stopping paradigm with *patience* 20) with a batch size of 100, and Adam optimizer along with a learning rate of 0.005; with nearly 2.1 million model parameters, this model took 15 minutes to train and 1.5 seconds to evaluate on a GPU. MSResNet utilizes both multi-scaled and residual learning frameworks to discover robust dynamics in gait motion.

2) Subject Generalization: Table III illustrates the mean and standard deviation of 5-fold cross-validation evaluation metrics for the top-3 ML and DL subject generalization classifier. Not surprisingly, the subject-wise metrics were superior to the sequence-wise performance measures. Overall, ResNet was the highest-performing classifier across all DL and traditional ML algorithms. All algorithms performed better with the *regress-N* data in contrast to the standard *size-N* data. The top subject generalization algorithm was ResNet with *regress-N* data attaining the mean accuracy, F_1 and AUC of 72.8%, 0.63 and 0.70 (resp.), at sequence-level; and 82.9%, 0.81 and 0.81 (resp.), at subject-level classification. However, the top-3 ML models, namely, LR, LSVM and DT, all ended up with a mean A_{sub}, F_{1sub} and AUC_{sub} of less than 75%, 0.73 and 0.74 (resp.). The top-performing ResNet architecture used a positional encoding layer followed by an initial convolutional block and 4 bottleneck blocks with {64, 128, 256, 512} filters (resp.); each bottleneck residual block had 3 convolutional layers with kernel sizes {3, 5, 1} (resp.). It was trained for 5 epochs with a batch size of 128 and AdamW optimizer (learning rate: 0.01, weight decay: 0.01);

TABLE IV
Ablation Study for Task and Subject Generalization Frameworks

	Task generalization						Subject generalization					
Feature set	Top DL algorithm	A_{sub}	P_{sub}	R_{sub}	F_{1sub}	AUC_{sub}	Top DL algorithm	A_{sub}	P_{sub}	R_{sub}	F_{1sub}	AUC_{sub}
Temporal	RNN	0.83	0.79	0.88	0.83	0.87	ResNet	0.71±0.16	0.47±0.20	**0.90±0.20**	0.58±0.19	0.65±0.13
Spatial	GRU	0.83	0.87	0.76	0.81	0.90	ResNet	0.66±0.15	0.34±0.31	0.49±0.42	0.39±0.33	0.58±0.15
Spatiotemporal	LSTM	0.97	0.94	**1.0**	0.97	**1.0**	ResNet	0.80±0.07	0.63±0.24	0.87±0.19	0.69±0.18	0.77±0.08
Kinetic	ResNet	0.86	1.0	0.71	0.83	0.86	ResNet	0.77±0.15	0.56±0.46	0.50±0.45	0.51±0.43	0.69±0.25
Temporal-kinetic	GRU	0.91	0.94	0.88	0.91	0.90	RNN	0.74±0.11	0.65±0.13	0.73±0.27	0.68±0.19	0.68±0.13
Spatial-kinetic	MSResNet	0.94	**1.0**	0.88	0.94	0.97	ResNet	0.77±0.11	0.65±0.18	0.82±0.19	0.71±0.14	0.74±0.12
All	MSResNet	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**	ResNet	**0.83±0.11**	**0.83±0.21**	0.81±0.19	**0.81±0.17**	**0.81±0.16**

*Note: The numbers in **bold** represent the highest model performance.*

with nearly 433 K trainable parameters, training took around 1.5 minutes on GPU. AdamW typically helps models to train faster and generalize better. See Fig. 3 for an illustrative ResNet architecture.

This diverse performance across model frameworks and normalization schemes by different classifiers is attributed to the no free lunch theorem for supervised algorithms, stating there is no universally accurate algorithm across all datasets and evaluation metrics [46]. For further post hoc analysis in Sections VI-C to VI-E, we used *regress-N* data and top-performing DL algorithms as they revealed better performance over both task- and subject generalization frameworks.

C. Ablation Study on Gait Features

Next, we perform an ablation study in an attempt to comparatively evaluate the importance of various feature subcategories for classification, particularly 7 temporal, 4 spatial, 13 spatiotemporal, 8 kinetic, 15 temporal-kinetic and 12 spatial-kinetic features relative to all 21 features. All subject-wise evaluation metrics for the top model per data subset are presented in Table IV across both the task- and subject generalization schemes. Note that we train and tune all ML and DL architectures entirely from scratch on these feature subsets as part of our post hoc analysis. DL, especially all recurrent architectures along with ResNet and MSResNet, exceeded the performance of traditional ML algorithms over all feature subsets and both generalization frameworks. It is interesting to note that there is an overlap between top architectures in Section VI-B and Table IV. The optimal task generalization metrics were observed by utilizing the entire 21-dimensional feature set with MSResNet (A_{sub}: 1.0, F_{1sub}: 1.0), closely followed by spatiotemporal subset with LSTM (A_{sub}: 0.97, F_{1sub}: 0.97), and then by spatial-kinetic parameters again with MSResNet (A_{sub}: 0.94, F_{1sub}: 0.94). Analogously for subject generalization, employing all features with ResNet resulted in top mean cross-validation performance (A_{sub}: 0.83, F_{1sub}: 0.81), followed by spatiotemporal (A_{sub}: 0.80, F_{1sub}: 0.69) and spatial-kinetic (A_{sub}: 0.77, F_{1sub}: 0.71) feature subsets, both also with ResNet. Across all feature designs for both model frameworks, CNN and TCN were never the top performers. Note that task generalization had a distinct variety of models that were top performers for each feature stream, whereas ResNet was the only top performer for subject generalization across all subsets, except for temporal-kinetic features where RNN was better. Within both task- and subject generalization designs, results when utilizing the complete feature set surpassed all other examined subset combinations; moreover, we observed that the spatiotemporal subset presents superior performance to any other composition with kinetic features. Further, comparing these results in Table IV with the performance in [13], we infer that using DL with multiple strides outweighs single-stride performance across both model designs and all data subgroups. Overall, this analysis backs our use of all spatiotemporal and kinetic features for MS classification.

D. Feature Importance

In this section, we attempted to demonstrate the interpretability of our DL models by means of 1) permutation feature importance (VI-D1) that defined the most and least informative features in our gait classification models, and 2) visualizing the neural network's inner feature maps at the penultimate layer (VI-D2) that gave insights about model's complex internal processing.

1) Permutation Feature Importance: Having fixed the *regress-N* normalization scheme, we permuted each of the 21 gait features, one at a time, and assessed the reduction in evaluation metrics for our top performing models, i.e., MSResNet for task generalization and ResNet for subject generalization. The inherent randomness in shuffling might bias our findings, thus this procedure was repeated 20 times for the test set and the corresponding metrics over these reiterations were averaged out for relatively robust results. This shuffling procedure broke the relationship between the shuffled feature and the corresponding target, and thus the drop in model performance after feature permutation was indicative of how much the model depended on the feature. Therefore, a reduced model performance after permuting a feature signified higher dependence of our model on the associated feature for classification and consequently, a greater importance of the respective feature. For task generalization, force at MidSSR followed by right single support and right foot progression angle (in order) were the most informative features; however, walk ratio was the least predictive of labels. For subject generalization, the most informative features with the least accuracy after permutation were stride time and lateral shift followed by stride length. A few features, namely, cadence and lateral deviation, had very little effect on model performance for subject generalization.

2) Visualizing Penultimate Layer Feature Vectors: Given a DL model's involved architecture comprising of several layers with numerous neurons and correspondingly large number of learnt parameters, we visualize our top DL network's feature vectors at the penultimate step, in an attempt to comprehend

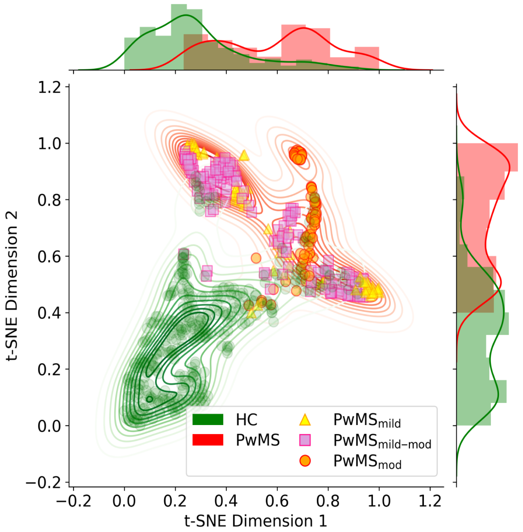

Fig. 7. 2D t-SNE visualization for task generalization. Two natural clusters, shown in green and red, grouping the 5-stride sequences of HC and PwMS (resp.), are identified in last layer embedding for the top task generalization model. Mild, mild-to-moderate and moderate severity subgroups within PwMS are marked in yellow triangles, pink squares and orange circles (resp.).

its complex processing to some extent. Considering high dimensional feature space at our model's last layer, we firstly used a non-linear dimensionality reduction technique, namely, t-distributed stochastic neighbor embedding (t-SNE) [47], to collapse the multidimensional feature maps into a 2D t-SNE embedding space and subsequently visualize it. Primarily, t-SNE is an iterative optimization algorithm to assign multidimensional data points to a lower dimensional space such that closer points in the higher dimensions still stay close and likewise, farther points remain distant in the reduced space as well. More formally, t-SNE minimizes the difference, as statistically measured by the Kullback-Leibler (KL) divergence, between the two probability distributions measuring the pairwise similarities of the original data points and the corresponding points in the low-dimensional t-SNE embedding (resp.). Our ultimate objective is to visualize any natural clusters of data points that may emerge in the penultimate space. Fig. 7 visualizes the 2D t-SNE of the 512-dimensional last layer embedding for the top task generalization model (MSResNet). Clearly, two inherent clusters, green and red, categorizing the HC and PwMS 5-stride sequences (resp.), originate in the penultimate layer; further, there appears to be a progression among sequences of mild, mild-to-moderate and moderate severity subgroups within PwMS. These native arrangements demonstrate the robustness of our predictions and in fact validate that our feature space is well optimized by backpropagation to classify sequences in HC and PwMS. In conclusion, these results verify that our model extracted necessary information from the data that enabled t-SNE to clearly identify two distinct classes of sequences.

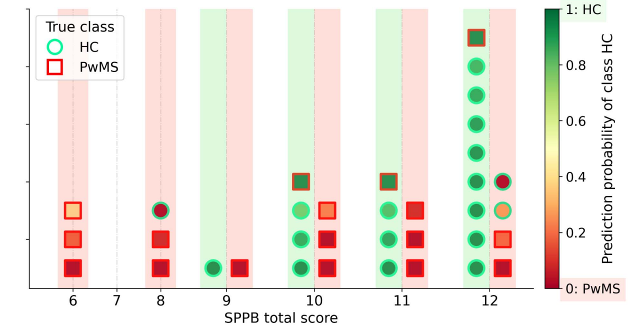

Fig. 8. Visualizing the predictions of subject generalization model w.r.t the corresponding subject's lower extremity function. Green-edged circles and red-edged squares represent actual HC and PwMS (resp.), where marker's face color shade denotes the corresponding prediction probability for class HC. Horizontal axis displays the overall SPPB score and background stripe's color depicts the prediction surface, where markers on green (or red) stripe are predicted as HC (or PwMS) by the model.

E. Association of Predictions With Lower Extremity Function

We examined a possible correspondence between the lower extremity physical function in middle-aged to older adults and our top DL model's prediction probabilities. We used the short physical performance battery (SPPB) assessment [48] to measure the physical performance of middle-aged to older adults on a scale of 0 (worst) to 12 (best). SPPB examined 3 areas that emulate day-to-day tasks essential for independent living, namely, static balance, gait speed and getting in and out of a chair. A higher summary score on SPPB signified none to mild mobility limitations and a lower score implied severe limitations. Our dataset had subjects with frailty ranging from minimal to moderate, i.e. SPPB: 10.37 ± 1.85 $[6–12]$. Fig. 8 depicts the predictions made by the top-performing subject generalization model, ResNet, w.r.t the corresponding subject's SPPB. The markers, i.e. green-edged circles and red-edged squares, represent actual HC and PwMS (resp.), where marker face-color denotes the corresponding prediction probability for class HC. Horizontal axis displays the overall SPPB score and background stripe color depicts the predicted class by the model, i.e., markers on green and red stripes are predicted as HC and PwMS (resp.). Note that markers in each SPPB and prediction stripe are sorted in order of prediction probability of their true class. For instance, one true PwMS with SPPB of 10 was misclassified as healthy with a particularly high HC confidence probability and no true HCs were misclassified at SPPB 10. Fig. 8 depicts that PwMS with a higher SPPB, i.e. better physical performance, had majority of 5-stride sequences incorrectly predicted as belonging to the healthy cohort; this was quite likely as gait characteristics of PwMS with none to mild mobility limitations could be difficult to discern from healthy gait. In summary, SPPB seemed to have some correspondence with our model predictions, but, no significant correlations were observed.

VII. Discussion

We studied a DeepMS2G framework employing data driven DL on 21 multi-stride spatiotemporal and kinetic gait features to classify middle-aged to older adults with and without MS. Our workflow is end-to-end open source, available at https://github.com/kaurrachneet6/DeepMS2G.git. Our proposed system offered an automated, accurate and remote monitoring mechanism for neurological gait classification and was quicker, utilizing only 5-stride sequence or a brief gait length than most typical clinical gait assessments. Past works [14], [15] have explored DL with domain knowledge-based gait features for low vs. high fall risk assessment. However, compared to our exhaustive experimental comparison with 16 diverse models across 2 different designs, namely, task- and subject generalization, these past works have only examined LSTM and 4 traditional ML classifiers for subject generalization. Further, these studies focused on 4 kinematics-based gait parameters, whereas we utilized dynamics from 21 kinematic as well as kinetic features. Additionally, we comprehensively investigated the explainability of our top-performing algorithms via post hoc analysis (Sections VI-C to VI-E), which was absent in previous analogous studies. Although our prior research [13] using traditional ML frameworks on individual strides provided utility in the identification of MS-related changes in gait, our current approach employing DL with multi-stride data provided a tool to extract additional information from gait dynamics and variations across temporally ordered strides. Moreover, using DL and multi-stride dynamics for MS classification exceeded the subject-wise performance metrics presented in [13] across both task- and subject generalization designs. A few additional past works have explored using classical ML to classify MS based on gait data [26], [49], however, to the best of our knowledge, ours is the first study extensively examining modern DL algorithms on multi-stride spatiotemporal and kinetic gait features for MS classification.

In contrast to prior work using wearable inertial measurement unit sensors [15], [20], [26], [50], [51] electromyography (EMG) [21], [52], and motion capture systems [22] to predict neuromuscular changes in neurological gait, our approach requires no sensors to be placed on the participant, which simplifies data collection. However, there is the need for an instrumented treadmill, which limits usability in smaller or rural medicine practices. Further, depth cameras capturing 3D movement patterns have been explored for gait assessment [53], [54], but these systems are relatively costlier, have some limitations when used outdoors and are constrained by the camera to object distance. Most studies explored either an end-to-end DL framework that demanded larger datasets or a traditional ML approach on hand engineered features that more suited smaller datasets. We studied a hybrid approach utilizing our domain-knowledge based spatiotemporal and kinetic feature space with advanced DL methods in an effort to overcome the challenges of limited clinical data, which exist in most medical scenarios.

The advantages of using regression normalized gait features were apparent when *regress-N* improved the accuracy of identifying pathological gait than the standard *size-N* strategy in both task and subject generalization frameworks. ResNet-based models, namely MSResNet for task generalization and ResNet for subject generalization, were top-performers across both model designs, which might guide model selection in future studies on neurological gait classification. Also, using a 5-stride data sample allowed for frequent conclusions, which might assist in the deployment of future clinical applications based on this work. Moreover, our analysis in Sections VI-C to VI-E helped establish the interpretability of our top DL algorithms, which might facilitate gait practitioners to comprehend and trust the findings from our proposed system. An ablation study on the set of features in Section VI-C supported using all the extracted gait features for better predictability in both task and subject generalization model designs. Moreover, we observed that the spatiotemporal subset presents superior performance to any other composition with kinetic features. When only including a subset of features to examine the most relevant features driving the DL performance, we found that stride and single support times, force at midstance, and butterfly diagram-based lateral shift were the most valuable features across both classification frameworks. Further, we observed that PwMS with none to mild mobility limitations had the majority of 5-stride sequences incorrectly predicted as belonging to the healthy cohort. This is in line with observations in some past studies on MS, where gait parameters were noticed to worsen for severely affected MS patient groups compared to the control group and are not seen in PwMS with mild disability [55].

A larger study would allow better understanding of the dependence of the regression function on demographics, and also better understanding of confidence intervals. A broader sample of neurological disability levels in PwMS and subtypes might assist in making more generalized predictions for the heterogeneous MS community. Assessing gait in additional concurrent cognitive settings in future works might provide increased sensitivity to distinguish between MS related changes. Recently, transformer-based DL models have achieved outstanding performance on several vision and language tasks [56], [57]. Given higher model complexity in transformers and our relatively smaller dataset for classification, we did not consider transformer-based models for this work. Future work might involve evaluating the performance of transformers for neurological gait classification. Lastly, clinical applications of gait-based data to determine MS related changes might benefit from further examination using wearable gait-related data in real-world environments.

VIII. Conclusion

We proposed a DeepMS2G pipeline for classification of PwMS using DL and multi-stride dynamics across domain knowledge-based spatiotemporal and kinetic gait features. We evaluated DeepMS2G to generalize over distinct walking trials and new participants. We observed that ResNet-based models with regression-based normalization were top performers across both task and subject generalization designs. With no known cure and clinically unpredictable disease progression, our proposed framework might augment findings from standard clinical tests and aid clinicians in defining effective medication strategies.

References

[1] K. Sharma et al., “Epidemiology of multiple sclerosis in the United States (p1. 140),” *Neurology*, vol. 90, p. 15 Supplement, 2018.

[2] Atlas of ms faqs. ms international federation website, 2019. [Online]. Available: https://www.msif.org/about-us/who-we-are-and-what-we-do/advocacy/atlas/atlas-of-ms-faqs/

[3] R. Marrie et al., “The rising prevalence and changing age distribution of multiple sclerosis in manitoba,” *Neurology*, vol. 74, pp. 465–471, 2010.

[4] A. Compston and A. Coles, “Multiple sclerosis,” *Lancet*, vol. 372, no. 9648, pp. 1502–1517, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140673608616207

[5] L. Scheinberg et al., “Multiple sclerosis: Earning a living,” *N Y State J Med.*, vol. 80, no. 9, pp. 1395–400, 1980.

[6] J. Noseworthy et al., “Multiple sclerosis,” *New England J. Med.*, vol. 343, no. 13, pp. 938–952, 2000.

[7] K. J. Kelleher et al., “The characterisation of gait patterns of people with multiple sclerosis,” *Disabil. Rehabil.*, vol. 32, no. 15, pp. 1242–1250, 2010.

[8] J. P. Kaipust et al., “Gait variability measures reveal differences between multiple sclerosis patients and healthy controls,” *Motor Control*, vol. 16, no. 2, pp. 229–244, 2012.

[9] A. Kalron et al., “Quantifying gait impairment using an instrumented treadmill in people with multiple sclerosis,” *ISRN Neurol.*, vol. 2013, 2013, Art. no. 867575.

[10] M. J. Socie et al., “Examination of spatiotemporal gait parameters during the 6-min walk in individuals with multiple sclerosis,” *Int. J. Rehabil. Res.*, vol. 37, no. 4, pp. 311–316, 2014.

[11] N. M. Tahir and H. H. Manap, “Parkinson disease gait classification based on machine learning approach,” *J. Appl. Sci.*, vol. 12, no. 2, pp. 180–185, 2012.

[12] F. Wahid et al., “Classification of Parkinson’s disease gait using spatial-temporal gait features,” *IEEE J. Biomed. Health Inform.*, vol. 19, no. 6, pp. 1794–1802, Nov. 2015.

[13] R. Kaur et al., “Predicting multiple sclerosis from gait dynamics using an instrumented treadmill–A machine learning approach,” *IEEE Trans. Biomed. Eng.*, vol. 68, no. 9, pp. 2666–2677, Sep. 2021.

[14] C. Tunca, G. Salur, and C. Ersoy, “Deep learning for fall risk assessment with inertial sensors: Utilizing domain knowledge in spatio-temporal gait parameters,” *IEEE J. Biomed. Health Inform.*, vol. 24, no. 7, pp. 1994–2005, Jul. 2020.

[15] B. M. Meyer et al., “Wearables and deep learning classify fall risk from gait in multiple sclerosis,” *IEEE J. Biomed. Health Inform.*, vol. 25, no. 5, pp. 1824–1831, May 2021.

[16] C. Prakash et al., “Recent developments in human gait research: Parameters, approaches, applications, machine learning techniques, datasets and challenges,” *Artif. Intell. Rev.*, vol. 49, no. 1, pp. 1–40, 2018.

[17] A. S. Alharthi, S. U. Yunas, and K. B. Ozanyan, “Deep learning for monitoring of human gait: A review,” *IEEE Sensors J.*, vol. 19, no. 21, pp. 9575–9591, Nov. 2019.

[18] C. L. Rice et al., “Neuromuscular responses of patients with multiple sclerosis,” *Muscle Nerve: Official J. Amer. Assoc. Electrodiagnostic Med.*, vol. 15, no. 10, pp. 1123–1132, 1992.

[19] L. J. White and R. H. Dressendorfer, “Exercise and multiple sclerosis,” *Sports Med.*, vol. 34, no. 15, pp. 1077–1100, 2004.

[20] H. Zhao et al., “IMU-based gait analysis for rehabilitation assessment of patients with gait disorders,” in *Proc. IEEE 4th Int. Conf. Syst. Inform.*, 2017, pp. 622–626.

[21] I. Hussain and S.-J. Park, “Prediction of myoelectric biomarkers in post-stroke gait,” *Sensors*, vol. 21, no. 16, 2021, Art. no. 5334.

[22] E. Knippenberg et al., “Markerless motion capture systems as training device in neurological rehabilitation: A systematic review of their use, application, target population and efficacy,” *J. Neuroengineering Rehabil.*, vol. 14, no. 1, pp. 1–11, 2017.

[23] L. Comber et al., “Gait deficits in people with multiple sclerosis: A systematic review and meta-analysis,” *Gait Posture*, vol. 51, pp. 25–35, 2017.

[24] M. Psarakis et al., “Wearable technology reveals gait compensations, unstable walking patterns and fatigue in people with multiple sclerosis,” *Physiol. Meas.*, vol. 39, no. 7, 2018, Art. no. 075004.

[25] S. Tajali et al., “Predicting falls among patients with multiple sclerosis: Comparison of patient-reported outcomes and performance-based measures of lower extremity functions,” *Mult. Scler. Related Disord.*, vol. 17, pp. 69–74, 2017.

[26] A. P. Creagh et al., “Smartphone-and smartwatch-based remote characterisation of ambulation in multiple sclerosis during the two-minute walk test,” *IEEE J. Biomed. Health Inform.*, vol. 25, no. 3, pp. 838–849, Mar. 2021.

[27] J. Kamruzzaman and R. K. Begg, “Support vector machines and other pattern recognition approaches to the diagnosis of cerebral palsy gait,” *IEEE Trans. Biomed. Eng.*, vol. 53, no. 12, pp. 2479–2490, Dec. 2006.

[28] K. A. Welsh et al., “Detection of dementia in the elderly using telephone screening of cognitive status,” *Neuropsychiatry Neuropsychol., Behav. Neurol.*, vol. 6, pp. 103–110, 1993.

[29] J. F. Kurtzke, “Rating neurologic impairment in multiple sclerosis: An expanded disability status scale (EDSS),” *Neurology*, vol. 33, no. 11, pp. 1444–1452, 1983.

[30] R. Holtzer et al., “Performance variance on walking while talking tasks: Theory, findings, and clinical implications,” *Age*, vol. 36, no. 1, pp. 373–381, 2014.

[31] A. Karatsidis et al., “Validation of wearable visual feedback for retraining foot progression angle using inertial sensors and an augmented reality headset,” *J. Neuroeng. Rehabil.*, vol. 15, no. 1, 2018, Art. no. 78.

[32] A. Kalron and L. Frid, “The “butterfly diagram”: A gait marker for neurological and cerebellar impairment in people with multiple sclerosis,” *J. Neurological Sci.*, vol. 358, no. 1/2, pp. 92–100, 2015.

[33] A. L. Hof, “Scaling gait data to body size,” *Gait Posture*, vol. 3, no. 4, pp. 222–223, 1996.

[34] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning*, vol. 4. Berlin, Germany: Springer, 2006.

[35] R. Kaur et al., “A vision-based framework for predicting multiple sclerosis and Parkinson’s disease gait dysfunctions-a deep learning approach,” *IEEE J. Biomed. Health Inform.*, vol. 27, no. 1, pp. 190–201, Jan. 2023.

[36] N. Srivastava et al., “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[37] A. Vaswani et al., “Attention is all you need,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[38] K. He et al., “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[39] R. Liu et al., “Multiscale kernel based residual convolutional neural network for motor fault diagnosis under nonstationary conditions,” *IEEE Trans. Ind. Inform.*, vol. 16, no. 6, pp. 3797–3806, Jun. 2020.

[40] S. Bai et al., “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018, *arXiv:1803.01271*.

[41] T. Salimans and D. P. Kingma, “Weight normalization: A simple reparameterization to accelerate training of deep neural networks,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 901–909.

[42] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[43] K. Cho et al., “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *Proc. Conf. Empir. Methods Natural Lang. Process.*, Doha, Qatar, 2014, pp. 1724–1734,.

[44] I. Goodfellow et al., *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[45] J. L. Ba et al., “Layer normalization,” *Neural Inf. Process. Syst. - Deep Learn. Symp.*, 2016. [Online]. Available: https://arxiv.org/pdf/1607.06450v1.pdf

[46] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, 1997.

[47] L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[48] J. M. Guralnik et al., “A short physical performance battery assessing lower extremity function: Association with self-reported disability and prediction of mortality and nursing home admission,” *J. Gerontol.*, vol. 49, no. 2, pp. M85–M94, 1994.

[49] M. Alaqtash et al., “Automatic classification of pathological gait patterns using ground reaction forces and machine learning algorithms,” in *Proc. IEEE Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2011, pp. 453–457.

[50] A. P. Creagh et al., “Interpretable deep learning for the remote characterisation of ambulation in multiple sclerosis using smartphones,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–14, 2021.

[51] P. Schwab and W. Karlen, “A deep learning approach to diagnosing multiple sclerosis from smartphone data,” *IEEE J. Biomed. Health Inform.*, vol. 25, no. 4, pp. 1284–1291, Apr. 2021.

[52] K. Akhmadeev et al., “SVM-based tool to detect patients with multiple sclerosis using a commercial EMG sensor,” in *Proc. IEEE 10th Sensor Array Multichannel Signal Process. Workshop*, 2018, pp. 376–379.

[53] D. Kastaniotis et al., “Using kinect for assesing the state of Multiple Sclerosis patients,” in *Proc. 4th Int. Conf. Wireless Mobile Commun. Healthcare-Transforming Healthcare Through Innov. Mobile Wireless Technol.*, 2014, pp. 164–167.

[54] F. Gholami et al., “A microsoft kinect-based point-of-care gait assessment framework for multiple sclerosis patients,” *IEEE J. Biomed. Health Inform.*, vol. 21, no. 5, pp. 1376–1385, Sep. 2017.

[55] M. Coca-Tapia et al., “Gait pattern in people with multiple sclerosis: A systematic review,” *Diagnostics*, vol. 11, no. 4, 2021, Art. no. 584.

[56] K. Han et al., “A survey on vision transformer,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023.

[57] Y. Xu et al., “Transformers in computational visual media: A survey,” *Comput. Vis. Media*, vol. 8, no. 1, pp. 33–62, 2022.