Article

Kinematics-based predictions of external loads during handcycling

Griffin C. Sipes [1], Matthew Lee [1], Kellie M. Halloran [1], Ian Rice [2], and Mariana E. Kersh [1,3,4]*

[1] Department of Mechanical Science and Engineering, University of Illinois Urbana-Champaign
[2] Department of Health and Kinesiology, University of Illinois Urbana-Champaign
[3] Beckman Institute for Advanced Science and Technology, University of Illinois Urbana-Champaign
[4] Carle Illinois College of Medicine, University of Illinois Urbana-Champaign
* Correspondence: mkersh@illinois.edu

Abstract: The increased risk of cardiovascular disease in people with spinal cord injuries motivates work to identify exercise options that improve health outcomes without causing risk of musculoskeletal injury. Handcycling is an exercise mode that may be beneficial for wheelchair users but further work is needed to establish appropriate guidelines and requires assessment of the external loads. The goal of this research was to predict the 6-degree-of-freedom external loads during handcycling from data similar to that which can be measured from inertial measurement units (segment accelerations and velocities) using machine learning. Five neural network models and two ensemble models were compared against a statistical model. A temporal convolutional network (TCN) yielded the best predictions. Predictions of forces and moments in-plane with the crank were the most accurate (r = 0.95 - 0.97). The TCN model could predict external loads during activities of different intensities, making it viable for different exercise protocols. The ability to predict the loads associated with forward propulsion using wearable-type data enables the development of informed exercise guidelines.

Citation: Sipes, G.C.; Lee, M.; Halloran, K.M; Rice, I.; Kersh, M.E. Kinematics-based predictions of external loads during handcycling. *Sensors* **2024**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

Copyright: © 2024 by the authors. Submitted to *Sensors* for possible open access publication under the terms and conditions of the Creative Commons Attri- bution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Full-time manual wheelchair users, such as persons with spinal cord injuries, are at increased risk of chronic diseases such as cardiovascular disease (CVD), diabetes, and metabolic syndrome [1–3]. Although the evidence supporting the benefits of physical activity is clear, many individuals with spinal cord injuries are physically inactive and sedentary leading to worsening symptoms and increased levels of disability [4,5]. An optimal exercise for improving cardiopulmonary function requires sufficient taxation of cardio-respiratory fitness levels without risking musculoskeletal injury. Unfortunately, the transition to manual wheelchair use presents challenges as the arms become the source of all activities of daily living. For example, as many as 68% of manual wheelchair users have rotator cuff tendon tears [6].

Handcycling has emerged as a promising exercise mode for wheelchair users [7] because the loads on the shoulder are lower than everyday wheelchair propulsion [8]. However, exercise guidelines (intensity, frequency, etc.) for handcycling remain to be established and requires assessment of both cardiovascular stimulation and musculoskeletal safety. Musculoskeletal assessments, such as shoulder torque, provide objective measures of shoulder function but requires specialized equipment to measure the external hand loads.

Custom-built instrumented crank handles have been used to measure handcycling loads [8,9], but are not readily accessible. In contrast, kinematic data collection is commonly used in biomechanics research and can be related to external loads via direct or indirect methods. The ground reaction force during legged locomotion can be solved for directly using Newton's 2nd law if the kinematics of major body segments are measured [10,11]. However, tracking all segments can be difficult and may not be feasible for every activity thus motivating the use of statistical or machine learning models to predict external loads from subsets of kinematic data [12,13].

Both inertial measurement units and optical motion capture have been used to estimate ground reaction forces during walking, stair-use, and running [10–12,14,15]. Joint-level kinetics have been estimated from kinematic data for everyday wheelchair propulsion [13]. However, to our knowledge, no study has reported kinematic-based predictions of the external loads during handcycling and is therefore the objective of this study. To mimic data that could be measured via an inertial measurement unit we based our analyses on data from the radius segment. We evaluated two classes of machine learning models (neural network and ensemble) and compared their prediction performance against a statistical regression model.

2. Materials and Methods

2.1. Dataset and Model inputs

Biomechanics data of full-time wheelchair users (n=20) completing a handcycling exercise protocol as part of a separate study [9] were used. For each participant, six propulsion cycles were extracted during moderate-intensity (45% peak power) handcycling and six cycles during high-intensity (90% peak power). A custom instrumented handle was used to measure the three forces and three moments applied at the right handcycle handle (sampling frequency = 2 kHz). Kinematic data of the right arm was recorded using a 10-camera motion capture system (Vicon, sampling frequency = 100 Hz). A rigid-body model [16] was scaled to each participant and used to determine the radius segment center of mass position and orientation using inverse kinematics (Figure 1). Data from each propulsion cycle was interpolated into 360 data points corresponding to the 360 degrees of crank rotation.

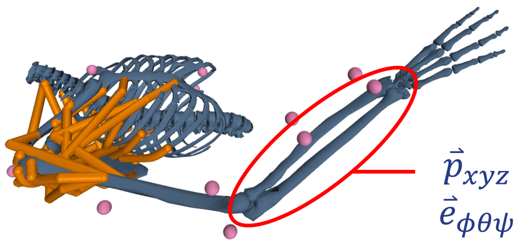

Figure 1. Motion-capture markers (pink spheres) used with subject-scaled rigid-body musculoskeletal model of upper arm. Radius segment (red circle) center of mass data derived from the radius position $\vec{p}$ and orientation $\vec{e}$ with respect to the global reference frame.

The segment positions were numerically differentiated twice using the central difference formula to obtain linear accelerations (Eq. 1):.

$$\vec{a}(t) = \frac{\vec{p}(t+\tau) - 2\vec{p}(t) + \vec{p}(t-\tau)}{\tau^2} \tag{1}$$

where $\vec{p}$ is the radius center of mass position, t is the current time, and τ is the interval between data points. Orientations (Euler angles) were differentiated once to obtain angular velocities (Eq. 2):

$$\vec{\omega}(t) = \frac{\vec{e}(t+\tau) - \vec{e}(t-\tau)}{2\tau} \tag{2}$$

where $\vec{e}$ is the vector of Euler angles denoting the radius center of mass in the global coordinate system.

The training data for all models was based on 90% of participants (n=18, randomly selected) while data from the remaining two participants were used as the test set. Data for model validation was based on a further split of the training dataset using n=2 participants' data. Participant weight (lbs), wingspan (in), and exercise intensity were included with the kinematic data as model input features. Exercise intensity was defined as as 0.5 or 0.9 to denote the varying protocol intensities. Prior to model training, each feature was standardized by subtracting the mean and dividing by the standard deviation. This process was used to standardize the training, test, and validation datasets.

2.2. Model Descriptions

Linear regression: We first assessed the predictive performance of an elastic net linear regression model (scikit-learn [17]) herein referred to as "Linear". The Linear model was trained with least absolute shrinkage and selection operator (L1) and Tikhonov (ridge) (L2) regularization of coefficients. The Linear model parameters were $\alpha = 0.003$, $\rho = 0.5$, and maximum training iterations = 2000. The model was fit by minimizing the objective function (Eq. 3):

$$\min_{\vec{\beta}} \frac{1}{2n_{samples}} \left\| \mathbf{X}\vec{\beta} - \vec{y} \right\|_2^2 + \alpha\rho \left\| \vec{\beta} \right\| + \frac{\alpha(1-\rho)}{2} \left\| \vec{\beta} \right\|^2 \tag{3}$$

where $\vec{\beta}$ is the vector of coefficients, $\mathbf{X}$ is the design matrix of predictors, $\vec{y}$ is the vector of outputs, α is a constant that multiplies the penalty terms, and ρ is the ratio of L1 regularization to L2 regularization.

Five neural network and two ensemble models (described below) were evaluated. Our choice of models was motivated by the work of others in predicting external loads [12,14] in addition to testing models of varying complexity. Each of the neural network models was compiled using mean squared error as the loss function and the Adam algorithm as the optimizer (learning rate = 0.0001). Models were trained for a maximum of 1000 epochs with an early stop callback, which stopped training early if the loss function did not decrease by 0.01 in 50 epochs. A batch size of 16 propulsion cycles was used during training. All neural network models were created using Tensorflow[18], the random forest model was created with scikit-learn [17], and the gradient boosting model was created with XGBoost [19]. All models were implemented using Python (v 3.9) .

Dense: A dense neural network (Figure 2A) model was created with one hidden dense layer containing 200 nodes and L1/L2 regularization of 0.00001 applied to the kernel weights matrix. Batch normalization and dropout (45%) were applied after the hidden layer. The output layer was a dense layer with six nodes denoting the six output features.

Bidirectional long short-term memory (BiLSTM): two layer BiLSTM model (Figure 2B). Each BiLSTM layer had 200 nodes with L1/L2 regularization (0.001) on the kernel weights matrix, recurrent kernel weights matrix, and on the bias vector. After each BiLSTM layer, batch normalization and dropout (45%) were applied. The output layer was a dense layer with six nodes.

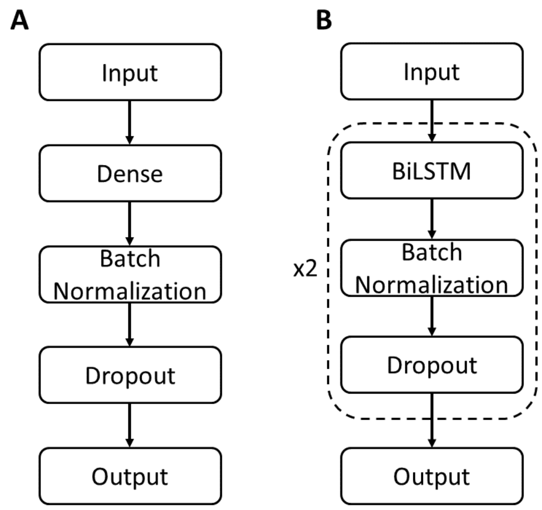

Figure 2. Neural network model architectures: (A) Dense (B) Bi-directonal long short-term memory (BiLSTM)

Convolutional neural network (CNN): Two 1D convolutional layers (Figure 3A) used to mimic the structure of the BiLSTM model. Each convolutional layer had 200 filters, a kernel size of 64, causal padding, rectified linear unit (ReLU) activation function, and L1/L2 kernel regularization (0.00001). After each convolutional layer, batch normalization and dropout (50%) were applied. The output layer was a time-distributed dense layer with six nodes.

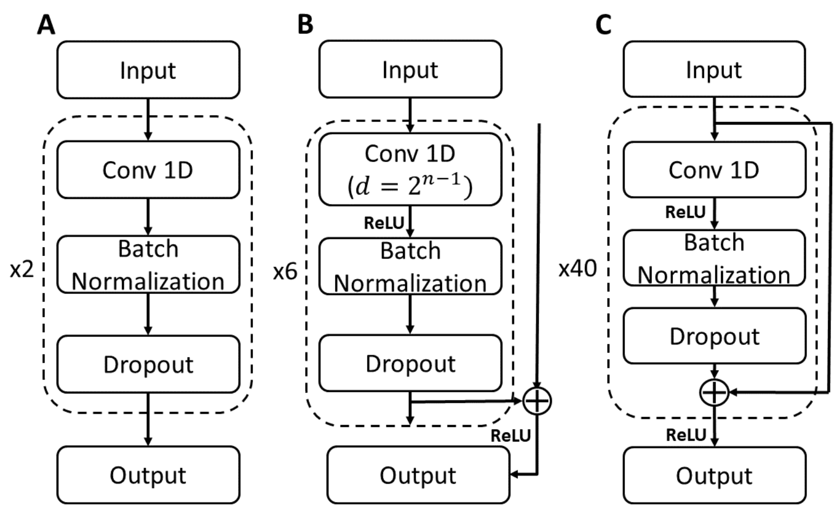

Figure 3. Convolution neural network architectures (A) Convolutional neural network (CNN) (B) Temporal convolutional network (C) Residual neural network (ResNet)

Temporal convolutional network (TCN): Six hidden layers with 2^{n-1} dilation rate, where n is the layer number (Figure 3B). Each layer had 64 filters with a kernel size of 32 and L1/L2 regularization (0.01). After each layer was an ReLU activation layer followed by a dropout layer (40%). The output layer was a time-distributed dense layer with 6 nodes and L1/L2 kernel regularization (0.01).

Residual neural network (ResNet): 40 layers of residual blocks (Figure 3C) . Each residual block was made of a 1D convolutional layer starting with 32 filters and a kernel size of 32. The kernel size in the convolutional layer doubled every five layers. The convolutional layers used 'same' padding and had L2 regularization (0.001). After each convolutional layer, batch normalization and an ReLU activation layer were used. The output layer was a time-distributed dense layer with 6 nodes and L2 kernel regularization (0.001).

Random Forest (RF) and Gradient Boosting Model (GBM): Finally, two ensemble models were evaluated. A random forest regressor [17] model was created with 400 decision trees, with each tree having a maximum depth of 10 levels. The second ensemble model was a Gradient Boosting Machine (GBM) implemented using XGBoost [19]. This model sequentially built 2000 decision trees, each with a maximum depth of 10, using a learning rate of 0.01. It employed both row and column subsampling at 70% and included L1 and L2 regularization (both set to 0.2). The GBM was trained for 1000 boosting rounds.

2.3. Statistical Analysis

For each of the 24 propulsion cycles in the test set, the median absolute percentage error (MdAPE) was calculated for each kinetic degree of freedom predicted. This analysis was repeated for the eight models tested. The Linear model was used as the baseline for comparison as it was the simplest model tested.

The distribution of model errors was non-normal for some degrees of freedom based on Shapiro-Wilkes tests for normality. Therefore, for each degree of freedom, a Kruskal-Wallis test was used to determine whether the errors differed between the models tested. If the differences were significant, a Dunn's test for multiple comparisons was used to determine which models differed in performance compared to the linear model.

The correspondence of the model prediction to the ground truth was also compared qualitatively by evaluating the shape of the predictions and quantitatively using Pearson's correlation coefficients. Kruskal-Wallis tests were used to determine whether the distributions of correlation coefficients for each degree of freedom differed between models and a post-hoc Dunn's test was used to determine differences compared to the linear model.

Wilcoxon Rank Sum tests were used to compare predictions made by the best performing model between the moderate and high-intensity propulsion cycles for each of the output features. Statistical analyses were performed using the SciPy package [20] in Python.

3. Results

Overall, errors in predictions of kinetics in the x and y directions, which contribute to the tangential and radial loads during handcycling, were lower (range: 32 to 56%) than kinetics in the out-of-plane z-direction (range: 45 to 101%). The Dense and RF models had significantly worse prediction performance (errors of 101 and 100%, respectively) than the Linear model for F_z (66% error), and the RF model predictions were also significantly worse for T_z. Most of the neural network models performed significantly better (error range: 38 - 42 %) than the Linear model (error = 54%) for predicting F_x (Table 1). The ResNet and TCN models performed better than the Linear model for F_y (errors of 38 and 36%, respectively) and T_x (errors of 32%). The GBM model predicted better than the linear model for T_y only (error = 39%). There were no other significant differences in prediction error compared to the Linear model.

Table 1. Comparison of model performance (median absolute percentage error (MdAPE)) across output features. Significance levels shown for models with lower error than the linear regression model.

Model Type	Model Name	Kinetic Prediction Error (MdAPE [%]) Fx	Fy	Fz	Tx	Ty	Tz
Statistical	Linear	53.97 (8.02)	51.65 (4.77)	66.36 (9.99)	44.84 (4.42)	51.05 (11.31)	44.86 (12.52)
Neural Network	Dense	**41.72 (7.16)***	42.44 (24.87)	101.45 (22.24)	40.6 (10.31)	55.74 (9.2)	55.74 (20.42)
	BiLSTM	45.02 (6.22)	41.97 (12.07)	84.66 (14.89)	35.14 (10.55)	45.11 (7.28)	61.67 (19.5)
	CNN	**39.83 (5.56)*****	43.18 (5.16)	85.64 (12.83)	40.27 (13.55)	45.32 (7.72)	51.53 (9.41)
	TCN	**37.53 (6.26)*****	**35.88 (5.17)*****	80.79 (6.66)	**31.64 (4.31)****	40.44 (6.9)	46.79 (16.77)
	ResNet	**41.58 (6.1)***	**37.83 (7.58)****	88.82 (18.71)	**31.96 (8.68)***	41.57 (6.88)	49.23 (8.85)
Ensemble	RF	47.35 (7.26)	43.35 (18.97)	99.53 (16.05)	41.65 (6.81)	49.45 (8.42)	65.4 (17.05)
	GBM	42.51 (6.36)	47.71 (6.73)	85.15 (5.72)	35.6 (8.12)	**38.9 (7.92)***	53.45 (9.61)

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$. Data are expressed as: Median (Median Absolute Deviation).

Of the neural network models, the TCN model had the lowest overall error and the GBM model had the lowest error of the ensemble models. The range of error values for the best statistical, neural network (TCN), and ensemble (GBM)) models were smaller in the x and y-directions compared to the z-direction (Figure 4).

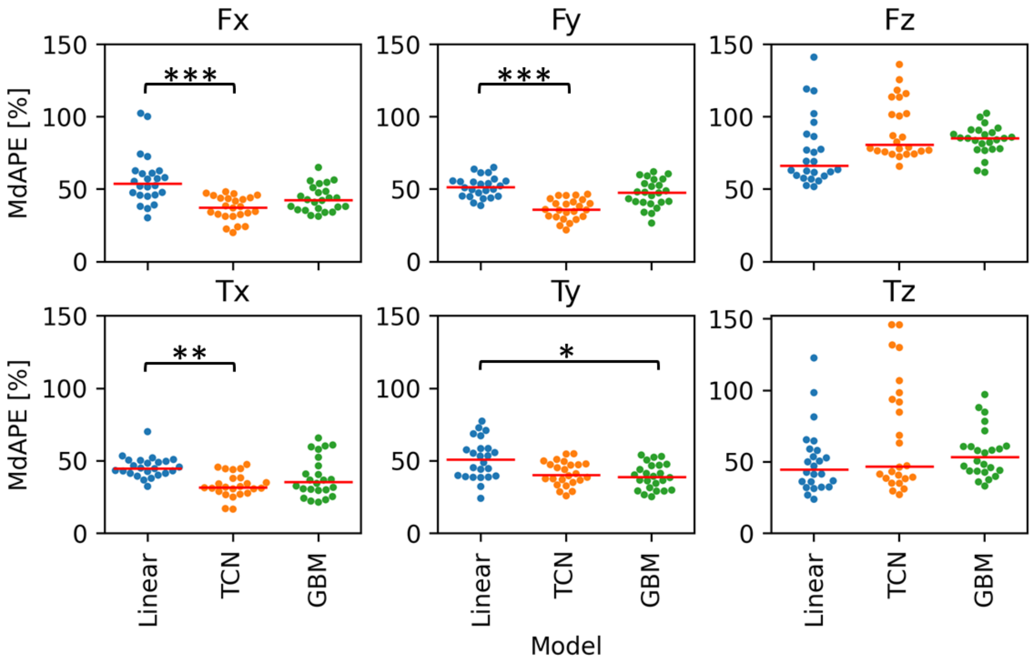

Figure 4. Comparison of model prediction median absolute percentage error (MdAPE) for each output feature across the 24 propulsion cycles of the test set. *** $p < 0.001$, ** $p < 0.01$, *$p < 0.05$

The shape of the average prediction from the TCN model matched well with the ground truth for the in-plane kinetic degrees of freedom (Figure 5, A1) with correlation coefficients ranging from 0.95 - 0.97 (Table 2). However, the model tended to underestimate the minima and maxima of each feature. For the x and y degrees of freedom, the errors tended to be highest at the beginning and end of the propulsion cycle. Force and torque profiles in the z-direction were poorly predicted.

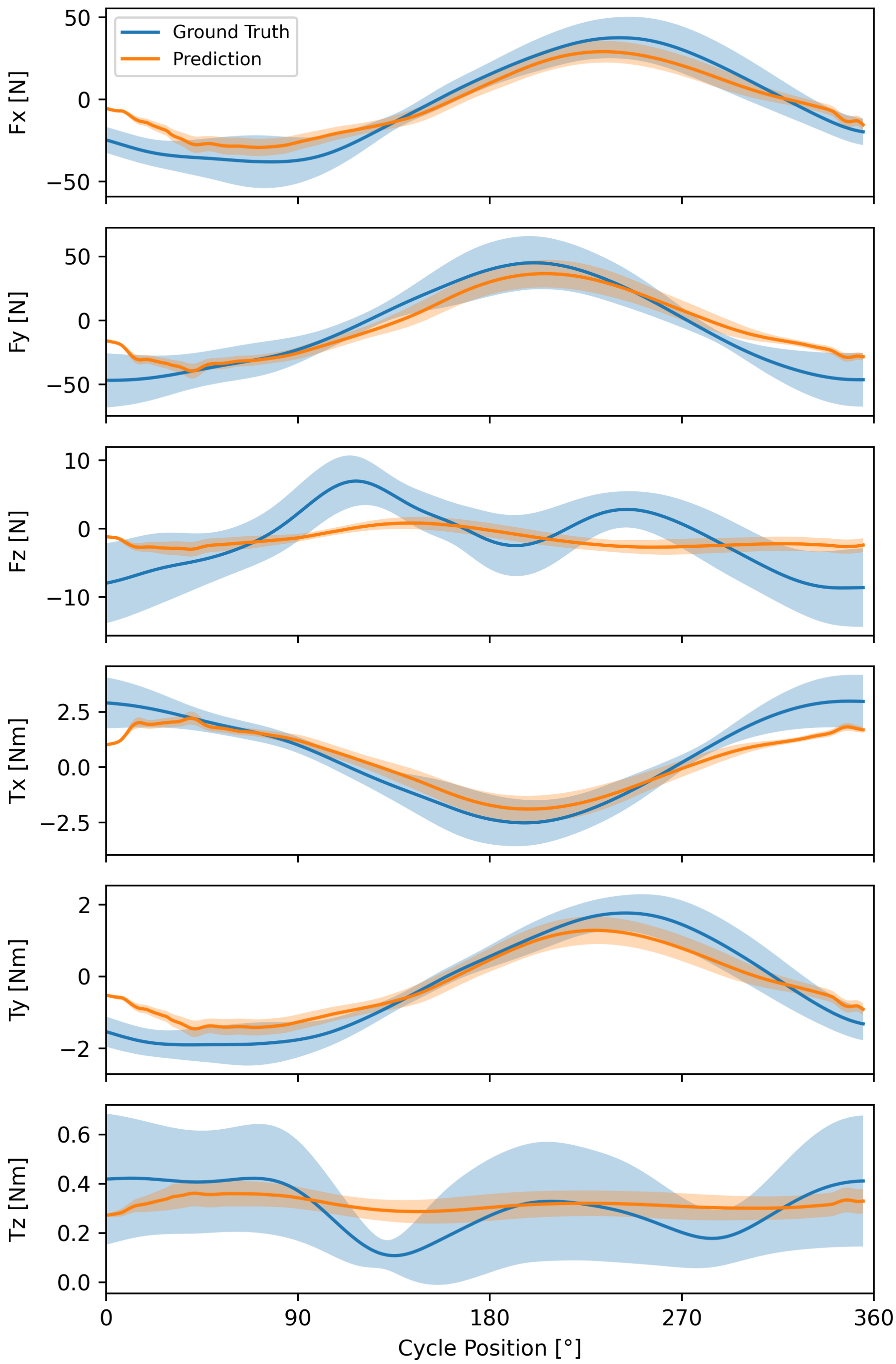

Figure 5. Average ground truth and average prediction of a propulsion cycle from the TCN model across each output feature. Standard deviation ribbons are shown.

Table 2. Comparison of model performance (Pearson's Correlation Coefficient (r)) across output features. Significance levels shown are compared to the linear regression model.

Model Type	Model Name	Output Feature Prediction Pearson's Correlation Coefficient (r)					
		Fx	Fy	Fz	Tx	Ty	Tz
Statistical	Linear	0.81 (0.05)	0.86 (0.02)	0.78 (0.06)	0.89 (0.01)	0.84 (0.05)	0.3 (0.09)
Neural Network	Dense	**0.92 (0.02)*** **	**0.95 (0.01)*** **	0.47 (0.24)	**0.94 (0.02)** **	0.88 (0.04)	0.41 (0.16)
	BiLSTM	**0.91 (0.03)*** **	**0.94 (0.02)*** **	0.4 (0.14)	**0.94 (0.02)*** **	0.89 (0.04)	0.3 (0.23)
	CNN	**0.94 (0.02)*** **	**0.97 (0.01)*** **	0.49 (0.21)	**0.94 (0.03)*** **	**0.93 (0.03)*** **	0.43 (0.34)
	TCN	**0.97 (0.01)*** **	**0.95 (0.02)*** **	0.56 (0.09)	**0.95 (0.01)*** **	**0.96 (0.02)*** **	**0.51 (0.2)* **
	ResNet	**0.91 (0.04)*** **	**0.96 (0.01)*** **	0.41 (0.41)	**0.97 (0.01)*** **	**0.93 (0.01)*** **	0.55 (0.16)
Ensemble	RF	0.83 (0.06)	0.86 (0.04)	-0.01 (0.24)	0.86 (0.03)	0.86 (0.04)	0.14 (0.5)
	GBM	**0.92 (0.03)*** **	**0.93 (0.02)*** **	0.63 (0.1)	0.96 (0.01)	**0.93 (0.01)*** **	0.54 (0.18)

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$. Data are expressed as: Median (Median Absolute Deviation).

Finally, we evaluated whether model performance was sensitive to exercise intensity level. Errors in forces in the y-direction were moderately higher during high-intensity exercise (mean error = 41%) compared to moderate intensity (mean error = 32%) (Figure 6). There were no significant differences in prediction performance between the two exercise intensities for the remaining degrees of freedom.

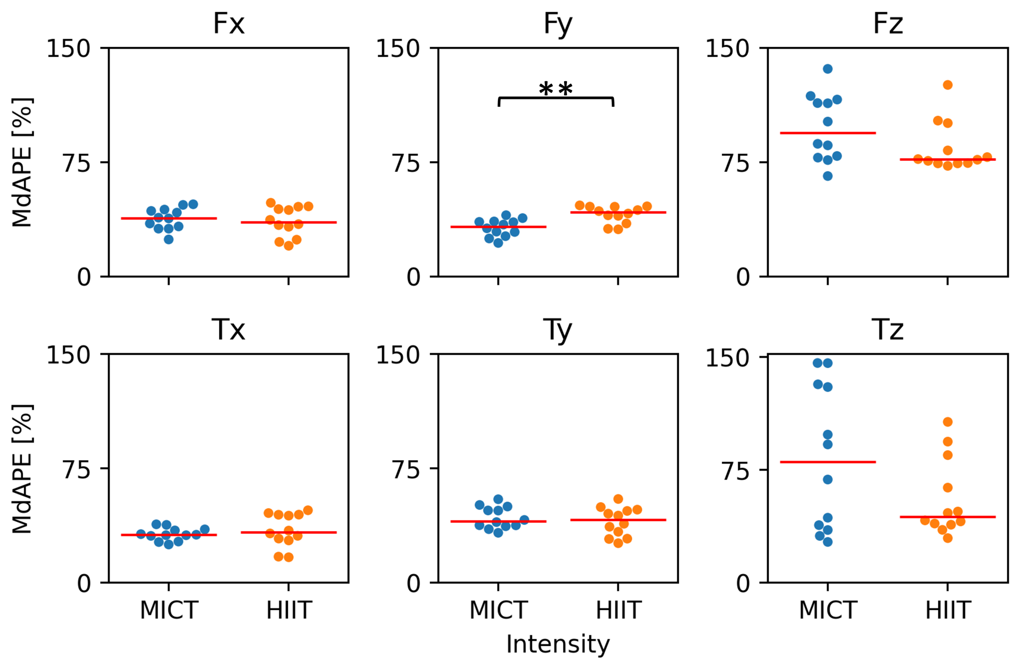

Figure 6. Comparison of TCN model prediction median absolute percentage error (MdAPE) between moderate intensity and high intensity propulsion cycles for each output feature. **: p=0.00323)

4. Discussion

We have shown that machine learning models can improve predictions of external loads during handcycling compared to a simpler elastic net linear regression model. Neural network models tended to perform better than ensemble models. Neural network predictions can leverage a longer time range of values, whereas tree-based ensemble models make instantaneous predictions at each timestep. Among the neural network models, convolutional neural networks outperformed recurrent (BiLSTM model) and dense neural network models.

Of all models tested, the predictive performance of the TCN was the most accurate for the kinetic parameters F_x and F_y. During handcycling, most force is applied in the tangential and radial directions with respect to the crank center, and is a combination of the F_x and F_y forces. In contrast, forces applied in the lateral direction (F_z) are lower in magnitude. Thus, while the lateral forces and torques had higher percent errors, the error magnitude was relatively small. The poor ability to predict lateral forces and torques may be due to the variability in propulsion technique (shaded blue regions in Figure 5) [9] despite the fact that handcycling is a closed chain task and relatively constrained in terms of motion. Individual technique has been classified for manual wheelchair propulsion but has not been used in handcycling. The inclusion of propulsion style, either as a categorical variable or via the inclusion of additional segment kinematics as inputs, in future machine learning models may prove useful for improving kinetic predictions. Another option for improving model performance is to tune the hyperparameters of the neural network via grid searching though requires significant computational resources. Finally, although there were some differences in prediction performance between moderate intensity and high intensity propulsion cycles, a single model is likely sufficient for multiple exercise intensities provided the intensity data is used as an input to the model.

The prediction performance for handcycling loads was similar to other studies that predicted external loads from kinematic data during locomotion. Johnson et al. used a combination of optical motion capture kinematics and inertial measurement unit data to predict the ground reaction forces and moments during various athletic maneuvers and reported Pearson's correlation coefficients from r=0.55 - 0.90 [12], similar to our results (r-range: 0.51 and 0.97) using the TCN model (Table 2). Liu et al. used joint angles from rigid body models to predict normalized ground reaction forces during stair use and reported Pearson's correlation coefficients between r=0.908 and r=0.991. Normalizing the external loads reduces individual variability, which may increase model predictions, but is challenging to apply to data from persons with spinal cord injury.

There are a few limitations of this study. The size of the dataset used was relatively small with 18 independent datasets used for training models. Remarkably, despite this small dataset, our best predictions were similar to others that used data from 80 subjects [14] for legged locomotion. Although the kinematic data used in this study was created to mimic that of a smartwatch-based inertial measurement unit, the data was derived from optical motion capture. Validation of the model predictions using IMU-derived kinematic data is needed. We also restricted our input data to the radius segment of the arm. It is possible that with additional training data from the adjacent segments (upper arm or hand), model performance could be improved. However, our goal was to evaluate the predictive performance of different models using a minimal dataset.

The results of this study indicate that a TCN model could be used in place of an instrumented crank handle in future studies of handcycling propulsion biomechanics. However, it should be noted that the model is best suited for predicting loads that contribute to forward propulsion with more work needed to improve predictions in the lateral direction. Nonetheless, the ability to predict propulsive loads during handcycling now enables future research aimed at providing evidence-based exercise recommendations for people with spinal cord injuries by quantifying the external loads of different protocols. Importantly, the use of wearable-type data combined with machine learning predictions of hand loads allows for the assessment of handcycling outside of the lab environment.

Author Contributions: Conceptualization, Griffin C. Sipes, Ian Rice, Mariana E. Kersh; Methodology, Griffin C. Sipes, Mariana E. Kersh; Software, Griffin C. Sipes, Matthew Lee; Validation, Griffin C. Sipes, Matthew Lee; Formal analysis, Griffin C. Sipes; Investigation, Griffin C. Sipes, Matthew Lee; Resources, Griffin C. Sipes, Kellie Halloran; Data curation, Griffin C. Sipes, Matthew Lee; Writing - original draft preparation, Griffin C. Sipes, Matthew Lee; Writing - review and editing, Griffin C. Sipes, Matthew Lee, Kellie Halloran, Ian Rice, Mariana E. Kersh; Visualization, Griffin C. Sipes, Mariana E. Kersh; Supervision, Ian Rice, Mariana E. Kersh; Project administration, Mariana E. Kersh; Funding acquisition, Griffin C. Sipes, Mariana E. Kersh. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation Graduate Research Fellowship Program and the De Luca Foundation Seed for Science Initiative.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: We thank Prof. Sanmi Koyejo for helpful guidance regarding best practices for machine learning during the early stages of this study. We are also grateful for the assistance of Dr. Joey Peters and Michael Focht in data collection.

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

Abbreviations

The following abbreviations are used in this manuscript:

CVD	Cardiovascular disease
PPO	Peak power output
IMU	Inertial measurement unit
BiLSTM	Bi-directional long short-term memory
CNN	Convolutional neural network
ReLU	Rectified linear unit
ResNet	Residual neural network
TCN	Temporal convolutional network
RF	Random forest regressor
GBM	Gradient boosting machine
MdAPE	Median absolute percentage error

Appendix A

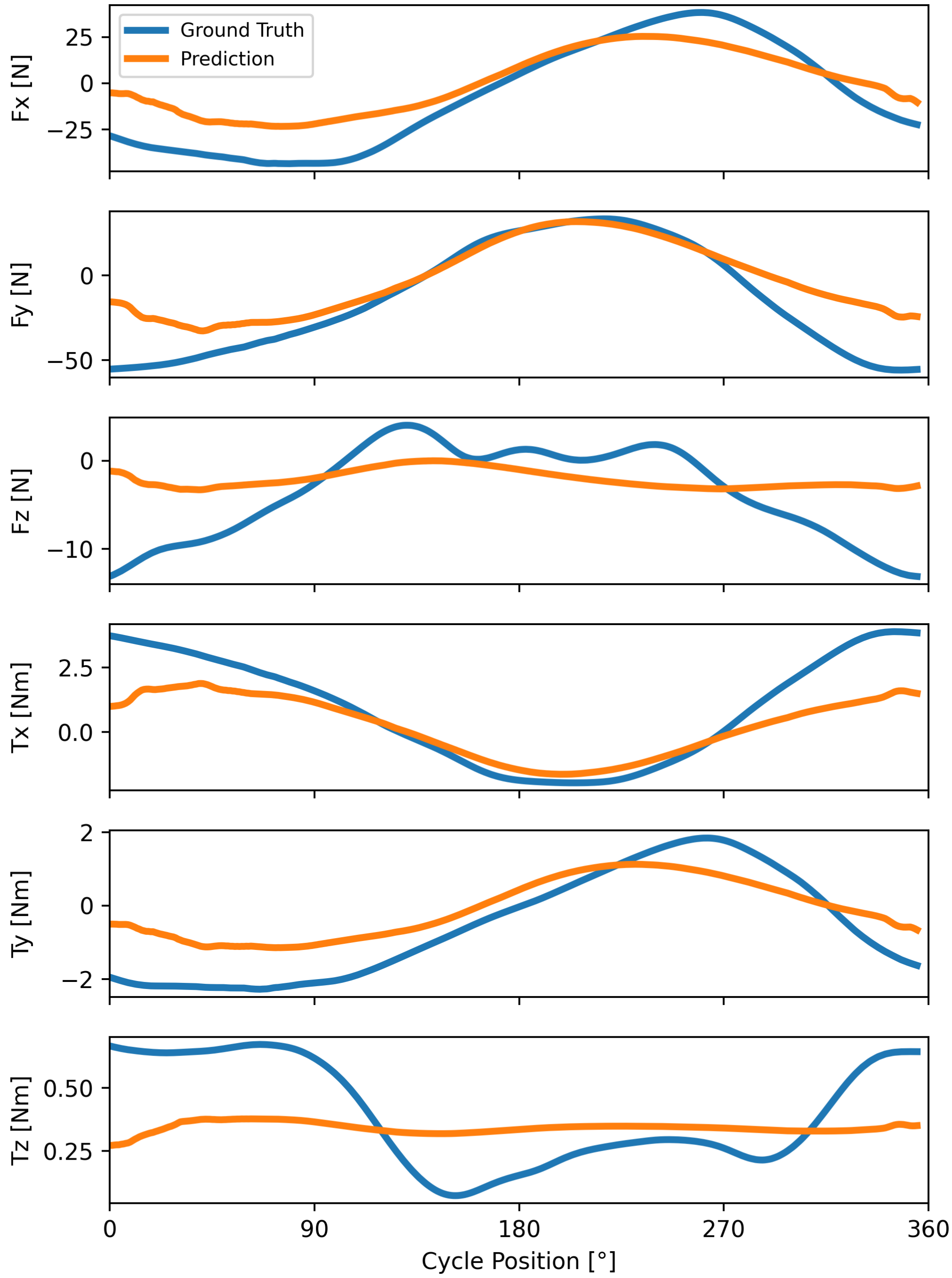

Figure A1. Representative (median error) propulsion cycle showing ground truth and TCN model prediction.

References

1. LaVela, S.L.; Evans, C.T.; Prohaska, T.R.; Miskevics, S.; Ganesh, S.P.; Weaver, F.M. Males Aging with a Spinal Cord Injury: Prevalence of Cardiovascular and Metabolic Conditions. *93*, 90–95, [22200386]. https://doi.org/10.1016/j.apmr.2011.07.201.
2. Barry, W.; St Andre, J.; Evans, C.; Sabharwal, S.; Miskevics, S.; Weaver, F.; Smith, B. Hypertension and antihypertensive treatment in veterans with spinal cord injury and disorders. *Spinal Cord* **2013**, *51*, 109–115.

3. Gater Jr, D.R.; Farkas, G.J.; Berg, A.S.; Castillo, C. Prevalence of metabolic syndrome in veterans with spinal cord injury. *The journal of spinal cord medicine* **2019**, *42*, 86–93.
4. Persson, R.; Lee, S.; Yood, M.U.; Wagner, M.; Minton, N.; Niemcryk, S.; Lindholm, A.; Evans, A.; Jick, S. Incident cardiovascular disease in patients diagnosed with multiple sclerosis: A multi-database study. *Multiple Sclerosis and Related Disorders* **2020**, *37*, 101423.
5. McPhee, P.G.; Claridge, E.A.; Noorduyn, S.G.; Gorter, J.W. Cardiovascular disease and related risk factors in adults with cerebral palsy: a systematic review. *Developmental Medicine & Child Neurology* **2019**, *61*, 915–923.
6. Jahanian, O.; Van Straaten, M.G.; Goodwin, B.M.; Lennon, R.J.; Barlow, J.D.; Murthy, N.S.; Morrow, M.M. Shoulder Magnetic Resonance Imaging Findings in Manual Wheelchair Users with Spinal Cord Injury. *45*, 564–574, [33166207]. https://doi.org/10.1080/10790268.2020.1834774.
7. Halloran, K.; Focht, M.; Teague, A.; Peters, J.; Rice, I.; Kersh, M. Moving Forward: A Review of Continuous Kinetics and Kinematics during Wheelchair and Handcycling Propulsion. https://doi.org/10.31224/2236.
8. Arnet, U.; Drongelen, S.V.; Scheel-Sailer, A.; Woude, L.H.V.V.D.; Veeger, D.H.E.J. Shoulder Load during Synchronous Handcycling and Handrim Wheelchair Propulsion in Persons with Paraplegia. *44*, 222–228, [22367531]. https://doi.org/10.2340/16501977-0929.
9. Halloran, K.M.; Peters, J.; Focht, M.D.K.; Rice, I.; Kersh, M.E. Propulsion Kinetics of Recumbent Handcycling during High and Moderate Intensity Exercise. *156*, 111672. https://doi.org/10.1016/j.jbiomech.2023.111672.
10. Faber, G.S.; Chang, C.C.; Kingma, I.; Dennerlein, J.T.; van Dieën, J.H. Estimating 3D L5/S1 Moments and Ground Reaction Forces during Trunk Bending Using a Full-Body Ambulatory Inertial Motion Capture System. *49*, 904–912. https://doi.org/10.1016/j.jbiomech.2015.11.042.
11. Karatsidis, A.; Bellusci, G.; Schepers, H.M.; De Zee, M.; Andersen, M.S.; Veltink, P.H. Estimation of Ground Reaction Forces and Moments During Gait Using Only Inertial Motion Capture. *17*, 75. https://doi.org/10.3390/s17010075.
12. Johnson, W.R.; Mian, A.; Robinson, M.A.; Verheul, J.; Lloyd, D.G.; Alderson, J.A. Multidimensional Ground Reaction Forces and Moments From Wearable Sensor Accelerations via Deep Learning. *68*, 289–297. https://doi.org/10.1109/TBME.2020.3006158.
13. Amrein, S.; Werner, C.; Arnet, U.; de Vries, W.H. Machine-Learning-Based Methodology for Estimation of Shoulder Load in Wheelchair-Related Activities Using Wearables. *23*, 1577. https://doi.org/10.3390/s23031577.
14. Liu, D.; He, M.; Hou, M.; Ma, Y. Deep Learning Based Ground Reaction Force Estimation for Stair Walking Using Kinematic Data. *198*, 111344. https://doi.org/10.1016/j.measurement.2022.111344.
15. Dorschky, E.; Nitschke, M.; Martindale, C.F.; family=Bogert, given=Antonie J., p.d.u.; Koelewijn, A.D.; Eskofier, B.M. CNN-Based Estimation of Sagittal Plane Walking and Running Biomechanics From Measured and Simulated Inertial Sensor Data. *8*.
16. Wu, W.; Lee, P.V.S.; Bryant, A.L.; Galea, M.; Ackland, D.C. Subject-Specific Musculoskeletal Modeling in the Evaluation of Shoulder Muscle and Joint Function. *49*, 3626–3634. https://doi.org/10.1016/j.jbiomech.2016.09.025.
17. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
18. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. Software available from tensorflow.org.
19. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 785–794. https://doi.org/10.1145/2939672.2939785.
20. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2.

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.