Binary Classification of Vertical Ground Reaction Force Time Series in Human Gait

Austin L. Mituniewicz, Tamer A. O. Abdelfatah, Caleb R. Darr

***Abstract*—Individuals who have survived a stroke are much more likely to fall than age-matched peers. These falls are often caused by internally generated trips where the foot leaves the ground and then prematurely contacts it prior to the desired location for the succeeding step. Another characteristic of stroke gait can is known as "scuffing," which is when the paretic side foot completely clears the ground during swing but often at a delayed toe-off time. This may also be accompanied by a change in slope or reloading of the vertical ground reaction force's (vGRF) descending limb. In current gait analyses, these steps are often averaged together with the other stroke gait toe-off archetype where toe-off occurs earlier than that of an age-matched peer. This grouping of different toe-off types may contribute not only to variability reported between individuals in studies of stroke gait, but also high intra-subject variability that may occur when different classes are averaged together. Therefore, the goal of this study is to test if scuffs can be separated from non-scuffs, using meaningful features extracted from the vGRF timeseries or the timeseries itself.**

***Index Terms*—Binary Classification, Ground Reaction Force, Fandom Forest, Logistic Regression, Long Short-Term Memory, Time Series Classification**

I. Introduction

FALLS are a leading cause of injury and hospitalization for individuals following stroke [1]. The detrimental consequences of falls include fractures, soft tissue injury, hospitalizations, decreased mobility, and negative psychological outcomes such as increased fear and anxiety [2]. Falls after stoke are particularly troubling due to the compounding levels of debility that yield worse rehabilitation outcomes, loss of independence, and chronic disability [3]. For community-dwelling individuals with chronic stroke, falls occur most often during walking [1] and the majority of falls are due to trips [4]. A trip-related fall could occur because of either 1) external factors: gait remains unchanged, but the environment changes (i.e., slippery floor, carpet, or obstacle), or 2) internal factors: the environment remains unchanged, but altered step-to-step gait kinematics elicit an unintentional foot contact with the ground during swing phase [5], causing a stumble. For individuals who have had a stroke, falls are most often 'self-induced', by internal factors, rather than induced by external factors [6].

A. Background on Dataset

The data presented here is from one individual who has suffered a stroke (Male; Wt; Ht; Age; Right Side hemiparesis; TYPE OF STROKE). The participant completed 22 walking trials from 0.2 to 0.4 m/s under while performing an additional cognitive tasks (e.g. casual conversation, counting by 7s, backwards spelling) to promote internally caused trips. Of the 22 trials, data from 19 were available. These 19 trials had a total of 564 gait cycles of the paretic limb. Gait events were found using a vertical ground reaction force (vGRF) threshold and each gait cycle was normalized to heel-strike. An original MATLAB GUI then separated trips from non-trips by defining a trip as a vGRF signal above 3N after the vGRF was found to be 0N. All non-trips gait cycles were then manually classified by a single researcher into one of three groups: "non-scuff," "scuff," or unsure. The scuff classification was observationally defined as a prolonged offloading period that often included some reloading prior to toe-off. Whereas the non-scuff classification was given to all gait cycles that were largely absent of the features seen in scuffs. The samples put into the unsure classification exist somewhere along the non-scuff to scuff spectrum that the researcher did not feel confident classifying into that binary. In total, the number of samples in the non-scuff and scuff classes were 224 and 104, respectively. It is estimated that the researcher would change about 1-5% of the samples in each class to that of another class if they were to manually relabel. Twenty-four features were defined, from which different subsets were tested. An original feature space of 24 features were defined by the researcher using temporal, vGRF, vGRF , and anterior-posterior GRF data.

B. Problem Definition

For individuals recovering from a stroke, an injurious fall can occur due to a stumble or "intrinsically generated" trip while walking (i.e., the swinging foot contacts the ground). Current approaches to prevent falls either teach reactive responses to a trip or train individuals post-stroke to minimize the impairments associated with falls (e.g., strength, balance, ROM). Although preventive training can reduce intrinsically generated trips for otherwise healthy older adults, deficits in voluntary muscle activation limit the efficacy of such training in individuals following stroke. Furthermore, whereas teaching compensatory

This work was completed to fulfill the course requirements of COMP 562, Introduction to Machine Learning and was supported in part by the National Institutes of Health (NIH) under Grant 12345678910

Austin L. Mituniewicz is with the Joint Department of Biomedical Engineering, University of North Carolina (UNC) at Chapel Hill, Chapel Hill, NC 27599 USA, and North Carolina State University (NCSU), Raleigh, NC27695 USA (e-mail: amituniewicz@unc.edu).

Tamer A. O. Abdelfatah is with the Computer Science Department, University of North Carolina (UNC) at Chapel Hill, Chapel Hill, NC 27599 USA, (email: tameros@unc.edu

Caleb R. Darr was with the Joint Department of Biomedical Engineering, University of North Carolina (UNC) at Chapel Hill, Chapel Hill, NC 27599 USA, and North Carolina State University (NCSU), Raleigh, NC 27695 USA (email: cdarr@unc.edu).

reactions can minimize the impact of a fall, such techniques cannot eliminate falls. Since even a single trip can result in a damaging fall, we are seeking a method of classifying normal vs trip-prone walking conditions by utilizing the vertical ground reaction force time series.

C. Motivation

The primary motivation for this work is to reduce the time burden required in the manual classification of non-trip toe-off events. The rationale behind implementing a machine learning solution is driven both by the high inter and intra-subject data variability and the relatively large number of potential features/input sources. Lastly, the implementation of any solution can be used to classify toe-off events that an experimenter is unsure of the classification.

D. Related Works

The methods by which data scientists and researchers have classified time series data are varied in approach and application. Various machine-learning models have been investigated for general-purpose time series classification along with specific models optimized for classifying vertical ground reaction force data.

1) General Time Series Classification Review: The classification of time series data can be subdivided into feature-based, model-based, and distance-based approaches [7]. Each comes with its distinct advantages and limitation; however, each can be utilized for general-purpose time series classification. A feature-based model transforms the data into feature vectors and then utilizes conventional classification methods to separate the data. Common classifiers are neural networks and decision trees where the data may be augmented with feature extraction using spectral methods [8]. Alternatively, a model-based method makes the assumption that all time series in a single class are characterized by an underlying model. Auto-regressive and Markov models are common examples of such model-based methods [9], [10]. The final approach commonly used for time series classification is distance-based methods. This technique uses (dis)similarity measures between series that are then used to separate each class. Common distance-based classification methods are k-nearest neighbors, support vector machines, and dynamic time warping [11], [12].

2) Ground Reaction Force Classification Review: The application of time series classification in human biomechanics commonly utilizes the vertical component of the ground reaction force to classify various human conditions that result in a manifestation in the gait cycle. Recent work by [13] utilized a time-frequency spectrogram and a deep-learning neural network to provide early identification of patients with neurodegenerative diseases. By using principal component analysis features were extracted and using a convolutional neural network a deep learning classifier was developed [13]. Another author used ground reaction force data gathered from wearable sensors to classify standing and walking states [14]. Similar classification methods were also employed by [15] to classify human vertical jump performance using the vertical ground reaction forces in linear models with dynamic time warping to reduce error.

II. Methods

Using three different classification types: Logistic Regression, Random Forest, LSTM the binary classification of normal vs abnormal gait cycles is analyzed. By approaching the classification problem from three different angles the optimal model type can be identified and then refined in future work.

A. Logistic Regression

A logistic regression (LR) is a supervised machine learning technique that can be used to calculate the probability of an observation, with predefined feature space, falling into a particular class. This method was chosen given the relative simplicity of the problem, binary classification, and the ability of the algorithm to use user defined features that have physical meaning and intuition. This space was then reduced to 3 features for the LR Figure 1. The LR was created in MATLAB and the was validated using a k=5 fold cross-validation method to be consistent with the other classifiers developed in this study. The first five folds had 45 non-scuff samples and 21 scuff samples, all of which were randomly assigned, while the last fold included 1 less sample from both classes.

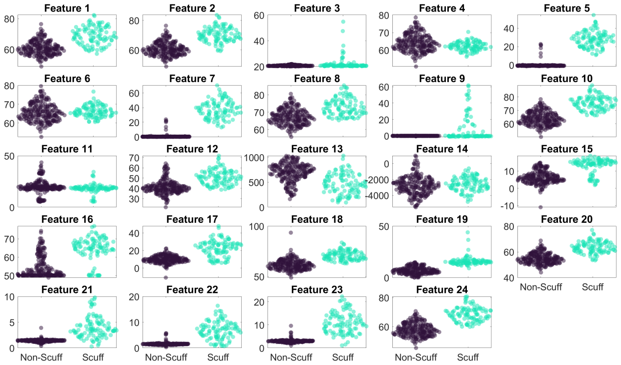

Fig. 1. User-defined feature space prior to any normalization. The logistic regression used features 7, 9, and 10; while the random forest classifications used these as well as 13, 14, 21, and 24.

B. Random Forest Classification

This model was created using the ensemble learning method, random forest binary classifier, where several decision trees are created, and the output of the forest is that which the majority of trees select. Fifty-six decision trees were utilized with a maximum depth of six. The minimum number of samples in a leaf was set to two, and the minimum number of samples required to split an internal node was set to two as well. No upper boundary was set for the number of features the random forest was allowed to try in an individual tree. Bootstrapping was used to control the level of variance, avoid overfitting and increase the general accuracy of this model. 3-fold, 5-fold, and 10-fold cross-validation were tried, all reaching the same results with the given dataset. This model was built using the scikit-learn RandomForestClassifer algorithm. Finally, hyperparameter tuning was done using the GridSearchCV algorithm, where 5760 different combinations were tried to reach the optimal parameters previously mentioned.

C. Long Short-Term Memory Network Classification

To provide a binary classification of the vertical ground reaction force a long short-term memory (LSTM) network can be used to learn long-term dependencies between time steps of the sequential data. LSTM's are a type of recurrent neural network (RNN) [16]. The advantage of RNNs is the ability to express temporal behavior from the direct connections between units of individual layers. A hidden vector **h** is updated at time steps *t* as described by [17] where tanh is the hyperbolic tangent function, W is the recurrent weight matrix, I is a projection matrix, and the hidden state h is used to make a classification:

$$h_t = tanh(Wh_{t-1} + Ix_t). \tag{1}$$

Then a softmax provides a normalized probability distribution function over each class.

$$y_t = softmax(Wh_{t-1}). \tag{2}$$

$$h_t^l = \sigma(Wh_{t-1}^l + Ih_t^{l-1}). \tag{3}$$

An improvement to a RNN a long short-term memory RNN contains a vanishing gradient problem [17]. This is addressed by incorporating gating functions into state dynamics by maintaining a hidden and memory vector at each time step to control state outputs and updates [18]. Additionally, LSTMs have difficulty with learning long-term dependencies; therefore, for the purpose of our models, the ground reaction time series will be limited to the last twenty-five percent of the time series where important features are located [19].

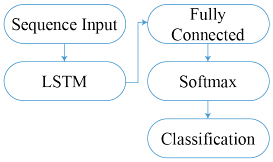

Fig. 2. LSTM network architecture for classification. The sequence input layer is followed by an LSTM layer then a fully connected layer and softmax before classification output.

The LSTM model used in this manuscript follows the network architecture illustrated in Figure 2 and the LSTM Layer Architecture illustrated in Figure 3.

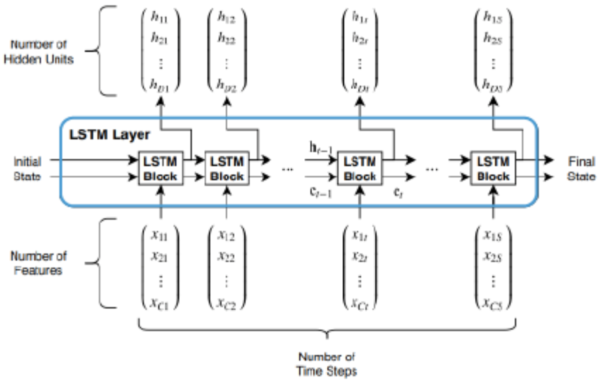
Number of Hidden Units
h_{11} h_{21} ⋮ h_{D1}
h_{12} h_{22} ⋮ h_{D2}
h_{1t} h_{2t} ⋮ h_{Dt}
h_{1S} h_{2S} ⋮ h_{DS}
LSTM Layer
Initial State
LSTM Block
h_{t-1}
c_{t-1}
c_t
Final State
Number of Features
x_{11} x_{21} ⋮ x_{C1}
x_{12} x_{22} ⋮ x_{C2}
x_{1t} x_{2t} ⋮ x_{Ct}
x_{1S} x_{2S} ⋮ x_{CS}
Number of Time Steps

Fig. 3. Time series B with C features of length S within an LSTM layer. Where ht and represents the hidden state while ct is the cell state.

To train a binary classification LSTM model for ground reaction force time series k-folds were used with k = 5, a varying number of hidden layers: 5,10,...95,100, five gradient thresholds: 0.5,0.75,1,1.25,1.5, and five minibatch sizes: 15,20,25,30,35. This cross-validation method results in a total number of 5*20*5*5 + 1 = 2501 models to be trained including the final model. Five folds were chosen so each fold would have n=69 samples and to allow for sufficient testing data after model training (276 observations in training data, 69 observations in testing data). Using MATLAB™ (R2022b, The MathWorks, Inc., Natick, MA), to perform cross-validation for its LSTM package and for the author's familiarity with the language.

III. Results

To compare the three model types for classifying human gait cycles the confusion matrix of each will be utilized. This will give an indication of the overall performance of each model with their individual strengths and weaknesses in classification.

A. Logistic Regression

Across the 5-fold cross-validation, the LR had an average overall accuracy of 97.3%, (Figure 4). The true positive rate of the scuffing class ranged from 85.7% to 100% with a standard deviation of 5.8%.

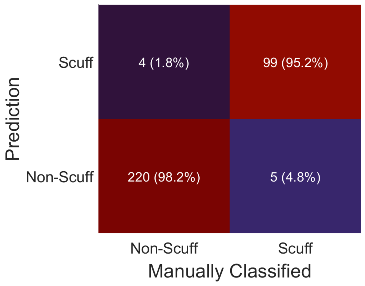

Fig. 4. Confusion matrix of the logistic regression. Numbers outside of the parenthesis are the total number of observations classified for each matrix entry and the numbers inside the parentheses are the percentages of the class that were correctly or incorrectly classified.

B. Random Forest Classification

This random forest binary classifier model seems to have performed relatively well (5 reaching an accuracy score of 98.5%. A metric that is especially important in this model is its recall score which is 100 for the positive samples given that the number of false negatives is 0 as shown in the confusion matrix. The confusion matrix is plotted using a 20 – 80 dataset split with no cross-validation. Given the context of this model and the significance of minimizing false negatives to avoid medical risk to patients, this result, having 0 false negatives, is favorable. Arguably, no compromise was made as the number of false positives is only 1 in the given test sample. The results were validated using cross-validation which showed a standard deviation in the accuracy of about 1% across several folds ranging from 3 to 10 using k-fold cross-validation.

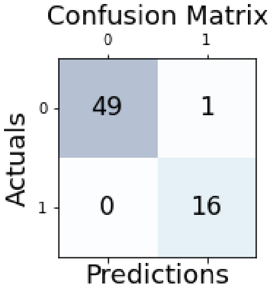

Fig. 5. Confusion Matrix for Random Forest Classifier.

C. Long Short-Term Memory Network Classification

Using the model and cross-validation scheme described in the LSTM method section Figure 6 was constructed to determine the model the performance over 5-folds.

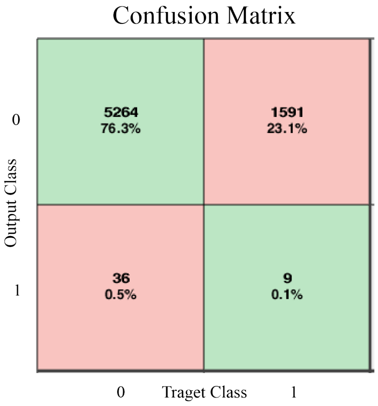

Fig. 6. Confusion matrix for a subset of cross-validated LSTM-RNN binary classification method.

IV. Discussion

Given the inherent error present in the data set, no algorithm should be perfect. The error in the data set was intentionally not cleaned to assess if any of the algorithms overfit their training data.

A. Logistic Regression

The LR performed very well given the estimated error inherent in the manually defined labels. This method was somewhat sensitive to the training data, particularly true for the scuffing class. If implemented, these issues can likely be overcome by ensembling the LR weights together and retesting or using it with combination with other methods. Overall, the LR appears the most promising of the methods tested, presenting the highest average accuracy and also potentially identifying the erroneously labeled samples in both classes. Looking forward, this method can potentially be expanded to include additional classes. Additional temporal features could be used to separate the non-scuff class into "premature" and "delayed" steps. These delayed steps could then be grouped together with those in the scuff class that exhibited little to no vGRF reloading. The LR's posterior probability could then be used to understand how "well" the algorithm estimates a given sample fits into the predefined classes. This may be important given that it is thought the classes do not exist on a binary, but rather a spectrum.

B. Random Forest Classifier

At first, the random forest binary classifier was trained with different feature spaces using 328 samples from the time series. This resulted in a low accuracy score of 75% and a very low recall score as almost all the positive samples were incorrectly classified. After adjusting the feature space to use only 7 samples from the time series, the model's performance improved dramatically as mentioned in the results section. Given the imbalanced nature of the dataset having fewer positive samples than negative ones, resampling of the data was attempted. However, both under-sampling and over-sampling did not show noticeable improvement. This model could potentially be overfitted due to mislabelling of some of the raw data samples provided estimated to be in the range of 5%.

C. Long Short-Term Memory Network Classification

Using k-fold cross-validation, an average model accuracy of 80.1% was achieved. Furthermore, a peak single model accuracy of 95.65% was achieved using the testing and training data split in the second fold of the 5 k-folds. The type two error is highly possible due to the imbalance between positive and negative gait classification in the training data.

To further improve the LSTM classification methodology transfer learning could be used to gain understanding from training models for new models. Additionally, a method developed by [16] called refinement could allow transfer learning within the same dataset. This method begins with an initial phase of optimizing hyperparameters before the refinement phase where transfer learning is coupled with a larger learning rate to overcome the local minima previously encountered in the initial phase. Using this technique our LSTM classification could be further improved.

V. Conclusion Future Work

As a tool to aid in gait cycle classification, the random forest model had the lowest type I and II error. With a type II error of zero, the random forest could be used as a first classify all of one class and then only the other classification would have to be confirmed by the researcher. This could potentially reduce the amount of time required for classification by a factor of two or three. For further research on that topic, it is perhaps worthwhile to examine the effect of gradient boosting in such an application. With gradient boosting, the trees are not going to be built independently as in random forests but rather built one after the other to complement the deficiency of the former thus resulting in an overall coherent model. This mainly falls to gradient boosting aggregating the result of each tree while building the model, not just adding the results at the end. Overall, gradient boosting performs better than random forests, but they are prone to overfitting, so that must be taken into consideration throughout. Gradient boosting also works best with minimal noise. Since the feature space used for the random forest binary classifier consisted of only 7 samples from the time series, it is perhaps going to improve on the performance currently reached

VI. Code Respository

The code used to complete this project can be found at the following repository: https://github.com/crdarr/COMP562_Mituniewicz_Abdelfatan_Darr

Acknowledgment

Thank you to Dr. Jorge Silva for providing instruction in the basic principles of machine learning models and for guidance on the specifics of time series classification. Without his assistance, this work would not have been possible.

References

[1] A. Forster and J. Young, "Incidence and consequences offalls due to stroke: a systematic inquiry," *Bmj*, vol. 311, no. 6997, pp. 83–86, 1995.

[2] V. Weerdesteijn, M. d. Niet, H. Van Duijnhoven, and A. C. Geurts, "Falls in individuals with stroke.," 2008.

[3] A. Ramnemark, M. Nilsson, B. Borssén, and Y. Gustafson, "Stroke, a major and increasing risk factor for femoral neck fracture," *Stroke*, vol. 31, no. 7, pp. 1572–1577, 2000.

[4] L. Cohen, T. Miller, M. A. Sheppard, E. Gordon, T. Gantz, and R. Atnafou, "Bridging the gap: bringing together intentional and unintentional injury prevention efforts to improve health and well being," *Journal of safety research*, vol. 34, no. 5, pp. 473–483, 2003.

[5] J. L. Burpee and M. D. Lewek, "Biomechanical gait characteristics of naturally occurring unsuccessful foot clearance during swing in individuals with chronic stroke," *Clinical biomechanics*, vol. 30, no. 10, pp. 1102–1107, 2015.

[6] L. Jørgensen, T. Engstad, and B. K. Jacobsen, "Higher incidence of falls in long-term stroke survivors than in population controls: depressive symptoms predict falls after stroke," *Stroke*, vol. 33, no. 2, pp. 542–547, 2002.

[7] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *ACM Sigkdd Explorations Newsletter*, vol. 12, no. 1, pp. 40–48, 2010.

[8] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *Acm Sigmod Record*, vol. 23, no. 2, pp. 419–429, 1994.

[9] A. Bagnall and G. Janacek, "A run length transformation for discriminating between auto regressive time series," *Journal of classification*, vol. 31, no. 2, pp. 154–178, 2014.

[10] P. Smyth, "Clustering sequences with hidden markov models, advances in neural information processing, mc mozer, mi jordan, t. petsche, eds," 1997.

[11] A. Abanda, U. Mori, and J. A. Lozano, "A review on distance based time series classification," *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 378–412, 2019.

[12] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data mining and knowledge discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[13] F. Setiawan and C.-W. Lin, "Identification of neurodegenerative diseases based on vertical ground reaction force classification using time–frequency spectrogram and deep learning neural network features," *Brain Sciences*, vol. 11, no. 7, p. 902, 2021.

[14] J. S. Park, S.-M. Koo, and C. H. Kim, "Classification of standing and walking states using ground reaction forces," *Sensors*, vol. 21, no. 6, p. 2145, 2021.

[15] M. G. White, J. Neville, P. Rees, H. Summers, and N. Bezodis, "The effects of curve registration on linear models of jump performance and classification based on vertical ground reaction forces," *Journal of Biomechanics*, p. 111167, 2022.

[16] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "Lstm fully convolutional networks for time series classification," *IEEE access*, vol. 6, pp. 1662–1669, 2017.

[17] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *arXiv preprint arXiv:1312.6026*, 2013.

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

Austin L. Mituniewicz is a proud Long Islander who applied for an analytics job with the New York Mets, his lifelong favorite team, prior to receiving his acceptance into the Joint Biomedical Engineering PhD program of the University of North Carolina at Chapel Hill and North Carolina State University. His qualifications for this job were fantastic, except for the primary one: expertise in machine learning and predictive models. Although his current plan post-graduation is to become a professor and teach biomechanics, he wants to be ready in-case life throws him a curveball.

Tamer A. O. Abdelfatah Tamer Abdelfatah was born in Cairo, Egypt in 2001. He is a Computer Engineering student from the American University in Cairo. Tamer's projects include a computer memory hierarchy simulator, a ride sharing application prototype, a RISC-V processor simulator, and a Huffman coding compression algorithm. Tamer is currently researching different machine learning models for his Egypt based healthcare start-up, Besta.

Caleb R. Darr a Hoosier at heart, was born in South Bend, IN, USA in 1998. He received a B.S. in nuclear engineering from Purdue University, West Lafayette, in 2021 and a M.S. degree in biomedical engineering from the University of North Carolina, NC, in 2022. From 2021 to 2022, he pursued new interests as a Research Assistant with the Gallippi Ultrasound Laboratory at the University of North Carolina. After finishing his MS by checking a Machine Learning course off his bucket list he is returning to the mid-west to work as a nuclear engineer in the power generation industry.