# Using cross model analysis to recognize patterns in IMU data of a wheelchair basketball player

Martijn Kok, Alejandro Oliver Llorente, Collin van Adrichem, Daan Gielen, Jake van der Valk
Applied Data Science, The Hague University of Applied Sciences

## Abstract

Fitness tracking is something people use daily. But for wheelchair users, this is hardly possible as they “don't take steps”. Wheelchair users move different in daily life and during sports, but this is barely implemented in health tracking apps. This paper focuses on using inertial measurement unit (IMU) data, from the Dutch national wheelchair basketball team, to track sprints in matches. As the goal is to classify movements random forest classifier (RFC) and recurrent neural network (RNN) are used as machine learning models to predict the actions. There is no definition of sprinting within wheelchair basketball, so the dataset is partially defined. Comparing the RFC with the RNN positive sprint predictions, it is possible to add data and define sprints in the dataset. An RFC model was designed, using the dataset, to classify sprints from IMU recordings.

## Introduction

Many sportsmen use fitness trackers to improve their insight into their performance. In [1], fitness trackers are used in sports like rugby and football with gratifying rewards. This helps players to determine their fitness by the number of steps taken in a game and in daily activities. But like this wheelchair athlete mentioned in research [2]: “But, I don’t take steps”, the current methods of fitness tracking are not built for wheelchair users. In research [2] is shown that there are currently no good fitness trackers for wheelchair sports like basketball, rugby and tennis. Even though 80% of the researched wheelchair athletes were interested in tracking their activities to gain more insights into their game and daily activities. So, this paper will focus on the possibility of using machine learning in combination with IMU’s (Inertial Measurement Unit), to help wheelchair athletes to get more grip on their physical activities. This research will discuss how machine learning can recognize sprints from wheelchair IMU recordings.

To get a better insight into the current situation, research was made into related studies. There are barely any studies done into wheelchair sports movements with IMU data, but there have been studies focused on recognizing activities with IMU data. though these mainly focused on more basic movements like walking and running as seen in [3] and [7].

The idea of using machine learning to classify movements thanks to machine learning is nothing new, as can be seen in [4], a Recurrent Neural Network (RNN) was used to successfully classify walking and sprinting in IMU recordings. This was also done in [7], here they used an RNN to classify movements with IMU sensors. Another commonly used model for classifying single things was a Random Forest Classifier. In [3], an RFC was used to classify different sports based on wearable sensors. In [6], Random Forest was used to classify crops. While these aren’t the same as a wheelchair movement it still is something that must deal with a lot of variety in the classifying options. This is also the case for the sports movements because just like crops, none are the same. Provided the positivity of these results, RFC and RNN are chosen to be used in this paper to detect wheelchair basketball movements.

The Dutch national wheelchair basketball team is one of the sports teams that use IMU data to get insights into their games and trainings. The team is at the frontier of developing better analyzation methods [8]. Rotational and straight-line speed and acceleration are some of the key aspects of this data. With the use of RFC and RNN models, stated before, it is possible to classify different in-game movements. This will not only increase the insight in gameplay but also the efficiency and accuracy of match analysis. As of now, the games are inspected by trainers through video. This paper will be focusing on the modifications and implementations of the data and models to (predict/analyze) wheelchair sprints in-game.

# Dataset

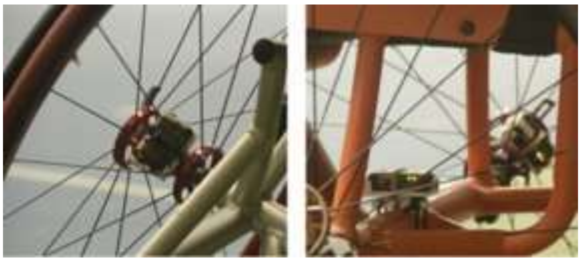

*Figure 2: Wheelchair with gyroscopes*

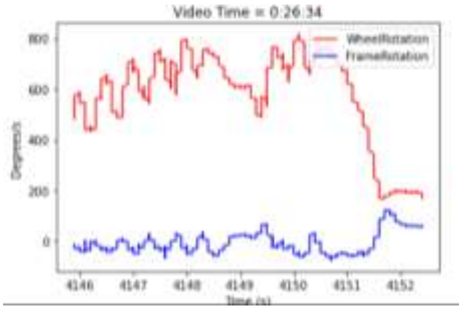

*Figure 1: Graph of WheelSpeed and FrameRotation*

This research features 24 unique datasets, from twelve different players over a total of two played basketball matches. Each dataset contained 113 minutes of IMU sensor data. The corresponding videos of the two played matches were also acquired for evaluation purposes.

The dataset consists of IMU sensor data of two separate sensors. These sensors contain a gyroscope that measures over three-axis (XYZ). These gyroscopes measure the rotational speed of the frame and the right wheel separately see Figure 2. As of today, the raw data is processed by performing several calculations [8]. This results in the dataset that contains sixteen features with a measuring frequency of 100Hz. The four most important features of this dataset are the speed and acceleration of the wheel and the rotation of the frame.

The graph in Figure 1 shows the Wheel rotation (red) and frame rotation (blue). This graph is a perfect example of sprinting behaviour the beginning of the graph shows little peaks that increase in magnitude, which means the player is accelerating and the speed is increasing. Each peak is a push-off in the wheelchair. While the wheel rotation increases the frame rotation fluctuates around zero, which indicates that this player is going in a straight line. These are the characteristics that show a player is sprinting.

When comparing the dataset to the corresponding video some flaws were detected. The first is that the dataset is not synchronized with the video's timeline. During warm-up the sensors are turned on, it already starts to measure data points during warm-up, also the quarter timeouts are cut out of the video but are still present in the dataset. This does not only mean that the sensor data does not run parallel to the time in the video, but also that a big part of the sensor data cannot be used since data cannot be validated by checking the video. These irregularities will cause issues with the accuracy of the model. 52,7% of the dataset is useable match time. The unusable data is removed from the data set.

The dataset for both the RFC and the RNN the dataset was prepared. This was done by:

- Changing all the 'NaN' values to 0.
- Expanding the dataset with three more features. First is the differential of the rotational wheel speed X, this feature gives a better understanding of the increase of the acceleration. The second and third are the low pass filtered signals of the wheel rotational speed X and the frame rotational speed Z, these features are added to smooth out the high peaks in the signal. These high peaks are a disturbance to the model and can cause false readings.
- The dataset is split into chunks of one second with an overlap size of half a second. These chunk sizes are chosen since there are no sprints shorter than one second, the overlap size of half a second is set to enhance the accuracy of detecting sprints.
- Balancing the dataset for the RFC and the RNN, both models have a different way of balancing data. The RFC model used the function class_weight = 'balanced' to balance the classes based on class frequency in the dataset. This ensured the RFC model had no preference for a specific result. The RNN model has a balancing function to improve the results of the model. This balancing function used oversampling of the positive sprint samples to improve the number of positive sprints in the training set.

# Solutions

## Training

The training for both the RFC and RNN model is set to maximize the recall score. By analyzing this score, we would know if the model is accurate. The precision score is not usable because the dataset contains a incomplete number of tagged sprints. Therefore it is impossible to compare the classification results with only the tagged sprints. To increase the precision and recall of the training and validation, the number of positive sprints in the dataset needed to be increased by finding undefined sprints.

The first collected dataset had 2.3%. Although the goal of the project is to detect sprints it is not possible to balance the dataset by copying and pasting a lot of the positive values. Wheelchair basketball is a dynamic sport so no sprint/rotation will be executed the same way as the previous. So when the data is balanced as previously mentioned the model is very likely to overfit and miss a lot of the movements. A Decision tree was created to try and increase the positive results in the dataset, this model was chosen for its simplicity and usability. The model searches for sprints in the dataset, the false positive classifications of this model were reviewed by hand to see if the results were actually (unknown) sprints.

To speed up this process, the random forest classifier and recurrent neural network are used together. Both models are trained and validated with the same data. The results are plotted in a confusion matrix to see the results of each model. Running this resulted in two prediction lists with a lot of false-negative timestamps. The false negatives of both models are used to improve the positive sprints in the partially defined dataset see Figure 3. This is done by comparing the false-negative timestamps of both models. If these timestamps match there is a big chance that both models predict it right. These timestamps are added to the dataset as sprints. This method is validated by randomly checking detected sprints, before adding all detected sprints to the dataset. This check is done by watching the corresponding video of the timestamps. This method proved to be efficient and accurate and was repeated seven times on the training and validating dataset, where each time the training and validating parts were changed.

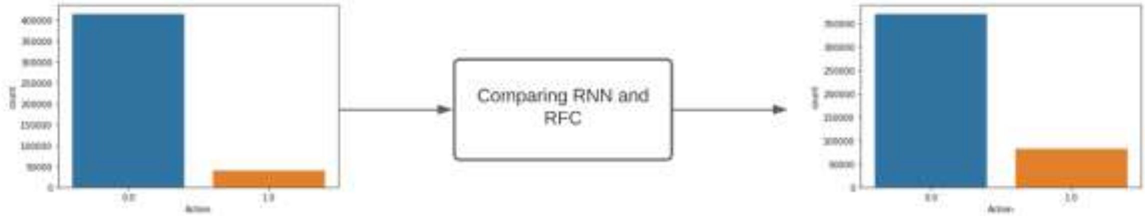

*Figure 3: Completing Dataset through comparing RNN AND RFC*

## Validating

The validation of the dataset is done through cross-validation. This meant the dataset is divided into the quarters of the game: Q1, Q2, Q3 and Q4. Quarters Q1, Q2 and Q3 are used as a training set and Q4 is used as the valid set. This is done because the RNN requires a synchronous data set to work. The model will use a separate player to test the model. This makes it a 75% to 25% split see Figure 4**Error! Reference source not found.**. To have an accurate comparison between the models RFC and RNN will both use this data-split setup.

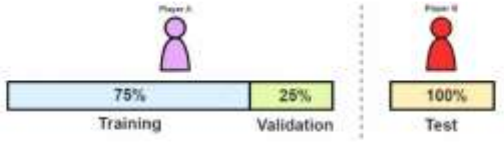

*Figure 4: Dataset division*

## Machine learning models

The implementation of the RFC and RNN used different methods. These methods are used to ensure the models have the best recall with precision above 0.50 and as little overfitting as possible.

### RFC

The feature engineering for the RFC model was done by calculating the importance of each feature this importance was calculated based on the mean decrease in impurity from Sklearn [9]. The following features showed the best results: Timeline, Frame speed, Frame acceleration, Differential of wheel rotational speed X, Frame rotational acceleration, Frame rotational speed Z, Wheel rotational speed X, Filtered frame speed Z, Filtered wheel speed X. These features each consists of chunks of one second, these chunks need to be summarized by a single integer for each feature to make sure the RFC can use this as an input. This is done by taking the mean or max value of the data in chunks. The operation that is used for a feature is based on the influence on the outcome of the model. After the feature engineering Grid Search was used to tune hyperparameters for the number of trees (number of classifiers), the criterion (measures the quality of a split) and the max depth of the model.

### RNN

The RNN model, see Figure 6, uses different functions in the training process that were not used by the RFC models. One of these functions is the data loader. The data loader will feed the training data to the model in chunks of 64 units, this will help the model prevent overfitting. This will also lower the use of GPU memory and improve how much the model learn in every epoch.

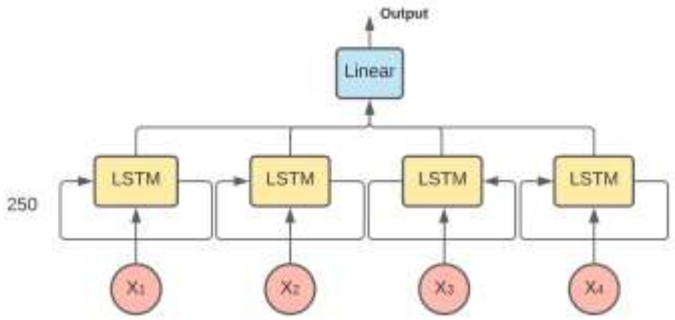

*Figure 6: RNN model*

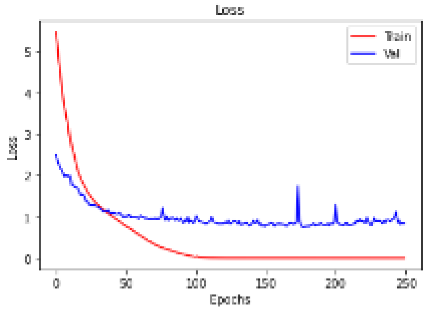

*Figure 5: Epoch vs Loss RNN*

The features of the RNN are: Wheel Speed X, Frame Speed Y, Frame Acceleration, Differential of Wheel speed X. These features were selected by setting a benchmark when the model was using all features. This benchmark was used to calculate the impact of each feature. The features with the most impact on the RNN model were chosen. In Figure 5 is the loss over epochs shown to prove the RNN model is not overfitting or underfitting.

## Results

### Dataset

The first method of getting more positive sprints in the dataset was by the use of a decision tree. This method increased the positives sprints in the dataset from 2.3% to 8.6%. To further improve the results the method of comparing the RFC and RNN model was used. This increased the positive sprint samples in the dataset from 8.6% to 17.1%.

### Machine learning

All models were trained and validated with a dataset that contained the data of one specific match of one specific player. The training was done on a Jupyter notebook server owned by the Hague University of Applied Science.

#### *Decision tree*

The decision tree that was used to improve the amount of positive sprints in the dataset was also validated on the dataset with a positive sprint count of 8.6% see Figure 7. The accuracy of this model is 0.881, the precision is 0.436 and the recall is 0.943.

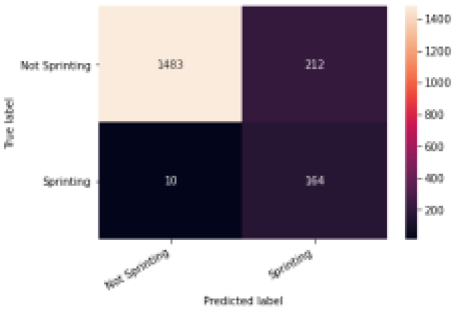

*Figure 7: Confusion matrix Decision tree*

### Random Forest Classifier

The RFC was validated two times, one time with the dataset with 8.6% positive sprint in the dataset and one with 17.1% positive sprint in the dataset. The results are shown in Table 1Table 1.

*Table 1: Accuracy/Recall/Precision from RFC model with 75% train and 25% valid*

| | Accuracy | Recall | Precision |
|---|---|---|---|
| Dataset with 8.6% positive sprints (Figure 8) | 0.950 | 0.822 | 0.694 |
| Dataset with 17.1% positive sprints (Figure 9) | 0.962 | 0.904 | 0.890 |

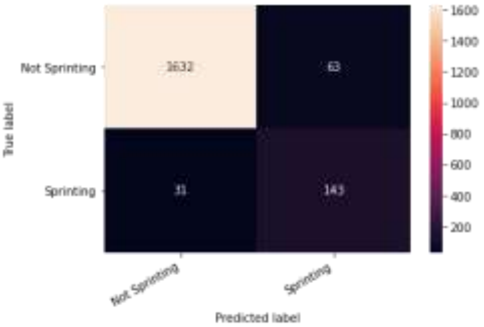

*Figure 8: Confusion matrix RFC 8.6%*

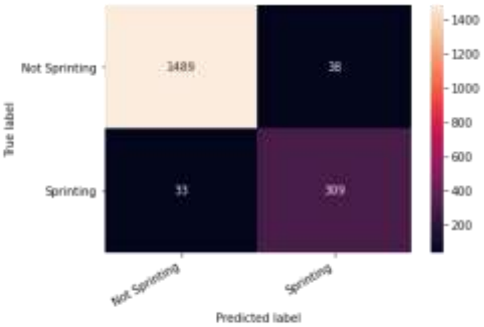

*Figure 9: Confusion matrix RFC 17.1%*

### Recurrent Neural Network

The RNN model was also validated the same way as the RFC, with a dataset of 8.6% positive sprint and with a dataset of 17.1% positive sprints.  The results are shown in Table 2Table 2.

*Table 2: Accuracy/Recall/Precision from RNN model with 75% train and 25% valid*

| | Accuracy | Recall | Precision |
|---|---|---|---|
| Dataset with 8.6% positive sprints (Figure 10) | 0.892 | 0.908 | 0.459 |
| Dataset with 17.1% positive sprints (Figure 11) | 0.925 | 0.911 | 0.634 |

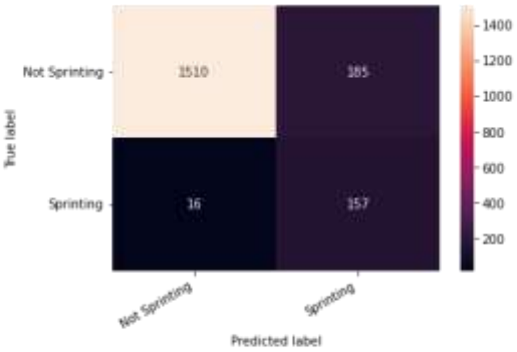

*Figure 10: Confusion matrix RNN 8.6%*

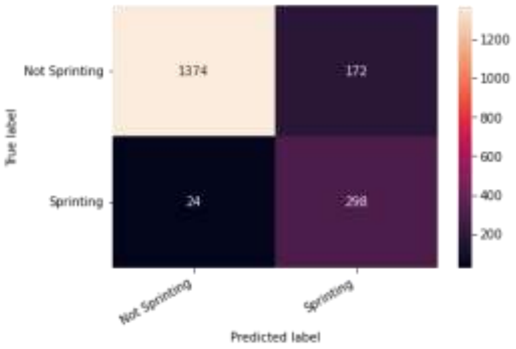

*Figure 11: Confusion matrix RNN 17.1%*

### Test set

From the validation the RFC has the best accuracy, recall and precision, this is done with a dataset of one specific player, named player A. To ensure the RFC model does not just work on one specific player the model has been given a completely new test dataset, named player B. After running the RFC on this new data set, the detected sprints are plotted. In Figure 13 a plot of a sprint of player A is shown and in **Error! Reference source not found.Error! Reference source not found.** a detected sprint of player B. Both graphs show the patterns that match a sprint. To ensure the timestamp the model tagged as sprints are correct, the video data is also compared to every timestamp the RFC tagged in the dataset of player B to check whether this player is actually sprinting at that moment. After checking all the detected sprints, it has been concluded that the model had a precision of 91.67 % and an unknown recall. The recall is unknown because not all sprints in the dataset are known.

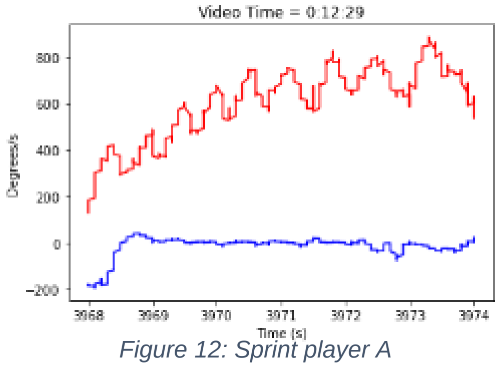

*Figure 12: Sprint player A*

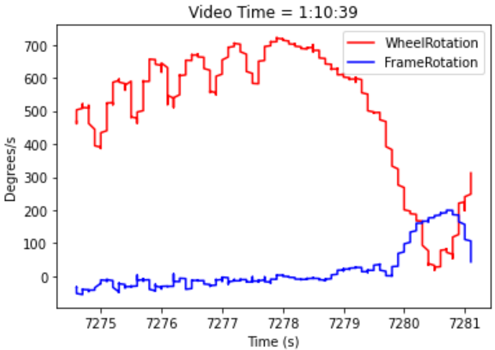

*Figure 13: Sprint player B*

## Conclusion

In this paper we investigated, developed and applied a method to classify wheelchair basketball sprints through RNN and RFC models. The method classifies sprints from partially defined IMU data. The models cannot correctly classify sprints with the starting dataset due to the lack of tagged sprints. By comparing and reviewing the false positives in the RFC and RNN model, undefined sprints are detected and corrected. Through adding these sprints to the dataset, it gets a more defined ground truth. The completed dataset has a feature competent “action” for sprint identification. Because of this feature component the RFC model classifies the sprints correct. We can conclude that this proposed method is able to classify sprints in wheelchair sports.

## Discussion and Recommendation

The study set out to classify wheelchair basketball sprints within IMU data using machine learning models. Model validation and testing showed that it is possible to classify sprints in match data. The validation process shows that the RFC model is better than the RNN model, using the finalized dataset RFC has significantly better accuracy and precision while RNN had a slightly better recall of 0.7%. Using an RFC model to analyze the player data gives a precision result of 91.67%. These findings show that wheelchair team sports can effectively monitor in-game sprints through IMU data using machine learning.

The dataset was found to be insufficient while training the RFC and RNN. The dataset didn’t contain a strong enough ground truth, this is because of the lack of tagged sprints. The ground truth was improved by expanding the number of tagged sprints with new sprints. These sprints were found with a technique where the results of the RFC and RNN were compared. The ground truth was improved by increasing the positive sprint samples from 2.3% to 17.1%. Training the RFC and RNN models with the improved dataset improved the accuracy of both models by around 2%. Both models use the features with the most impact and can detect the length of a sprint with a deviation of one second. Due to time limitations, these techniques couldn’t be applied to more than 2 players.

The research question of this paper was: How can an RFC and an RNN be used to classify sprints in partially defined IMU data? This research paper shows that it is in fact possible to detect sprints out of IMU data with the use of an RFC and an RNN. Currently, we have used those models on two players. These players both have a similar paralysis level. Because of the nature of the sport, multiple different paralysis levels occur. We believe that it might be possible to see some patterns between the different paralysis levels. It might even be possible to predict a paralysis level. This paper focuses on sprints, it would be interesting to apply these techniques to different movements like Sudden change in possession (Fast Breaks and Fast Defense) and collisions, or predict the state of a player, they are experiencing fatigue or is even overloaded. Another aspect that would be interesting to look into whether this technique can also be implemented in the daily life of a wheelchair user, to track and improve their overall health. Looking back at this project, a big step is taken into improving the wheelchair department of fitness tracking. But there are still a lot of interesting unanswered questions, as mentioned before. We do recommend someone to investigate these questions.

# Sources

[1] Smith, N. B. K. (2016, 15 april). The wearable tech giving sports teams winning ways. BBC News. Geraadpleegd op 4 januari 2022, van https://www.bbc.com/news/business-36036742

[2] Carrington, P., Chang, K., Mentis, H., & Hurst, A. (2015). “But, I don’t take steps”. Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS ’15. https://doi.org/10.1145/2700648.2809845

[3] Badar Ud Din Tahir, S., Jalal, A., & Batool, M. (2020). Wearable Sensors for Activity Analysis using SMO-based Random Forest over Smart home and Sports Datasets. 2020 3rd International Conference on Advancements in Computational Sciences (ICACS). https://doi.org/10.1109/icacs47775.2020.9055944

[4] Sherratt, F., Plummer, A., & Iravani, P. (2021). Understanding LSTM Network Behaviour of IMU-Based Locomotion Mode Recognition for Applications in Prostheses and Wearables. Sensors, 21(4), 1264. https://doi.org/10.3390/s21041264

[5] Fernández-Delgado, M., Cernadas, E., Barro, S., & Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems?. The journal of machine learning research, 15(1), 3133-3181.

[6] Reuß, F., Greimeister-Pfeil, I., Vreugdenhil, M., & Wagner, W. (2021). Comparison of Long Short-Term Memory Networks and Random Forest for Sentinel-1 Time Series Based Large Scale Crop Classification. Remote Sensing, 13(24), 5000. https://doi.org/10.3390/rs13245000

[7] Drumond, R. R., Marques, B. A. D., Vasconcelos, C. N., & Clua, E. (2018). An LSTM Recurrent Network for Motion Classification from Sparse Data. In Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (Vol. 1, pp. 215-22).

[8] van der Slikke, R. (2018). Out of the lab, onto the court: Wheelchair Mobility Performance quantified.

[9] *Feature importances with a forest of trees*. (z.d.). Scikit-Learn. Geraadpleegd op 10 december 2021, van https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html#:%7E:text=A%20random%20forest%20classifier%20will%20be%20fitted%20to,accumulation%20of%20the%20impurity%20decrease%20within%20each%20tree.