Table of Contents[J(1]

Introduction

Over the past 18 weeks, multiple groups of 6 students have worked on different projects with a focus on applied data science. During these 18 weeks, the students got classes on applied data science related topics and project-related topics. This research paper describes the multiple angles the project group took, to answer the research question defined in the plan of approach.

Here follows a little background story of the project wheels:

A new 3-year cycle begins. With the Paralympic games finished every country will look at innovations and better training methods to stay or get on top at the 2024 Paralympic games. Compared to the Olympic Games where top sport is the crucial factor, innovation plays a significant role in the Paralympic games.

This year the female Dutch wheelchair basketball team won the 2020 Paralympic games in Japan. This is now the team to beat. Rienk van der Slikke works as an embedded scientist for the women's basketball team. The use of IMU sensors (internal measurement units) was introduced by Rienk. A IMU contains accelerometers and gyroscopes these sensors measure linear acceleration and rotational speed[2]. These are used improve the statistical insights from training sessions and matches of the players. Right now, Rienk works with MATLAB to process the extracted data from the IMU's. This MATLAB file calculates the speed, acceleration, and rotation of the wheelchair. This data already gives the team insights to improve on. To possibly get even more out of the data Rienk contacted the Hague university applied data science to try and help him. In the next 18 weeks, a group of 6 students will help Rienk achieve this goal and help the national wheelchair basketball team so they can also be the champion at the next Paralympic games.

- Who are we
- Research question
- What is the goal
- For the gold winning Paralympic basketball players

1. Problem description

The goal of this project is to make a wheelchair basketball-specific analysis of both trainings and matches. Movements like fast offence, fast defense, collisions or ball possession can improve the team's insight. At the moment the complete video gets analyzed by a human in order to extract the important timestamps. The goal of the project is to automate this process by finding patterns with the use of machine learning. The data consists of different variables measured by an IMU. Multiple machine learning techniques will be researched and tested on the IMU data to conclude the best training and testing results. With these techniques the project group will try to identify the following wheelchair basketball-specific movements: sprinting, collisions and rotations.

1.1. Research question and sub questions

To get a greater view of the result a research question with sub-questions were defined in the plan of approach. These questions are:

- Research question:
  - How can IMU data be used to identify wheelchair basketball-specific movements?
- Sub-questions:
  - Which form of data processing will be used?
  - Which specific movements can be detected?
  - Which sensor data is used for each movement?
  - Can movements be used to predict fatigue?
  - Can movements be used to detect overload?

- Match data gets tagged by hand takes a lot of time
- No easy statistical insight in the complete game

2. Dataset

The dataset contains 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. [insert picture?] As of today, the problem owner processes the IMU data by performing several calculations. [refer to rienks paper] This results in the starting dataset that contains 16 features. For this project the project group received these IMU datasets of several players over 2 played games with the corresponding videos. For this project the project group received the sensor data of several players of 2 played games with the corresponding videos. The group decided to use one player to train and validate the models on and another player to test this model with.

Being IMU sensor data, this brings some difficulties that need to be solved before using, chapter 2.1. goes deeper into these problems. A big part of solving these issues is done by data preparation. Chapter 2.2. goes deeper into the preparation of the dataset.

2.1. Problems with the dataset

The provided dataset is raw data from the IMU's combined with the calculations of the problem owner and an excel sheet with the tagged timestamps. The first issue with the dataset is that the data from the IMU sensor is not synchronized with the video. The team starts the measurements during the warm up. This means the sensor data does not run parallel toe the time in the video.

The second problem is that there are not enough true positives in the data set, reason being that at the moment specific wheelchair actions like collisions are tagged by human eye with the help of video material. This means that the tagged timestamps are all subjective and that there is no real definition of movements like sprints, rotations and collisions.

Solutions

In order to solve the first problem, the video needed to be synchronized with the IMU data. This was done by searching the video timestamps in the plotted data of the IMU sensor. The IMU data was plotted with a range of 20 seconds (10 before and after timestamp), so that the precise time could be noted in the IMU data. Once this was done for all the quarters the video was synchronized to the IMU data.

The second problem the project group encountered was an insufficient amount of true positives. This was solved with the use of two different techniques.
The first action that the project group wanted to detect was sprinting. Before this action could be detected the dataset needed to be improved[A(2]. This was done by creating a machine learning model that searches the data for sprints, the false positive results of this model were reviewed by hand. This process took a lot of time. In order to speed up this process, two new machine learning models were created that could detect sprints (see chapter X). The positive results of the two models, were compared. When both models gave a true on the same timestamp, this timestamp was set to sprinting and was added to the training dataset. In order to validate this method of training the models, the found sprints were plotted and randomly checked by a group member with the help of the match video. This method was repeated several times till the group found the results sufficient.

2.2. Data preparation

Before the data can be loaded into the different machine learning models, it needs to be prepared. The data preparation for this project can be divided into two parts, data preparation for predictive models like decision tree and random forest and data preparation for neural networks. The first step of the data preparation is equal for both the predictive models and the neural networks. This step is changing al the 'NaN' values to 0. The next steps for the different machine learning models can be found bellow.

Predictive models

The dataset for both the predictive models goes through a data preparator. This data preparator splits the data set into chunks of 1 second, with an overlap of half a second. After splitting the data, the preparator returns for every feature either the max value or the mean value, depending on what gives a more precise outcome for the models.

For detecting sprints one feature was added by the project group. This was the differential of the wheel rotational speed in the axis X. The reason being that the differential of this feature gives a clearer view on acceleration and deceleration. Due to the method of training the dataset of sprinting there was no need to balance this dataset for predictive models.

Neural networks

Neural networks use a data set which is divided in chunks of 1 second with an overlap of half a second. After splitting up the dataset in chunks, the complete data set is divided in 75% train data and 25% test data. The training set needs to be balanced. This is done through oversampling on the X and y -train by duplicating all the positive samples once. The complete training data set is then put in a tensor[A(3] which goes into a data loader. This data loader, loads the training chunks of the data set in baches of 64 chunks in the model, till the complete till all the chunks are loaded into the model.

3. Predictive models

In the beginning of the project, the group researched multiple predictive models and neural networks[X]. After this research 4 machine learning models were chosen. In the testing phase where all the models were build and tuned to the IMU dataset, 2 machine learning models stood out. The Random forest and the recurrent neural network. Since these two models had the most potential these were chosen to continue on by further improving the data set and testing on an other player. Below the four chosen machine learning models with there results and a little description can be found.

3.1. Predictive models

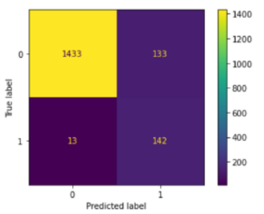
The first found predictive model was the decision tree. After some more research the group stumbled up on the random forest. Since this model showed more potential the group decided to continue with this model. Underneath both the decision tree and random forest results are shown.

Decision tree

The decision tree showed the best results with the use of the following features:

-

This model has an accuracy of 0.915, a precision of 0.753 and a recall of 0.915. These numbers are this high due to the amount of true negatives. In order to get a more representative view also an confusion matrix is made (Figure X).



Random forest

In this project a random forest classifier was chosen as machine learning model because it is widely proven to be a model of high accuracy over a wide range of applications.[…]. In the research paper made by the Vienna University of Technology, it shows how RFC is used to classify crops over time series classification where it got an overall accuracy of 81%.[Remote Sensing | Free Full-Text | Comparison of Long Short-Term Memory Networks and Random Forest for Sentinel-1 Time Series Based Large Scale Crop Classification (mdpi.com)] Also a study performed by dept. of Computer Science Air University Islamabad in Pakistan showed great results from classifying different sports based on wearable sensors and a random forest classifier.[Sci-Hub | 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) | 10.1109/ICACS47775.2020.9055944]

- Confusion matrix
- Accuracy
- Precision
- Recal
- Explanation why these are high (lots of true negatives)

3.2. Neural networks

Recurrent neural network

In a research paper made by the University of Bath in the UK they showed how a Recurrent Neural with LSTM (Long-Short-Term-Memory) could be used to classify walking and sprinting through IMU data with a accuracy of more then 90% [https://www.mdpi.com/1424-8220/21/4/1264/htm].Furthermore a research done by a the Federal Fluminense University in Brazil also showed promising results for the use of a RNN with LTSM for classifying movements with IMU sensors [https://pdfs.semanticscholar.org/6e43/75e997c8aa96175efcf3cddf0f1147157362.pdf]. These research papers convinced us to make a Recurrent Neural Network with a Long-Short-Term-Memory.

- Confusion matrix
- Accuracy
- Precision
- Recal
- Explanation why these are high (lots of true negatives)

Implementing

Both models were trained to maximize recall score. This way the project group would know if the model is accurate. If the model, then shows false positives it is likely those are still undefined sprints. To improve the validation the project group needed to increase the positive sprints in the dataset, this would prove to be a lot of work. The first collected dataset had 2.3%. Although the goal of the project is to classify specific movements it is not possible to balance the dataset by copying and pasting a lot of the positive values. Wheelchair basketball is a dynamic team sport so no sprint/rotation will be executed the exact same way as the previous. 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 positives results in the dataset, this model was chosen for its simplicity and usability. The model searches for sprints in the dataset, the false positive results of this model were reviewed by hand to see if the predictions were actually (unknown) sprints. this work resulted in 8.6% sprint positives.

In order to speed up this process, the random forest classifier and recurrent neural network were used together. Both models were trained with the same data and also validated with the same data. This resulted in two predictions with a lot of false negatives. The false negatives of both models were used to improve the positives sprints in the validating set. This was done by comparing timestamp of the false negatives of both models. If these timestamps matched there was a big chance that both models predicted it right. These timestamps were added into the dataset as sprints. This method was validated by a randomly checking sprint that were going to be added to the dataset. This check was done by watching the corresponding video of the timestamps. This method proved to be efficient and accurate and was repeated 7 times on the training and validating dataset, were each time the training and validating parts were changed. This resulted in an increase from 8.6% into 17.1% sprint-positives in the dataset. This updated dataset was used for the training and validating of both models.

Write small introduction to model implementation//preparation.

The two previously mentioned models were used to predict sprints.

RFC

The RFC model used a function class_weight = ‘balanced’ to balance the classes based on their frequency. Feature engineering for RFC models was done by trial and error. The following features showed the best results: Time line, 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. after the feature engineering Grid Search was used to tune hyper parameters for the number of trees(number of classifiers), the criterion(measures the quality of a split) and the max depth of the model.

The RNN model used some 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. Furthermore this will lower the use of GPU memory and improve how much the model learn in every epoch. The other function used by the RNN model is a balancing functions to improve the results of the model. This balancing functions used oversampling of the positives sprint samples to improve the amount of the positives sprints in the training set.

The features of the RNN are: Wheel Speed X, Frame Speed Y, Frame Acceleration, Differential of Wheel speed X. These features were selected through trial and error, the features with the most impact on the RNN model were chosen. Further tuning of the RNN model was done through the graphs seen in figure [X].

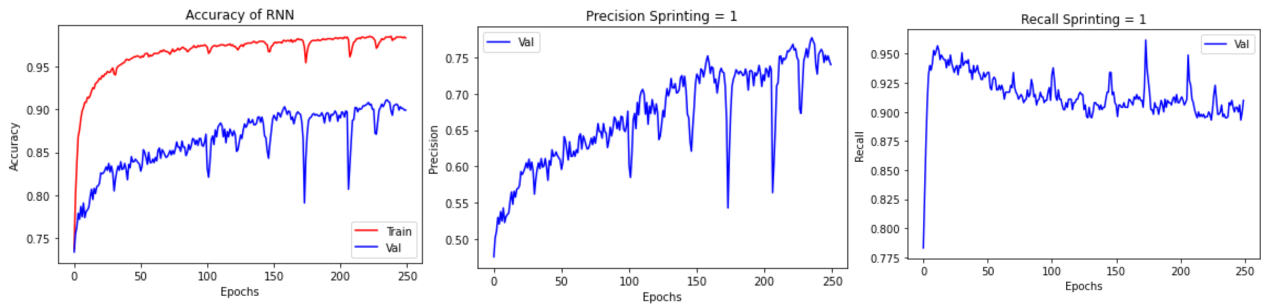
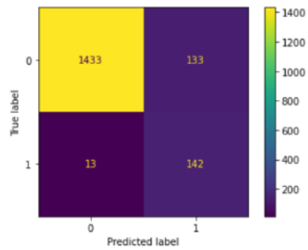


Figure 1: Epoch vs (Accuracy, Recall, Precision)

Results

All models were trained with the same training and test set. The training was done on a Jupyter server owned by the Hague University of Applied Science

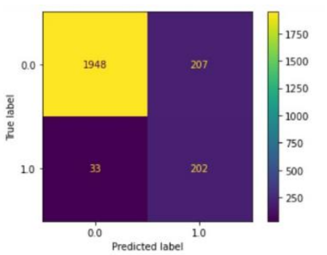


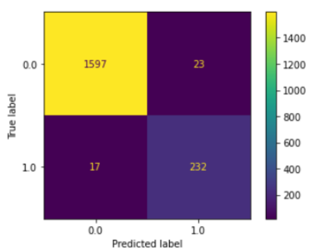
Decision tree

This model has an accuracy of 0.915, a precision of 0.516 and a recall of 0.916. These numbers are this high due to the amount of true negatives. In order to get a more representative view also an confusion matrix is made (Figure X).

Random Forest Classifier

This model has an accuracy of 0.900, a precision of 0.493, and a recall of 0.859. Due to having more true positives and detecting more points the group chose to continue improving the dataset for this model. After further optimizing the data set, as described in chapter X. The model had an accuracy of 0.979, a precision of 0.910 and a recall of 0.932. The confusion matrix belonging to this optimized data set is shown in figure X.





Random Forest Classifier

The RNN model was first trained with a dataset that was not optimized. The results of this model were: a accuracy of 0.901, a recall of 0.935 and a precision of 0.627. The precision of this model is not great. This was solved by optimizing the dataset by comparing the false positives of the RNN with the RFC as explained in Chapter X. With this optimized dataset the model was trained, these results are: a accuracy of 0.892, a recall of 0.911 and a precision of 0.723.

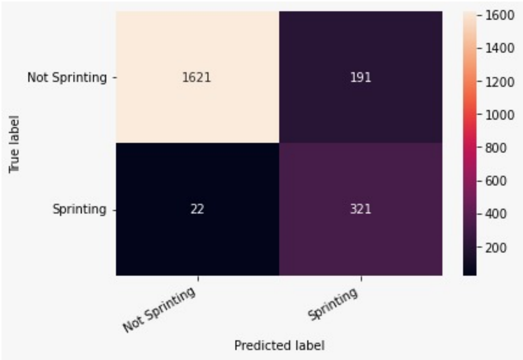


Figure 2: CM RNN not optimized

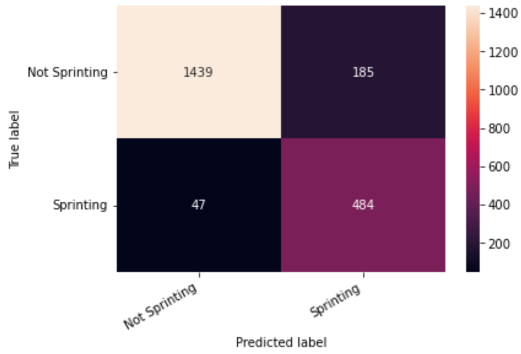
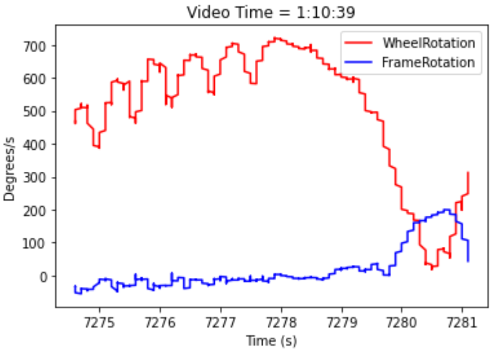
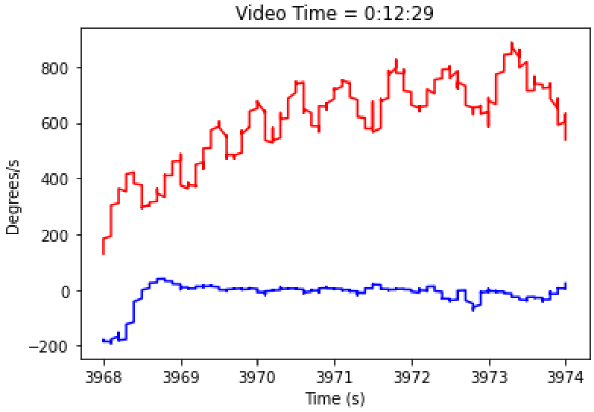


Figure 3: CM RNN optimized

4. Validation

During the training and validating of the random forest classifier, the data set of 1 player was used, player A. In order to verify that the random forest classifier worked on not only 1 specific player, the model was given a completely new test dataset, player B. After running the model on this new data set, the detected sprints were plotted. In Figure X a plot of a sprint of player A is shown, in figure X a plot of a sprint of player B is shown (same text). As these figures suggest the patterns look the same. After comparing the patterns of the graphs, the video data was also compared to the graphs of player B to check whether this player was really sprinting. After checking all the detected sprints, it was concluded that the model had a precision of 91.67 % and a unknown recall. The recall is unknown because in the test dataset no sprints are tagged.





This is a little short maybe add some details

- first manual now automatic
- Player a and player b
- Refer back to dataset

Conclusion

- Refer back to research question
- What did we achief

Discussion and Recommendation

- What went well
- what could have gone better
- how can a follow up project improve
- what are the next steps

Sources

Appendix