SPAITR Data Processing

Joey Neleber, Nicholas Bridges, Jacob Easterling, Alex Valhos
Computer Science Department, University of New Hampshire
Project Sponsor: Nicolas Silberstein Camara

SPAITR Sports AI Tracker

Introduction

SPAITR is a sports analytics company that has developed a Neuro™ (shown below) to track stick motion data.

The SPAITR app allows players and coaches to see the speed of a shot, angle of release, shot duration, and the number of repetitions completed.

Coaches can hold players accountable by tracking time spent practicing off the field, and by reviewing session data of team practices and games.

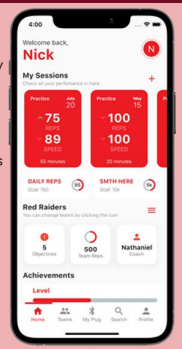
Users can view growth over time, participate in leader-boards, and compete against their teammates.

Project Requirements

Develop a method of isolating and distinguishing shots from a data stream obtained from the Neuro™, allowing SPAITR to more accurately present a user's data to them after using their product.

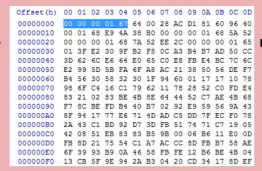
Develop several discrete code sections; security, data parsing, isolation, and classification of areas of interest.

Isolate 95% of shots correctly, calculate shot speeds within 7.5 mph, and calculate shot angle within 10 degrees of their actual values.



Design and Implementation

Once transferred to the phone, the data is uploaded to the cloud as a binary data file.



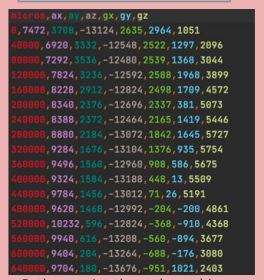
Data is authenticated by comparing the uploaded session to previously uploaded sessions, and if they are too similar then the session is considered a duplicate.

Data Authentication

If the data is flagged as a duplicate the database is updated to remove the duplicate session.

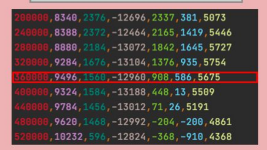
Database

BINtoCSV



Python script that takes a binary file created by the Neuro sensor and gives converts it to CSV files.

Isolator



Data from the converted CSV is isolated into sections of interest through an assortment of tools including SciPy's regression and statistical analysis sub-libraries and Sci-kit learn's machine learning sub-libraries.

Classifier

Left Shots: 3
Right Shots: 5
Ground Balls: 2
Stick Checks: 3
Faceoffs: 1

Isolated sections of interest are then inputed into various proprietary methods to determine which game-specific event is occurring. Session totals are then calculated and sent to the database to update the user's profile.

At the end of a play session, a player will tap their stick to their phone and the most recent session data will be transferred over Bluetooth.

Data Acquisition

Data was acquired from a diverse group of participants including children, teenagers, and male and female collegiate lacrosse players.

During every session, each player had their shot speeds, sides of shots, and the number of shots recorded by a SPAITR team member to ensure data authenticity and accuracy.

Bushnell Speed Gun. (n.d.). [Photograph]. Bushnell. https://www.bushnell.com/dw/image/v2/BDBJ_PRD/on/demandware.static/-/Sites-shootAccessory-master-catalog/default/dw9e6b9c4b/images/large/101911_CAT2.jpg?sw=800&sh=800&sm=fit

Tools

The isolation and classification processes occur entirely in Python - using Numpy, Pandas, Scipy, and Scikit-learn.

Numpy: Enables dataset-wide boolean logic, dataset-wide math operations, and manipulation of parallel datasets.

Pandas: Enables storage and rapid dataset-wide transformations and analysis.

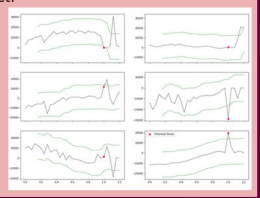
Scipy: Enables reading and manipulation of the waveform data from our CSV input form.

Scikit-learn: Enables machine learning, dimensionality reduction, and regression models.

Testing

Testing is done by an isolation algorithm on different sets of lacrosse data. Data sets fed into the algorithms have a known amount of shots, containing anywhere from one shot to one hundred shots per data set.

The number of shots identified by the algorithm is compared to the number of shots known to be in the set, and the difference from actual to calculated is used to determine the accuracy of the isolation algorithm.



Security

The primary security concern in this context is the upload of illegitimate data. That is, we must ensure each upload of user data is an actual recording of lacrosse practice that this user has performed.

Our data integrity algorithm attempts to sanitize a new upload by reviewing all of the data that has been previously uploaded. The Neuro™ records data precisely, so any similarities between data uploads are measurable and reportable.

If the similarity between the current upload and the previous sessions is too great, the algorithm will flag the upload and report it to the server, potentially invalidating the upload.

The algorithm performs this security check in an amount of time that grows polynomially based on the amount of previous data a user has. As we are dealing with data that is precise and not compressed, such checks take some time to finish. Due to the data being stored on a database, the algorithm can perform its checks asynchronously.

Each security characterization is individually toggleable to change the sensitivity, flag rate, and previous sessions accounted for - each corresponding directly to program runtime.

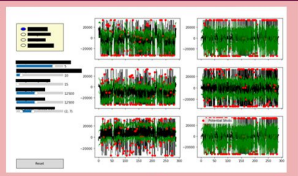
Results

Our evaluation of accelerometer and gyroscopic time series data, representing events in a lacrosse game, performed using our algorithm can:

Correctly detect repetitions at an accuracy of 97.5%.

Correctly analyze speed within 7.5mph.

Correctly analyze the angle of release within 10°.



0	174
1	372
2	418
3	451
4	504

97	4372
98	4427
99	4464
100	4507
101	4562

Length: 102, dtype: int64

*Each red dot on the graphs represents a point of interest (shown below the graph) identified by the isolation algorithm.