Using Machine Learning to Classify Volleyball Jumps

Miki Jauhiainen[1], Michael Jones[1,*]

[1] *Brigham Young University, Provo, Utah, USA 84602*

Abstract

In this study, inertial measurement units (IMUs) were used to train a random forest classifier to correctly classify different jump types in volleyball. Athlete motion data were collected in a controlled setting using three IMUs, one on the waist and one on each ankle. There were 11 participants who at the time played volleyball at the collegiate level in the United States, seven male and four female. Each performed the same number of jumps across the eight jump types—five BASIC jumps and three each of the other seven—resulting in 26 jumps per subject for a total of 286. The data were processed using a max-bin method and trained using a leave-one-out cross-validation method to produce a classifier that can determine jump type with an accuracy of 0.967, as measured by an F_1-score.

Keywords

sports, wearable sensors, supervised machine learning, volleyball

1. Introduction

In this paper, we investigate classification of blocking jumps in volleyball through supervised machine learning using inertial measurement unit (IMU) data. Jump classification could be used to create novel analysis tools for coaches and athletes.

IMU sensors are inexpensive and can be easily attached to volleyball players in both practice and game settings. A single sensor can collect more than 100 readings per second and each reading contains nine data points representing linear acceleration, rotational velocity and magnetic field values. When used to collect motion data for volleyball players, the challenge is turning IMU readings into useful insights for coaches, athletes, and others.

In order to use sensors to improve performance as part of sports training, we will need to find specific events in the data and classify jumping movements, which is not a trivial task. Finding events and classifying movements in data represented using a graph is hard for the untrained human eye, as exemplified in Figure 1. Figure 1 contains data that we collected from an IMU attached to a volleyball player in a practice setting. The IMU measures linear acceleration, rotational velocity, and magnetic field in three dimensions, all of which are displayed in Figure 1. The different lines represent the values for the x, y, or z axes for either the accelerometer, the gyroscope, or the magnetometer. For the gyroscope, the x, y, and z axes correspond to roll, pitch, and yaw.
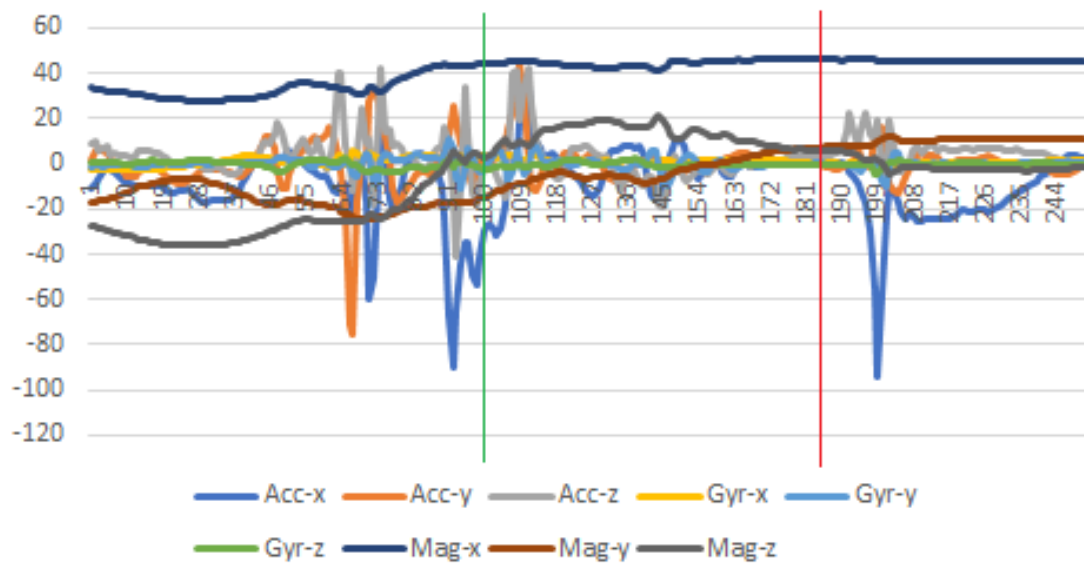
Figure 1: A graph of an X3L jump using the waist sensor. The green line is the takeoff and the red is the landing.

The data in this graph were collected during a blocking move, which consists of movement along a volleyball net, and a jump. We measured this data because blocking is an important skill in volleyball. It is possible for a person to spot this event and that movement with some training, depending on the movement type, but it is not easy.

Training a classifier to classify movements in the data could generate a more usable description of the data. Training a classifier involves two tasks: processing the data for use as input and setting classifier parameters. There are many ways to process the data and there are many settings for classifier parameters.

Classifying jump types in volleyball motion data has value for both players and coaches. One of the authors played volleyball at both the collegiate and international level. In that author's experience, athletes and coaches care about tracking and improving jumping skills while avoiding injury. To validate this perspective, we talked to two collegiate volleyball coaches about tracking jumps. One coach stated that measuring differences in jump height between different types of jumps would allow for more specific training programs. One coach suggested aligning sensor data with film from the practice or match. This would allow looking up the jump on the film using the timestamp. Building a system that matches sensor data with video from practice is a promising direction, and the work done in this report contributes to the future construction of such a system.

However, the first step in both of these ideas is to identify jumps in the data itself. Once jumps can easily and accurately be identified, systems can be built to measure and compare jump heights and match up specific jumps in sensor data and video.

Collecting and analyzing data can also help coaches and athletes by leading to better injury prevention protocols. It is in the best interest of everyone (such as coaches, athletes, and team

owners) for athletes to achieve better longevity through injury prevention: the athletes can continue to do what they love doing while getting paid to do it, the coaches do not lose their star players to chronic injuries as quickly, and the owners get to enjoy the profit from ticket sales that their best players continue to boost.

Injuries incur both financial and personal costs. In one collegiate program in the United States, an MRI to diagnose a knee injury due to overuse, or other injuries, costs about $1,000. The actual surgical repair is another $10,000 on top of that. Furthermore, it is extremely hard to recover from surgeries and get back to the same level of play, which is bad for both the player and the team they represent. Anterior cruciate ligament (ACL) injuries are fairly common in volleyball [1], and they are one of the injuries that require the expensive surgery. Counting jumps in data collected during training could be part of an injury prevention protocol.

The following scenario illustrates the need for the proposed work. Jack is a 20-year-old sophomore in college, and he aspires to play professional volleyball after graduating. His position is middle blocker. He is great at attacking, but not so good at blocking. He recognizes that not being able to block well could hinder his chances of making it as a professional. He asks his coach to help him with blocking, so they start using a sensor-based app to monitor Jack's training. Jack starts using the app during practice and reviews the data after practice with his coach. Because the app can distinguish between different types of jumps, Jack and his coach can easily find the jumps going left and right and compare them on video. They notice that when going left, his steps are too small, so he does not make it far enough in time.This results in the coach being able to assign specific workouts to balance out Jack's leg strength, as well as monitor his footwork to make sure he takes big enough steps.

Others have studied the problem of volleyball action detection and classification but with limited accuracy. Using computer vision, Ibrahim et al. [2] attempted to classify blocking, hitting, and setting, among other things, but they only achieved 51.1% accuracy. Kautz et al. [3] used an IMU strapped to a wristband to identify different volleyball actions with near-perfect recall, but only 34.8% overall accuracy. Their work with IMUs is encouraging, but use for performance improvement requires more accuracy. Furthermore, our top priority, blocking, had the lowest accuracy among the actions they targeted.

To attempt to solve the problem of classifying volleyball jumps, we labeled our collected data and processed them using a max-bin approach so that it could be used to train a classifier and validated using leave-one-out cross validation (LOOCV). We gathered the data using IMUs and labeled it with the help of our IMU-synchronized video. We processed the data using a max-bin approach, which allowed us the aggregate the data while saving the peak values. We then trained a random forest classifier using the aggregated data, only including the parts with jumps. Finally, we used LOOCV to measure accuracy with an F_1-score.

We were able to achieve an F_1-score of 0.97 using the combination of the left and right foot sensors, window size of 360, and bin size of 25, with a random forest. Most results for any combination of sensors were between 0.85 and 0.95, as long as the bin size stayed under 100. These results suggest that we were able to successfully solve our problem, as F_1-scores in the 90s are typically accepted as good results, as demonstrated in [4, 5, 6, 7, 8]. This classifier could be used in the future to build applications that measure jump height and can be synchronized with video for more efficient coaching.

2. Related Work

There exist ways, such as VERT [9] to measure jump height in sports like volleyball, but, to the best of our knowledge, there are no existing ways to accurately determine what kinds of jumps volleyball players are performing. There have been no previous attempts in the research literature to classify jumps using data from IMUs in volleyball, but similar work has been done in other sports that involving jumping such as figure skating [10]. Similar to our work, they used an IMU strapped to the waist together with synchronized video to gather and annotate data. They labeled the takeoff and landing times of the jumps and then used those labeled jumps as input to a supervised classification algorithm that learns to recognize those jumps, which is the same exact approach we will be using. Like figure skating, volleyball involves jumping and rotating in the air, which gives us confidence that this can be done. The jumps in figure skating involve more spinning, but the basic concept of movement followed by a jump is the same in both sports.

Others have studied the problem of identifying volleyball movements in video. Those efforts have not yet achieved accuracy needed to improve performance outcomes in training. In [2], Ibrahim et al. attempted to pinpoint actions such as blocking, hitting, and setting, but they only achieved 51.1% accuracy. In [11], Azar et al. recognized the activity fairly accurately through recognizing what individual players are doing, but important pieces like information about the ball and the net were missing. Using computer vision would require multiple expensive cameras and visibility of the whole volleyball court. This might not be as feasible as using IMUs due to financial reasons and possible venue limitations–it might not be viable to set up the cameras in good enough places to be able to use the system. Kautz et al. [3] recognized different volleyball-specific actions, like passing or serving, using an IMU strapped to a wristband. Using a decision tree, they achieved high recall, but only 34.8% overall accuracy, meaning that there were many false positives. Their work suggests that machine learning is a reasonable approach, but more accuracy is needed for use in performance improvement. Additionally, the action identified with the lowest accuracy was blocking, which is our top priority since we are studying primarily blocking jumps.

Salim et al. [12] performed a study similar to [3], but with slightly better results. Both studies used an IMU strapped to the wrist, but in [12], they had one on each wrist, whereas in [3] it was only on the dominant hand. The F_1-scores and accuracy scores ranged between 20-90%, although for most actions they were around 70-80%. Once again, however, blocking actions were not recognized accurately enough for performance improvement. Furthermore, attaching an IMU to the wrist of a volleyball player would be like wearing a smart watch, which is generally not recommended for for volleyball.

There is also a body of work related to IMUs and swimming [13]. In [13], sensor placement seems to be significant and the accuracy of the results when classifying stroke type look promising. Distinguishing swim strokes based on motion is similar to classifying different volleyball blocking movements because both activities include the position and motion of the hips, which is where we placed one of the sensors. Results in [13] suggest that working with several sensor locations will be needed to find an optimal placement.

3. Volleyball Background

In order to fully understand this research, it is important to have some knowledge of how volleyball is played. Although one of the most popular team sports in the world, the difficulty of the actions required and of the rules makes it hard for people unfamiliar with the sport to grasp.

Volleyball is played on a court with two sides of 9 x 9 meters that are divided by a net that stands at 243 centimeters for men and 224 centimeters for women. There is a line on both sides, three meters from the net, that separates the court into front court and back court. Both teams have six players on the court at once, although seven play actively. The seven comprise of one setter, one opposite hitter, two outside hitters, two middle blockers, and one libero (a defensive specialist). Three players play at the net and three in the back court. The player who has most recently rotated into the back court is always the one to serve. The middle blockers and outsides, respectively, are also positioned across from each other in a similar manner. The lineup of one team on their half of the court is illustrated in Figure 2, and the setter would be the one serving in this situation.

There are three touches allowed on each possession. Ideally, the setter always gets the second touch and sets the ball to an attacker, meaning that the setter decides who gets to attack the ball over the net. The defending side usually attempts to block the attack with as many players as possible (which is three) but at least with one player. Players in the back court can not put the ball over the net–or prevent it from coming over the net–if they step inside the three-meter line. Hence, only three players can block. Because the blockers are spread out across the net, but they all try to end up blocking the ball in the same spot, they have to use different footwork to get there. That is why there are several different types of blocking jumps that are recognized and taught on the highest levels of volleyball.

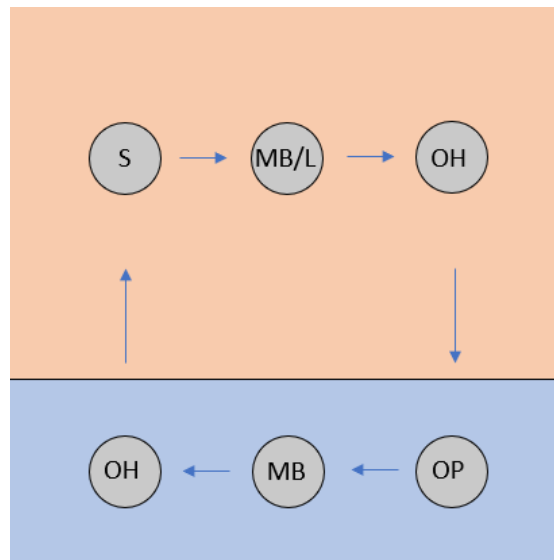The blocking jumps studied in this research are: BASIC, a jump straight up; Q3, a quick

Figure 2: Volleyball lineup. The net is at the bottom, and the front court is blue.
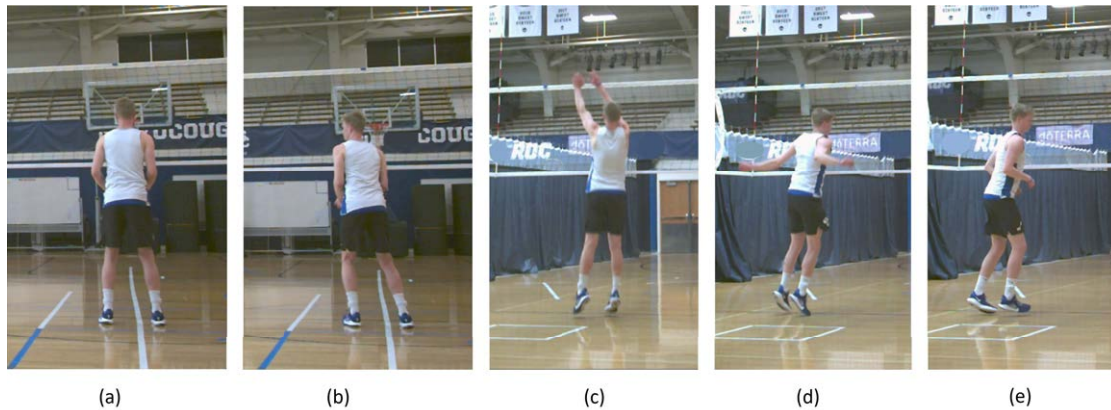
Figure 3: Five elements of a volleyball blocking jump starting in the neutral position (a). Take-off occurs when the player's feet leave the ground (c) and landing when the player's feet touch the ground again (d)

shuffle-step move with three steps left or right; X3, a crossover 3-step move left or right; X2, a crossover 2-step move left or right; and ATTACK, an attacking jump with typically a 3- or 4-step approach. Left and right are indicated by an "L" or an "R" after the jump type. During a Q3, your chest is facing the net the whole time, and the jump happens off both feet. While taking the first step of an X3 and an X2, you turn and face the direction you are going. Furthermore, the jump happens off one foot for an X2 and both feet for an X3, and the chest starts turning back towards the net again on takeoff so that at the peak of the jump you are facing the net. All the movement in the blocking jumps happens parallel to the net, the attacking jump is the only one that happens perpendicular or at an angle.

4. Methods

This research consists of five major components: data collection, data labeling, data processing, training, and testing.

In this report, we focus on classifying jump types from a wearable 9-axis IMUs attached to the athletes ankles and waist (but not wrists). We assume that jumps can be detected using threshold-based algorithm. This means that every segmented jump in training and testing contains a jump. We had experimentally determined earlier that a value of 24.5 $\frac{m}{s^2}$ for the x-axis of linear acceleration indicates a jump, but that is outside the scope of this report. Nevertheless, the classifier classifies jump type assuming the data contain a jump.

4.1. Data Collection

For gathering the data we recruited 11 NCAA Division I volleyball players at a university in the United States with 7 male and 4 female. There were players from every position group except libero (because liberos do not perform jumps in games). Every participant was between 18 and 24 years old. All participants performed the same 26 jumps: five of type BASIC, and three each

of types Q3L, Q3R, X3L, X3R, X2L, X2R, and ATTACK. These jumps are defined in Section 3. Even though the focus of this study is blocking, attacking is such a common occurrence in volleyball that it is important to include it so that the classifier is trained on the complete set of jump types.

During the jumps, each subject wore 3 IMUs[1]: one around the waist so that the IMU was centered on the small of the back, and one on the lateral part of each ankle right above the shoe. These IMUs were configured to measure linear acceleration, rotational velocity, and magnetic field on 3 different axes at a rate of 120 samples per second. Jumps were filmed with a Qualisys Miqus Video camera synchronized with the IMUs so that it recorded 120 frames per second, with a resolution of 1280 x 720. The two systems were hardware synchronized using a common trigger that was wired to sync inputs from both systems.

Two different courts were used to perform the jumps, both of which were empty except for the athlete jumping at the time. The courts used were side by side and all jumps were performed on the same side of the net. The jumps happened at the net and were filmed from the service line. Each athlete was allowed adequate warm-up time according to their needs. In order to allow full focus on the blocking motion, no balls were used.

The athletes performed the jumps one by one. To decrease the risk of having the sensors and camera become unsynchronized, we only recorded for a couple minutes at a time. The recordings were split up by jump type. We did not collect the dominant foot (left or right) for each athlete. Because for blocks the approach and jump motion are the same regardless of the athlete's dominant foot.

4.2. Data Labeling

Once the data had been collected, each jump was annotated with four events: motion started, feet left ground, one foot back on group, and motion done. Every jump starts from a stationary neutral position, as shown in Figure 3 (a) For example, for a X3L, we would label the moment the subject's left foot starts moving to the left as the start of the movement (Figure 3 (b)), the moment their toes leave the ground as the takeoff (Figure 3 (c)), the moment the toes touch the ground again as the landing (Figure 3 (c)), and the moment they return to a relatively stable position (hard to pinpoint exactly) after landing as the end of the movement (Figure 3 (e)). Because the camera and the IMUs were synchronized, we could now pinpoint the exact moments in the raw motion data where the jumps happened. After this initial round of annotation, a volleyball expert reviewed all labels to confirm that they were accurate, and fixed any potential errors.

4.3. Data Processing

We processed the data using a max-bin approach, because it smooths out high frequency noise while preserving peaks. Peaks are important because they show when landing and take-off happen.

The max-bin approach, given, for instance, a window size of 100 and a bin size of 10, works as follows. First, we take the 100 rows of data and split into 50 in the past and 50 in the future, with the current row arbitrarily assigned to be in the "past." We then apply a filter that first takes

---

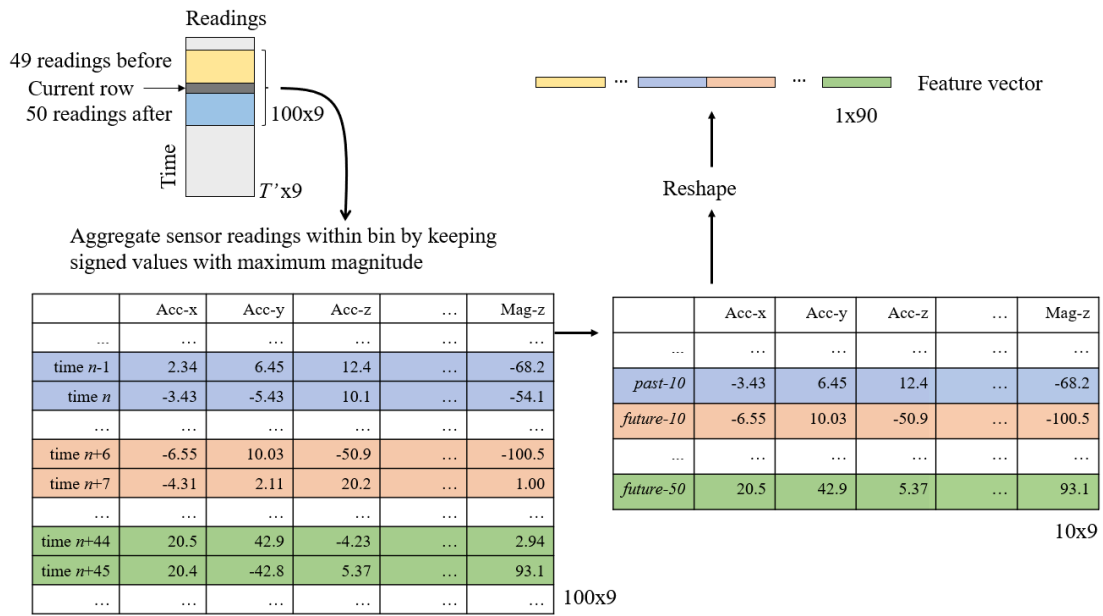[1]Opal model, APDM, Inc., Portland OR, USA

Figure 4: The aggregation process for a single point in time, or row of data.

the value with the maximum magnitude of each of 9 columns for a single sensor for the first 10 values in the past and adds those 9 values to the feature vector. Next, we take the max of the next 10 values in the past and concatenate those to the feature vector. We repeat the process for the 50 rows in the past a total of five times. The same process is then repeated for the 50 rows in the future. This creates one input vector with 9 x (5 + 5) = 90 values per sensor. This process is pictured in Figure 4. If the bin size does not divide the window evenly, the remaining rows are treated as their own bin. The aggregation process is started from the middle of the window, so the partial bins, if any, are at the beginning and end of the window. For instance, with a window size of 100 and a bin size of 15, the process begins with two halves of the window with 50 rows each. 15 goes into 50 three times with 5 left over. The bin process starts from the center of the window and works to either end. Any leftover rows in an incomplete bin are treated as a single partial bin. Thus, for a window size of 100 and a bin size of 50 the window would be split up into bins as follows: 5-15-15-15-15-15-15-5.

The whole process of creating a window and computing an input vector then "slides" across every row in the data frame, as shown in Figure 5. All of the feature vectors stacked together make up the rows in the final set of feature vectors. Since there are about 70-150 rows per jump–depending on the movement type–this process creates about a hundred slightly altered copies of a single jump, increasing the number of feature vectors. This way there is enough data to train a reasonably general classifier even with a smaller original data set.

This process is repeated for each labeled data frame, and they are all concatenated to each other to form one massive preprocessed data frame with dimensions N x T, where N is the number of columns and T is the total number of rows resulting from adding all the smaller data
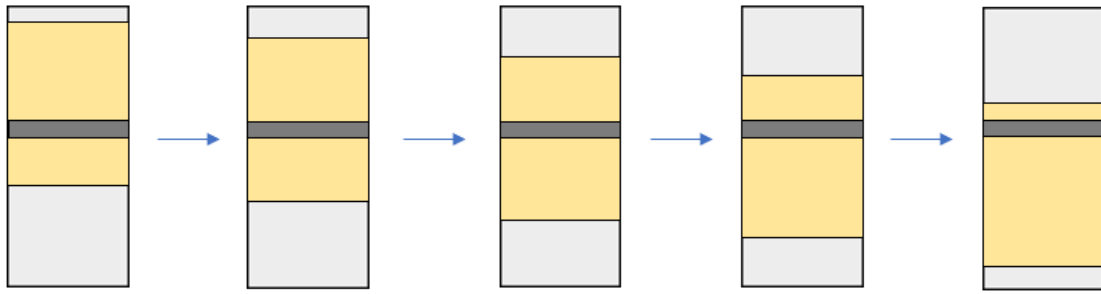
Figure 5: The way the aggregation window "slides" through the data to form the preprocessed data frame. Each window is processed as shown in Figure 4. The gray row is the same row each time, visualizing how the window shifts around it.

frames together.

4.4. Preliminary Study

Before running extensive experiments to find the best processing and training parameters for a classifier, we ran a preliminary study to compare performance across a group of supervised learning algorithms. The independent variables for the preliminary study were window size, bin size, algorithm, and sensor combination, and the dependent variable was accuracy, measured as an F_1-score (defined in detail in Section 4.5). There are multiple supervised learning algorithms in the Python library scikit-learn that handle multi-class classification problems. The ones we tested were random forest, decision tree, AdaBoost, logistic regression, multilayer percepton (MLP), k-nearest neighbors (KNN), naive Bayes, and support vector machine (SVM). To compare the different algorithms, we ran tests using a window size of 350 and a bin size of 25. We tested with all three sensors combined, as well as each of them separately. The summarized results are in Table 1. The first row contains average accuracy across each of the 4 conditions (all sensors, left ankle, right ankle and waist). The second row contains the maximum observed accuracy in the same 4 conditions.

We got results ranging from 0.040 all the way to above 0.90, and random forest consistently producing the best results. Some algorithms, like SVM and naive Bayes, performed poorly across all tests. We did not expect accurate results using naive Bayes because it is a fairly simple classifier, but the poor accuracy of SVM surprised us. It is possible that the implementation of SVM we used was not equipped to handle the complexity of the input data.

As the results of the preliminary study show, random forest was more accurate than the other algorithms (for the chosen parameter settings), so further testing involved just the random

Result type	RF	DT	AB	LR	KNN	NB	SVM	MLP
Average	**0.898**	0.721	0.266	0.809	0.524	0.565	0.040	0.629
Highest	**0.970**	0.753	0.288	0.845	0.607	0.670	0.040	0.704

Table 1
Algorithm comparison results. The random forest (RF) generated the most accurate average results as well as the most accurate single result as shown in bold.

forest algorithm.

4.5. Variables

There are four independent variables in the second study: window size, bin size, movement type, and sensor combination. For movement type, there are only two options: full movement or jump only. There are seven different sensor combinations: waist, left foot, right foot, waist + left, waist + right, left + right, and all three. There are a large but finite number of options for window and bin sizes, but we imposed some restrictions on them to keep the experiment tractable. Since we sampled at 120 frames per second, and each row of data is one frame, 120 rows represents one second in real time. It takes about one second to perform a BASIC jump, and all the other ones take longer, so we decided to not use window sizes smaller than 200 to allow fitting the entire jump sequence in the window. The jumps, including the approach motion, should not take longer than 3-4 seconds, so we used 440 as the biggest window size. We used a step size of 20 (i.e., to obtain window sizes of 200, 220, 240, ... 440). There is likely little benefit to trying every single window size, and going through all the results would have been extremely time-consuming. The bin sizes we used were 5, 10, 15, ..., 75, 90, 110, 130, ... all the way up to the size of the window. To keep the experiment and analysis tractable, we limited the number of combinations by choosing 5 as the bin size interval up until 75, at which point the bin size is already so big that using every 5 would most likely have been redundant, hence the switch to using every 20. It does not make sense to have a bin size larger than the window size, so that is the upper limit.

The dependent variable is accuracy, as measured by an F_1-score using a macro average over the eight jump types. An F_1-score is defined as the harmonic mean of precision and recall as follows:

$$F_1 = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \tag{1}$$

where tp, fp, and fn stand for true positive, false positive, and false negative, respectively. Precision measures the ratio of relevant items picked to irrelevant items picked, and recall measures the ratio of relevant items picked to all relevant items. Further, true positive means the number of correctly picked items, false positive means the number of incorrectly picked items, and false negative means the number of items that should have been picked but were not. We chose this measure because it penalizes extremes (aggressive/timid classifying), ensuring that the classifier is balanced.

4.6. Training & Testing

As a result of the preprocessing, the data are organized into a collection of input vectors of dimensions N x T with each input vector labeled as a type of jump or a non-jump. The input vectors all contain a jump.

We initially chose the training and testing data randomly, but decided to switch to LOOCV to simulate testing on completely new data from an unseen athlete. We used 10 out of 11 athletes for the training and the remaining one for testing. This way the classifier had not seen any of the jumps from the specific athlete before testing, which combats overfitting. This process was
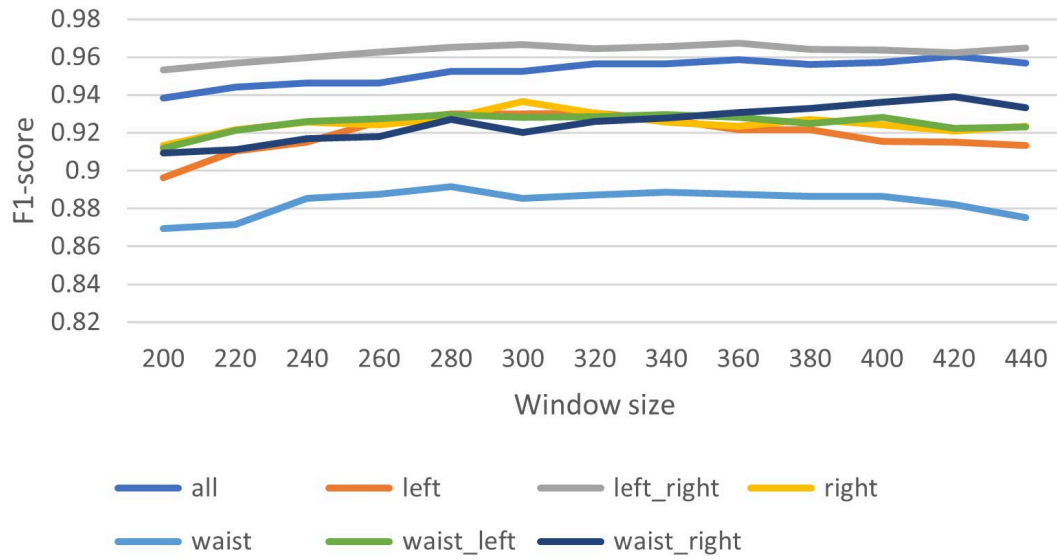
Figure 6: Best scores across all bin sizes for all sensor combinations. The combination of sensors on the left and right ankles, as shown by the grey line, consistently produced the best results.

repeated for every athlete so that the classifier was exhaustively tested on each athlete. All the results presented are averages from doing this for all 11 athletes so that results for a specific athlete do not dominate.

5. Results

Figures 6 through 8 show key results.

Figure 6 shows the best scores by sensor combination across all bin sizes, for each window size, and for both jump types. In the graph, the horizontal axis represents the window size. Each line of data in the graph represents F_1-scores for a different combination of sensors as shown in the legend at the bottom of the graph. The vertical axis represents the maximum F_1-score averaged across all bin sizes for a given window size and sensor combination. A combination of both the left and right ankle sensors produced the best results while the waist sensor alone produced the least accurate results.

Figure 7 also shows F_1-scores for different window sizes and sensor combinations but for a single bin size of 25. As in Figure 6, the left and right ankle sensors produce the best results while the waist alone produced the least accurate results.

Figure 8 shows F_1-scores for all sensor combinations and bin sizes with window size 360. The vertical axis still represents the F_1-score and the horizontal axis is the bin size. Note that the gap between bin sizes varies in the horizontal axis. Larger bin sizes produce less accurate results as might be expected.
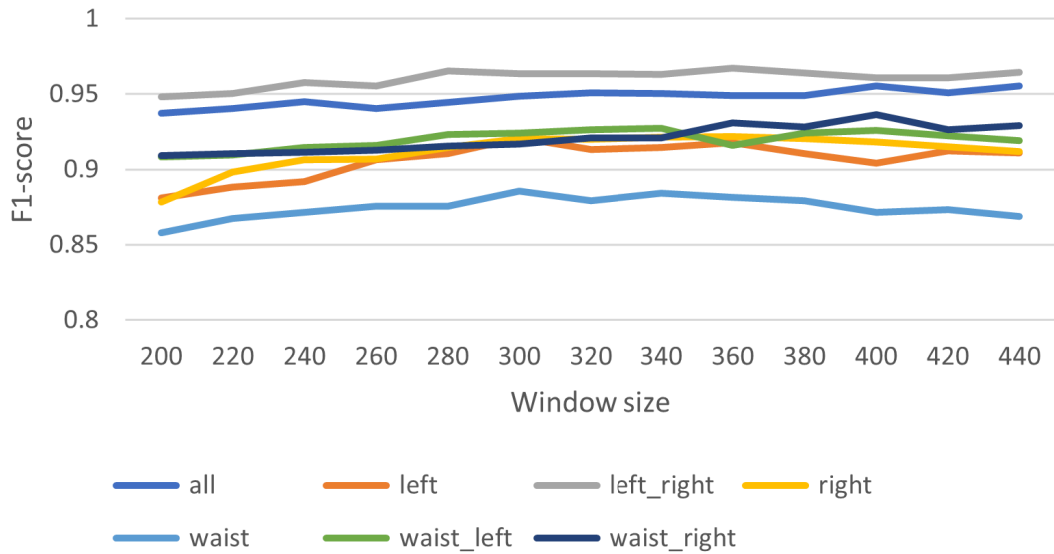
Figure 7: Results for all sensor combinations and window sizes with bin size 25. The left & right feet together performed the best.

6. Discussion

We obtained accurate jump type classifiers by training a random forest classifier on input vectors generated from volleyball blocking jumps using window size 360, bin size 25, and left and right ankle sensors together. Feature importance analysis did not indicate that any single feature was significantly more important than others.

Compared to [14] which uses a similar approach, we obtained more accuracy on a larger set of jump classes. There are three factors that may explain our increased accuracy. First, we generated more input vectors by sliding the feature vector window over jumps. We went from having 26 jumps per athlete to about 1500 input vectors per athlete based on those jumps. The reason a single jump can be turned into many useful input vectors without creating redundant noise is that the jump itself moves around in the window. Because we are looking at windows larger than the duration of the jumps, they can slide around inside the windows, making each window unique, even though the jump is the same. Additionally, depending on the bin size and how the values line up across the bins, the peak values around take-off and landing could end up being slightly different after the aggregation process, altering the critical pieces of the jump each time.

Second, our data are collected in a highly controlled setting while data in [14] were collected in a more general practice setting. Moreover, figure skating motion data may include more motion that is not directly related to a jump because the athlete is always in motion on the ice. In contrast, volleyball players in our data collection process remained stationary until performing the actual jumping motion.

Third, we used data from sensors on the ankles rather than the waist. The better accuracy achieved by the ankle sensors could be because the waist moves in similar ways across the jumps,
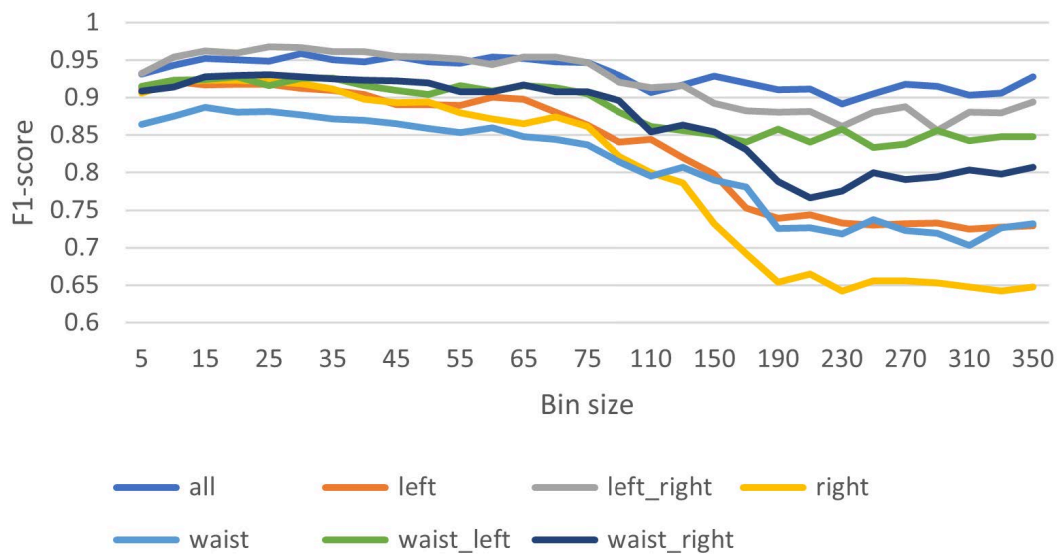
Figure 8: Results for all bin sizes and sensor combinations with window size 360. The scores drop off significantly once the bin sizes get past 150.

while the feet do something different every time. This could create additional inconvenience in practice, because the jump detection algorithm we are relying on primarily uses the waist sensor, which means that usage in a live setting would require all three sensors. Ideally we would only need one sensor, because having to strap them on can be annoying for the athletes.

Collecting more input data from more athletes would likely increase the accuracy of our classifiers. This would involve recruiting more athletes and organizing more data collection sessions.

One weakness of this study is that we were not able to collect data and test the classifier in a live volleyball setting. We did our best to simulate one with our testing method, but nothing compares to testing during an actual game or practice, especially since the collected data–and hence, the jumps that were left out for testing–were so clean and from a controlled setting. This could leave to overfitting, which is when a classifier fits exactly to its training data, but struggles to generalize to unseen data. Overfitting is a problem because game and practice settings involve more movement than our tightly controlled data collection sessions. That extra motion may prevent an overfit classifier from recognizing a jump; and may also include false positives. Overfitting may be exacerbated by the combination of max-bin and a small data set.

Another limitation is that the orientation of the only two courts we used to collect data was the same, meaning that the values of the magnetometer, which tracks orientation relative to the magnetic north pole, were always similar. It is possible that a classifier trained like this could confuse left and right directions if it was used on jumps performed on the opposite sides of these nets or on a net with a different orientation. To avoid having this problem, we could zero out the magnetometer values at takeoff to "reset" the orientation so it only accounts for the rotation in the air. We decided to test this approach with the best parameters we found (i.e.,

window size 360, bin size 25 and the left and right feet together). We achieved an F_1-score of 0.97 with this method, which is about the same as what we had before, so at least the impact was not negative. This suggests that court orientation may not be a significant factor.

Overall, these results could support the implementation of an app that tracks volleyball jumps, which could be useful in a coaching setting. For example, tracking different types of jumps in a game or practice and being able to search for them would make film study a lot easier, and it could help spot aspects that need to be worked on in a player's game. Additionally, identifying and classifying jumps could become the foundation for a recommendation system that identifies trends or issues in a specific athlete's training. For example, such a system might notify a coach and athlete that the athletes jumps to the left have lost power. The coach and athlete can then follow up to determine why.

References

[1] D. Xu, X. Jiang, X. Cen, J. S. Baker, Y. Gu, Single-leg landings following a volleyball spike may increase the risk of anterior cruciate ligament injury more than landing on both-legs, Applied Sciences 11 (2021). URL: https://www.mdpi.com/2076-3417/11/1/130. doi:10.3390/app11010130.

[2] M. S. Ibrahim, S. Muralidharan, Z. Deng, A. Vahdat, G. Mori, A hierarchical deep temporal model for group activity recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 1971–1980. doi:10.1109/CVPR.2016.217.

[3] T. Kautz, B. H. Groh, J. Hannink, U. Jensen, H. Strubberg, B. M. Eskofier, Activity recognition in beach volleyball using a deep convolutional neural network, Data Mining and Knowledge Discovery 31 (2017) 1678–1705. URL: https://doi.org/10.1007/s10618-017-0495-0. doi:10.1007/s10618-017-0495-0.

[4] F. Magalhães, G. Vannozzi, G. Gatta, S. Fantozzi, Wearable inertial sensors in swimming motion analysis: A systematic review, Journal of Sports Sciences 33 (2014). doi:10.1080/02640414.2014.962574.

[5] D. Dalmazzo, S. Tassani, R. Ramírez, A machine learning approach to violin bow technique classification: A comparison between IMU and MOCAP systems, in: Proceedings of the 5th International Workshop on Sensor-Based Activity Recognition and Interaction, iWOAR '18, Association for Computing Machinery, New York, NY, USA, 2018. URL: https://doi.org/10.1145/3266157.3266216. doi:10.1145/3266157.3266216.

[6] T. E. Lockhart, R. Soangra, J. Zhang, X. Wu, Wavelet based automated postural event detection and activity classification with single IMU, Biomedical sciences instrumentation 49 (2013) 224–233. URL: https://pubmed.ncbi.nlm.nih.gov/23686204.

[7] Z. Zhang, D. Xu, Z. Zhou, J. Mai, Z. He, Q. Wang, IMU-based underwater sensing system for swimming stroke classification and motion analysis, in: 2017 IEEE International Conference on Cyborg and Bionic Systems (CBS), 2017, pp. 268–272. doi:10.1109/CBS.2017.8266113.

[8] D. Yang, J. Tang, Y. Huang, C. Xu, J. Li, L. Hu, G. Shen, C.-J. M. Liang, H. Liu, Tennismaster: An IMU-based online serve performance evaluation system, in: Proceedings of the 8th Augmented Human International Conference, AH '17, Association for Comput-

ing Machinery, New York, NY, USA, 2017. URL: https://doi.org/10.1145/3041164.3041186. doi:10.1145/3041164.3041186.

[9] Player management system for injury prevention and player load management, ???? Https://www.myvert.com/ Accessed July 2022.

[10] D. A. Bruening, R. E. Reynolds, C. W. Adair, P. Zapalo, S. T. Ridge, A sport-specific wearable jump monitor for figure skating, PLOS ONE 13 (2018) 1–13. URL: https://doi.org/10.1371/journal.pone.0206162. doi:10.1371/journal.pone.0206162.

[11] S. Azar, M. Ghadimi Atigh, A. Nickabadi, A multi-stream convolutional neural network framework for group activity recognition, ArXiv (2018).

[12] F. A. Salim, F. Haider, D. Postma, R. van Delden, D. Reidsma, S. Luz, B.-J. van Beijnum, Towards automatic modeling of volleyball players' behavior for analysis, feedback, and hybrid training, Journal for the Measurement of Physical Behaviour 3 (2020) 323 – 330. URL: https://journals.humankinetics.com/view/journals/jmpb/3/4/article-p323.xml. doi:10.1123/jmpb.2020-0012.

[13] R. Mooney, G. Corley, A. Godfrey, L. R. Quinlan, G. ÓLaighin, Inertial sensor technology for elite swimming performance analysis: A systematic review, Sensors (Basel, Switzerland) 16 (2015) 18. URL: https://pubmed.ncbi.nlm.nih.gov/26712760. doi:10.3390/s16010018.

[14] M. D. Jones, S. T. Ridge, M. Caminita, K. E. Bassett, D. A. Bruening, Automatic classification of take-off type in figure skating jumps using a wearable sensor, in: ISEA Engineering of Sport 14, 2022.