Handball throw type detection and classification with different machine learning models based on wrist IMU data during a handball practice.

By:

Ayla Engels

4246233

In Partial Fulfilment of the Requirements for the Degree of
MSc Biomedical Engineering

Thesis Committee:
Prof. dr. H.E.J Veeger, TU Delft, Biomechanical Engineering
Prof. dr. Roland van den Tillaar, Nord University, Sport Science and Physical Education
Dr. M. Kok TU Delft, Delft Center for Systems and Control

Preface

This thesis is the final work as a student at the TU Delft. After the bachelor Nanobiology, I wanted a more technical master with still applications in the medical field and with the possibility to do research in sports engineering. With the Biomedical Engineering master I found both. How engineering can help in the medical field became very clear during my internship at the Microsoft Research Lab in Cambridge, where we worked on glasses which could help blind children become more aware of their surroundings. For my master thesis, I wanted to do more with sports engineering, so I went to Nord University in Levanger in Norway to work with data science in handball.

During both projects, I loved to do the experiments and look at the data which those experiments provided. Especially in my time in Norway, it wasn't always easy to communicate with the handball players, but they always wanted to help. I'm very grateful to all the players and the coaches of LHK who helped me with the data acquisition. Also in the last part of the project, the handball players of the Dutch Handball Academy provided data, with the physical therapist Linda van Maanen-Coppers who did the experiment in the Netherlands and helped the manual classification, which I'm also very thankful for.

There are of course many other who helped me during the project and my time abroad and who I would like to thank. First of all Roland van den Tillaar, who was not only my supervisor in Norway and the person I could come to at every moment with questions, but who also organised board games and activities with fellow students. I would also like to thank DirkJan Veeger and Bart van Trigt who guided me through the project and provided very helpful advice. Lastly I would like to thank my friends and family who kept in touch while I was in Norway, helped me through the tough times and gave invaluable advise.

Ayla Engels, July 2022

Contents

Abstract

Activity classification in sports is a powerful tool for athlete monitoring, enhancing performance and injury prevention. In handball, detection and classification of throws during a practice or a (practice) match has not been done. Therefore, the aim of this study is to use machine learning algorithms to detect handball throws and to classify between different throwing types and wind ups in handball based on wrist IMU data during a practice or practice match. A total of 2475 throws from 16 players were used for the detection and classification. Multiple algorithms were tested for the binary (throw versus no throw) event detection. The k-Nearest Neighbours algorithm provided the highest accuracy and F1 score and is therefore the best fit. For classification, all throws were labelled with one of the 17 throw types. Five categories were made to test on what scale the classification is possible. The categories consisted of all 17 throw types, shots versus passes, wind up type, a 7-class category and an intensity-based category. Even though multiple algorithms were tested, for all categories Support Vector Machines gave the highest F1-score and accuracy and was therefore the best fit. The categories based on intensity and wind up type scored higher than the categories with all 17 throw types and with 7 classes. Future research should focus on balancing out and enlarging the dataset, preferably with lab data.

1. Introduction

Activity classification can be a powerful tool in sports. It is helpful in enhancing performance, athlete monitoring and injury prevention [1]. One way to obtain data during sport activities is with an inertial measurement unit (IMU). IMUs are sensors with an accelerometer, gyroscope and optionally a magnetometer. The accelerometer provides the linear acceleration data in g-force, the gyroscope measures the angular velocity in degrees per second and the magnetometer gives data about the magnetic field. The IMU can be incorporated into a small wearable device that does not limit normal movement [1]. IMU studies were conducted in many sports, including basketball [2], [3], tennis [4]–[6], volleyball [7]–[10] and handball [11], [12].

Anand et al. [4] and Bai et al. [2] used IMU data for detecting shots in respectively swing sports and basketball. With precisions over 90%, both studies showed that it is possible to detect shots with machine learning based on IMU data in a game scenario. Anand et al. [4] also classified between multiple shot types with an accuracy above 90% in tennis and squash. In handball, no classification has yet been done on during a game scenario. Handball is a popular, Olympic sport where shoulder injuries are highly prevalent [13]. One of the factors for obtaining those shoulder injuries is training load [14]. Quantifying the training load can therefore be beneficial for reducing injury risk. In handball, both passes and shots on goal occur. The shots can be done standing, running or jumping. Furthermore, there are two types of throws based on the wind up: whip-like and circular. With a whip-like wind up, the ball moves from the front of the body upwards and then backwards, while with a circular wind up, the ball moves from the front first down and backward, making a circular motion [15]. Until now, one study has tried to classified the different throws in handball based on IMU data [11] and this was in a lab setting.

Therefore, the aim of this study is to use machine learning algorithms to detect handball throws and to classify between different throwing methods and wind ups in handball based on wrist IMU data during a practice or practice match.

2. Methods

2.1 Participants

Data were collected from nine handball players in six practices. All players were right-handed and injury-free during data collection. The players were female semi-professional players in Norway or female youth elite players in the Netherlands. The study complied with the current ethical regulations for research, conformed to the latest revision of the Declaration of Helsinki.

2.2 Procedure

The measurements were performed during regular team training sessions. During practice, the players wore a sweatband with IMU on the distal dorsal side of the throwing arm. An Axivity ax6 IMU containing a 3-axis accelerometer (16 g range, 1600 Hz sampling frequency) and 3-axis gyroscope (2000 °/s, 1600 Hz) was used. The session was recorded using a camera.

Throws were manually classified based on the videos and recorded in a excel file with time of the throw (to a second accurate) and the throw type. The possible throw types are given in table 1. The calibration time was also noted to later align the video and IMU data.

The IMU data was loaded in by the Open Movement GUI application (version 1.0.0.43, Newcastle University UK) and saved as an csv file. The data was pre-processed and analysed with MATLAB (version 2019a). A lowpass filter with cut-off frequency of 7 Hz was used to pre-process the data. In order to obtain individual throws, the data was segmented by finding the local maxima in the filtered gyroscope x-axis data which were at least 500 °/s in prominence and 0.75 s after the last peak and a window of 1.5 s around the peak was extracted.

The features calculated within each window included axis mean, minimum, maximum, difference, standard deviation, variance, skewness, kurtosis, root-mean square and coefficient of variation. These were calculated for each axis (x, y, z) and the vector magnitudes ($\sqrt{x^2 + y^2 + z^2}$), resulting in a total of 60 features. The events were normalized between 0 and 1 for equal contribution in the machine learning algorithms.

Table 1 – Handball throw types

Shot	Pass
jump circle	pass left circle
jump whip-like	pass left whip-like
running circle	pass right circle
running whip-like	pass right whip-like
standing circle	pass forward circle
standing whip-like	pass forward whip-like
underarm shot	pass underarm
	pass right push
	pass left push
	pass forward push

2.3 Modelling

2.3.1 Event detection

In separating throws and non-throws, a binary classification was needed. With the classification app on MATLAB, Quadratic Discriminant, kernel Naïve Bayes, cosine k-Nearest Neighbours and RUSBoosted Trees provided the highest initial accuracies. In section 2.3.4 all algorithms will be explained.

2.3.2 Classification

All throws were manually classified into one of the 17 classes of table 1. To test if less than 17 throw types or a different division of the throws give different results, a total of five categories were created. Table 2 lists all the categories and which machine learning algorithms were used. The algorithms were chosen based on the highest accuracies provided by the classification application on MATLAB and varied per category. The first category uses all 17 throw classes as presented in table 1. The second category consists of two classes: shots and passes, table 1 shows which throws are considered shots and which are considered passes. Category three consists of seven classes: jump shots, running shots, standing shots, underhand throws, passes to left, passes to right, passes forward. Category four focusses on the wind up type and consists of three classes: whip-like, circular and other wind up. With a whip-like wind up, the ball moves from the front of the body upwards and then backwards, while with a circular wind up, the ball moves from the front first down and backward, making a circular motion [15]. Category five consists of two classes: high intensity and low intensity, where underhand passes and push passes are considered low intensity and all other throws high intensity since underhand passes and push passes are very minor movements compared to the other throws and passes and therefore considered as low intensity throws.

Table 2 – ML algorithms per category

Category	17 throw types	Shots and passes	7 throw types	Wind up type	Intensity
ML model					
Linear Discriminant				X	
Quadratic Discriminant		X			X
Kernel NB					X
Fine kNN	X				X
Weighted kNN				X	
Subspace kNN	X	X	X		
Quadratic SVM	X			X	
Cubic SVM	X	X	X		X
Bagged Trees	X		X	X	
RUSBoosted Trees		X			X

2.3.2 Machine learning algorithms

Linear and Quadratic Discriminant (LDA and QDA) analysis classifier works with a linear or quadratic decision surface which are easily computed and can work for multiclass problems [16], [17].

Naïve Bayes (NB) is, just like the Quadratic Discriminant classifier a supervised machine learning model, but based on Bayes theorem of probability to predict the class of data sets [18]. Where normal Naïve Bayes works with a Gaussian distribution, kernel NB uses a continuous distribution and is thereby more flexible [19].

K-Nearest Neighbour (kNN) is also a supervised machine learning model but can be used for both classification and regression. It assumes that similar data points belong to the same class, so new data is compared to the training data and classified to the class with the highest similarities [20]. While the standard kNN uses the Euclidean distance between points, with cosine kNN one minus the cosine of the angle between observations is used. With fine kNN, a small number of training samples which are close to the test samples are found before identifying the k-nearest neighbours, therefore also the distance based on representation is taken into account [21]. With weighted kNN, the nearest k points are given a weight based on distance and distance-weighted voting provides the class [22]. Lastly, the subspace kNN is trained on a random set of features instead of the entire set of features to reduce the correlation between features [23].

With the Support Vector Machine (SVM) classifier, the training examples are marked to belong to one of two categories [24]. New examples are assigned to one of the categories based on the training data. It is therefore a non-probabilistic binary linear classifier. It can be altered to work for multiclass problems, where the multiclass problem is reduced into multiple binary classification problem. With quadratic SVM, instead of using a line to separate the categories, a quadratic function is used [25]. With cubic SVM, a cubic function is used for the separation [26].

Decision Trees (DT) can be used for classification and regression. It separates the input into classes in a hierarchic way. If finds the best attribute to split the data, and continues doing this until it cannot be separated anymore [27]. With a Random Forest (RF) classifier, many individual decision trees are used that each give a class prediction. The most prevalent class prediction becomes the model's classification [28]. The bagged trees is an algorithm that fits in between a standard DT and the RF model. Where bagged trees make multiple decision trees and choses the most prevalent class, Random Forest goes a bit further in making sure the sub-trees are uncorrelated [29]. The RUSBoost algorithm is designed for imbalanced databases where with random undersampling (RUS) examples from the majority class are removed until the dataset is balanced [30]. In combination, the RUSBoosted Trees algorithm runs boosted training data repeatedly on weak learners and combine it into a strong classifier.

2.4 Evaluation of event detection & classification

In order to evaluate the event detection and classification, cross-validation was applied. After training and testing the algorithm, a confusion matrix was made which provided the true and false positive and negative results of the prediction (Table 3).

Table 3 – Confusion matrix

		Reference	
		Positive	Negative
Prediction	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

With the values from the confusion matrix, the accuracy, sensitivity, specificity, precision and F1-score were calculated (Eq. 1-4) for both the event detection algorithms and the classification algorithms. For the binary classification (event detection and categories 2 and 5 of the classification), the results could be calculated directly, for the multi-class algorithms (categories 1, 3 and 4 of the classification), the weighted average was taken of the results of the individual classes.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$F1 - score = \frac{2\ x\ precision\ x\ sensitivity}{precision+sensitivity} \quad (4)$$

3. Results

In total, 16 players were measured doing a total of 2475 throws. All throws that were recorded as one of the shot types from table 1 were included. Other annotated throws were either changed to a corresponding shot type from table 1 or excluded due to not being a handball throw. Figure 1 shows the number of throws per type in each category.

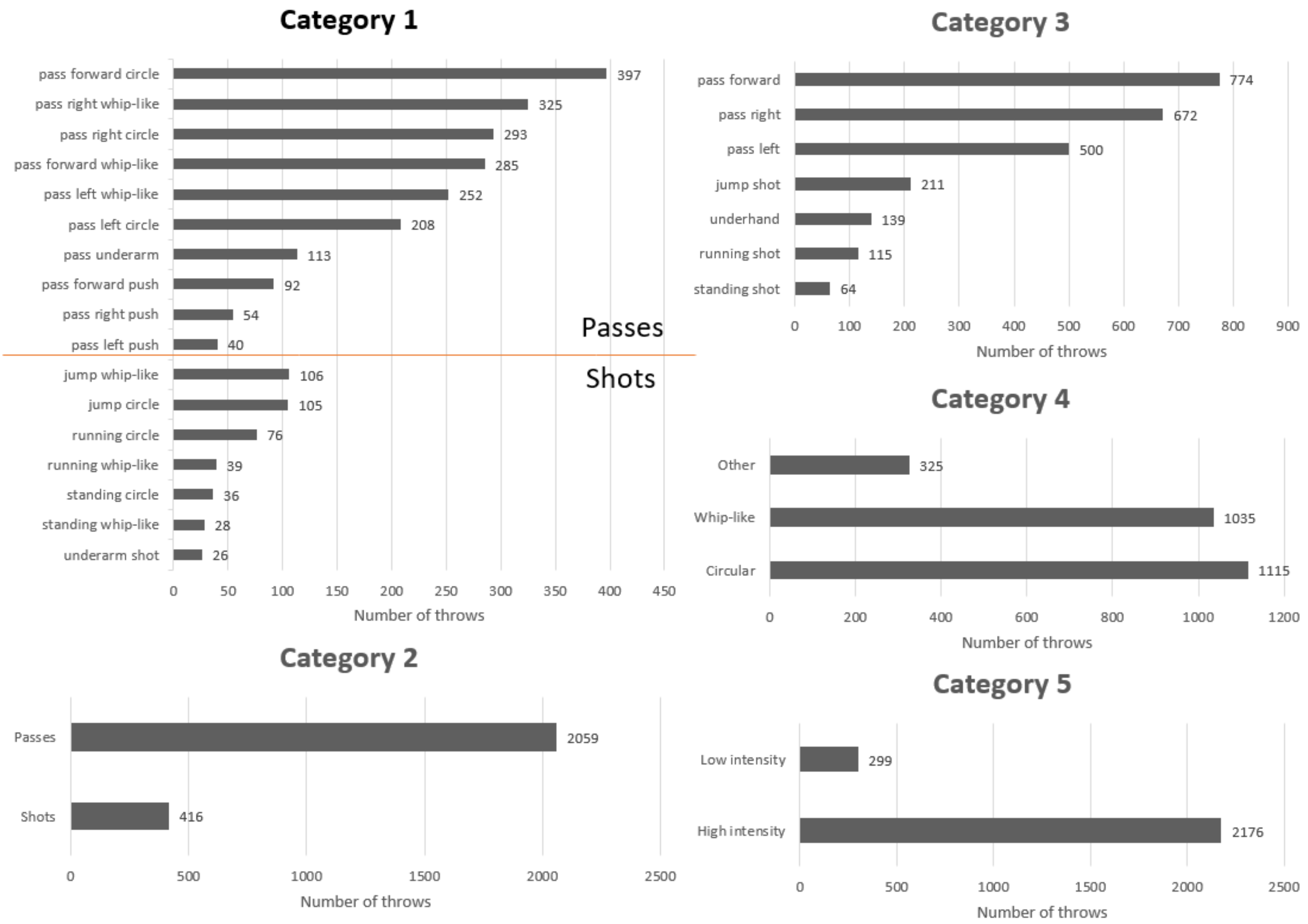

Figure 1 - Number of throws per category

3.1 Event detection

The evaluation scores for the four algorithms used for event detection are shown in figure 2.

This shows that the medians of the accuracy, F1-score and sensitivity were highest with the kNN algorithm (accuracy of 0.94 for k-Nearest Neighbour (kNN), 0.92 for Quadratic Discriminant analysis (QDA) and Decision Tree (DT), 0.90 for Naïve Bayes (NB); F1-score 0.81 for kNN, 0.76 for QDA and DT, 0.73 for NB; sensitivity 0.79 for kNN, 0.68 for QDA and DT, 0.62 for NB). For the precision, NB had the highest median (0.88 for NB, 0.86 for QDA and DT, 0.83 for kNN).

Since the kNN algorithm provides the highest F1-score, the confusion matrix for this algorithm is shown in table 4. This shows that, over all 20 rounds, 3378 throws were missed (FN) and 4418 non-throws were classified as throws (FP). Appendix 1 gives the confusion matrices of all the algorithms.

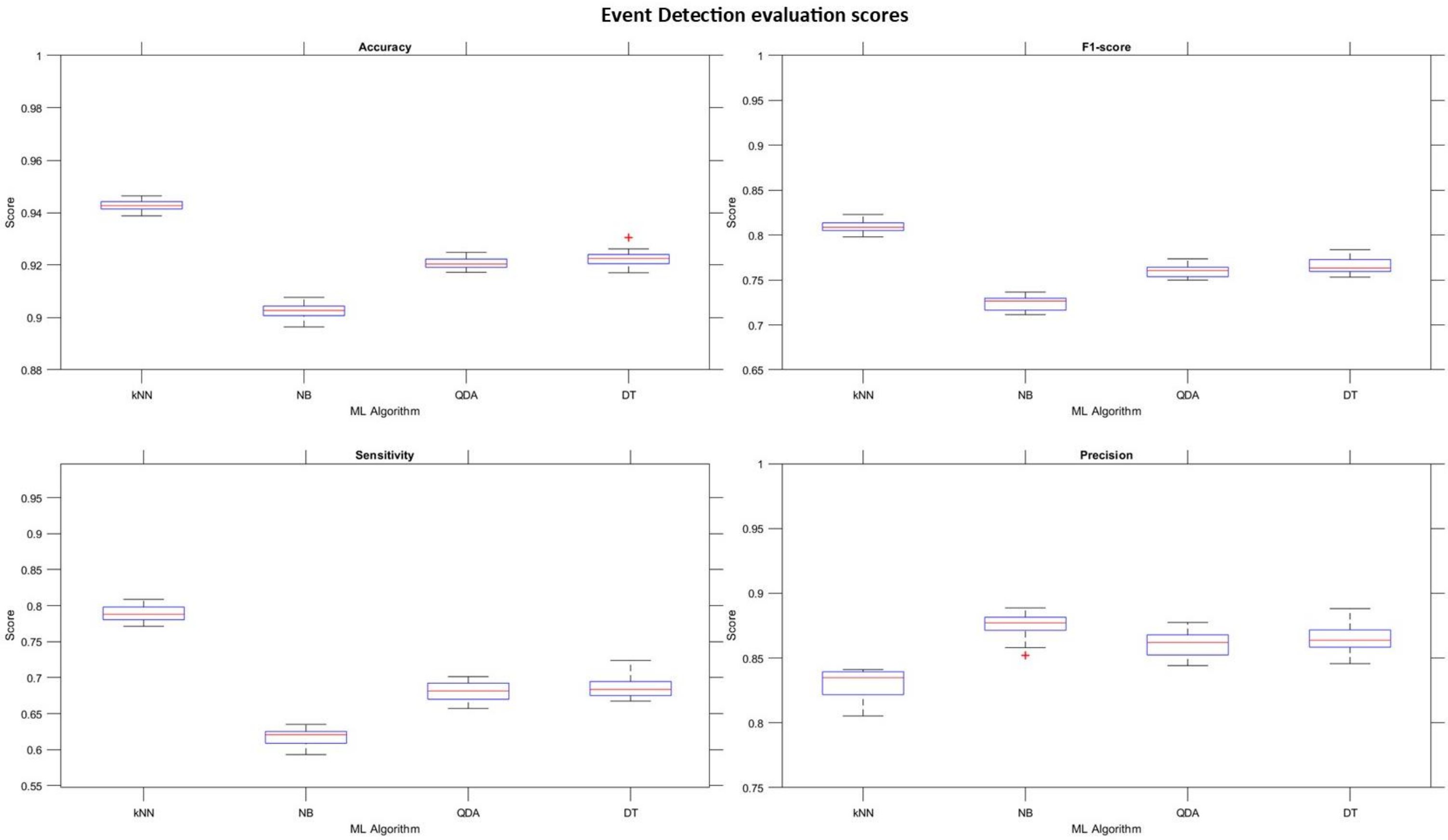

Figure 2 - Evaluation scores event detection. kNN = cosine k-Nearest Neighbours, NB = Kernel Naïve Bayes, QDA = Quadratic Discriminant Analysis, DT = RUSBoosted Decision Trees.

Table 4 – Confusion matrix of kNN algorithm over all 20 rounds

		Reference	
		Throw	Non-throw
Prediction	Throw	16518	4418
	Non-throw	3378	111926

3.2 Shot classification

Appendix 2 shows figures with the evaluation scores for all 5 categories and the different algorithms. All the median scores with corresponding algorithm are provided in table 5.

As shown in table 4, cubic SVM was the algorithm with the highest evaluation scores for category 1 (all throw types) (accuracy of 0.97, F1-score of 0.62, precision of 0.57 and sensitivity of 0.69). For category 2 (shots vs. passes), cubic SVM was the algorithm with the highest accuracy, F1-score and sensitivity (accuracy of 0.93, F1-score of 0.77 and sensitivity of 0.80), while the RUSBoosted DT gave the highest precision (0.76). For category 3 (7 throw classes), cubic SVM gave the highest accuracy, F1-score and precision (accuracy of 0.95, F1-score of 0.69 and precision of 0.64), while the subspace kNN had the highest sensitivity (sensitivity of 0.76). For category 4 (wind up type), quadratic SVM gave the highest scores (accuracy of 0.81, F1-score of 0.79, precision of 0.78 and sensitivity of 0.81). For category 5 (intensity), cubic SVM gave the highest accuracy, F1-score and precision (accuracy of 0.90, F1-score of 0.94 and precision of 0.94), while the RUSBoosted DT gave the highest sensitivity (sensitivity of 0.98).

Based on the F1-scores, category 5 (intensity) presented the highest score (0.94), category 4 (wind up type) had the second highest score (0.79), category 2 (shots vs. passes) the third highest (0.77), category 3 (7 throw classes) the fourth highest score (0.69) and category 1 (all throw types) the lowest F1-score (0.62).

The confusion matrices for all categories for the algorithm with the highest F1-score are shown in appendix 3.

Table 5 – Evaluation scores all categories and algorithms. The algorithms with the highest scores are bold.

	Category 1					**Category 2**				**Category 3**			**Category 4**				**Category 5**				
	qSVM	**cSVM**	kNN	DT	skNN	QDA	**cSVM**	skNN	DT	**cSVM**	DT	skNN	LDA	**qSVM**	wkNN	DT	QDA	NB	**cSVM**	kNN	DT
Accuracy	0.96	**0.97**	0.96	**0.97**	**0.97**	0.90	**0.93**	**0.93**	0.89	**0.95**	0.94	**0.95**	0.78	**0.81**	0.78	0.78	0.87	0.77	**0.90**	0.89	0.82
F1	0.54	**0.62**	0.54	**0.62**	0.60	0.70	**0.77**	**0.77**	0.69	**0.69**	0.66	0.67	0.76	**0.79**	0.76	0.76	0.93	0.85	**0.94**	**0.94**	0.89
Precision	0.53	**0.57**	0.53	**0.57**	0.55	0.71	0.73	0.71	**0.76**	**0.64**	**0.64**	0.61	0.74	**0.78**	**0.78**	0.75	0.92	0.76	**0.94**	**0.94**	0.82
Sensitivity	0.55	**0.69**	0.59	0.68	0.63	0.69	**0.80**	0.84	0.63	0.74	0.68	**0.76**	0.78	**0.81**	0.75	0.75	0.94	0.97	0.94	0.93	**0.98**

4. Discussion

The aim of the study was to use machine learning algorithms to detect handball throws and to classify between different throwing types and wind ups in handball based on wrist IMU data during a practice or practice match. For the detection of handball throws, the cosine k-Nearest Neighbours (kNN) algorithm was the best with an accuracy of 0.94 and a F1-score of 0.81. For the classification between different throwing types and wind ups, five categories were created with all 17 throw types (category 1), a binary category which classified between shots and passes (category 2), a category with 7 different throw types (category 3), a category with the different wind up types (category 4) and an intensity-based category (category 5). For all categories, support vector machines provided the highest evaluation scores (accuracy and F1-score). As shown in table 4, category 5 gave the highest scores (accuracy of 0.90 and F1-score of 0.94), followed by category 4 (accuracy of 0.81 and F1-score of 0.79). Category 2 (accuracy of 0.93 and F1-score of 0.77) had a higher accuracy than category 4, but a lower F1-score. Category 3 (accuracy of 0.95 and F1-score of 0.69) and category 1 (accuracy of 0.97 and F1-score of 0.62) had clearly lower F1-scores.

In this study, the F1-scores were deemed more important than the accuracy, since the accuracy is based as much on the true positives as on the true negatives, while the F1-score focusses more on the true positives. Since the multiclass categories have a lot more true negatives than true positives and since we find it more important to get the amount of true positives right, the focus was mainly on the true positive values and thereby the F1-score.

Limitations

Although category 1 scored lowest off all, it is the most important category since it contains all the possible throw types that we want to classify. Therefore, it is important to look at the limitations of this study to see how the results of especially this category, but preferably all categories can be improved.

There are two crucial parts with machine learning: the database on which machine learning is performed and the machine learning algorithms used. In this study, the database seems the main limitation since the database is imbalanced. For the event detection, this is shown in table 3 where there are only 20936 throws versus 115304 non-throws. For the classification, the imbalance becomes clear when looking to figure 1. This figure shows that in all categories there is a large difference in the number of throws in a category. To find ways to reduce these imbalances, there are many parts of building the database that need to be discussed: collecting the data, labelling it, preprocessing and segmentation the data and lastly feature extraction.

The collection of the data was done during handball practices or practice matches instead of the more common lab gathering of data. The main drawback to this was that there was no control over what throws were done. During the warming up, many passes were done and not as many shots on goal, which resulted largely in the imbalance. The advantage of this way of gathering data is that the movements are natural and closer to the movements which need to be classified in the end: real movements with defenders and in-game scenarios. Also, for the event detection, the real life data is more interesting, since during a practice, there are many more movements which are not done in the lab such as bouncing the ball, defending or high fiving a teammate. This study shows that even with the practice data event detection is possible. Another issue with the collection of the data was that some of the throw types are rarely done. There were for example fewer than 50 throws for the underarm shot (26), standing whip-like shot (28), standing circular shot (36), running whip-like shot (39) and left push pass (40). For machine learning, these amount of datapoints for one class is too low [31].

Furthermore, labelling of the data was done manually by three different persons at different times. Although for most throws the labelling was very clear, there were in all practices some controversial shots or shots that were hard to put into one category. Since the labelling was done at different times and by different people, there might be some similar throws that were classified differently. Despite this being the case, it is expected that on the total number of throws, this makes only a marginal impact. There were also some throws (e.g. jump passes) that didn't fit into one category. These were labelled as 'other' or described in words and later put into the category which resembled the throw the best.

For the preprocessing and segmentation, a variety of methods were tried, before concluding the lowpass filter and the peak with threshold method worked best. The chosen threshold was quite conservative (500 °/s on the x-axis gyroscope data) since we wanted to be sure none of the shots were missed due to be filtered out. The event detection was based on the machine learning and not on filtering. But to balance out the large difference of throws and non-throws, it might be better to first make a selection based on one or multiple thresholds before using the machine learning. The main drawback of this would be that some of the less intense throws (e.g. underhand passes and push passes) will get filtered out. The question arises if this would really be a problem. If using classification for injury prevention than probably not, since these movements are less forceful. But for performance and counting which players do what types of throws, it might still be interesting to also show the less intense throws. On that same note, there are also movements which are high intensity and resemble throws that are not taken into account in this study, like fake throws.

The last part of building the database is the feature extraction. For this study, a large number of features (60) were used over all axis and the vector magnitudes of both the accelerometer and the gyroscope. It has not been tested if different, fewer or more features would provide better results, nor has it been investigated which features were mainly used by the machine learning algorithms to make the predications.

Next to the imbalance in the database, the machine learning was also a vital part of this study. Although the F1-scores were deemed more important in this study, the choice of which algorithms to use were based on which provided the highest accuracies in the classification application in MATLAB. For all categories, 3-5 different algorithms were chosen to be used. All these algorithms were used in with their default settings. Results might improve if the settings were adjusted to better fit each of the categories. For all different categories, SVM came out as the best classifier, although it must be said that the results were often very close as can be seen in table 4. Therefore, different algorithms might provide better results than the SVM if different settings are used.

Future work

The most important point to improve is the database. Although using data obtained during practices has advantages, it results in such an imbalanced database that a lab-based database is preferred. In the lab, all shots can be done in equal numbers. The lab-based database can then be used for training the machine learning algorithms after which it can be tested with datasets from practices in order to find out if the system works.

Also, the categories chosen might need to be looked at. Some of the categories were less informative than the other, like the intensity-based category where almost all throw types were in the high intensity category. This also leads to the before mentioned point of the goal of classifying the throws. Is it desired to measure all the different throws, or are the low intensity throws less important than for example shots on goal?

For the building of the database, it might be interesting to see how the event detection works with more filters. Either on the raw data, or on some of the features. With this, the imbalance in the event detection database would be less. Another thing to investigate is which features are used and if these could be used in a better way.

Another step which can improve the results is the machine learning algorithms. As mentioned before, in this study the default settings were used. It would be good to experiment with different settings.

Lastly, the computation and loading time was quite long, up to 30 minutes for one practice. The measurements were done with a frequency of 1600 Hz, resulting in a lot of datapoints. Lowering this frequency might result in similar results in a shorter time.

5. Conclusion

For the detection of throws during a handball practice, the k-Nearest Neighbours algorithm is the best machine learning algorithm based on accelerometer and gyroscope data of the dominant wrist. For the classification of the different throws, support vector machine is the best machine learning algorithm, although other machine learning algorithms gave similar results. The evaluation scores were highest for classifying between high and low intensity throws, classification based on wind up type and classification between throws and passes. The results were lower when trying to classify all throw types or classifying seven different throw types. In order to improve the results for all throw types, the dataset needs to be larger and more balanced with all types of throws in equal amounts.

Acknowledgements

A big thank you is in place for all players of Levanger Handball Club and the Dutch Handball Academy who participated in collecting data, as well as to Anouk Vergunst who sew the sweat band with pockets for the IMU and Linda van Maanen-Coppers for manually classifying video data.

References

[1] J. McGrath, J. Neville, T. Stewart, and J. Cronin, “Upper body activity classification using an inertial measurement unit in court and field-based sports: A systematic review,” *Proc. Inst. Mech. Eng. Part P J. Sport. Eng. Technol.*, p. 175433712095975, Oct. 2020.

[2] L. Bai, C. Efstratiou, and C. S. Ang, “WeSport: Utilising wrist-band sensing to detect player activities in basketball games,” in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2016*, 2016.

[3] B. Eggert, M. Mundt, and B. Markert, “Imu-Based Activity Recognition of the Basketball Jump Shot,” *ISBS Proc. Arch.*, vol. 38, no. 1, pp. 344–347, Jul. 2020.

[4] A. Anand, M. Sharma, R. Srivastava, L. Kaligounder, and D. Prakash, “Wearable motion sensor based analysis of swing sports,” in *Proceedings - 16th IEEE International Conference on Machine Learning and Applications, ICMLA 2017*, 2017, vol. 2017-Decem, pp. 261–267.

[5] C. Ó. Conaire, D. Connaghan, P. Kelly, N. E. O’Connor, M. Gaffney, and J. Buckley, “Combining inertial and visual sensing for human action recognition in tennis,” in *ARTEMIS’10 - Proceedings of the 1st ACM Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams, Co-located with ACM Multimedia 2010*, 2010, pp. 51–56.

[6] D. Whiteside, O. Cant, M. Connolly, and M. Reid, “Monitoring hitting load in tennis using inertial sensors and machine learning,” *Int. J. Sports Physiol. Perform.*, vol. 12, no. 9, pp. 1212–1217, Oct. 2017.

[7] F. Haider *et al.*, “Evaluation of dominant and non-dominant hand movements for volleyball action modelling,” in *Adjunct of the 2019 International Conference on Multimodal Interaction, ICMI 2019*, 2019, pp. 1–6.

[8] A. K. Holatka, H. Suwa, and K. Yasumoto, “Volleyball Setting Technique Assessment Using a Single Point Sensor,” in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2019*, 2019, pp. 567–572.

[9] K. Peng, Y. Zhao, X. Sha, W. Ma, Y. Wang, and W. J. Li, “Accurate Recognition of Volleyball Motion Based on Fusion of MEMS Inertial Measurement Unit and Video Analytic,” in *8th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, CYBER 2018*, 2019, pp. 440–444.

[10] Y. Wang, Y. Zhao, R. H. M. Chan, and W. J. Li, “Volleyball Skill Assessment Using a Single Wearable Micro Inertial Measurement Unit at Wrist,” *IEEE Access*, vol. 6, pp. 13758–13765, 2018.

[11] R. van den Tillaar, S. Bhandurge, and T. Stewart, “Can machine learning with imus be used to detect different throws and estimate ball velocity in team handball?,” *Sensors*, vol. 21, no. 7, Apr. 2021.

[12] C. Gençoğlu and H. Gümüş, “Standing Handball Throwing Velocity Estimation with a Single Wrist-Mounted Inertial Sensor,” *Ann Appl Sport Sci*, vol. 8, no. s1, p. 893, Oct. 2020.

[13] B. Clarsen, R. Bahr, S. H. Andersson, R. Munk, and G. Myklebust, “Reduced glenohumeral rotation, external rotation weakness and scapular dyskinesis are risk factors for shoulder injuries among elite male handball players: a prospective cohort study,” *Br. J. Sports Med.*, vol. 48, no. 17, pp. 1327–1333, 2014.

[14] M. A. Mohseni-Bandpei, R. Keshavarz, H. Minoonejhad, H. Mohsenifar, and H. Shakeri,

"Shoulder pain in Iranian elite athletes: the prevalence and risk factors," *J. Manipulative Physiol. Ther.*, vol. 35, no. 7, pp. 541–548, Sep. 2012.

[15] R. Van den Tillaar, A. Zondag, and J. Cabri, "Comparing performance and kinematics of throwing with a circular and whip-like wind up by experienced handball players," *Scand. J. Med. Sci. Sports*, vol. 23, no. 6, pp. e373–e380, Dec. 2013.

[16] A. Tharwat, "Linear vs. quadratic discriminant analysis classifier: a tutorial," *Int. J. Appl. Pattern Recognit.*, vol. 3, no. 2, pp. 145–180, 2016.

[17] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, "Linear discriminant analysis: A detailed tutorial," *AI Commun.*, vol. 30, no. 2, pp. 169–190, Jan. 2017.

[18] G. I. Webb, E. Keogh, and R. Miikkulainen, "Naïve Bayes," in *Encyclopedia of machine learning*, 2010, pp. 713–714.

[19] A. Pérez, P. Larrañaga, and I. Inza, "Bayesian classifiers based on kernel density estimation: Flexible classifiers," *Int. J. Approx. Reason.*, vol. 50, no. 2, pp. 341–362, Feb. 2009.

[20] H. S. Sahak Kaghyan, "Activity recognition using k-nearest neighbor algorithm on smartphone with tri-axial accelerometer," *Int. J. "Information Model. Anal.*, vol. 1, pp. 146–156, 2012.

[21] Y. Xu, Q. Zhu, Z. Fan, M. Qiu, Y. Chen, and H. Liu, "Coarse to fine K nearest neighbor classifier," *Pattern Recognit. Lett.*, vol. 34, no. 9, pp. 980–986, 2013.

[22] W. Zuo, D. Zhang, and K. Wang, "On kernel difference-weighted k-nearest neighbor classification," *Pattern Anal. Appl. 2008 113*, vol. 11, no. 3, pp. 247–257, Jan. 2008.

[23] G. Tremblay, R. Sabourin, and P. Maupin, "Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm," *Proc. - Int. Conf. Pattern Recognit.*, vol. 1, pp. 208–211, 2004.

[24] V. Jakkula, "Tutorial on Support Vector Machine (SVM)."

[25] I. Dagher, "Quadratic kernel-free non-linear support vector machine," *J. Glob. Optim. 2007 411*, vol. 41, no. 1, pp. 15–30, Jun. 2007.

[26] U. Jain, K. Nathani, N. Ruban, A. N. J. Raj, Z. Zhuang, and V. G. V. Mahesh, "Cubic SVM classifier based feature extraction and emotion detection from speech signals," *Proc. - 2018 Int. Conf. Sens. Networks Signal Process. SNSP 2018*, pp. 386–391, Jan. 2019.

[27] Y. SONG and Y. LU, "Decision tree methods: applications for classification and prediction," *Shanghai Arch. Psychiatry*, vol. 27, no. 2, p. 130, Apr. 2015.

[28] G. Biau and E. Scornet, "A random forest guided tour," *TEST 2016 252*, vol. 25, no. 2, pp. 197–227, Apr. 2016.

[29] R. T. Guy, P. Santago, and C. D. Langefeld, "Bootstrap Aggregating of Alternating Decision Trees to Detect Sets of SNPs that Associate with Disease," *Genet. Epidemiol.*, vol. 36, no. 2, p. 99, Feb. 2012.

[30] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A Hybrid Approach to Alleviating Class Imbalance," *Syst. HUMANS*, vol. 40, no. 1, 2010.

[31] R. Zaman Khan, H. Allamy, W. Hong, and H. D. Khalaf Jabbar Rafiqul Zaman Khan, "Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study)," 2015.

Appendix 1 – Confusion Matrices all event detection algorithms

Table 6 - Confusion matrix cosine kNN over all 20 rounds in amount of throws

		Reference	
		Throw	Non-throw
Prediction	Throw	16518	4418
	Non-throw	3378	111926

Table 7 - Confusion matrix cosine kNN over all 20 round in percentages

		Reference	
		Throw	Non-throw
Prediction	Throw	83.0%	3.8%
	Non-throw	17.0%	96.2%

Table 8 - Confusion matrix kernel Naïve Bayes over all 20 rounds in amount of throws

		Reference	
		Throw	Non-throw
Prediction	Throw	17420	10814
	Non-throw	2476	105530

Table 9 - Confusion matrix kernel Naïve Bayes over all 20 rounds in percentages

		Reference	
		Throw	Non-throw
Prediction	Throw	87.6%	9.3%
	Non-throw	12.4%	90.7%

Table 10 - Confusion matrix Quadratic Discriminant analysis over all 20 rounds in amount of throws

		Reference	
		Throw	Non-throw
Prediction	Throw	17133	8046
	Non-throw	2763	108298

Table 11 - Confusion matrix Quadratic Discriminant analysis over all 20 rounds in percentages

		Reference	
		Throw	Non-throw
Prediction	Throw	86.1%	6.9%
	Non-throw	13.9%	93.1%

Table 12 – Confusion matrix RUSBoosted Trees over all 20 rounds in amount of throws

		Reference	
		Throw	Non-throw
Prediction	Throw	17232	7891
	Non-throw	2664	108453

Table 13 – Confusion matrix RUSBoosted Trees over all 20 rounds in percentages

		Reference	
		Throw	Non-throw
Prediction	Throw	86.6%	6.8%
	Non-throw	13.4%	93.2%

Appendix 2 – Shot classification results

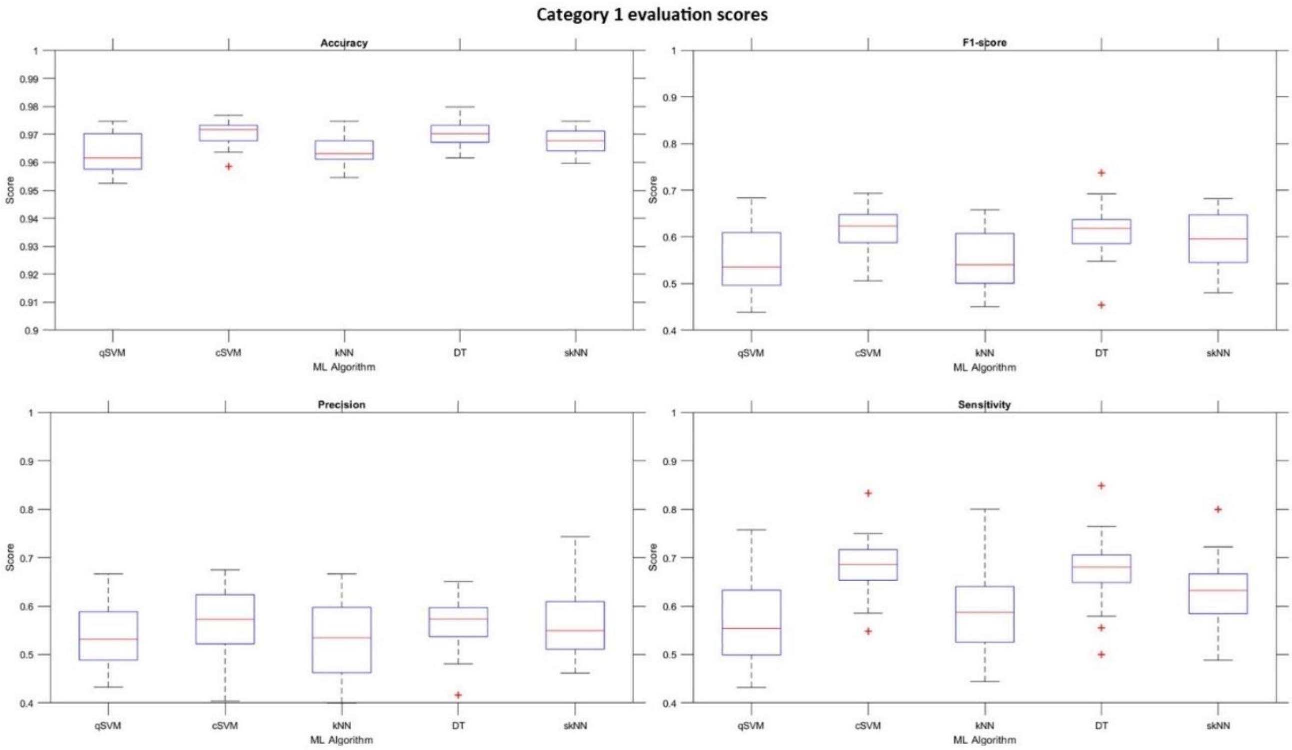

Figure 3 - Evaluation scores category 1 (all throws)

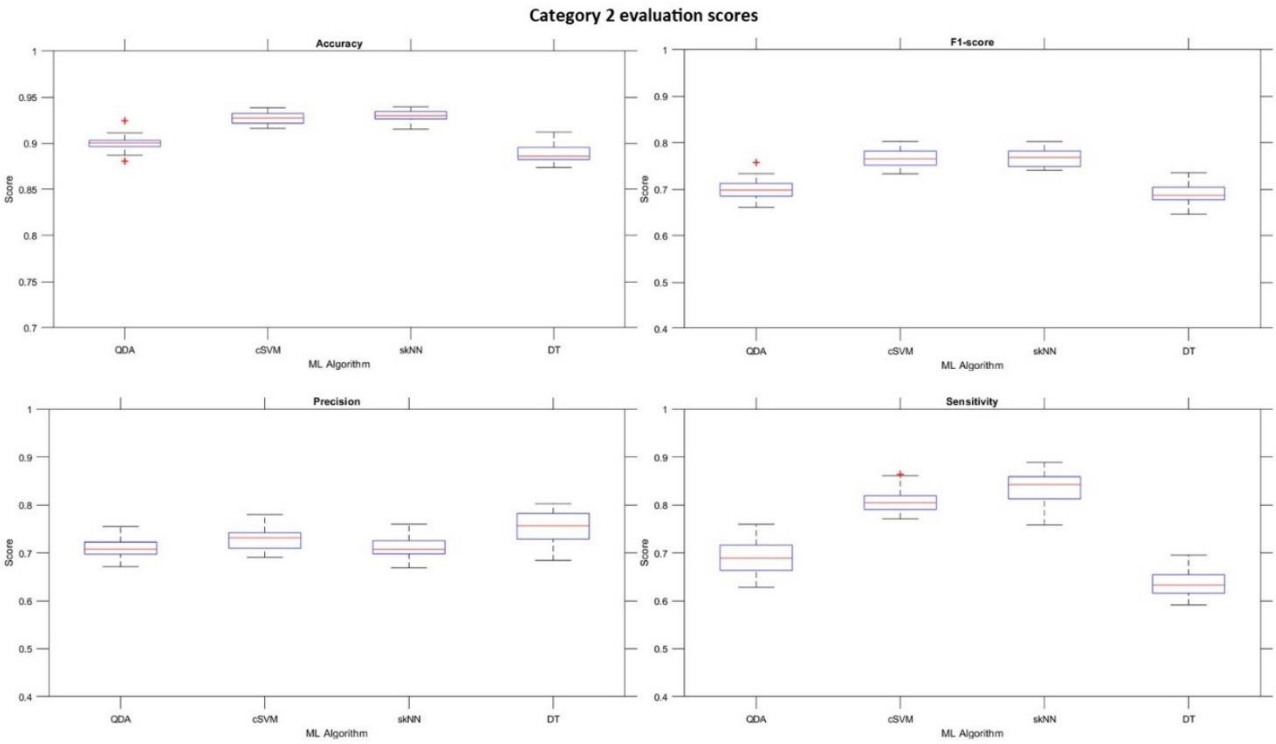

Figure 4 - Evaluation scores category 2 (shots vs. passes)

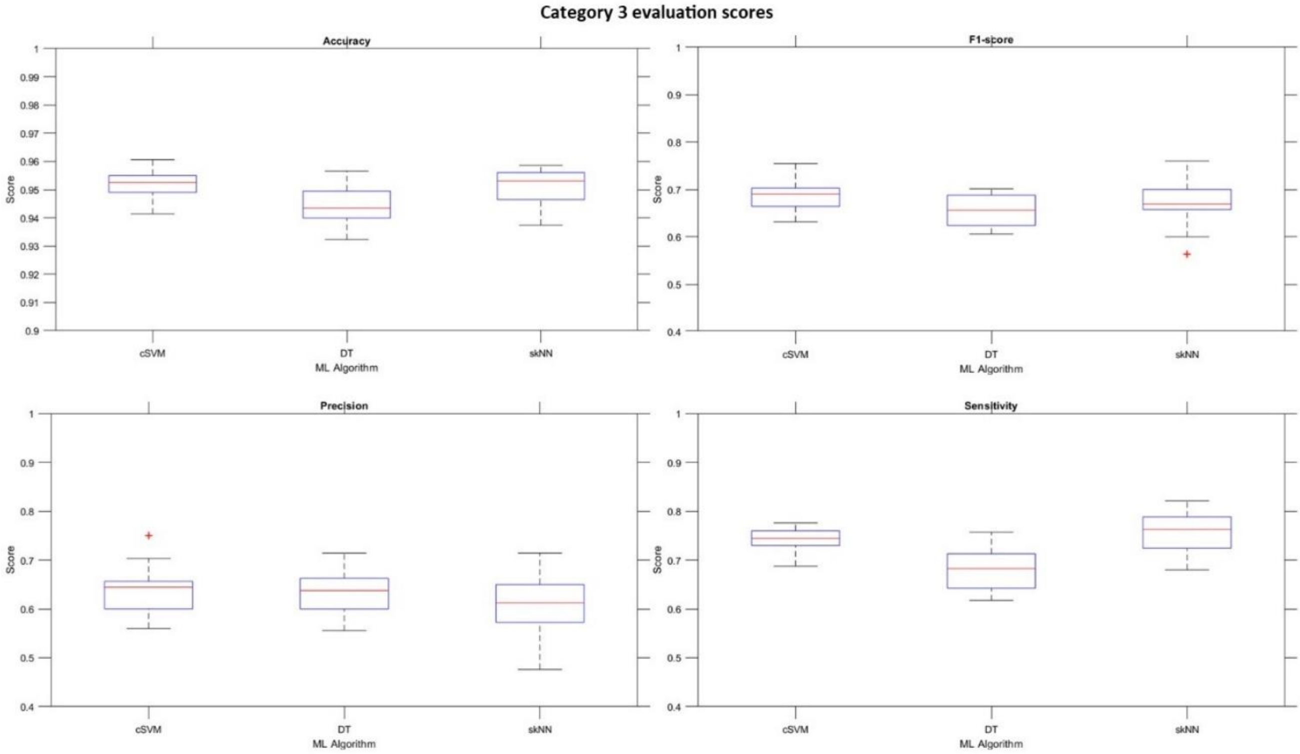

Figure 5 - Evaluation scores category 3 (7 throw classes)

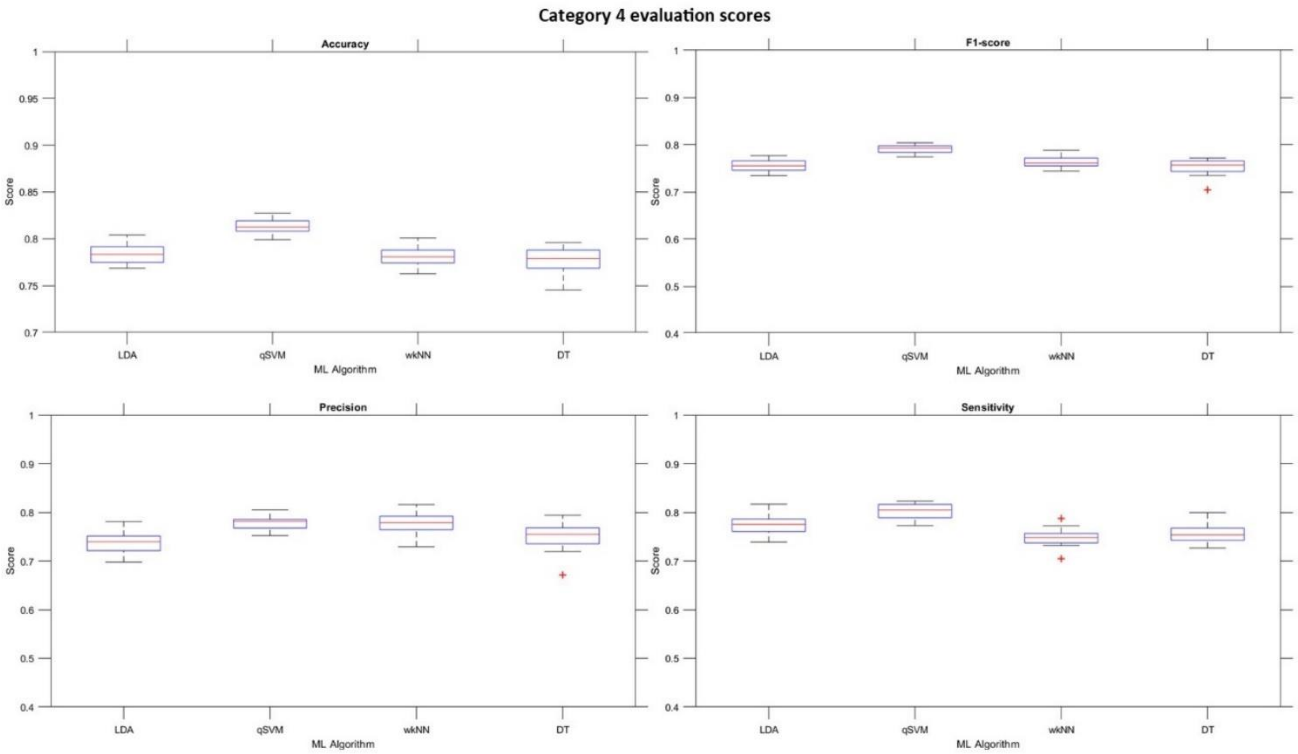

Figure 6 - Evaluation scores category 4 (wind up type)

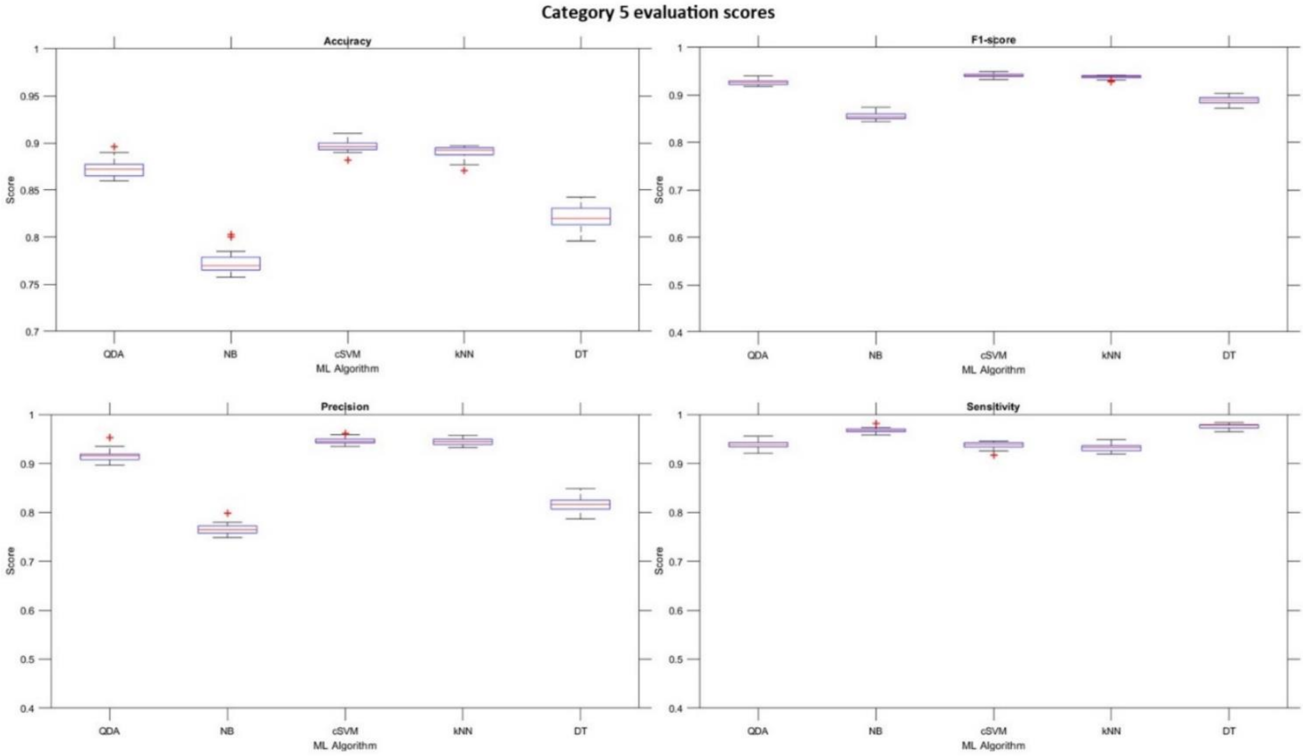

Figure 7 - Evaluation scores category 5 (intensity)

Appendix 3 – Confusion Matrices shot classification

Table 14 – Confusion matrix category 1 (all throw types) in percentages over 20 rounds[1]. In green the true positives. (JSC = jump shot circular, JSW = jump shot whip-like, RC = running circular, RW = running whip-like, SC = standing circular, SW = standing whip-like, US = underarm shot, PLC = pass left circular, PRC = pass right circular, PLW = pass left whip-like, PRW = pass right whip-like, PFC = pass forward circular, PFW = pass forward whip-like, PU = pass underarm, PRP = pass right push, PLP = pass left push, PP = pass forward push)

		Prediction																
		JSC	JSW	RC	RW	SC	SW	US	PLC	PRC	PLW	PRW	PFC	PFW	PU	PRP	PLP	PP
Reference	JSC	56.7	8.9	8.9	1.6	2.3	0.3	0.9	0.8	6.3	0.2	1.9	4.2	2.7	2.1	1.1	0.9	0.3
	JSW	13.1	40.1	2.6	7.5	0.9	1.1	0.3	3.5	2.6	2.4	4.4	11.1	3.4	3.2	0.5	0.3	3.1
	RC	7.6	2.5	50.3	5.0	2.5	1.0	0.8	2.8	6.8	2.9	4.1	7.2	3.5	0.0	1.1	0.0	1.9
	RW	2.5	1.7	13.6	44.1	2.8	5.4	2.8	1.7	1.7	4.5	1.1	14.1	4.0	0.0	0.0	0.0	0.0
	SC	10.6	0.6	9.4	4.4	40.1	0.9	0.0	2.1	13.3	0.6	2.9	11.8	0.0	1.5	0.3	0.9	0.6
	SW	0.0	7.5	2.3	21.8	0.0	36.5	2.6	8.3	0.8	1.9	12.0	3.8	1.1	0.4	0.0	0.0	1.1
	US	3.7	4.9	13.9	7.3	5.3	2.0	33.1	1.2	5.3	6.5	2.0	3.7	2.4	1.2	0.8	0.0	6.5
	PLC	0.3	0.0	0.9	0.7	0.4	1.4	0.1	31.9	20.1	11.0	13.2	11.4	2.2	2.8	1.0	1.0	1.3
	PRC	0.7	0.1	1.2	0.4	0.6	0.0	0.0	11.4	51.0	5.1	11.0	9.5	4.8	0.8	1.4	1.1	1.0
	PLW	0.2	0.0	0.0	0.8	0.2	0.7	0.2	12.2	6.7	32.8	25.9	2.2	9.2	2.6	1.6	2.6	1.8
	PRW	0.5	0.1	0.0	0.2	0.3	0.2	0.0	7.2	11.5	16.8	47.9	2.6	6.5	1.2	1.2	1.5	2.2
	PFC	0.3	0.9	0.3	0.7	1.0	0.2	0.1	4.1	4.4	1.9	2.1	72.1	9.4	1.1	0.3	0.4	0.6
	PFW	0.8	0.7	1.5	0.2	0.0	0.4	0.0	1.1	3.4	8.0	7.8	15.9	51.5	3.3	1.0	0.8	3.8
	PU	1.1	1.2	0.4	0.0	0.0	0.2	0.0	3.8	4.0	8.2	7.6	5.5	11.4	37.8	8.4	4.5	6.0
	PRP	1.8	1.6	0.2	0.0	0.4	0.0	0.0	5.7	6.3	6.3	15.8	1.4	6.8	22.3	11.7	6.6	13.3
	PLP	0.8	0.8	0.0	0.8	1.7	0.3	0.0	6.6	7.8	20.8	16.3	4.7	5.5	10.8	9.1	5.3	8.6
	PP	0.8	3.5	0.6	0.1	0.0	0.5	0.0	4.1	5.7	9.9	15.8	4.2	10.6	10.4	6.4	4.2	23.2

Table 15 – Confusion matrix category 2 (shots vs. passes) in percentages over 20 rounds[1]. In green the true positives.

		Prediction	
		Shot	Pass
Reference	Shot	76.8	23.2
	Pass	8.7	91.3

[1] Different 20 round than used in Appendix 1 and the results

Table 16 – Confusion matrix category 3 (7 throw classes) in percentages over 20 rounds[1]. In green the true positives. (JS = jumping shot, RS = running shot, SS = standing shot, UH = underhand, PL = pass left, PR = pass right, PF = pass forward)

		Prediction						
		JS	RS	SS	UH	PL	PR	PF
Reference	JS	61.6	8.8	2.5	3.2	4.1	7.3	12.5
	RS	9.8	56.5	5.2	1.8	6.0	8.1	12.6
	SS	9.7	18.7	40.3	2.6	8.5	10.9	9.3
	UH	3.3	3.8	0.7	34.8	13.2	19.5	24.7
	PL	0.3	1.1	1.4	3.7	47.6	32.5	13.3
	PR	0.8	0.7	0.7	2.8	22.3	58.7	14.0
	PF	1.3	1.0	0.8	2.6	9.1	11.5	73.6

Table 17 – Confusion matrix category 4 (wind up type) in percentages over 20 rounds[1]. In green the true positives.

		Prediction		
		Circular	Whip-like	Other
Reference	Circular	75.9	20.7	3.4
	Whip-like	22.3	71.3	6.4
	Other	18.4	35.5	46.1

Table 18 – Confusion matrix category 5 (intensity) in percentages over 20 rounds[1]. In green the true positives.

		Prediction	
		High	Low
Reference	High	82.5	17.5
	Low	13.7	86.3

[1] Different 20 round than used in Appendix 1 and the results