Towards an Artificial Training Expert System for Basketball

Yasin Acikmese[1], Baris Can Ustundag[2], Edin Golubovic[3]

[1] Department of Industrial Engineering, Galatasaray University, Istanbul, Turkey
yasin.acikmese@ogr.gsu.edu.tr
[2] Department of Computer Engineering, Istanbul Technical University, Istanbul, Turkey
ustundag16@itu.edu.tr
[3] Inovatink, Istanbul, Turkey
edin@inovatink.com

Abstract

Low power wearable devices equipped with powerful embedded processors found their place in applications such as personalized health, wellness/sports/fitness, rehabilitation, personal entertainment, social communications and lifestyle computing. In sports, the use of these devices together with powerful machine learning algorithms have opened up opportunity to create so called artificial training expert systems. In this work, an example of such system is discussed with the application in basketball training. The designed system is capable of recognizing basketball training type automatically by sampling data from battery powered wireless wearable device equipped with motion sensors. Support Vector Machine (SVM) is used as classification algorithm. The algorithm is capable of achieving 99.5% accuracy on the examined dataset.

1. Introduction

New generation of computation devices have emerged nearly every 10 years ever since the main frame computers in 70's [1]. The latest generation of computation devices, collectively named as IoT, are smaller, more energy efficient, more computationally capable, deployed in bigger numbers (ubiquitous) and cheaper than the previous generation - smart mobile devices. As a part of new generation of ubiquitous and connected computation devices, wearable devices, augment our capabilities as humans and bring benefits and amenity not available through any other technological means. Wearable devices are primarily used as monitoring devices nowadays, however, their full-benefit utilization is still to be explored through context-related data-intense application development. State of the art currently points out growing popularity of wearable devices application in the areas of personalized health, wellness/sports/fitness, rehabilitation, personal entertainment, social communications and lifestyle computing [2].

The increase of processing capacity and power efficiency and decrease in size of embedded processors have created opportunity of implementation of advanced real-time signal processing and machine learning algorithms in miniature battery powered wearable devices. Abilities of wearable devices to perform sensor input based detection, classification, regression and prediction in the context of their use have put them to the frontier of many different applications [2, 3].

For the purposes of this research the utilization of wearable devices and machine learning algorithms in sports training sessions are of particular interest. Sports training sessions are generally prepared, evaluated and monitored by coaches. Due to such intense involvement, coaches have a great influence on the performance qualities of trainees and, ultimately, on the competition results. Next to the sport specific knowledge the top-class coaches require the knowledge from areas like, anatomy, physiology, biomechanics, psychology, sociology, and didactics to make the best use of trainees' abilities [4]. Generally top-class coaches are hard to find and expensive to employ. It is therefore an imperative to create artificial training expert systems that are cheap, accessible and capable of emulating human coaches. Such endeavours require careful integration of advanced perception systems, modern computing paradigms and systems capable of reasoning and inference.

Research presented in this paper is an initial building block of an artificial trainer system to be applied in the basketball training. The system consists of a wearable device equipped with wireless transceiver and sensors for sensing of basketball trainee arm motion parameters (acceleration, angular rate and orientation). The arm motion data is transmitted from wearable device to a stationary hub. This hub receives, stores, prepares and processes the data to ultimately identify the current state of the progress of basketball training. Using described technology at the current state of the progress, the artificial trainer system is capable of correctly classifying 6 different types of basketball training.

The main focus of research presented in this paper is to correctly classify the type of basketball training using the arm motion data obtained through wearable device. The problem formulated in this way actually becomes typical problem of motion classification using accelerometer and gyroscope signals previously studied by researchers and experts. Leg motions are classified into 8 distinct movement classes by authors in [5]. To create feature set, Discrete Wavelet Transform of gyroscope signals is used. Using these features and Multi-Layer Feed-Forward Artificial Neural Network as classifier 97.7% accuracy is achieved. Authors assessed different classification algorithms in separate study of classification of sports and daily activities [6]. In this study Naive Bayesian, Artificial Neural Networks (ANN's), dissimilarity-based classifier, three types of decision trees, Gaussian mixture models and Support Vector

Work was conducted as a part of the Wearable Experts Collaboration Platform (WECooP). Authors wish to thank to Inovatink for supplying Bibi wearable sensor (http://bibi.inovatink.com).

Machines (SVMs) are considered. The highest correct differentiation rates are achieved with ANNs and SVMs, 99.2%.

Yang et.al. [7] presents classification of human activities using 3-axis accelerometer. Multi-Layer Feed-Forward Neural Network classifier is used for classification. The approach of classification of human movement type consists of two phases. In the first phase, the system recognizes presence of activity. In second stage, it classifies the type of activity. To choose the best features that represent activity types, Common PCA is used, and 95.24% accuracy is reached.

Recognition of walking and running can be considered as a most trivial movement recognition. However, more complex classification of different movements could improve exercise experience. Authors in [8] propose a system that automatically tracks different types of strength-training exercises, such as weight training and calisthenics.

Well tuned machine learning algorithms can deliver state-of-the-art results and classification accuracy above 95%. However, their performance is mostly depends on the quality of input data, features. Authors in [9] perform feature extraction from accelerometer signal data with a novel linear-time method. 3-axis accelerometer signals converted to a 2 dimensional data with Tilt Invariance Calculations, namely, Tilt and Gravity Compensated Signals (Horizontal and Vertical). Energy is used as a feature, and Hjorth Mobility & Complexity values are calculated to be used as features.

Research was done on the classification of activities during the basketball game as well. Bai et.al. [10] present tracking service in team-based sports that detects player's activities during a one-to-one basketball game.

The rest of the paper is organized as follows; next section introduces methods and procedures used to create the system. Section 3 presents the results. The results are discussed in Section 4 and paper is concluded in Section 5.

2. Methods and procedures

2.1. Overview of the system

The main aim of the proposed system is to facilitate simple and efficient collection and processing of relevant motion signals during the training session. The system consists of battery powered wearable device equipped with motion sensors and Bluetooth Connectivity, a stationary hub, able to receive, store, prepare and process data sent from wearable delivers the result on site and as third component, cloud system, is used for storage of training session data.

Wearable device is a compact and custom-built electronic device used for monitoring of specific movement activities during training. It is capable of recording acceleration using a 3-axis accelerometer, angular velocity using 3-axis gyroscope and orientation in the form of quaternions. The data recorded by this sensor is sent to the hub using Bluetooth Low Energy communication protocol. The wearable device, made by Inovatink, is based on Atmel's low power ARM Cortex M0 microcontroller, SAMD21. This MCU coordinates all functions of the wearable device. The accelerometer and gyroscope are integrated in Invensense MPU9250 motion sensor. The wearable device is shown in Figure 1. The device is intended to be worn on the arm as an armband. However, it might be placed anywhere on the body by using a different size strap. The processor reads signals from all three sensors at a rate of $100Hz$. Currently, no computation is done on-board in order to retain simplicity of the device and to increase the battery life.

The role of the hub is to communicate with the wearable device, store data and prepare it further processing (preprocessing). Additionally, the hub is responsible for running proposed classifier algorithm. Additionally, hub is connected to internet via Wi-Fi and can store all of the training data on cloud. In the future, the hub will be able to offload part of the processing to the cloud. Hub was implemented using Raspberry Pi 3 Model B. Overall system scenario is shown in Fig.2.

2.2. Preprocessing and feature extraction

The accuracy of basketball training type classification heavily depends on which features will be used during the classification process. For the purpose of this work statistical features from signal data were extracted to be used in classification algorithms. For the training purposes data from each sensor is stored in separate files. Each file consists of time, x-axis, y-axis and z-axis data for accelerometer and gyroscope signals. Orientation is calculated using sensors advanced features in the form of quaternions which are further converted to Euler angles using geometric transformation, then these angles are stored in a file together with timestamp. Thus, every trial creates three different files containing data from nine different signals. These data are filtered using *Moving Average Filter* to remove noisy signals and prepare them for further processing.

In order to classify the types of training features that are input to the classifier need to be obtained from data. For the purposes of this research, most known statistical features such as *Max, Min, Mean, Median, Standard Deviation, and Variance* are computed. Similarly, *Skewness, Kurtosis, Root Mean Square, Mean Absolute Deviation, Mean Crossing Rate, Percentiles (25th, 50th, 75th) and Hjorth Parameters* [9] are calculated.

The duration of the basketball training may be long and it is desired to classify the training type in real-time, due to frequent changing of training types, dribble and pass, dribble and shoot, etc. To achieve classification the features are calculated for specific time intervals. The windowing technique is performed on data with overlapping factor of 50%. The window duration is set to be 6 seconds which is enough to capture the type of training exercise. The sensor data rate is $100Hz$ therefore windows result in 600 data points for each sensor.

Statistical calculations for each 6-second window are added as a row to the dataset matrix (Fig. 3). The feature extraction process results in a set containing 153 features (9 signals x 17 statistical metrics) for each 6-second window (Fig. 3). Dataset is prepared by labeling each row according to training type (output). After feature extraction is performed for each 6-second window, rows are labeled regarding their training type. Table 1 contains the training labels.

Performing classification on total set of features with 153 elements is computationally expensive and may be deteriorat-

Figure 1. Wearable device used for data collection

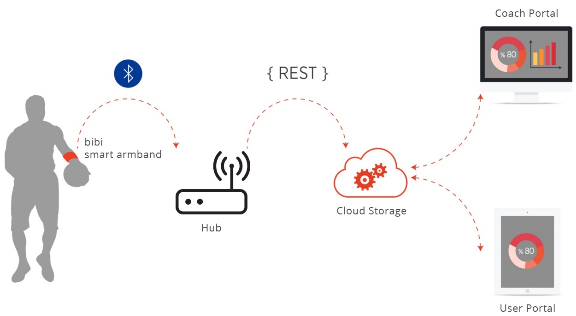

Figure 2. Overall system

Table 1. Exercise Type Labels

Label	Exercise Type
1	Forward-Backward Dribbling
2	Left-Right Dribbling
3	Regular Dribbling
4	Two Hands Dribbling
5	Shooting
6	Layup

ing for performance of the classifier. In such cases it is recommended to eliminate the features with the least contribution to the classification process. To reduce the number of features, it must be known which features are more valuable to classification process. This is evaluated by three different algorithms, Information Gain [11], Fisher Score [12], and T-Test. These techniques are widely used in importance ranking for datasets [11]. Top contributing features are selected using following procedure; Firstly, Information Gain, Fisher Score, and T-Test scores are calculated for each of 153 features; after calculation, 3 separate sets of top 30 ranking features are obtained. These resulting sets can be denoted as, I, F, T for Information Gain, Fisher Score, and T-Test respectively; the final feature set $\mathcal{F}$ is then obtained using following set relation:

$$\mathcal{F} = (I \cap F) \cup (I \cap T) \cup (F \cap T) \quad (1)$$

2.3. Classification and Verification

Classification is one of the most common tasks of machine learning, and represents the problem of classifying unknown instances into one of the known categories - classes. Classification of an object is based on finding similarities with predetermined objects belonging to different classes, with the similarity of two objects being determined by the analysis of their attributes (features). During classification, each object is grouped into one of the classes with a certain accuracy.

The classification process consists of two phases, in which the first stage builds a model based on the features of objects whose classes are known. Each instance of data takes only one class value. Classification algorithm learns on the basis of known classes. Therefore, based on the value of their attributes and class attributes, a set of rules can be established based on which classification will be made later. After learning stage, the model is tested i.e. its accuracy is evaluated, whereby the accuracy is percentage of instances that are correctly classified. The class attribute value of each test instance is compared to the class attribute value that is determined by the model.

In the proposed artificial training expert system the incoming window of data contains features that need to be classified into 6 different classes (training types), namely *Forward-Backward Dribbling, Left-Right Dribbling, Regular Dribbling, Two Hands Dribbling, Shooting* and *Layup*. In the first attempt, the intention is to predict the training type whether it's *dribbling or not*. The dataset is divided as dribbling and other. *Forward-Backward, Left-Right, Regular and Two Hands Dribbling* exercises labeled as *Dribbling*. *Shooting* and *Layup* exercises are labeled as *Not Dribbling*.

In order to perform classification SVM algorithm [13] is chosen, as an algorithm widely used in classification problems to predict exercise types [5–7]. As a starting point, the goal is set to classify the type of exercise into two discrete classes, *(Dribbling vs. Not Dribbling)*. This implementation of SVM algorithm employs the Linear Kernel [13];

$$\kappa(x, y) = x^T y + c \quad (2)$$

where x and y are vectors of features computed from training samples and $c \geq 0$ is a free parameter trading off the influence of higher-order versus lower-order terms in the polynomial.

The c parameter trades off misclassification of training examples against simplicity of the decision surface. A low c makes the decision surface smooth, while a high c aims at classifying all training examples correctly by giving the model freedom to select more samples as support vectors.

The work on training classification is expanded and goal is shifted to classify each of the involved exercises. The classification of data into 6 discrete classes is also performed using SVM. In this case SVM shows poorer performance with Linear Kernel, thus, Gaussian Radial Basis Function (RBF) Kernel is selected;

$$\kappa(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}} \quad (3)$$

where $\|x - y\|^2$ is Euclidean distance between the two feature vectors. The parameter defined as $\gamma = {}^1/_{2\sigma^2}$ is used to set *how far* the influence of a single training example reaches, with low values meaning *far* and high values meaning *close*.

As with the Linear Kernel, the *complexity* c parameter also exists in the implementation of RBF Kernel, in this case c sets the amount of allowable missclassification during the SVM optimization. For large values of c, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training examples classified correctly. Conversely, a very small value of c will cause the optimizer to look for a larger margin separating hyperplane, even if that hyperplane misclassifies more examples.

When dealing with problems in machine learning, it is often the case that models need to be developed based on the small set of data. The construction process of the classification model as well as its evaluation are then particularly difficult. The evaluation method using the test set in this case may be imprecise, especially if it relies on one possible non-characteristic partition of a set of learning examples. Random selection is one of the basic requirements when forming data partitions for learning and testing. However, the possibility that selected data do not represent a population pattern is increased when the total number of available data is reduced. Therefore, evaluation by using a test data can result in an imprecise error estimate, due to the specificity of the set of learning data or tests that are not the property of the population.

Multiple repetition of the evaluation process using different randomly selected training and testing sets, as well as the

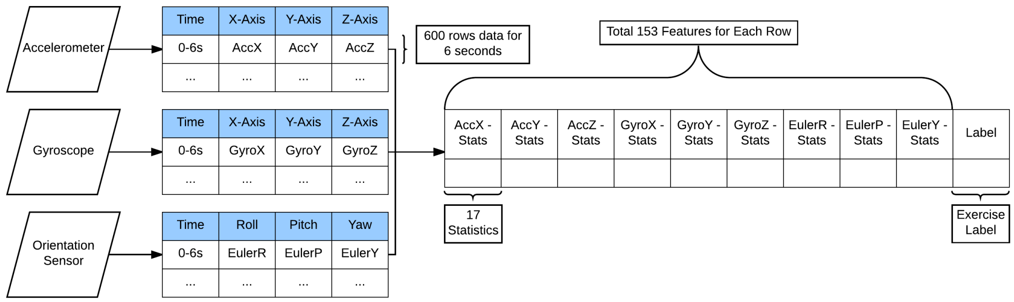

Figure 3. Data collection from each sensor and creating feature set according to 6 second windows.

averaging of the error estimation obtained, can overcome these anomalies. Cross-validation is based on this principle, with the corresponding substitution of the training set and the test set in each iteration [14]. In this work results are validated using 10-fold cross-validation method.

3. Results

3.1. Feature Selection

As a result of operation from eq.1, a set of 28 features is obtained as shown in Table 2. To get a better overview of how features ranked in the overall I, F, T sets the boxplot from Fig. 4 can be observed.

Table 2. Best 28 Features

Feature Names	
AccX - Max	AccX - Mean
AccX - Median	AccX - 50%
AccX - 75%	AccX - Hjorth
AccY - Mean	AccY - RMS
AccZ - Mean	GyroX - Max
GyroX - Std	GyroX - Var
GyroX - Skew	GyroX - 25%
GyroX - Hjorth	GyroX - RMS
GyroX - MAD	GyroY - Hjorth
OrientX - Std	OrientX - MAD
OrientX - MCR	OrientY - Mean
OrientY - Median	OrientY - Std
OrientY - Var	OrientY - 50%
OrientY - MAD	OrientY - MCR

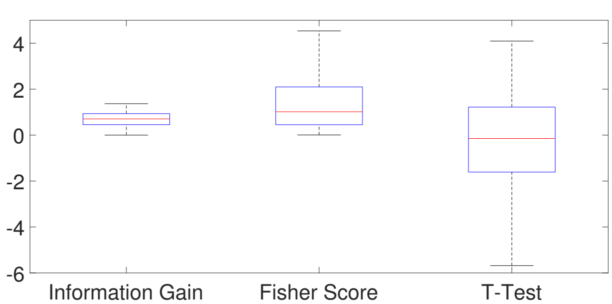

Figure 4. Boxplot of 3 feature ranking algorithms

3.2. Training Type Classification

Four metrics are used to evaluate the performance of training classification, which are Accuracy, Precision, Recall, and F1 Score [11]. Each recorded training consists of $\sim$30 seconds performed by 4 trainees with 2 or 3 repetitions. In evaluation, different individuals are not regarded because the classification of training type is desired to be independent from individuals. After extraction of statistical data from every 6 seconds of each training, the dataset matrix's size is 202 rows by 153 columns. Additional column for labels (training type) for each row is appended to the dataset finally resulting in 37 *Forward-Backward Dribbling*, 37 *Left-Right Dribbling*, 36 *Regular Dribbling*, 36 *Two Hands Dribbling*, 29 *Shooting*, 27 *Layup* instances. The classification is performed for both feature sets of 153 and 28 features. The metrics related to the classification is shown in Table 3. Separate confusion matrices are also shown in Table 4 for 153 features set and in Table 5 for *best* 28 features set ($\mathcal{F}$).

Table 3. Validation results for different feature sets

Metrics	**6 - Class (153 Features)**	**6 - Class (28 Features)**
Accuracy	98.51%	99.5%
Precision	0.9849	0.9940
Recall	0.9865	0.9955
F1 Score	0.9856	0.9947

Table 4. Confusion matrix for classification without feature ranking

	FBD	**LRD**	**RD**	**THD**	**S**	**L**
FBD	36	1	0	0	0	0
LRD	1	35	0	0	0	0
RD	0	0	36	0	0	0
THD	0	0	0	36	0	0
S	0	0	0	0	29	0
L	0	1	0	0	0	27

4. Discussion

The importance (*quality*) of elements in the feature set plays an essential role in machine learning methods. Having large numbers of features does not necessarily guarantee good algorithm performance. On the other hand, the lack of features may

Table 5. Confusion matrix for classification with feature ranking

	FBD	LRD	RD	THD	S	L
FBD	37	0	0	0	0	0
LRD	0	36	0	0	0	0
RD	0	0	36	0	0	0
THD	0	0	0	36	0	0
S	0	0	0	0	29	0
L	0	1	0	0	0	27

cause underfitting of data. Similarly, too many features may overfit it. There are some methods for overcoming this problem.

In this work, statistical features are calculated for the given data and a large set of 153 features is obtained. SVM model is trained using those features and shows satisfactory result with 98.5% accuracy. This result satisfies the state-of-the-art accuracy however it can be improved. At the same time having 153 features as the training input may create unnecessary computational load for rich dataset with many more training instances.

After reaching a satisfactory result, a performance improvement in the model is desired (fine tuning). With many features and less data to train, overfitting may occur. To overcome overfitting and improve computational performance, the feature set was reduced to 28 features from 153 features. Generally, at this stage Principal Component Analysis (PCA) is used for feature reduction. However, the reduced features set via PCA does not directly represent extracted features, i.e. the resulting feature set is not explicitly representable and real influence of individual features cannot be determined. Due to this reason, Information Gain, Fisher Score and T-Test feature ranking techniques are used to evaluate the contribution of each feature in classification.

Compared to 153 feature set, 28 feature set results show that classification with small number of feature set can be performed with better accuracy (99.5%). This feature set with 28 elements is not optimal feature set either. Further research needs to be done to find optimal feature set for this type of data and classification algorithm.

Gaussian Kernel for SVM algorithm is used due to its performance on this particular dataset. For the purpose of comparison other machine learning algorithms were tested. SVM with Linear Kernel, Random Forest and Logistics Regression resulted in accuracies of 95.5%, 97.5% and 97.5% respectively. Hence, SVM with Gaussian Kernel is identified as the best algorithm for this particular dataset.

5. Conclusion

This paper is initial step towards the development of an artificial training expert system for basketball. A dataset containing six types of basketball training was introduced. Data was collected using wearable device with accelerometer and gyroscope sensors on board. Collected data was preprocessed and labeled to use in training type classification process. Specifically, SVM with Gaussian Kernel model was adopted to classify the training type. Comprehensive experiments were performed to evaluate the stability of our approach, which are Accuracy, Precision, Recall, and F1 Score. To improve the performance of the model, feature reduction was done using Information Gain, Fisher Score and T-Test. SVM with Gaussian Kernel resulted in 99.5% Accuracy, 0.994 Precision, 0.9955 Recall, and 0.9947 F1 Score for classification of six types of basketball exercises.

Further research will be directed towards classification of other training exercises in basketball. With growing number of training types, more data will help to model and understand the differences between each training better. The process would be improved by decreasing the number of sensors or feature set. In addition, the entire process could be automatized and the system could give feedback to trainee.

6. References

[1] G. Bell, ”Bell’s law for the birth and death of computer classes: Atheory of the computer’s evolution”, *IEEE Solid-State Circuits Society Newsletter,*, vol.13, no.4, pp: 8-19, 2008.

[2] E. Sazanow, M.R. Neuman (Eds.), ”Wearable Sensors: Fundamentals, Implementation and Applications”, *Academic,* Cambridge, MA, USA, 2014.

[3] H. Banaee, M. Ahmed, A. Loutfi, ”Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges”, *Hindawi Sensors,*, vol.13, no.12, pp: 17472-17500, 2013.

[4] I. Fister, K. Ljubic, P. Suganthan, M. Perc, ”Computational intelligence in sports: challenges and opportunities within a new research domain”, *Applied Mathematics and Computation,*, vol.262, no.1, pp: 178-186, 2015.

[5] B. Ayrulu-Erdem, B. Barshan, ”Leg motion classification with artificial neural networks using wavelet-based features of gyroscope signals”, *Hindawi Sensors,*, vol.11, no.2, pp: 1721-1743, 2011.

[6] B. Barshan, M.C. Yuksek, ”Recognizing Daily and Sports Activities in Two Open Source Machine Learning Environments Using Body-Worn Sensor Units”, *The Computer Journal,*, vol.57, no.11, pp: 1649-1667, 2014.

[7] J.Y. Yang, J.S. Wang, Y.P. Chen, ”Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers”, *Pattern recognition letters,*, vol.29, no.16, pp: 2213-2220, 2008.

[8] D. Morris, T.S. Saponas, A. Guillory, I. Kelner, ”RecoFit: using a wearable sensor to find, recognize, and count repetitive exercises”, in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems,* Toronto, Canada, 2014, pp.3225-3234

[9] M. Khan, S. Ahmed, S.I. Rahman, R.O. Smith, ”A feature extraction method for realtime human activity recognition on cell phones”, in *3rd International Symposium on Quality of Life Technology,* Toronto, Canada, 2011, pp.

[10] L. Bai, C. Efstratiou, C.S. Ang, ”weSport: Utilising wrist-band sensing to detect player activities in basketball games”, in *IEEE international conference on Pervasive Computing and Communications,* Sydney, Australia, 2016, pp.1-6

[11] J. Han, J. Pei, M. Kamber. ”Data mining: concepts and techniques”, *Elsevier,* Waltham, MA, USA, 2011.

[12] R.O. Duda, P.E. Hart, D.G. Stork. ”Pattern classification”, *John Wiley & Sons,* NJ, USA, 2012.

[13] V. Vapnik, I. Guyon, T. Hastie ”Support vector machines”, *Machine Learning,*, vol.20, pp: 273-297, 1995.

[14] R. Kohavi, ”A study of cross-validation and bootstrap for accuracy estimation and model selection”, *Ijcai,*, vol.14, no.2, pp: 1137-1145, 1995.