A Feature-based Approach for Identifying Soccer Moves using an Accelerometer Sensor

Omar Alobaid and Lakshmish Ramaswamy
Department of Computer Science, University of Georgia, Athens, Georgia, U.S.A.

Keywords: Soccer, Activity Recognition, Accelerometer, Sensor.

Abstract: During the past decade, Human Activity Recognition (HAR) systems have been an evolving topic due to the popularity of smart devices. Recognizing soccer moves in real-time is an important research problem that has not yet been studied thoroughly in the literature. In contrast to daily physical activities, recognizing soccer moves poses a set of unique challenges, such as pattern irregularity and body positions when performing these moves. In this paper, our goal is to recognize soccer moves in real-time by utilizing accelerometer data. We explore three different feature-based algorithms: Time Series Forest, Fast Shapelets, and Bag-of-SFA-Symbols. We also examine different factors that can affect the performance of these algorithms, such as parameter tuning and accelerometer axis elimination. Additionally, we introduce a novel collaborative model consisting of the above-mentioned algorithms in a majority voting mechanism to further enhance the performance of the system. We also add a light-weight classifier to act as a tie breaker in case of disagreement between the classifiers. We experimentally choose the right parameters to reduce the training time drastically without forfeiting the level of accuracy. Our collaborative model outperforms the single model by 2% to reach 84% in accuracy with a decrease in the training time by one order of magnitude.

1 INTRODUCTION

In the past decade, there has been rapid development of Ubiquitous Computing involvement in our daily lives. Smart phones, smart watches, and smart clothes are examples of this technological explosion. These smart devices are usually equipped with sensors that can be utilized to serve numerous purposes. One of these sensors is the Accelerometer, which measures the change of velocity in m/s^2 in 3 dimensions. Because of its low power consumption, the accelerometer sensor is frequently used to achieve different tasks, such as device orientation and user's physical activity.

Accelerometer data can be viewed as a time series since it is a sequence of data points that are observed over time. Time Series analysis is a major subject of interest within the area of Data Mining. Time Series analysis tasks include forecasting (De Gooijer and Hyndman, 2006), querying (Hochheiser and Shneiderman, 2004), clustering (Liao, 2005), and classification (Xi et al., 2006). Time Series classification has been used by researchers in various domains, such as in medical (Kurbalija et al., 2014), biological (Tapinos, 2013), and geographical (Campbell and Diebold, 2005) sciences.

In recent years, there has been rising interest in recognizing human activity by using mobile sensors (Anguita et al., 2012) (Bayat et al., 2014) (Wan et al., 2015). Human Activity Recognition (HAR) systems can be delineated into three categories (Lara et al., 2013): external sensors, wearable sensors, and hybrid. Our focus in this paper is on wearable systems for the cost, size, and convenience of these systems compared to external sensor systems.

Soccer, as one of the most popular sports around the world, also offers a unique opportunity for real-time application software. There are many real-time applications that track human activity, such as Apple Health and Google Fit. However, none of these applications recognize soccer moves in real-time. Implementing an affordable system that recognizes and tracks basic soccer moves in real-time is desirable to develop players' soccer skills. As thousands of players practice soccer every day, having a real-time detection system would assist in tracking this multitude of training sessions. Our system can be extended to analyze players' performance. For example, coaches may instruct players to perform a number of shots in order to improve shooting skills. Instead of manually counting the number of shots, those players can use the proposed system to track number of shots taken in real-time.

Alobaid, O. and Ramaswamy, L.
A Feature-based Approach for Identifying Soccer Moves using an Accelerometer Sensor.
DOI: 10.5220/0008910400340044
In *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020) - Volume 5: HEALTHINF*, pages 34-44
ISBN: 978-989-758-398-8; ISSN: 2184-4305

Besides typical HAR system challenges, soccer poses other unique challenges. First, soccer moves have fast and irregular patterns, which make them harder to distinguish compared to other daily physical activities. Second, players might use either foot while playing, which makes the determination of sensor placement a crucial decision. Our research goal is to utilize accelerometer data to recognize basic soccer moves performed by a player in a training session in real-time without extra hardware.

In this paper, we applied feature-based algorithms which typically extract meaningful features from the data to be used in classification. The contributions of this paper include: first, evaluate three different feature-based algorithms: Time Series Forest, Fast Shapelets, and Bag-of-SFA-Symbols, which represent different approaches- interval, Shapelet, and Dictionary-based- to recognize soccer moves; second, analyze the performance of these algorithms in terms of accuracy and training time when the parameters are tuned; third, we propose a voting approach composed from the aforementioned algorithms to enhance accuracy. From this study, our results show that soccer moves can be recognized in real-time with an accuracy of 84%.

2 RELATED WORK

Wearable sensor systems can be divided into two types based on the learning approach: semi-supervised and supervised.

- Semi-supervised: the model uses labeled and unlabeled data in the training phase. This approach is attractive to researchers because having labeled data requires human resources to label it. Human labeling becomes an unpractical approach when the dataset is particularly large. However, obtaining unlabeled data is easier, and it eliminates labeling cost (Guan et al., 2007). The authors in (Radu et al., 2014) aimed to detect whether the user is indoor or outdoor by employing a semi-supervised approach as follows: use some of the labeled data to assign a label(s) to clustered unlabeled data; train a classifier on small label data, then tune the classifier based on the unlabeled data; and utilize collaborative learning, where classifiers can enhance their performance by mutual learning. In (Ghazvininejad et al., 2011), the authors used a small portion of the labeled data in a graph-based method. They calculated the association probability of each class using a k-nearest neighbor graph. Then, these probabilities were fed into the Hidden Markov Model to classify unseen examples.
- Supervised: the model uses labeled data only in the training. This approach is the most popular approach in HAR systems. In (Lee et al., 2017), walking and running were identified with an accuracy of 92% by training a Convolutional Neural Network. The researchers in (Kwapisz et al., 2011) recognized daily activities by training logistic regressions, decision trees, and multilayer neural network classifiers. Similarly, (Yazdansepas et al.,), the researchers utilized a Shapelet-based approach to recognize ambulatory activities in real time with a high accuracy compared to offline systems.

3 MOTIVATION & BACKGROUND

Nowadays, there are many commercial HAR systems that help users to track their physical activity in real-time, such as Fitbit, Apple, Garmin, and Android watches. However, these watches are usually limited to a number of general activities like running, walking, and swimming. Our goal in this research is to recognize a different and more intense type of sport in real-time. For this, we chose soccer, because it is the world's most popular sport (Dunning, 2013). With millions around the world playing this sport, an affordable system, that can track players' soccer moves during training sessions in real time, can help to improve players' performances. Furthermore, to the best of our knowledge, identifying soccer actions in real time using the accelerometer sensor has not yet been discussed in the literature. It is worth mentioning that soccer moves are harder to recognize compared to daily activities, because soccer moves are fast and irregular, compared to activities like walking and running. A related point to consider is that players have different techniques to perform these moves. Building a player-independent platform to recognize these moves is a significant feature of our system.

Our research objective is to build a player-independent platform to identify soccer moves in real-time utilizing accelerometer data. One of the possible approaches for real-time classification is to apply time series classification algorithms. We apply lightweight feature-based algorithms to classify streaming data on-the-fly with minimal overhead on resource-constrained mobile devices. Though machine learning algorithms are most popular in HAR studies, most of these algorithms require high

computational resources to extract and determine the most important features. As such, feature-based algorithms are more efficient when paired with accelerometer data on mobile devices.
In our research, we focused on the following moves: **shooting the ball**, **passing the ball**, **heading the ball**, **running**, and **dribbling**. We chose these moves because these are the basic moves in soccer.

Time series classification is a classic problem in the Time Series analysis domain. Classifying unknown instances is a desired goal in real-life applications. There are two main approaches to classify time series (Baydogan et al., 2013): instance-based and feature-based. In this section we will discuss the instance-based approach. The feature-based approach will be discussed in the next section.

3.1 Instance-based

This approach uses a similarity measure between the new instance and the training instances. Euclidean Distance and Dynamic Time Warping are examples of this approach. The instance-based approach is characterized by its simplicity. On the other hand, it only works well with short time series and does not generalize well to long and noisy time series (Schäfer, 2015).

3.1.1 Euclidean Distance

This method measures the straight distance between two data points in the Euclidean space.

In 3-dimensional space, calculating the Euclidean Distance between two points from the accelerometer sensor is shown in Equation 1:

$$ED(p,q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2 + (p_z - q_z)^2} \tag{1}$$

Euclidean Distance is widely used due to its efficiency and simplicity. However, it does not perform well in most cases due its sensitivity to distortion (Ratanamahatana and Keogh, 2004) and intolerance to time shifting.

3.1.2 Dynamic Time Warping (DTW)

Dynamic Time Warping (Ratanamahatana and Keogh, 2004) performs non-linear matching between two series by reducing the distance between the time series. This approach was proposed to overcome Euclidean Distance's weakness to distortion and time shifting. Dynamic Time Warping showed superb performance in many applications. Nonetheless, the main disadvantage of Dynamic Time Warping is its time complexity $O(n^2)$ where n is the length of the time series. There are variations of DTW, such as Weighted DTW (Jeong et al., 2011) and Time warp edit distance (TWED) (Marteau, 2009). Weighted DTW assigns less weight to points that have the largest difference between a training point and a testing point. This step aims to reduce the outliers effect on the classification. On the other hand, Time warp edit distance (TWED) introduces a parameter called *stiffness* which controls the flexibility of TWED. The *Stiffness* parameter compromises between *infinite stiffness* in Euclidean Distance and *zero stiffness* in Dynamic Time Warping (DTW).

4 FEATURE-BASED SOCCER MOVES IDENTIFICATION

The feature-based approach generates features from time series to be compared instead of similarity measures in the instance-based method. Feature-based is usually faster than instance-based when it uses fast feature extraction and classification algorithms (Baydogan et al., 2013). Feature-based can be divided into three types: Interval, Shapelets, and Dictionary. The following taxonomy and examples are adapted from (Bagnall et al., 2017).

4.0.1 Time Series Forest (TSF)

TSF (Deng et al., 2013) is a tree-ensemble that deploys Random Forest sampling approach to reduce the feature space for intervals. TSF samples $\sqrt{m}$ intervals where $m = TS.length$. Using this approach reduces the feature space drastically from $O(m^2)$ to $O(m)$. For each interval, *mean*, *standard_deviation*, and *slope* are calculated. These features are used to distinguish between different soccer moves. For example, *mean* can be used to distinguish between heading and passing the ball, since *mean* value of the vertical axis in heading will be higher than passing (i.e. most players jump to head the ball which results in higher values on the vertical axis). TSF uses Entropy and Distance to spot best splits in the trees if the node exceeds a threshold. The goal of the Entropy is to determine the most discriminated/ disparate nodes which can distinctly separate the classes. In many cases, there will be more than one possible split. To break this tie, the distance will be measured between the candidate threshold and the nearest feature value.

One of the main advantages of TSF is the ability to train trees independently, which allows for parallel training. To classify an unseen example, TSF assigns a label based on a majority voting approach.

4.0.2 Fast Shapelets

Fast Shapelet (Rakthanmanon and Keogh, 2013) is a heuristic algorithm. It converts the time series' real values into an alphabetical representation. The purpose of the conversion is to reduce the search cost, since the values range is limited. Another major purpose of the conversion is to hash the data to increase the search accuracy by utilizing collision history. In soccer, different moves have different signal amplitudes which will result in different SAX representations. For instance, passing, and shooting have roughly similar signal patterns, but in different amplitudes. Utilizing the SAX representation in this case helps to differentiate between the two classes. In addition, the SAX representation helps reduce the variety of move patterns from each player.
Fast Shapelet uses a sliding window technique to convert the data to a SAX representation. Because the window size has a length less than the time series segment, there will be multiple SAX words that refer to the same time series segment. However, this will lead to false dismissals where two nearly identical segments could create two different SAX words. Random Projection was proposed by the authors to solve this issue. For example, if there are words on a length of 5, then Random Projection masks 2 positions to produce words on a length of 3. This approach will increase the chance that two similar SAX words are mapped to the same masked word. After r iterations, distinguishing power will be calculated to differentiate between words that strongly represent the same class while hardly appearing in any other class.

4.0.3 Bag-of-SFA-Symbols

Bag-of-SFA-Symbols (BOSS) (Schäfer, 2015) is a bag-of-words model that uses a structural-based approach to extract representative features. BOSS converts the raw time series into various substructures by applying Symbolic Fourier Approximation (SFA). SFA utilizes Fourier Transform and Multiple Coefficient Binning to achieve approximation and quantization. SFA reduces the noise of these substructures by applying low pass filtering to facilitate the work of string matching algorithms.

BOSS applies a sliding window. Each window is normalized for amplitude invariance, and the mean normalization is enabled depending on the time series nature. In soccer, the mean is an important feature to separate between different moves. For instance, body movement in shooting is more intense than passing, which will increase the mean sensor reading values. In Section 6, we showed the importance of the mean on classification accuracy. For each window, SFA is applied. Because adjacent windows have a high chance of being identical windows, numerosity reduction is applied to avoid over-counting a substructure. Finally, BOSS creates a histogram of the SFA words. A modified version of Euclidean Distance is used to measure the distance between words. To classify any new unseen instance, BOSS deploys a 1-NN search algorithm to find the nearest hit.

5 ARCHITECTURE

In this paper, we conduct comprehensive experiments to recognize soccer movements and compare the performance between 3 different feature-based algorithms: Time series forest (TSF), Fast Shapelets (FS) from, and Bag-of-SFA-Symbols (BOSS). The main objective is to find the most accurate, yet efficient algorithm to achieve the recognition tasks. After we examine different algorithms, we propose a novel collaborative model composed from the above-mentioned algorithms. The following subsections explain the workflow of our system.

5.1 Data Collection

The data was collected from 16 players between 18 and 35 years old. Each player performed different soccer actions: shooting the ball, passing the ball, heading the ball, running, and dribbling. To collect the data, a Samsung Galaxy S5 was used to record the accelerometer data using the Sensor Kinetics Application (INNOVENTIONS®, 2017) at a 100 Hz sampling rate. Every player wore a belt to hold the phone in the abdomen area. The central abdomen area was chosen intentionally to help the system recognize moves from players regardless of their dominant foot.

5.2 Signal Denoising

The accelerometer sensor is sensitive to even the weakest movement, which leads to unwanted data in recognition tasks. To overcome this issue, signal denoising/smoothing is applied. There are many signal denoising methods (Lorenz,), such as Wavelt and

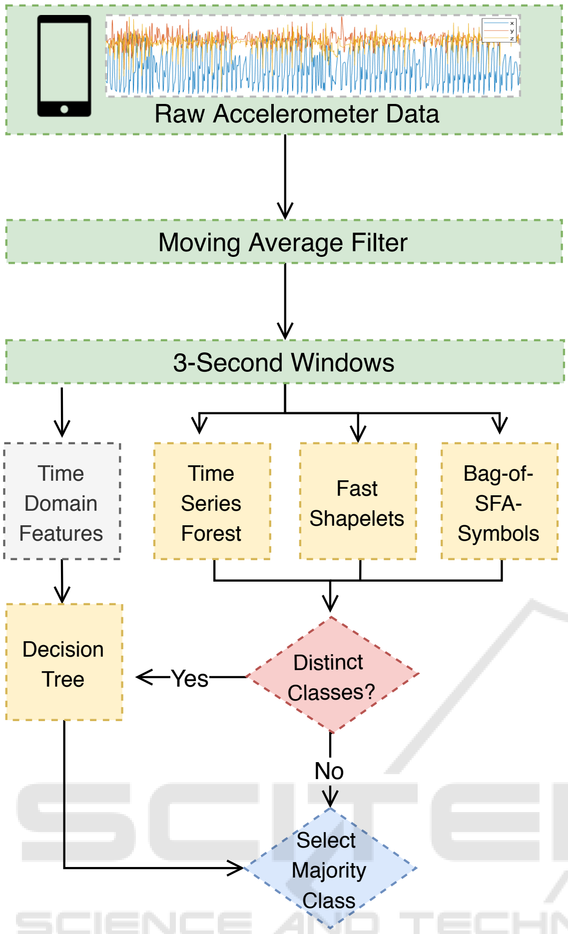

Figure 1: Architecture of the the proposed system.

Linear Fourier. Because our system is designed to run on portable devices with limited computational resources, we chose Moving Average Filter method (Lorenz,) as our smoothing method.

5.3 Signal Segmentation

Soccer practice sessions typically last between several minutes to one hour or more. As a result, dividing the incoming data into segments/windows is necessary to extract the important features. We conducted several preliminary experiments to find the best window size for this purpose. We found a 3-second window is the optimal size for our system to capture the characteristics of each soccer move without negatively affecting the accuracy.

5.4 Classification

Our goal is to examine 3 feature-based algorithms that use different approaches. We explored Time series forest (TSF) from the Intervals family, Fast Shapelets (FS) from the Shapelets family, and Bag-of-SFA-Symbols (BOSS) from the Dictionary family. The resulting segments from the signal segmentation step were then used to train the 3 classifiers. After tuning each algorithm to find the best parameters, we proposed a novel approach to combine the aforementioned algorithms in voting mechanism to improve the accuracy and reduce the training time. The main goals of this step are to enhance the classification accuracy and reduce the over-fitting probability. Our collaborative model is independent, which allows for parallel training. We also added a light-weight classifier, which will act as a tie breaker (e.g. when each classifier produces a distinct class). Decision Tree was chosen as the tie breaker classifier due to its speed and efficiency. To train the fourth classifier, we extracted 10 time domain features from the x and y axes.

6 EMPIRICAL EVALUATION

In all experiments, we used the open-source code from the Time Series Classification repository (Bagnall et al., 2017).

All of the experiments were done using a single machine with a dual-core processor (4 logical processors) and 8 Gigabytes memory. To evaluate the classification performance, we used 10-fold cross validation.

6.1 Time Series Forest (TSF)

In our preliminary experiment, we used 500 trees and 30 candidates. The accuracy of our system was is 81%. Figure 2 shows accuracy per activity. It can be clearly seen that shooting and passing have the highest accuracy compared to the rest. On the other hand, heading has the lowest accuracy. We believe the reason for these results is the nature of these activities. Passing and shooting have distinct natures. When a player passes the ball, the power applied on the ball is considerably less than shooting the ball, which makes body movements in passing less intense than shooting. Table 1 illustrates the confusion matrix.

Table 1: Confusion matrix of TSF.

	Running	Passing	Heading	Shooting	Dribbling
Running	**0.80**	0.04	0.06	0.00	0.10
Passing	0.00	**0.88**	0.02	0.10	0.00
Heading	0.08	0.17	**0.70**	0.06	0.00
Shooting	0.00	0.08	0.02	**0.91**	0.00
Dribbling	0.16	0.00	0.09	0.02	**0.73**

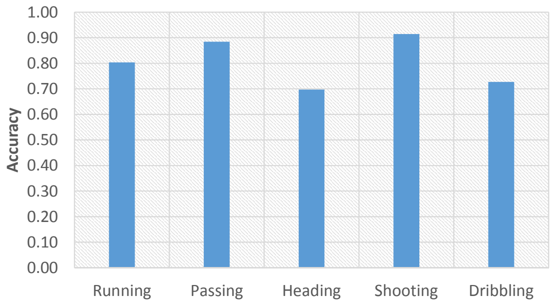

Figure 2: TSF Accuracy per activity.

6.1.1 Parameters Effect

TSF has two parameters to be entered by the user: number of trees and number of candidate thresholds. In this experiment, we attempted to find the best parameters combination to achieve best accuracy while lowering training time using a greedy approach (Frank Hutter,)(Matuszyk et al.,). We started with a varying number of trees while keeping the number of candidates fixed to 30. The forests' sizes were $50, 100, 250, 500(Default), 1000$. The results show that the execution time increases as the number of trees increases, while the improvement in accuracy is less than 1% which is insignificant. Figure 3 shows the detailed performance.

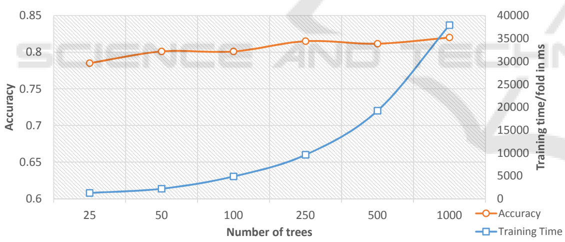

Figure 3: TSF performance with different number of trees.

After testing the impact of the number of trees, we evaluated the effect of the number of candidate thresholds by fixing the number of trees to 100 and 250, as these numbers showed the highest accuracy while carrying less training time. The tested number of candidate thresholds were $1, 3, 5, 10, 20(Default), 30, 50$. The results in Figure 4 show that increasing the number of candidates from 1 to 30 increases the accuracy by ∼4%, from 77% to 81% in the case of 100 trees, with a minimal rise in the execution time by ∼500 ms. However, there was is no difference in accuracy when the forest size is 250. Figure 4 shows the performance of our experiments.

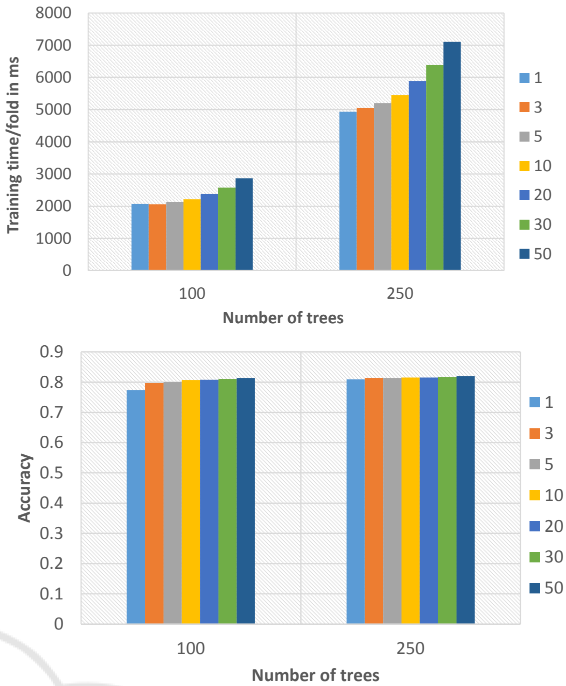

Figure 4: TSF performance with different number of candidate thresholds.

6.1.2 Accelerometer Axis Elimination Effect

One of our primary goals in this paper is to increase the efficiency of our system while maintaining the same level of accuracy. For this step, we tested removing one axis at a time from the accelerometer data in the forests with 100 and 250 trees while candidate thresholds were set to 30. In our first trial, the forest size at 100 has an accuracy of 81% with all axes. After eliminating the z axis, the accuracy decreases by only 1%, while the training time drops by 45% from ∼4800 to ∼2700 ms. When we remove the x and y axes, the accuracy sharply decreases to 75% and 73%, respectively.
In our second trial, the forest size was 250. Removing the z axis does not affect the accuracy, while the training time reduces by 23%, which is a significant improvement toward an efficient recognition system. However, using y, z and x, z axes affects the performance negatively by 4% and 5%, respectively. Figure 5 shows the performance of our experiments.

6.1.3 Parallel Training Effect

Constructing and training a large forest sequentially is not the optimal approach. Therefore, in this experiment, we converted the TSF implementation into a parallel approach in order to utilize the computation resources. When forest size is small (i.e. 50 trees), the

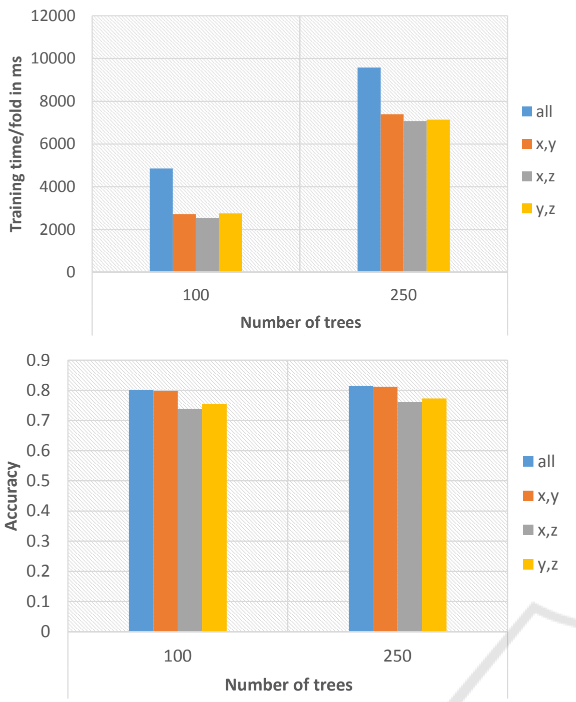

Figure 5: TSF Accuracy and training time based on the used axes.

training time is similar between the two approaches. An improvement of 60% is achieved when the forest size is larger than 250 trees. Figure 6 shows the performance of our experiments.

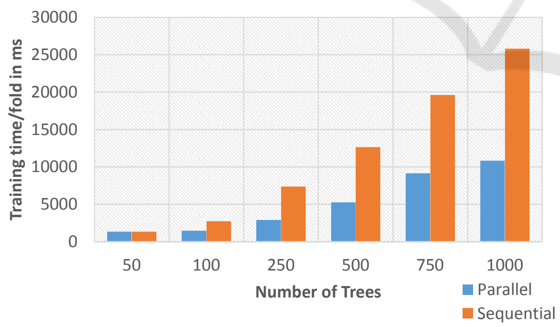

Figure 6: TSF performance with parallel training.

6.2 Fast Shapelets

In our exploratory experiments, we used the default parameters as shown in the next paragraph. The accuracy is 74%, but the training time is substantially longer (i.e. 110 mins/fold) compared to TSF. It is worth mentioning that our data is relatively small compared to other datasets, meaning that training time will increase dramatically with larger datasets. The reason behind this slowness is FS does an intensive search to construct Shapelet from all possible lengths. In the next section, we discuss the effect of reducing the possible Shapelet lengths. Figure 7 shows accuracy per activity, and Table 2 shows the confusion matrix.

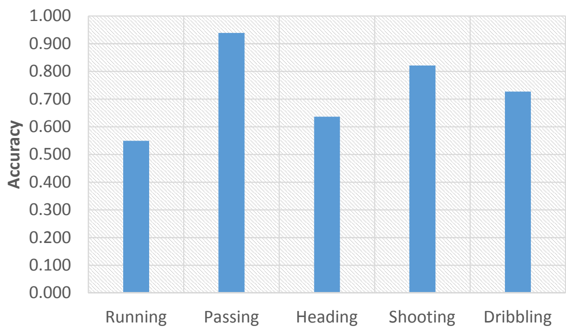

Figure 7: FS Accuracy per activity.

Table 2: Confusion matrix of FS.

	Running	Passing	Heading	Shooting	Dribbling
Running	**0.55**	0.00	0.10	0.00	0.35
Passing	0.00	**0.94**	0.03	0.02	0.02
Heading	0.06	0.09	**0.64**	0.21	0.00
Shooting	0.02	0.04	0.10	**0.82**	0.02
Dribbling	0.25	0.00	0.00	0.02	**0.73**

6.2.1 Parameters Effect

Fast Shapelet has many default parameters that can be modified: $r = 10$, which is the number of iterations to perform random masking, $top\_k = 10$, which is the top k^{th} subsequences that have the highest score, $min\_shaplet\_len = 10$, which is the minimum length of any Shapelet, and $step = 1$, which is the increment in the Shapelet discovery search.
Our initial experiments showed that $step$ parameter is the most influential factor on training time. As a result, we started by testing 10 various values for $step$ in the interval of $[1,500]$ while the other parameters were left unchanged. Our experiments showed that the accuracy is not drastically affected by increasing the $step$ in most cases, while the improvement in the training time is exponential. For example, when we increased the $step$ from 1 to 5, the accuracy was lowered by only 1%, while the training time was reduced by $\sim$80%. When $step = 30$ or 50, the accuracy is 74% and 75%, respectively, while the time is reduced by one order of magnitude. Figure 8 shows the impact of step size on the accuracy and training time.

Next, we set the value of $step$ to 50, since this value showed the best performance. We then tested 6 values for top_k which were $\{1, 3, 5, 10(Default), 20, 30\}$. Our results show that the default value, 10, is the highest value in

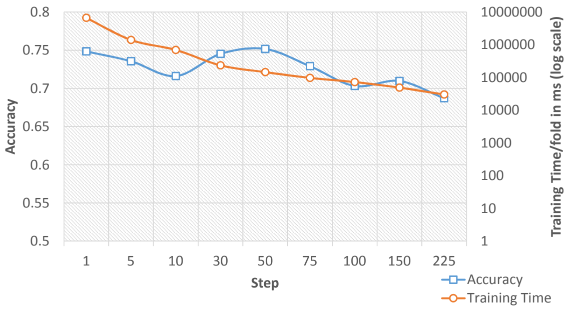

Figure 8: FS Accuracy and training time based on step size.

accuracy. Increasing the values to 20 or 30 affect the accuracy and training time negatively in both *step* values. On the other hand, reducing the value of *top_k* to 3 reduced the training time by 50% with only a 2% accuracy loss, which is suitable for limited computational resources. Figure 9 illustrates our results in regards to the *top_k* values.

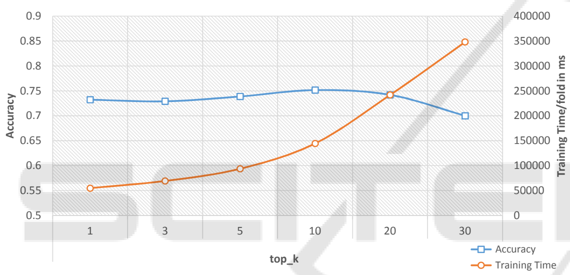

Figure 9: FS Accuracy based on step size (50) and *top_k* values.

Random Projection iteration r is another parameter that we can modify. We tested 3 values for r, which were $\{5, 10(Default), 20\}$. Our results show that r does not affect the accuracy, but when r increased, the training time also increases by 10%. Figure 10 illustrates the results of this experiment.

6.2.2 Accelerometer Axis Elimination Effect

In this section, we examined the impact of an axis elimination while fixing the parameters as follows: $step = 50$, $r = 5$, and $top\_k = 3, 5, 10$. When only x, y is used, the accuracy drops by 3%, from 73% to 70%, with a 66% reduction in the training time. In the case of using x, z and y, z, the accuracy drops by 10% and 30%, respectively. These results confirmed what we found in the TSF experiment about the importance of the x axis in the recognition task. Using only two axes, x, y, can lead to an acceptable accuracy in order to shorten the training time. Figure 11 shows the results of using different *top_k* values.

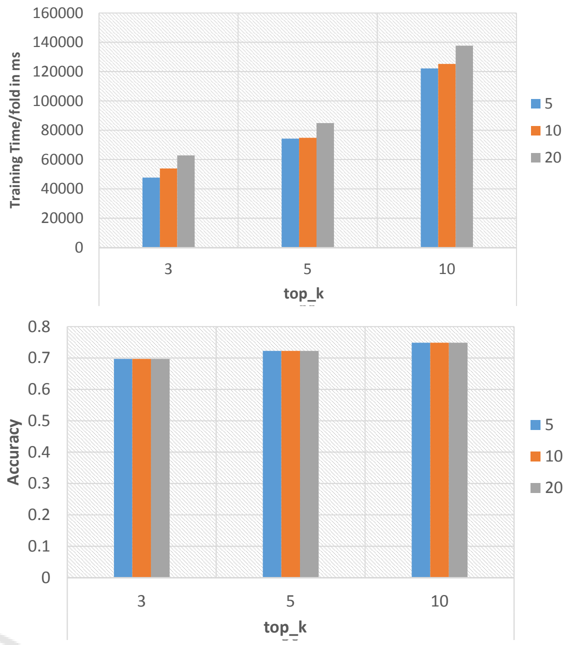

Figure 10: FS Accuracy and training time based on step size (50) and *top_k* values (3,5,10) with different r values.

6.3 Bag-of-SFA-Symbols

In our first BOSS experiment, we used the default parameters to train our model. The accuracy is 82%, which is similar to TSF, however, the training time is considerably longer than TSF (i.e. 52 mins/fold). Figure 12 shows accuracy per activity, and Table 3 shows the confusion matrix.

Table 3: Confusion matrix of BOSS.

	Running	Passing	Heading	Shooting	Dribbling
Running	**0.87**	0.00	0.00	0.04	0.09
Passing	0.00	**0.89**	0.03	0.08	0.00
Heading	0.00	0.17	**0.82**	0.02	0.00
Shooting	0.00	0.12	0.08	**0.79**	0.01
Dribbling	0.14	0.00	0.07	0.00	**0.80**

6.3.1 Parameters Effect

BOSS has few parameters that can be modified. $alpha\_size = 4$ refers to the number of letters in the string representation. *mean_norm* is a boolean variable to determine whether to perform mean normalization or not. *window_length* is a dynamic variable that can be affected by 3 different parameters: $min\_window\_length = 10$, $max\_window\_length = TS.length$, and $step = 1$, which is the increase in the window's size. The default parameters will conduct SFA transforming to all possible window sizes $1, 2, 3, ..., TS.length$, which is an expensive computation. In this experiment, we used different *step* values

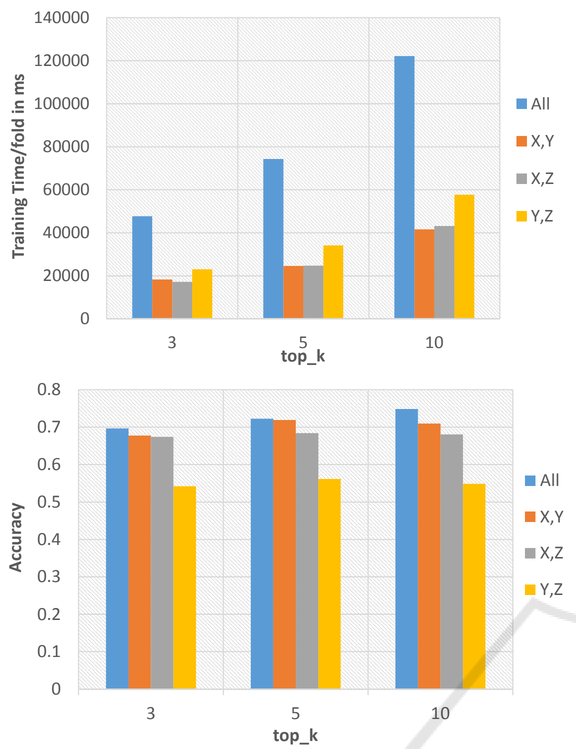

Figure 11: FS Accuracy and training time based on the used axes.

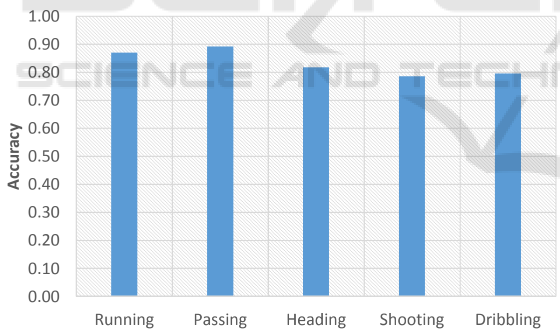

Figure 12: BOSS Accuracy per activity.

to measure the classification performance in terms of accuracy and training time. Our results show that increasing the *step* value to 15 does not affect the accuracy by more than 2%, however it reduces the training time by one order of magnitude (i.e. from 52 mins to 4 mins/fold). From Figure 13, we can conclude that any $step \leqslant 25$ can achieve an acceptable accuracy, while significantly reducing the training time.

The next experiment examined the effect of $alpha\_size$. We tested the following values $3, 4(Default), 6, 8, 10$ when $step = 5$ and $step = 15$. Our experiments show that 4 is the best alphabet size and increasing the alphabet size leads to a sharp de-

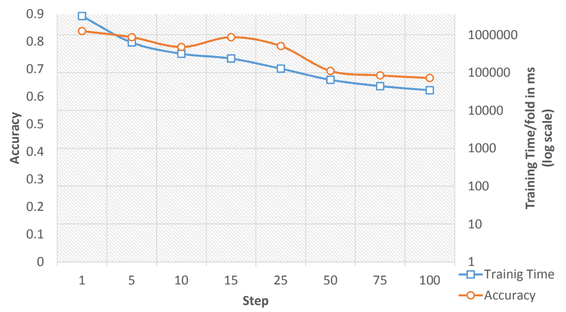

Figure 13: BOSS training time and accuracy based on step size.

crease in accuracy. When we increase the $alpha\_size$ from 4 to 6, the accuracy drops by 7% when $step = 5$, and by 11% when $step = 15$. These results are consistent with the findings in (Lin et al., 2012) (Schäfer, 2015), which suggest 4 is an appropriate alphabet size.

Our last experiment in parameter tuning was the mean normalization effect. For every SFA word, the first Fourier coefficient was not included. Our results show that the normalization crucially lowered the accuracy by more than 11%. Our interpretation for this result is that soccer movements are fast and have high amplitude, and the normalization reduces the signal amplitude which leads to confusion in the classification process. Figure 14 summarizes the results of the last two experiments.

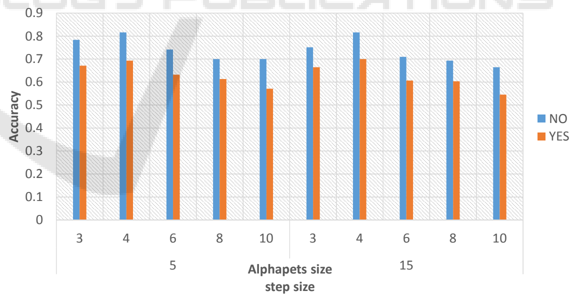

Figure 14: BOSS Accuracy based on the alphabet size (3, 4, 6, 8, 10) and mean normalization (yes,no).

6.3.2 Accelerometer Axis Elimination Effect

In this experiment, we tested the effect of eliminating one of the three axes. We used the following attributes: $alpha\_size = 4$, $mean_norm = false$, $step = 15$, $min\_window\_length = 10$, and $max\_window\_length = TS.length$. Our results show that eliminating the z axis causes a decrease of 3%, from 82% to 79%, while reducing the training time by more than 50%. Eliminating the y and x axes reduces the accuracy by 8% and 11%, respectively. These results align with the outcomes of TSF and FS experiments, which reveals the importance of x, y axes in the classification process, compared with the z axis, which can be omitted without crucially affecting the accuracy. Figure 15 shows the accuracy and training time of the axis elimination experiment.

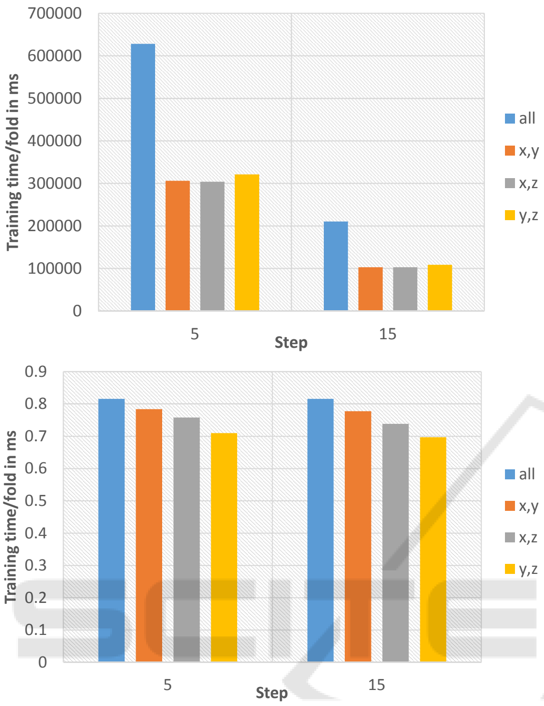

Figure 15: BOSS accuracy and training time based on the used axes.

6.4 Collaborative Feature-based Approach

Our collaborative model achieves better accuracy by 2% with less training overhead, because we selected the best possible parameters for these models. Figure 16 shows the accuracy of the collaborative model compared to TSF, FS, and BOSS classifiers.

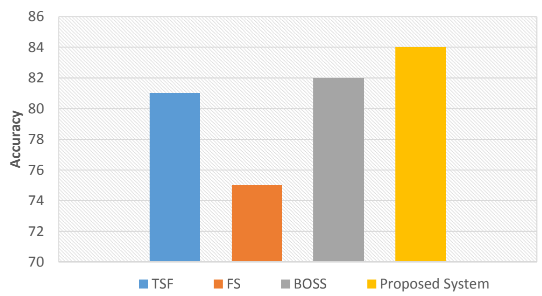

Figure 16: Collaborative Approach compared to single models.

7 CONCLUSION

In this paper, we aimed to recognize soccer moves in real-time. We comprehensively examined three different feature-based approaches, which are Time Series Forest (interval-based), Fast Shapelets (Shapelet-based), and Bag-of-SFA-Symbols (Dictionary-based). We studied different factors that might affect the accuracy and the training time, such as parameters tuning and axis elimination. We tuned our model to reduce the training time by one order of magnitude, in the case of Fast Shapelets and Bag-of-SFA-Symbols, without sacrificing the accuracy. We then proposed a collaborative model where we combined all three approaches in a voting mechanism using only two axes, which led to an increase in accuracy by 2% to reach 84%.

REFERENCES

Anguita, D., Ghio, A., Oneto, L., Parra, X., and Reyes-Ortiz, J. L. (2012). Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine. In *International workshop on ambient assisted living*, pages 216–223. Springer.

Bagnall, A., Lines, J., Bostrom, A., Large, J., and Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660.

Bayat, A., Pomplun, M., and Tran, D. A. (2014). A study on human activity recognition using accelerometer data from smartphones. *Procedia Computer Science*, 34:450–457.

Baydogan, M. G., Runger, G., and Tuv, E. (2013). A bag-of-features framework to classify time series. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2796–2802.

Campbell, S. D. and Diebold, F. X. (2005). Weather forecasting for weather derivatives. *Journal of the American Statistical Association*, 100(469):6–16.

De Gooijer, J. G. and Hyndman, R. J. (2006). 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473.

Deng, H., Runger, G., Tuv, E., and Vladimir, M. (2013). A time series forest for classification and feature extraction. *Information Sciences*, 239:142–153.

Dunning, E. (2013). *Sport matters: Sociological studies of sport, violence and civilisation*. Routledge.

Frank Hutter, M. L. Algorithm configuration: Hands-on tutorial. http://www.ml4aad.org/wp-content/uploads/2016/02/AC-Tutorial.pdf.

Ghazvininejad, M., Rabiee, H. R., Pourdamghani, N., and Khanipour, P. (2011). Hmm based semi-supervised learning for activity recognition. In *Proceedings of the 2011 international workshop on Situation activity & goal awareness*, pages 95–100. ACM.

Guan, D., Yuan, W., Lee, Y.-K., Gavrilov, A., and Lee, S. (2007). In *Embedded and Real-Time Computing Systems and Applications, 2007. RTCSA 2007. 13th IEEE International Conference on*, pages 469–475. IEEE.

Hochheiser, H. and Shneiderman, B. (2004). Dynamic query tools for time series data sets: timebox widgets for interactive exploration. *Information Visualization*, 3(1):1–18.

INNOVENTIONS®, I. ((accessed February 3, 2017)). *Sensor Kinetics Pro*.

Jeong, Y.-S., Jeong, M. K., and Omitaomu, O. A. (2011). Weighted dynamic time warping for time series classification. *Pattern Recognition*, 44(9):2231–2240.

Kurbalija, V., Radovanović, M., Ivanović, M., Schmidt, D., von Trzebiatowski, G. L., Burkhard, H.-D., and Hinrichs, C. (2014). Time-series analysis in the medical domain: A study of tacrolimus administration and influence on kidney graft function. *Computers in biology and medicine*, 50:19–31.

Kwapisz, J. R., Weiss, G. M., and Moore, S. A. (2011). Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82.

Lara, O. D., Labrador, M. A., et al. (2013). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys and Tutorials*, 15(3):1192–1209.

Lee, S.-M., Yoon, S. M., and Cho, H. (2017). Human activity recognition from accelerometer data using convolutional neural network. In *Big Data and Smart Computing (BigComp), 2017 IEEE International Conference on*, pages 131–134. IEEE.

Liao, T. W. (2005). Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874.

Lin, J., Khade, R., and Li, Y. (2012). Rotation-invariant similarity in time series using bag-of-patterns representation. *Journal of Intelligent Information Systems*, 39(2):287–315.

Lorenz, D. A comparison of denoising methods for one dimensional time series.

Marteau, P.-F. (2009). Time warp edit distance with stiffness adjustment for time series matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):306–318.

Matuszyk, P., Castillo, R. T., Kottke, D., and Spiliopoulou, M. A comparative study on hyperparameter optimization for recommender systems.

Radu, V., Katsikouli, P., Sarkar, R., and Marina, M. K. (2014). A semi-supervised learning approach for robust indoor-outdoor detection with smartphones. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 280–294. ACM.

Rakthanmanon, T. and Keogh, E. (2013). Fast shapelets: A scalable algorithm for discovering time series shapelets. In *proceedings of the 2013 SIAM International Conference on Data Mining*, pages 668–676. SIAM.

Ratanamahatana, C. A. and Keogh, E. (2004). Everything you know about dynamic time warping is wrong. Citeseer.

Schäfer, P. (2015). The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery*, 29(6):1505–1530.

Tapinos, A. (2013). *Time Series Data Mining In Systems Biology*. PhD thesis, The University of Manchester (United Kingdom).

Wan, J., O'grady, M. J., and O'hare, G. M. (2015). Dynamic sensor event segmentation for real-time activity recognition in a smart home context. *Personal and Ubiquitous Computing*, 19(2):287–301.

Xi, X., Keogh, E., Shelton, C., Wei, L., and Ratanamahatana, C. A. (2006). Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 1033–1040. ACM.

Yazdansepas, D., Saroha, N., Ramaswamy, L., and Rasheed, K. Towards efficient & real-time human activity recognition using wearable sensors: A shapelet-based pattern matching approach.