



# Improving a simulated soccer team's performance through a Memory-Based Collaborative Filtering approach



Pedro Henriques Abreu<sup>a,b,\*</sup>, Daniel Castro Silva<sup>c,d</sup>, Fernando Almeida<sup>e,f</sup>, João Mendes-Moreira<sup>c,g</sup>

<sup>a</sup> Department of Informatics Engineering, Faculty of Sciences and Technology, University of Coimbra, Pólo II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal

<sup>b</sup> Centre for Informatics and Systems, University of Coimbra, Pólo II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal

<sup>c</sup> Department of Informatics Engineering, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

<sup>d</sup> Artificial Intelligence and Computer Science Laboratory, University of Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

<sup>e</sup> Department of Informatics, Polytechnic Institute of Viseu, Campus Politécnico de Repeses, 3504-510 Viseu, Portugal

<sup>f</sup> Institute of Electronics and Telematics Engineering of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

<sup>g</sup> Laboratory of Artificial Intelligence and Decision Support, University of Porto, Rua Dr. Roberto Frias s/n, 4200-465 Porto, Portugal

## ARTICLE INFO

### Article history:

Received 17 March 2013

Received in revised form 17 June 2014

Accepted 18 June 2014

Available online 25 June 2014

### Keywords:

Collaborative filtering

Memory-based techniques

Factorized Regression DataSet

Nearest neighbors

Robotic soccer simulation

## ABSTRACT

Collaborative filtering techniques have been used for some years, almost exclusively in Internet environments, helping users find items they are expected to like by using the user's past purchases to provide such recommendations. With this concept in mind, this research uses a collaborative filtering technique to automatically improve the performance of a simulated soccer team. Many studies have attempted to address this problem over the last years but none has shown meaningful improvements in the performance of the soccer team. Using a collaborative filtering technique based on nearest neighbors and the FC Portugal team as the test subject (in the context of the RoboCup 2D Simulation League), several simulations were run for matches against different teams with much better, better and worse performance than FC Portugal. The strategy used by FC Portugal was to combine 8 set-plays and 2 team formations. The simulation results revealed an improvement in performance between 32% and 384%. In the future, there are plans to expand this approach to other contexts, such as the 3D Simulation League.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Using a popular sport like soccer, RoboCup is an international scientific and educational project that provides researchers with standard problems, such as those presented by the learning of individual agents and teams dealing with uncertain environments (as is also the case of financial environments [1]), multi-agent team planning and plan-execution in service of teamwork and opponent modeling [2]. One of the competitions within the RoboCup initiative is the 2D Simulation League where two teams of 11 virtual players each compete in a simulated soccer match. These

matches are run using a soccer simulation engine<sup>1</sup> which generates a log file at the end of the match. This file contains data related to<sup>2</sup>:

- 1 Perception** – This kind of information is divided into three distinct groups: audio, visual and sensory data. These groups involve information related to different aspects, such as the game time and position, speed and head direction of players.
- 2 Action** – This group includes information regarding low-level actions (e.g. kick, tackle, neck turn) that each player is executing at a given time of the game.

In soccer matches (be it robotic, simulated or human), players are required to make decisions for individual or collective purposes.

\* Corresponding author at: Department of Informatics Engineering, Faculty of Sciences and Technology, University of Coimbra, Pólo II, Pinhal de Marrocos, 3030-290 Coimbra, Portugal. Tel.: +351 933506734.

E-mail addresses: [pha@dei.uc.pt](mailto:pha@dei.uc.pt) (P.H. Abreu), [dcs@fe.up.pt](mailto:dcs@fe.up.pt) (D.C. Silva), [falmeida@estv.ipv.pt](mailto:falmeida@estv.ipv.pt) (F. Almeida), [jmoreira@fe.up.pt](mailto:jmoreira@fe.up.pt) (J. Mendes-Moreira).

<sup>1</sup> rcserver, available at <http://server.sourceforge.net>

<sup>2</sup> For more detailed information please go to <http://wwfc.cs.virginia.edu/documentation/manual.pdf>

Usually, these decisions are influenced by factors such as the current team strategy in the game, the behavior of the opponent team and the current game result. Assuming that the main objective of this sport is to score more goals than the opposing team, the central task of a coach is to choose the best strategy that allows his team to achieve this result against a given opponent. However, the task of identifying the best team strategy to use against the opponent is not trivial.

Throughout the years, FC Portugal has tried to develop methods to automatically improve its performance in the RoboCup 2D Simulation League. Some of the developed research tried to characterize opponent teams either by using a soccer ontology [3] or by abstracting its behavior during a match based on collected game statistics [4–6]. These studies involved 2 phases: an offline phase where the intention was to model the behavior of the opponent; and an online phase where the aim was to classify the opponent [7] based on the assumption that to adopt the best strategy for a given opponent one must use some previous knowledge. This approach has also been used by other authors [8–10] and it is justified by the fact that a soccer game has a limited and short duration. However, this approach cannot adapt to changes in the strategy of the opponent team that occur during the match and consequently will most likely lead to non-optimal results in these circumstances. This work addresses this issue by making use of a Memory-Based Collaborative Filtering (CF) approach applied to a Factorized Regression DataSet (FRDS) to choose the strategy (based on a set of 2 formations and 8 set-plays) that maximizes the difference of scored goals benefiting FC Portugal.

This approach has some important differences in comparison to classical applications of CF to market basket analysis (MBA) datasets, where CF is used to predict additional topics or products that a new user may appreciate [11]. CF for FRDS aims to select a factor that maximizes the target quantitative variable given a vector of independent variables. To the best of our knowledge, the application of Memory-Based CF to FRDS has never been used. This topic is fully detailed in Section 5.

The obtained results reveal improvements on the performance of FC Portugal varying from 32% to 384% for different opponent teams, comparatively to using the same strategy during the entire match, thus proving this approach to be better than a Model-Based approach [7]. These are outstanding results, which demonstrate the feasibility of using this technique in this specific context.

The remainder of this paper is organized as follows: Section 2 presents a brief review of the literature. Section 3 describes the tool that was developed to calculate final game statistics and Section 4 presents the strategies available to be used by FC Portugal. Section 5 outlines the methodological steps used in this project and also the algorithms used to detect the opponent's behavior. Section 6 reports the collected results and Section 7 discusses the differences between the used approach and Model-Based CF. Finally, Section 8 presents the conclusions and proposals for further studies.

## 2. Literature review

To the best of our knowledge, the use of a CF technique for online strategic recommendations in robotic or simulated soccer to defeat an opponent constitutes a novel approach. However, there are several works focusing on related problems, such as opponent modeling which is an abstract description of how the opponent plays (for a more complete overview about opponent modeling and its applications scenarios refer to [12,13]).

One of the most widely known approaches in this area was developed by Carmel and Markovitch [14]. They proposed a generalization of the minimax algorithm based on a model that captured the opponent's weaknesses. By characterizing the

opponents according to classes, the authors were capable of improving the team performance by up to 13%. With the same problem in mind, the authors [15] implemented a new approach capable of constructing an opponent model using a deterministic finite automaton (DFA). The basic idea of this approach is to find the best response for a given DFA representative of an opponent model. This was done by means of a second DFA which made it possible to improve the performance of a soccer team in polynomial time [16]. However, since finding the second DFA is a NP-complete problem, the authors created a polynomial method based on observed data. The model is learned in an unsupervised way due to some particular issues related to computational processing costs. The main problems with this approach are that: (1) there are dynamic and complex game situations that cannot be represented using a deterministic automaton, and (2) since decisions are supported by an opponent prediction model, it is impossible to classify new behaviors that can be used by an opponent team as in the approach proposed by Carmel and Markovitch [14].

In 2005, Markovitch and Reger [17] proposed a new approach in the opponent modeling domain. Bearing in mind that learning a full model of the opponent is a complex task, the authors decided to adopt a new strategy, which consisted of constructing a model set up only by certain weaknesses of the opponent. They were capable of improving the team performance by up to 12%. Even though this approach is theoretically reasonable, there are still major problems. After detecting the weaknesses of an opponent, the authors decided which were the main actions that should be taken by their players to improve their performance. It must be noted that this approach was only tested with the games of Chess and Connect Four which constitute static, accurate, episodic and fully observable environments. However, this approach does not scale to the soccer domain which is a continuous, inaccurate, dynamic and partially observable environment. In particular, in order to choose the best set of actions for their players to execute in the soccer domain the authors must also analyze the weaknesses that their own players exhibit while executing all possible actions. Moreover, many other relevant aspects are not discussed, such as how the features in a set were selected, why some of the feature sets performed better than others and how feature weights were calculated.

In 2006, Janssen and Ahn [18] evaluated a hybrid model of decision making for a public-good scenario. However, this kind of environment presents some unique characteristics such as: non-rival – if one person uses a good it does not prevent another person from using it (e.g. clean air) – and Non-excludable – once goods are provided they are available to all users. The main issue associated with this scenario is described as the free riding problem and consists in the tendency to avoid paying for a public-good from the perspective of a player who benefits from the use of this good.

In 2008, Busoniu et al. [19] presented a survey about multi-agent reinforcement learning (MARL) and stated that a MARL goal should be based on two important pillars: stability of the learning dynamics of the agent (normally achieved using Nash equilibria) and the adaptation to the dynamic behavior of the other agents. However, it is argued that in stochastic environments there is no clear relationship between the stability of the learning dynamics and the achieved performance [20].

Aler et al. [21] presented an agent modulation for capturing relationships between inputs and outputs (IOAM) to model a human playing soccer. This approach consisted on allowing a human to play soccer using the RoboCup 2D Simulation League simulator. Using the part machine-learning algorithm, the authors were able to build a human player model and implement it in a software agent. Results showed that the modulated agent was capable of scoring more goals than any other agents in the same environment. It is important to highlight that this approach constitutes the first imitation approach in the RoboCup environment. A

different domain is the classification of the opponent behavior, where the main goal is not to imitate opponents but to model them in order to choose the best strategy to defeat them. Normally these models are predefined rather than learned and are used to classify an opponent team rather than an individual player.

Some of the research in the RoboCup environment is related to opponent modeling and is mostly focused on how a coach agent that has constrained communication with his players can be used to improve the performance of his team. Stone et al. [22] presented a low-level positioning and interaction agent approach based on an ideal world in which the performance of the opponent team is the best possible. A severe limitation of this approach is that it assumes that the process of adapting player positioning does not change during the match and it does not depend on the opposing team. An extension of this work is proposed by Ledezma et al. [23] where the main goal is to improve the low-level skills of the modeled agent. Druecker et al. [24] used a neural network to identify the formation of the opposing team. However, this kind of information is very limited as it is not capable of improving the performance of a soccer team by itself. Similarly, Riley et al. presented a formation learning approach based on player position [25]. However, the limitations presented by this study are similar to the previous one.

Based on the premise that the behavior of a player is influenced by past events, Iglesias et al. [26] tried to classify those behaviors across supervised learning (Chi-square test) sequences. These sequences are composed of events such as pass, dribble, steal and goal. The authors implemented 4–5 behaviors, one per player in three different matches. The results are preliminary and in addition the component responsible for correlating the acquired knowledge with the strategy to be adopted by the team presented several limitations. In the same context, Abreu et al. [7] divide the input space offline using clustering, and associating a strategy to each cluster. However, these approaches can recommend no more strategies than the number of clusters used to divide the input data. This is an important limitation that will be further debated in Section 7.

### 3. Statistical extracting tool

In this project, a tool capable of extracting final game statistics from soccer game logs was built. This tool builds on SoccerScope 2 software<sup>3</sup> and implements new features in order to fulfill the expectations of soccer experts. Using a sequential temporal analysis, a set of almost 60 statistics were defined. Furthermore, a team strategy was also created according to 2 types of information (explained below).

The set of statistics previously validated by a board of sports experts can be divided into 5 groups: Pass, Shot, Goal, Set Piece and Ball Possession.

- **Pass:** A successful pass occurs when a soccer player kicks a ball and, after a few cycles, a teammate receives the ball without a player from the opposing team intercepting it. In this work, the number of successful passes in each part of the match is analyzed, as well as those that are intercepted by a player from the opposing team. Other variations of passes that were also detected include “wing chain” and “pass chain”. In order to detect the “wing chain” event, the soccer field was divided into three equal regions/corridors: left, middle and right. If the event algorithm detects a successful pass between two teammates and if this pass occurs between the left and the right regions (or vice-versa), the algorithm will classify it as a successful pass and a “wing chain”.

The “pass chain” event consists on identifying the number of successful consecutive passes that a team is capable of executing.

- **Shot:** A shot event occurs when a player, in his attacking midfield, kicks the ball in the direction of the goal line (with a 5 m margin) with enough force for the ball to reach it. After this kick, if a player from the opposing team intercepts the ball, the event is classified as an intercepted shot. Otherwise, two situations can occur: the player's kick results in a goal or the ball leaves the field. In this last situation, if the ball leaves the field very close to the goal line, the event is classified as a ‘shot on target’; otherwise it is considered a ‘shot’.
- **Goal:** To detect a goal using a temporal sequential analysis, 3 consecutive time cycles must be analyzed. In the first cycle, the ball needs to be on the playing field and behind the goal line. In the next cycle the ball needs to be over the goal line. Finally, in the last cycle the ball must have passed the goal line completely. If these 3 conditions occur during the match the event is classified as a goal. The number of goals scored in both parts of the game is registered as well as the region from where the player kicked the ball that resulted in a goal. In this work, the concept of “goal opportunity” is created and consists on identifying the situations where an attacking player has a large probability of scoring a goal. The probability calculus is based on the position of the player (inside or outside the penalty area) and the number of players in the view field, aligned with the goal line. For each player this probability decreases by 0.2, plus 0.2 if the player is outside the penalty area.
- **Set piece:** A set piece is an extremely important game situation [27] and can be characterized as Corners, Goal-Kicks, Throw-Ins and Fouls. All these groups are detected by this tool; however, within the Fouls group, only offside situations are considered.
- **Ball Possession:** A soccer team is considered to have the possession of the ball within a given time interval if, during that time, none of the players from the opposing team intercepts the ball and the ball does not leave the playing field. To classify ball possession more succinctly, the soccer field was divided into 12 equal parts (6 defensive and 6 offensive). Furthermore, a new concept was introduced which consists on evaluating the time a team takes to get to the last third of the field without losing the ball. This information is extremely relevant in order to classify the offensive style that a team uses during a game. This classification is divided into 4 levels: slow, medium, fast or break depending on the moment when the opposing team recovers the ball.

### 4. Team strategy definition

The FC Portugal team strategy (Fig. 1) describes the high-level behavior that its players should follow in order to achieve the best possible result in a soccer match. A strategy is composed of several tactics, which encapsulate the players' cooperative behaviors in the form of formations and set-plays to be used in specific game situations (e.g. corner kick, attack). A formation is composed by a set of positionings that assign each player an agent type and a base strategic position.

An agent type encompasses the individual actions the player should take depending on whether he considers the game to be on an active (e.g. intercept the ball) or strategic (e.g. move to strategic position) situation for his team. The positionings in a formation are assigned to players during the match using matching algorithms based on utility functions. A set-play consists on a predefined recipe for a multi-step course of action that a subset of players should follow to improve the team's collective performance.

In this research work, the goal was not to change the predefined FC Portugal strategies or the code. The goal is to change the high-level player behaviors according to the opponent behavior. Consequently, in this work 2 high-level variables were used to

<sup>3</sup> More information available at <http://web.archive.org/web/20080616072239/http://ne.cs.uec.ac.jp/~koji/SoccerScope2/index.htm>

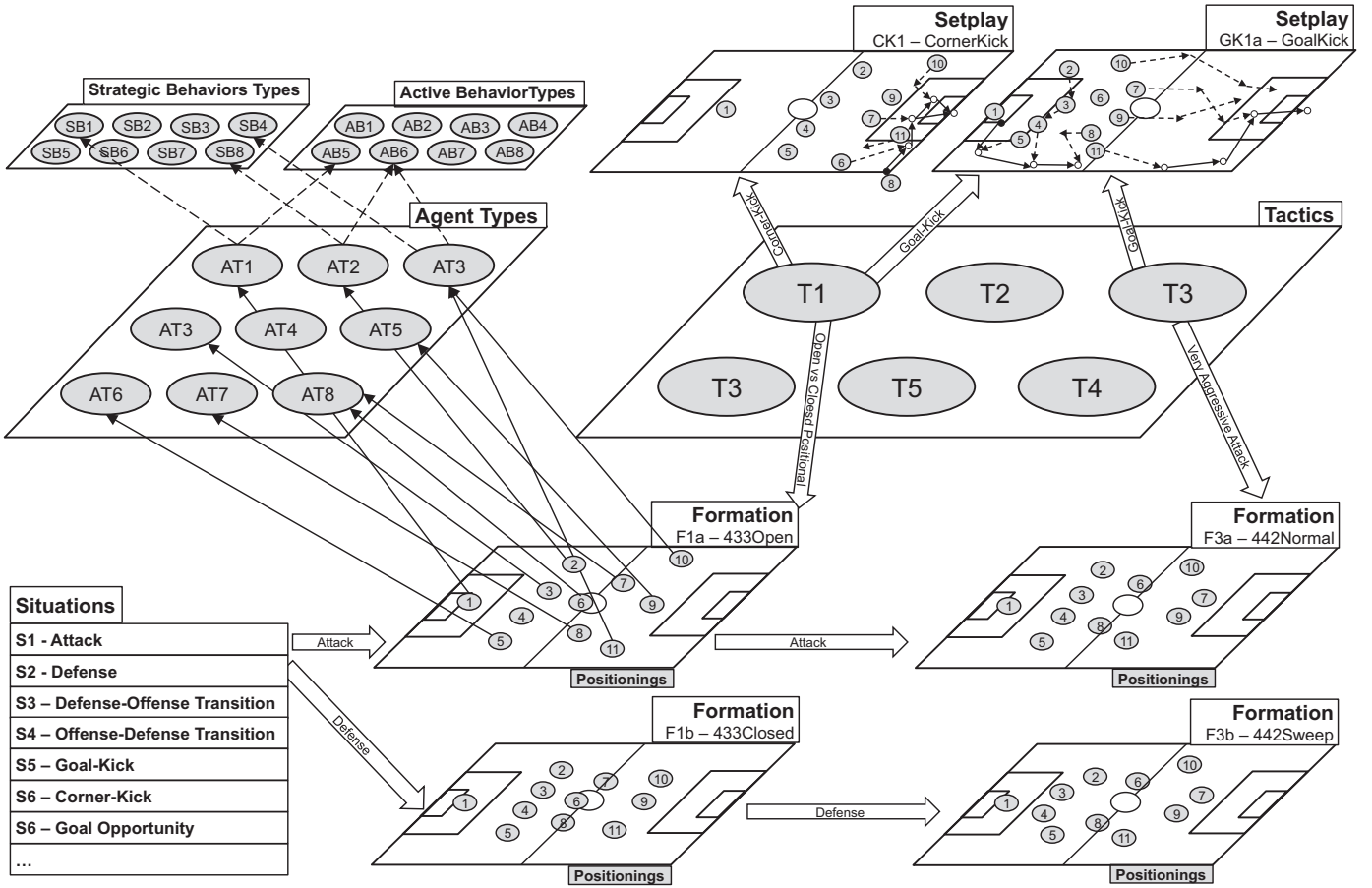


Fig. 1. FC Portugal internal architecture [28].

change the behaviors of players during the match: team formation (2); and set-plays (8). The Playmaker tool [29] was used to define the formations and set-plays used in the experiments.

#### 4.1. Team formation

A team formation defines the positions the players should occupy in the field for a given set of situations. In this tool, the player's positions are calculated using a Delaunay Triangulation [30] and a linear interpolation algorithm. The same algorithm used in the Gouraud Shading algorithm [31] was used for this task. Each situation included in a formation description consists on defining the players' positions depending on the ball position. Ball positions of all situations are used as vertices of a 2D Delaunay Triangulation. After determining the triangle (B1, B2, and B3) that encompasses the current ball position (B), player positions are calculated (P(B1), P(B2), and P(B3) are the positions of the players when the ball is in the corresponding point). Fig. 2 illustrates an example of the interpolation process [29].

The three steps that constitute this process are:

- 1 Calculate I (the intersection point between the B1B line and the B2B3 segment).
- 2 Calculate the interpolated target position for I as the current ball position and B2 and B3 as reference points (1):

$$P(I) = P(B2) + (P(B3) - P(B2)) * \frac{|B\vec{2}I|}{|B\vec{2}I| + |B\vec{3}I|} \quad (1)$$

- 3 Calculate the new player position in relation to the ball position B(2):

$$P(B) = P(B1) + (P(I) - P(B1)) * \frac{|B\vec{1}B|}{|B\vec{1}B| + |B\vec{1}I|} \quad (2)$$

In this research, 115 points were used to define each of the 2 used formations: 1-4-3-3 and 1-4-4-2. An example of a 1-4-3-3 formation definition using Playmaker is illustrated in Fig. 3.

The definition of all positions of team players for all relevant game situations can be time-consuming and cumbersome. As a general rule of thumb, players within a formation should be evenly distributed with regard to teammates and the soccer field boundaries. Consequently, to ease the definition of player positions in a formation their spatial relationship can be exploited. The most relevant spatial relationship to exploit consists on defining the position of a player based on the position of a specific teammate by mirroring

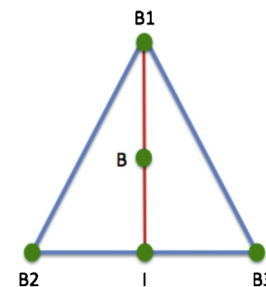


Fig. 2. Gouraud shading interpolation.



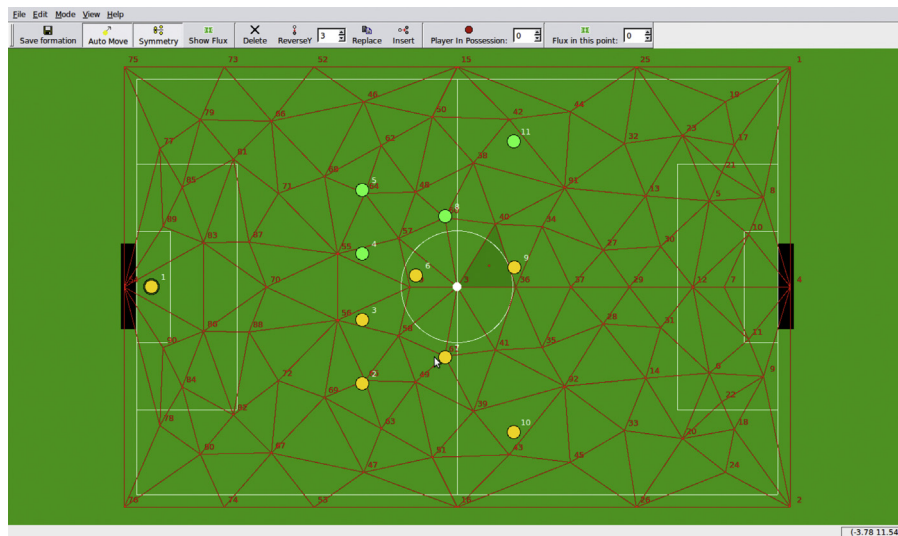


Fig. 3. Example of the 1-4-3-3 formation using the Playmaker tool.

it along the soccer pitch horizontal axis. Players represented with the green color have their positions dependent on players represented with the yellow color (e.g. the position of player 5 depends on that of player 2).

#### 4.2. Set-plays

The definition of set-plays can be seen as a flexible team plan for a specific game situation that may involve a specific game period, the number of scored goals, a game situation, such as a set piece, or even the opposing player's position on the field.

Usually, a set-play is identified by a name, a set of parameters (conditions) and players. Furthermore, a set-play can be seen as a list of states. The possible transition between these steps can be divided into 3 groups: (1) abort if all the conditions to continue with the set-play are not met; (2) go to the next step when all the conditions to continue are met; and finally (3) the finish transaction indicates that a set-play execution has been completed.

Eight different set-plays were used in this research. They can be divided into 4 groups relating to a specific game situation. Only one set-play for each group was used in each simulation. For each representative set-play figure, the black and white lines symbolize the movements of players and ball, respectively.

1. *Kick-off* is a situation that characterizes the start or resuming of a soccer match (e.g. after a goal is scored). Two set-plays relating to this game situation were used here:

- i. Kick-off to winger includes 4 steps (Fig. 4):
  - a. The kick-off taker passes the ball to the midfielder, while the defender positions himself to the left of the midfielder.
  - b. The midfielder passes the ball to the striker, while the winger moves closer to the offside line to the left. The defender stays on the left of the midfielder.
  - c. The striker passes the ball to the defender, while the winger keeps heading towards the offside line to the left.
  - d. The defender passes the ball to the winger.
- ii. Kick-off to winger includes 2 steps (Fig. 5):
  - a. The taker makes a direct pass to the midfielder. The winger positions himself near the offside line to receive a pass from the midfielder.
  - b. The midfielder passes the ball to the winger.

2. *Free Kick* is a situation that occurs when a player violates the game rules. For this situation, two distinct set-plays were used

taking into account that this situation takes place in the offensive region, from *our* team's perspective.

- i. Free Kick directly to the goal line (Fig. 6):
    - a. The taker passes the ball to a receiver near the goal.
    - b. The receiver shoots for the goal.
  - ii. Free Kick where the ball possession is privileged (Fig. 7):
    - a. The taker passes the ball to the striker.
    - b. The striker passes the ball to the left-winger or the right-winger; if this is not possible, the ball is passed back to the taker.
3. *Goal Kick* occurs when the ball crosses the goal line completely without a goal having been scored and having last been touched by an attacking team player. Two different Goal Kick plays were used:
- i. Goal Kick involving 7 steps (Fig. 8):
    - a. The goalkeeper and the left defender move to the left region of the field (the left defender more to the left and in front of the keeper).
    - b. The goalkeeper kicks the ball to the region where the left defender is running. If the left defender intercepts the ball, the left midfielder moves to the left and front of the left defender.
    - c. The left defender kicks the ball to the place where the left midfielder is running. If the left midfielder intercepts the ball, the left forward moves to the front and left of the midfielder.
    - d. The left midfielder kicks the ball to be intercepted by the left forward. A runner moves to the front of the left forward.
    - e. The left forward kicks the ball forward to be intercepted by the runner.
    - f. The runner kicks the ball to the right and it is received by a kicker. The runner keeps running forward.
    - g. The kicker kicks the ball in a left and forward direction to be intercepted by the runner.
  - ii. Goal Kick involves 3 steps (Fig. 9):
    - a. The players' position themselves in two rows perpendicularly to the goal line, and in front of the penalty area (except for the goalkeeper who stays inside the penalty area). The goalkeeper kicks the ball to the player closest to him on the right.
    - b. The player with the ball passes it to the player in front of him (on the other row) and slightly ahead (zig zag slalom).
    - c. The previous step is repeated until the last player receives the ball.

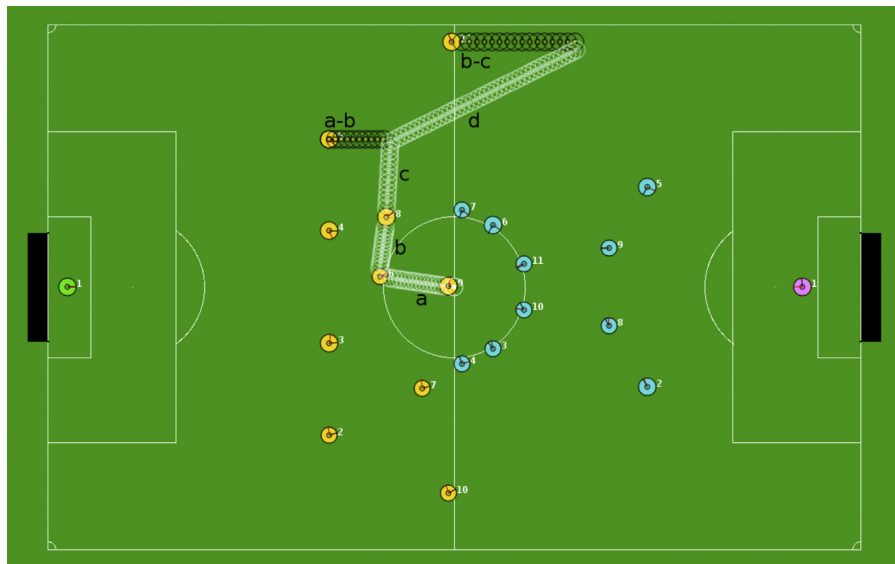


Fig. 4. Example of a Kick-off set-play involving 4 steps.

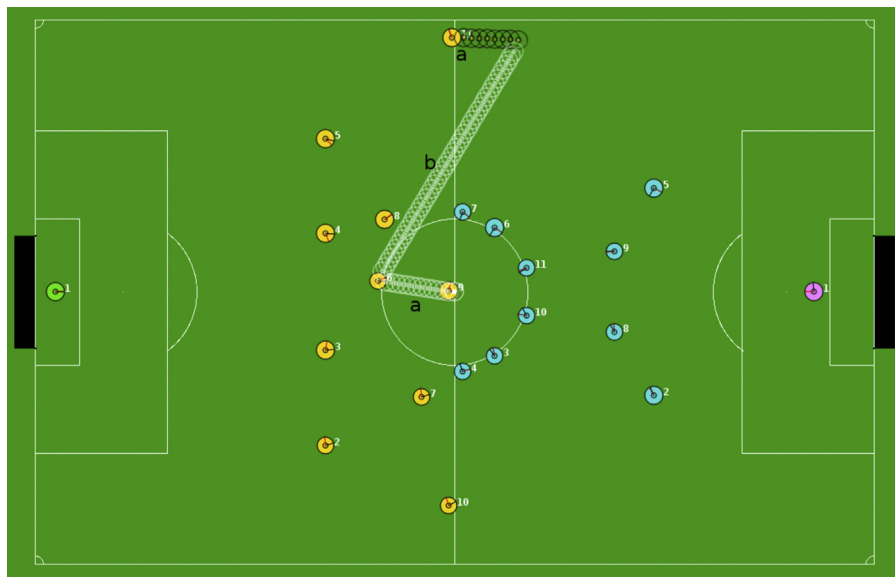


Fig. 5. Example of a Kick-off set-play involving 2 steps.

4. *Corner Kick* is allowed when the whole of the ball passes over the goal line without a goal being scored, and it was last touched by a player from the defending team. Two corner kick situations were used:

- i. *Corner Kick to the left striker* (Fig. 10):
  - a. The taker passes the ball to the nearest receiver.
  - b. The receiver passes the ball to the left striker who is in front of the opponents' goal line.
  - c. If the left striker has enough space to shoot the ball, the ball will be shot towards the goal line; if not, the ball will be passed to the center striker.
  - d. The center striker receives the ball and shoots.
- ii. *Corner Kick*:
  - a. The taker passes the ball to the nearest receiver.
  - b. The receiver dribbles the ball back to the middle of the penalty box.
  - c. The receiver passes the ball either to the left, to the right or to the center striker.
  - d. Whoever receives the ball shoots at the goal.

## 5. Methodology

Collaborative filtering (CF) is a well-known recommendation method for market basket analysis (MBA) datasets. An example of such approach is the recommendation of items to a user in an online shop based on his history as a buyer in that shop and that of buyers with similar tastes [11]. The problem under study, however, cannot be solved using the methods used for MBA datasets because the data under study has different characteristics. We name this kind of datasets Factorized Regression DataSet (FRDS). CF for FRDS aims to select a factor that maximizes the target quantitative variable given a vector of independent variables. In the case under study it aims to recommend a strategy (a factor) according to how the opponent team is playing in order to maximize the goal difference (a quantitative variable). The proposed methodology is a Memory-Based CF approach for FRDS.

This section starts by formalizing FRDS and factorized regression (FR) problems. Then, a brief description of Memory-Based CF is presented and the main differences between the application



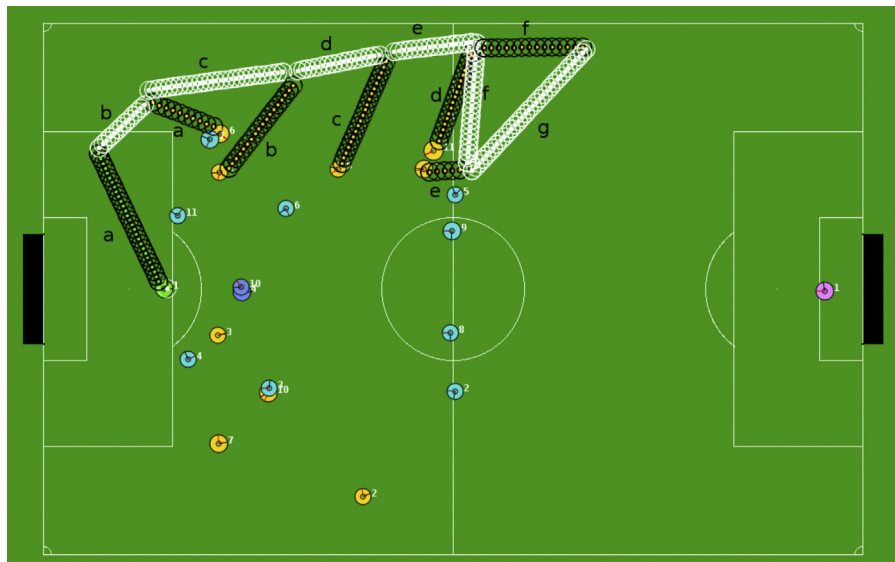


Fig. 8. Example of a Goal Kick set-play involving 7 steps.

In the given MBA problem, the available data is a matrix  $M(u \times i)$  where, in the given example,  $u$  is the number of users and  $i$  the number of items. The matrix is filled with binary values, where 1 represents that the item was bought by the user and 0 represents no action. The matrix is typically large in the two dimensions and sparse because users usually only buy a very small percentage of available items. Both the number of items and users can increase over time.

In the given FR problem the instances are games. The variables are the strategy used by our team (the factor), a set of statistics on how the opponent team plays and the final goal difference between our team and the opponent (the quantitative target variable). No variables have missing values and all but strategy have quantitative values. There are no more than a few dozens of variables and, once defined, the variables are fixed over time. Naturally, the number of games can increase over time.

The differences between the use of Memory-Based CF in MBA and FR problems are:

- In MBA problems, given a new user, the similarity measures are based on the items bought by the new user and past users [11]. In the FR problem the most similar games do not depend on having common values with the given game for a variable or set of variables, i.e., it can be obtained using the usual distance measures depending on the data type of the variables [32]. A possible distance measure for FR problems is the widely used Euclidean distance, assuming that all variables in  $\mathbf{x}$  are quantitative, as is the case under study.
- In MBA problems it is difficult to recommend items adapted to the user when the user or the item is new. The quality of the recommendations typically increases when the history of that user/item also increases. This problem is known as cold start. In the FR problem this issue does not exist. Indeed, while the similarity between two users depends on the history of both users in terms of common items bought, in the FR problem the similarity measure (using the Euclidean distance for instance) can always be calculated whichever the values of  $\mathbf{x}$  are.

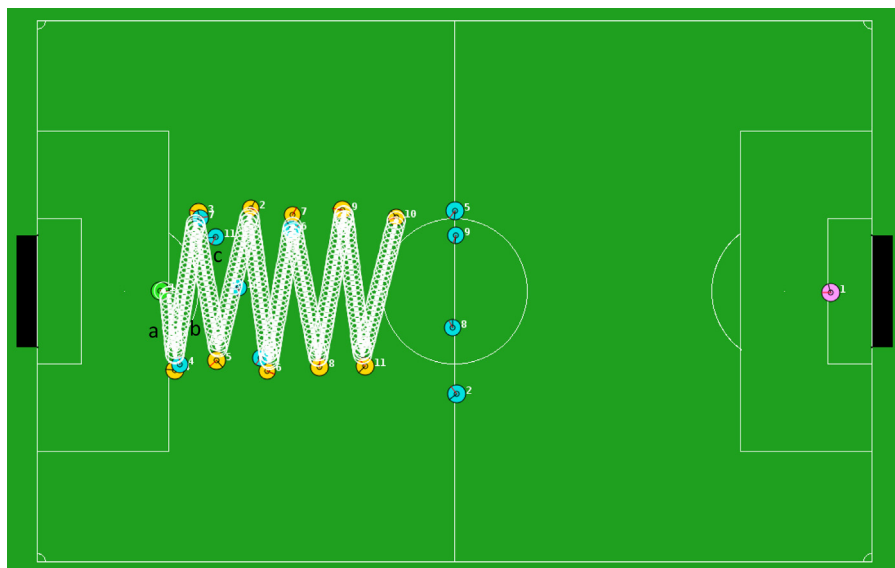


Fig. 9. Example of a Goal Kick set-play involving 3 steps.



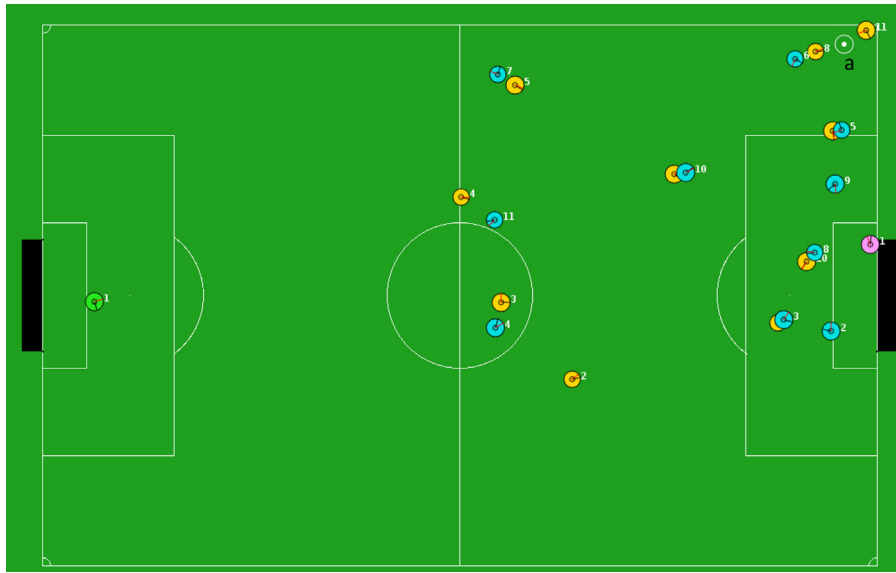


Fig. 10. Example of a Corner set-play.

- The data in MBA problems is much more constrained in terms of format than in FR problems. Consequently, despite the fact that ranking criterion can be different in MBA or FR problems, they are expected to be less diverse in MBA than in FR problems. Indeed, a typical ranking criterion in MBA datasets is the amount of common items bought by both users. Naturally, several other criteria exist but, due to the nature of the dataset, they do not vary too much. FRDS are more generic than MBA datasets and can be seen in very different contexts implying, typically, different ranking criteria.

### 5.3. Memory-Based CF for FR problems

To the best of the authors' knowledge the only work that addresses a problem like the one described in Section 5.1 is [7]. In that work, the games were clustered using the k-means algorithm and for each cluster a strategy is identified as the most promising one. Given the statistics of a current game, a classifier identifies the most adequate cluster for it. Then, the strategy identified for that cluster is the one recommended to be used by the team (additional information about this method is given in [7]). This is a Model-Based CF approach. The main drawback of this approach is that the number of strategies that can be recommended is, at most, the number of used clusters. This is a strong limitation because the number of strategies (16) is typically much larger than the number of clusters (in [7], 9 clusters were used).

For that reason, a new method is proposed using distance measures to define the neighborhood given the statistics of the current game. This neighborhood is used to identify the games where the opponent team has played the closest to the current opponent play style. This neighborhood is calculated only when the statistics of the current game are known. This proposed Memory-Based approach is, by comparison against the Model-Based one presented in [7], able to recommend more strategies because it is not constrained by the numbers of clusters used. Moreover, the distance is calculated against the data of the current game statistics. In the Model-Based approach, the neighborhood of the current game statistics is the data in the cluster identified as the most adequate for it. If the new game is in the border of that cluster, its neighborhood (games in the identified cluster) are not likely to be the best

ones to be used for recommending a strategy for the problem under study.

Five steps organized in two phases are used to solve the given problem (Fig. 11):

- The off-line phase is executed typically only once and is comprised of 3 steps that are executed before knowing the new statistics of a game:
  - (1) Simulation: simulated matches are run producing 56 statistics per match.
  - (2) Feature selection: features are selected according to its relevance to predict the target variable, i.e., the goal difference.
  - (3) Feature weighting: a weight is calculated for each selected feature according to its relevance to the target variable.
- The on-line phase is comprised of 2 steps that are executed when the statistics of a given game are known:
  - (4) Neighborhood definition: the nearest games in terms of statistics are selected using a weighted distance measure.
  - (5) Strategy recommendation: the best strategy according to a problem-defined criteria is selected.

Steps 1 and 2 have been previously described in [7] and are only briefly summarized here to clarify their role within the proposed methodology. The remaining steps constitute the core component of the proposed approach and are described in more detail below.

#### 5.3.1. Simulation

To better characterize the game, during the simulation step, 56 statistics were collected based on the information present in the log files (previously mentioned in Section 3). A brief description of these statistics arranged in groups is shown in Table 1.

#### 5.3.2. Feature selection

This step aims to identify the variables that should be used to calculate the distance for finding past games where opponents had the most similar way of playing as the current opponent. Despite this step being fully described in [33], the MARS (multivariate adaptive regression splines) algorithm and the main results obtained in this step are presented.

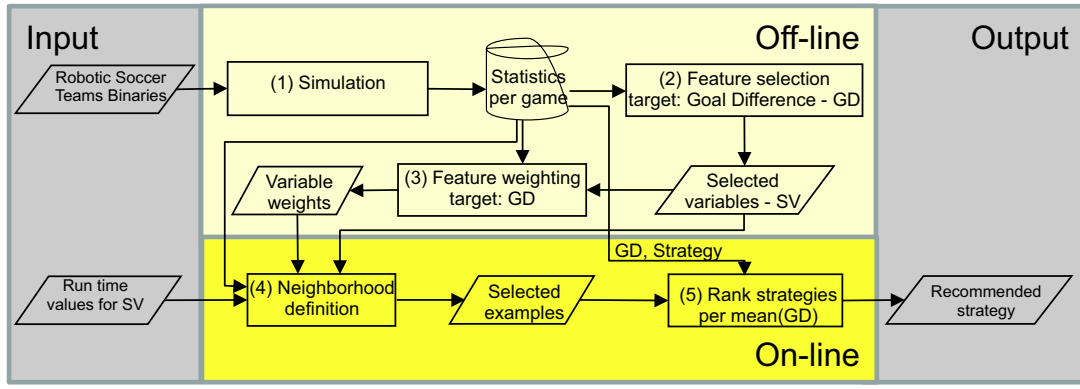


Fig. 11. Project architecture (where GD – goal difference; SV – selected input variables).

The MARS model is expressed as follows, where  $\beta_i (i=0, \dots, I)$  and  $c_{ij} (i=0, \dots, I; j=1, \dots, J_i)$  are constants determined by MARS:

$$\hat{y} = \beta_0 + \sum_{i=1}^I \beta_i \times \prod_{j=1}^{J_i} \begin{cases} \max(0, x[k_{ij}] - c_{ij}) \\ \max(0, c_{ij} - x[k_{ij}]) \end{cases} \quad (4)$$

The number of MARS terms ( $I+1$ , where the plus 1 refers the  $\beta_0$  coefficient) are determined on the fly. However, a maximum number can be specified. When a term has  $J_i > 1$  it means that it models interactions between two or more variables. The maximum degree of interaction can be also specified. At the end of this step MARS selected 18 of the 56 game statistics illustrated in Table 1, presented in bold. For more information please refer to [33].

### 5.3.3. Feature weighting

In order to use weighted distance measures for the neighborhood definition (Section 5.3.4) the weights for each selected variables must be calculated. These weights should be defined according to the relevance of each variable to the target. Algorithms of the Relief family of algorithms [34] are suitable for this purpose. Since the target variable is quantitative, the weights were calculated using the RRelief algorithm (algorithm 5.1).

#### Algorithm 5.1. The RRelief algorithm

**Require:**  $\{R_i : i=1, 2, \dots, M\}$ ,  $M$  vectors of length  $V$   
**Require:**  $\tau$ , the correspondent  $M$  target values  
**Require:**  $t$ , the number of iterations  
**Require:**  $k1$ , the number of nearest examples  
 set all  $N_{dC}$ ,  $N_{dA}[A]$ ,  $N_{dC\&dA}[A]$ ,  $W[A]$  to 0, where  $A=1, 2, \dots, V$   
**for**  $i=1$  to  $t$  **do**  
   randomly select example  $R_i$   
   select  $k1$  examples  $I_j$  nearest to  $R_i$   
   **for**  $j=1$  to  $k1$  **do**  
    $N_{dC} := N_{dC} + \text{diff}(\tau(\cdot), R_i, I_j) \times d(i, j)$   
   **for**  $A=1$  to  $V$  **do**  
    $N_{dA}[A] := N_{dA}[A] + \text{diff}(A, R_i, I_j) \times d(i, j)$   
    $N_{dC\&dA}[A] := N_{dC\&dA}[A] + \text{diff}(\tau(\cdot), R_i, I_j) \times \text{diff}(A, R_i, I_j) \times d(i, j)$   
   **end for**  
   **end for**  
**end for**  
**for**  $A=1$  to  $V$  **do**  
    $W[A] := N_{dC\&dA}[A] / N_{dC} - (N_{dA}[A] - N_{dC\&dA}[A]) / (t - N_{dC})$   
**end for**  
**return**  $W$

Robnik-Sikonja et al. [34] propose a deeper analysis of the proposed method. At the beginning of the calculation process, it is

important to define two variables:  $k1$  – representing the number of nearest neighbors – and  $t$  – representing the number of iterations. Following the advices of Robnik-Sikonja (to decrease computation costs and increase stability of weight estimations) in this research  $t$  and  $k1$  were defined as being equal to 50 and 10 respectively. It is important to state that  $k1$  is only used in the RRelief calculus and does not have any relationship with the calculus of the neighborhood definition described in Section 5.3.4. In addition to  $t$  and  $k1$  two additional parameters exist:  $\{R_i, i=1, 2, \dots, M\}$  that is, in this case, a set of  $M$  vectors  $R_i$  each having the selected statistics and the strategy of a match; and a vector  $\tau$  of size  $M$  with the correspondent goal difference in each match.

In order to calculate the distance between the variables, researchers have used Eq. (5), where  $teq$  and  $tdiff$  are 5% and 10% of the length of the input variable's interval value (as suggested by the authors), and  $d$  represents the absolute difference of the input variable  $A$  for the two examples,  $I1$  and  $I2$ .

$$\text{diff}(A, I1, I2) = \begin{cases} 0 & : d \leq teq \\ 1 & : d > tdiff \\ \frac{d - teq}{tdiff - teq} & : teq \leq tdiff \end{cases} \quad (5)$$

### 5.3.4. Neighborhood definition

To identify the similarities between examples, two well established distance measures were used to select a user-defined number of nearest examples: Euclidean (Eq. (6)) and Manhattan (Eq. (7)).

$$d_E(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (6)$$

$$d_M(x_1, x_2) = \sum_{i=1}^n |x_{1i} - x_{2i}| \quad (7)$$

These measures weigh all variables equally. However, the relevance for the output variable is expected to be different for distinct variables. Using the weights obtained in the third step, new distance measurements are calculated as illustrated in Eqs. (8) and (9), guaranteeing that the similarity is measured weighing the attributes differently, according to their relevance to the output variable. All four distance measures (2 with and 2 without weights) were tested in the experiments described in Section 6.

$$d_{WE}(x_1, x_2) = \sqrt{\sum_{i=1}^n w_i (x_{1i} - x_{2i})^2} \quad (8)$$

**Table 1**  
Game statistics (statistics marked in bold were selected using the MARS algorithm – explained below).

Group	Acronym	Definition
Pass	<b>Year</b>	Competition year
	<b>1-GoodPassTot</b>	Total number of successfully executed passes in the first half of the game
	<b>1-GoodDef</b>	Total number of successful passes in the first half of the game executed in the defensive zone
	<b>1-GoodOff</b>	Total number of successful passes in the first half of the game executed in the offensive zone
	2-GoodPassTot	Total number of successfully executed passes in the second half of the game
	<b>2-GoodDef</b>	Total number of successful passes in the second half of the game executed in the defensive zone
	2-GoodOff	Total number of successful passes in the second half of the game executed in the offensive zone
	<b>1-BadPassTot</b>	Total number of missed passes in the first half of the game
	1-BadDefDefensive	Total number of missed passes in the first half of the game executed in the defensive zone
	1-BadOff	Total number of missed passes in the first half of the game executed in the offensive zone
	<b>2-BadPassTot</b>	Total number of missed passes in the second half of the game
	<b>2-BadDef</b>	Total number of missed passes in the second half of the game executed in the defensive zone
	2-BadOff	Total number of missed passes in the second half of the game executed in the offensive zone
	PassChain	Total number of consecutive passes executed by a team
	WingChain	Total number of successful passes between two defined regions
Shot	1-Shot	Total number of shots executed in the first half of the game
	1-IntShot	Total number of intercepted shots in the first half of the game
	1-ShotTarget	Total number of shots on target executed in the first half of the game
	2-Shot	Total number of shots executed in the second half of the game
	<b>2-IntShot</b>	Total number of intercepted shots in the second half of the game
	2-ShotTarget	Total number of shots on target executed in the second half of the game
Goal	<b>GoalsTot</b>	Total number of goals scored
	1-Goals	Total number of goals scored in the first half of the game
	2-Goals	Total number of goals scored in the second half of the game
	PenBoxBack	Total number of goals scored from inside the Penalty Box Area
	PenArea	Total number of goals scored from inside the Penalty Area
	OutPenArea	Total number of goals scored from outside the Penalty Area
Set Piece	GoalsOpp	Total number of goal opportunities
	OutTot	Total number of outside situations
	1-GoalKick	Total number of goal kicks in the first half of the game
	1-Corner	Total number of corners in the first half of the game
	<b>1-ThrowIn</b>	Total number of Throw-ins in the first half of the game
	2-GoalKick	Total number of goal kicks in the second half of the game
	2-Corner	Total number of corners in the second half of the game
	2-ThrowIn	Total number of Throw-ins in the second half of the game
	1-Offside	Total number of offsides in the first half of the game
	2-Offside	Total number of offsides in the second half of the game
	<b>1-OffInt</b>	Total number of intercepted offsides in the first half of the game
	2-OffInt	Total number of intercepted offsides in the second half of the game
Ball Possession	BroAtt	Total number of broken attacks
	<b>FasAtt</b>	Total number of fast attacks
	MedAtt	Total number of medium attacks
	SloAtt	Total number of slow attacks
	<b>AttTot</b>	Total number of attacks
	<b>1-LeftBposs-Def</b>	Ball possession throughout the first left defensive field zone
	<b>2-LeftBposs-Def</b>	Ball possession throughout the second left defensive field zone
	3-LeftBposs-Attack	Ball possession throughout the first left offensive field zone
	4-LeftBposs-Attack	Ball possession throughout the second left offensive field zone
	1-MiddBposs-Def	Ball possession throughout the first middle defensive field zone
	<b>2-MiddBposs-Def</b>	Ball possession throughout the second middle defensive field zone
	3-MiddBposs-Attack	Ball possession throughout the first middle offensive field zone
	4-MiddBposs-Attack	Ball possession throughout the second middle offensive field zone
	1-RightBposs-Def	Ball possession throughout the first right defensive field zone
	<b>2-RightBposs-Def</b>	Ball possession throughout the second right defensive field zone
	3-RightBposs-Attack	Ball possession throughout the first right offensive field zone
	4-RightBposs-Attack	Ball possession throughout the second right offensive field zone

$$d_{WM}(x_1, x_2) = \sum_{i=1}^n w_i |x_{1i} - x_{2i}| \quad (9)$$

- 2 Choose the one that appears most frequently in the previously selected nearest neighbors.
- 3 Choose the one with the maximum value for the MARS function.

### 5.3.5. Ranking strategies

In this step the best strategy for a specific opponent is recommended. In this work the best strategy is the one with the maximum average of goal differences. To choose the strategy, the following criterion were used sequentially until only one strategy was selected:

- 1 Choose the one with the maximum average goal differences from FC Portugal's perspective.

## 6. Experimental results

To produce experimental results, 840 games were simulated (280 per opposing team) between FC Portugal and each of the following teams: Wright Eagle (the best team in the tournament), Nemesis (the team one place above FC Portugal in the final competition table) and Bahia 2D (the worst team in the competition). Simulations were run using different frequencies in a simulated soccer game composed of a total of 6000 cycles: every 500 cycles,

**Table 2**  
Ranks of the Friedman test.

CF approach	Distance measure	Number of NN	Number of cycles	Ranks												Mean
				1	2	3	4	5	6	7	8	9	10	11	12	
Model-Based	N/A	N/A	500	5	5	5	5	5	5	5	6	2	5	4	5	4.75
	N/A	N/A	2000	6	6	6	6	6	6	6	5	6	6	5	7	5.916667
Memory-Based	$d_M$	20	1000	2	4	4	1	2	3	1	4	1	1	2	2	2.25
	$d_{WM}$	10	1000	4	3	2	4	1	4	4	2	5	2	1	1	2.75
	$d_E$	15	500	1	1	1	3	4	1	2	1	4	3	6	4	2.583333
	$d_{WE}$	10	500	3	2	3	2	3	2	3	3	3	4	3	3	2.833333
			Baseline	7	7	7	7	7	7	7	7	7	7	7	6	6.916667

CF – collaborative filtering; NN – nearest neighbor; N/A – not applicable.

every 1000 cycles and every 2000 cycles. Also, different values for the number of nearest neighbors were tested: 5, 10, 15 and 20. Finally, the number of neighbors were combined with the two distance measurements (with and without weights) and used in different game periods which were recorded in 48 different combinations (3 frequencies  $\times$  4 nearest neighbors' values  $\times$  4 distance measurements). As the combination number was high and most of them did not award useful results, it was decided to present only the best combination of frequency  $\times$  nearest neighbors per distance measurement. As a consequence, 6 configurations (4 approaches using Memory-Based CF and 2 others using Model-Based CF, as proposed by Abreu et al. [7]), were compared to a Baseline algorithm. This approach (baseline) chooses the strategy that FC Portugal will use before the game starts and the chosen strategy is maintained throughout the game. The results were evaluated according to the number of wins, draws and defeats. If, for some reason, there was a draw between the results produced by at least two algorithms (same number of victories, defeats and draws), a draw scale is used. This scale consisted of assigning 4 points for a victory, 2 points for a draw and 1 point for a loss. These values increased 0.1 per goal scored (increasing with the difference of goals scored in case of victory and decreasing in case of defeat). For instance, if FC Portugal wins two games with a difference of 2 and 12 goals, these games will be ranked with 4.2 and 5.2 points respectively. Furthermore, if this team loses two games by a difference of 2 and 12 goals, these games will be ranked with 0.8 and  $-0.2$  points respectively.

The comparison is made using the Friedman rank test. The averages of the results of each of the 6 configurations were compared using the baseline approach (7 different options). The 280 simulations per opponent were divided into 4 distinct groups (1 per option). The obtained ranks are shown in Table 2. Following the work presented by Demšar [35], and as  $N > 10$  (number of split groups – 12) and  $k > 5$  (number of classifiers used – 7)  $F_f = 31.40$  was calculated and compared to the  $F$  distribution  $F(0.01)_{6,66} = 7.07$ . As a consequence, the null hypothesis of equivalence between the 7 predictors is rejected. Comparing the 6 configurations to the baseline for a 1% significance level using the Bonferroni–Dunn test [35], it was possible to obtain  $CD = 2.3265$ .  $CD$  is the critical value for the difference of mean ranks between the baseline and any of the other

6 predictors. It was proved that the four Memory-Based approaches (two using Manhattan Distance –  $d_M$  and  $d_{WM}$  – and two using Euclidean Distance –  $d_E$  and  $d_{WE}$  – in both cases using simple and weighted variant) perform better than the baseline.

As far as the scale presented in Table 3 is concerned, it was possible to improve FC Portugal's performance by over 58% (59.11%, 61.77%, 63.29% and 58.93%) for the 4 distance measurements. Performing a deeper analysis of the opponent (using the same scale), for the best team in the tournament (WE) all Memory-Based approaches improved the performance of the baseline approach (by more than 48%). For the Nemesis team, once again all Memory-Based approaches beat the baseline approach (over 377% of improvement). The FC Portugal team presented a smaller improvement against the Bahia 2D (around 32%). This result is explained by the fact that the Bahia team is a rookie in the RoboCup competition and, because of that, teams like FC Portugal normally win the games scoring many goals. For that reason, the improvement will always be somewhat limited (when compared to the other 2 teams).

In terms of the number of wins, defeats and draws (Table 4), the four Memory-Based approaches presented outstanding results. Globally (without individual team analysis), the  $d_M$ -20-1000 (distance measure-number of nearest neighbors-frequency) increased the number of wins by 40%, decreased the number of losses by 21.67% and increased the number of draws by 227.59%. As far as  $d_{WM}$ -10-1000 is concerned, there was an increase in wins of 42.67%, a decrease in defeats of 21.5% and an increase in 203.45% in the number of draws. The  $d_E$ -15-500 increased the number of wins by 42.22%, decreased the number of losses by 26.28% and increased the number of draws by 303.45%. Finally,  $d_{WE}$ -10-500 presented an increase in wins (36%), a decrease in defeats (21.33%) and an increase in draws (251.72%).

Finally, in terms of the difference in goals scored (always from the FC Portugal perspective), the numbers presented were also excellent: 151.21%, 149.79%, 154.59% and 150.86% for the  $d_M$ -20-1000,  $d_{WM}$ -10-1000,  $d_E$ -15-500 and  $d_{WE}$ -10-500, respectively (Table 5). This constitutes an outstanding result when compared to other studies, such as the one conducted by Stone et al. [22], which presented an increase of 9 goals on average, and the study

**Table 3**  
Percentage of performance improvement for the FC Portugal team using a predefined scale as criteria.

	Baseline	WE 120.8	Nemesis 78.8	Bahia 1051.4	Total 1251	Total percentage
Score by opponent	$d_M$ -20-1000	185.2	396.9	1408.36	1990.46	59.11%
	$d_{WM}$ -10-1000	189	437.6	1397.2	2023.8	61.77%
	$d_E$ -15-500	179	451.3	1412.4	2042.7	63.29%
	$d_{WE}$ -10-500	191.4	376.5	1411.1	1979	58.93%
Percentage by opponent	$d_M$ -20-1000	53.31%	403.68%	33.95%		
	$d_{WM}$ -10-1000	56.46%	455.33%	32.89%		
	$d_E$ -15-500	48.18%	472.72%	34.34%		
	$d_{WE}$ -10-500	58.44%	377.79%	34.21%		

**Table 4**

Percentage of FC Portugal performance improvement using the number of victories, defeats and draws as criteria.

Baseline	WE	Nemesis	Bahia	Total	Total percentage
	Victory/defeat/draw				
	0/277/3	1/276/3	224/33/23	225/586/29	
$d_M$ -20-1000	1/267/12	35/192/53	279/0/1	315/459/66	40%/-21.67%/227.59%
$d_{WM}$ -10-1000	2/272/6	41/187/52	278/1/1	321/460/59	42.67%/-21.5%/203.45%
$d_E$ -15-500	3/270/7	42/160/78	275/2/3	320/432/88	42.22%/-26.28%/303.45%
$d_{WE}$ -10-500	4/266/10	25/195/60	277/0/3	306/461/73	36%/-21.33%/251.72%

**Table 5**

Percentage of FC Portugal performance improvement using the difference of scored goals scored as criterion.

	Baseline	WE	Nemesis	Bahia	Total	Total percentage
		-1619	-2069	790	-2898	
Score by opponent	$d_M$ -20-1000	-1048	-376	2908	1484	151.21%
	$d_M$ -10-1000	-1030	-372	2845	1443	149.79%
	$d_E$ -15-500	-1162	-295	3039	1582	154.59%
	$d_{WE}$ -10-500	-1106	-385	2965	1474	150.86%
Percentage by opponent	$d_M$ -20-1000	35.27%	81.82%	368.10%		
	$d_{WM}$ -10-1000	36.38%	82.02%	360.13%		
	$d_E$ -15-500	28.23%	85.74%	384.68%		
	$d_{WE}$ -10-500	31.69%	81.39%	375.32%		

conducted by Ledezma et al. [23], with an average increase of one goal (in one hundred games), or even the one performed by Abreu et al. [7] which improved less than 100%. In conclusion, it was proved that the approach used here is capable of improving the performance of a simulated soccer team.

## 7. Memory-based vs. model-based discussion

In this research a Memory-Based CF algorithm (using nearest neighbors) was used to automatically improve the performance of a soccer team. This work was compared to Model-Based approaches developed by Abreu et al. [7] and presented a meaningful advantage towards Model-Based approaches, as shown in the previous section. In this section, the results are further discussed. The aim is to provide some clues on why Memory-Based approaches consistently beat Model-Based ones. An important difference between these two approaches is that Memory-Based approaches delay the construction of the model to the moment where the recommendation is necessary (this is a lazy approach [36]); Model-Based approaches filter the examples offline reducing the online effort of the recommendation task (this is an eager approach).

Decision trees can give a good understanding of lazy and eager approaches. While the C4.5 [37] or the CART [38] algorithms are eager, which means that the decision tree is built before the test example is known, the lazy decision trees [39] are built only at the prediction moment, when the test example is known. As a consequence, eager trees are built in order to cover all the input space, while lazy trees only build the branch needed for the particular test case. The eager version is more time consuming offline and less so online, while the lazy version does all the work online, and thus is less time consuming offline and more so online. The same happens to Memory-Based CF vs. Model-Based CF, as can be seen in Table 6, which shows the execution times for both approaches.

Additionally, the lazy version adapts the leaf node to the test example and, consequently, the example is at the center of the leaf. In the eager approach the example can fall to the border of the leaf. Moreover, eager decision trees typically have a smaller number of leaf nodes comparatively to the potential number of different leaf nodes that the lazy version can create. Even though the lazy version develops only one leaf node, this node is potentially different for

each test example. For all these reasons, the prediction ability of lazy decision trees is, at least theoretically, better than in the eager version, but with an additional online computational cost.

The reason to compare eager and lazy decision trees is that the majority of these arguments are also applicable to the construction of local ranks. In the eager version, the regions and the ranks are built offline and once they are built they can be used multiple times. In this case, the number of regions (clusters) is given as input and each region has a strategy associated to it. Consequently, the number of recommended strategies is lower than or equal to the number of regions (clusters). In the lazy version, everything is performed online but only one region and one rank are defined for a given example. However, the region and the rank must be defined again for each new example. In this approach, all strategies can be recommended, at least potentially.

## 8. Conclusions and future work

In this project, a Memory-Based CF approach applied to a FRDS was used to recommend the strategy that maximizes the difference of scored goals in simulated soccer competitions. As far as the authors can tell, this constitutes one of the first approaches regarding Memory-Based CF and FRDS. This approach was compared to a Model-Based CF approach [7], which consists of creating a given number of clusters and ranking each of these clusters (using k-means [40]). The authors rank the strategies according to the average difference of scored goals by decreasing order. Each cluster has an attached strategy – the first in the ranking. The maximum number of different strategies recommended by this system is therefore limited to the number of clusters. The number of different strategies can be less than the number of clusters if there are two or more clusters with the same strategy attached. The final results proved that this approach far exceeded the eager approach. Even for the best team of the tournament (WE), the lazy approach allowed an improvement of more than 48% using a pre-defined scale, which is unprecedented in the literature.

There are two different directions for future work: use this kind of approach in other dynamical environments, evaluating whether the final results maintain the conclusions achieved in this work; incorporating a new (and more complex) set of set-plays to further improve the performance of the team.



**Table 6**

Time consuming comparison for different types of collaborative filtering in seconds (average, median).

Distance measure	CF approach	Number of NN	Cycles		
N/A	Model-based [7]	N/A	500 0.11, 0.10	1000 0.11, 0.12	2000 0.12, 0.13
$d_M/d_{WM}$	Memory-Based	5	1.69, 1.70/2.85, 2.80	1.80, 1.79/2.89, 2.91	1.99, 2.01/2.91, 2.88
	Memory-Based	10	1.69, 1.71/2.86, 2.84	1.86, 1.82/2.90, 3.05	1.95, 1.97/2.91, 2.86
	Memory-Based	15	1.71, 1.73/2.89, 2.97	1.91, 1.97/2.95, 3.01	1.99, 2.01/3.02, 3.10
	Memory-Based	20	1.77, 1.74/2.91, 3.02	1.89, 1.91/2.99, 3.06	2.02, 2.13/3.04, 3.14
$d_E/d_{WE}$	Memory-Based	5	1.66, 1.63/2.61, 2.71	1.73, 1.87/2.62, 2.78	1.79, 1.88/2.69, 2.83
	Memory-Based	10	1.67, 1.65/2.63, 2.75	1.75, 1.92/2.65, 2.84	1.87, 1.90/2.68, 2.77
	Memory-Based	15	1.67, 1.68/2.62, 2.70	1.99, 2.11/2.67, 2.72	2.72, 2.73/2.77, 2.92
	Memory-Based	20	1.67, 1.71/2.62, 2.71	2.47, 2.52/2.68, 2.71	2.93, 2.83/2.75, 2.96

CF – collaborative filtering; NN – nearest neighbor; N/A – not applicable.

## Acknowledgments

This work is part-funded by project “NORTE-07-0124-FEDER-000059” a project financed by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

This work is also partially supported by iCIS project (CENTRO-07-ST24-FEDER-002003) which is co-financed by QREN, in the scope of the Mais Centro Program and FEDER.

## References

- [1] Y. Peng, G. Wang, G. Kou, Y. Shi, An empirical study of classification algorithm evaluation for financial risk prediction, *Appl. Soft Comput.* 11 (2) (2011) 2906–2915.
- [2] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, RoboCup: the robot world cup initiative, in: *Proceedings of the International Conference on Autonomous Agents*, 1997, pp. 340–347.
- [3] P. Henriques Abreu, M. Faria, L.P. Reis, J. Garganta, Knowledge representation in soccer domain: an ontology development, in: *Proceedings of the Iberian Conference on Information Systems and Technologies*, 2010, pp. 460–465.
- [4] I. Gonzalez, P. Henriques Abreu, L.P. Reis, Using a datawarehouse to extract knowledge from RoboCup teams, in: *Proceedings of the International Conference on Enterprise and Information Systems*, 2008, pp. 511–514.
- [5] P. Henriques Abreu, J. Moura, D. Silva, L. Reis, J. Garganta, Performance analysis in soccer: a Cartesian coordinates based approach using RoboCup data, *Soft Comput. Fusion Found. Method. Appl.* 16 (1) (2012) 47–61.
- [6] P. Henriques Abreu, J. Mendes-Moreira, I. Costa, D. Castelo, L.P. Reis, J. Garganta, Human versus virtual robotics soccer: a technical analysis, *Eur. J. Sport Sci.* 12 (1) (2012) 26–35.
- [7] P. Henriques Abreu, D.C. Silva, L. Portela, J. Mendes-Moreira, L.P. Reis, Using model-based collaborative filtering techniques to recommend the expected best strategy to defeat a simulated soccer opponent, *Intell. Data Anal.* 18 (5) (2014) 21.
- [8] P. Riley, M. Veloso, Recognizing probabilistic opponent movement models, in: *RoboCup 2001: The Fifth RoboCup Competitions and Conferences*, Springer, Berlin, Heidelberg, 2002, pp. 453–458.
- [9] T. Steffens, Feature-based declarative opponent-modelling in multi-agent systems, in: *RoboCup 2003: Robot Soccer World Cup VII*, Springer, Berlin, Heidelberg, 2004, pp. 125–136.
- [10] J. Iglesias, J. Fernández, I. Villena, A. Ledezma, A. Sanchis, The winning advantage: using opponent models in robot soccer, in: *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*, Springer-Verlag, 2009, pp. 485–493.
- [11] X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, *Adv. Artif. Intell.* 2009 (2009) 19.
- [12] A. Kott, W.M. McEneaney, *Adversarial Reasoning: Computational Approaches to Reading the Opponent's Mind*, Chapman & Hall/CRC, New York, 2006.
- [13] Y. Shoham, K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press, Cambridge, 2009.
- [14] D. Carmel, S. Markovitch, Incorporating opponent models into adversary search, in: *Proceedings of the National Conference on Artificial Intelligence*, AAAI Press, Portland, Oregon, 1996, pp. 120–125.
- [15] D. Carmel, S. Markovitch, Learning models of intelligent agents, in: *Proceedings of the National Conference on Artificial Intelligence*, AAAI Press, Portland, Oregon, 1996, pp. 62–67.
- [16] C. Papadimitriou, J. Tsitsiklis, The complexity of Markov decision processes, *J. Math. Oper. Res.* 12 (3) (1987) 441–450.
- [17] S. Markovitch, R. Reger, Learning and exploiting relative weaknesses of opponent agents, *Auton. Agents Multi-agent Syst.* 10 (2) (2005) 103–130.
- [18] M.A. Janssen, T.K. Ahn, Learning, signaling, and social preferences in public-good games, *Ecol. Soc.* 11 (2) (2006) 138–160.
- [19] L. Busoniu, R. Babuska, B. De Schutter, A comprehensive survey of multiagent reinforcement learning, *IEEE Trans. Syst. Man Cybern. C: Appl. Rev.* 38 (2) (2008) 156–172.
- [20] Y. Shoham, R. Powers, T. Grenager, *Multi-Agent Reinforcement Learning: A Critical Survey*, Technical Report, Computer Science Department of Stanford University, 2003.
- [21] R. Aler, J. Valls, D. Camacho, A. Lopez, Programming Robosoccer agents by modeling human behavior, *Exp. Syst. Appl.* 36 (2) (2009) 1850–1859.
- [22] P. Stone, P. Riley, M. Veloso, Defining and using ideal teammate and opponent agent models, in: *Proceedings of the Innovative Applications of Artificial Intelligence Conference*, 2000, pp. 1040–1045.
- [23] A. Ledezma, R. Aler, A. Sanchis, D. Borrajo, Predicting opponent actions by observation, in: *RoboCup 2004: Robot Soccer World Cup VIII*, Springer, Berlin, Heidelberg, 2005, pp. 286–296.
- [24] C. Druecker, S. Huebner, H. Neumann, E. Schmidt, U. Visser, H. Weland, Virtualweder: Using the Online-Coach to Change Team Formations, Technical Report, TZI-Center for Computing Technologies, University of Bremen, 2000.
- [25] P. Riley, M. Veloso, G. Kaminka, Towards any-team coaching in adversarial domains, in: *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, 2002, pp. 1145–1146.
- [26] J.A. Iglesias, A. Ledezma, A. Sanchis, G. Kaminka, A plan classifier based on Chi-square distribution tests, *Intell. Data Anal.* 15 (2) (2011) 131–149.
- [27] J. Garganta, J. Maia, F. Basto, Analysis of goal-scoring patterns in European top level soccer teams, in: J.B. Thomas Reilly, M. Hughes (Eds.), *Science and Football III*, Taylor & Francis Group, Cardiff, 1997, pp. 246–250.
- [28] L.P. Reis, N. Lau, E. Oliveira, Playmaker: graphical definition of formations and setplays, in: *Proceedings of the Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, 2001, pp. 175–197.
- [29] L.P. Reis, R. Lopes, L. Mota, N. Lau, Playmaker: graphical definition of formations and setplays, in: *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems*, 2010, pp. 582–587.
- [30] H. Akiyama, I. Noda, Multi-agent positioning mechanism in the dynamic environment, in: *RoboCup 2007: Robot Soccer World Cup XI*, Springer, Berlin, Heidelberg, 2008, pp. 377–384.
- [31] H. Gouraud, Continuous shading of curved surfaces, *IEEE Trans. Comput.* 20 (6) (1998) 87–93.
- [32] D. Wilson, T. Martinez, Improved heterogeneous distance functions, *J. Artif. Intell. Res.* 6 (1) (1997) 1–34.
- [33] P. Henriques Abreu, D.C. Silva, J. Mendes-Moreira, L.P. Reis, J. Garganta, Using multivariate adaptive regression splines in the construction of simulated soccer teams behavior models, *Int. J. Comput. Intell. Syst.* 6 (5) (2013) 893–910.
- [34] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of ReliefF and RReliefF, *Mach. Learn. J.* 53 (1–2) (2003) 23–69.
- [35] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (2006) 1–30.
- [36] D.W. Aha, Editorial, *Artif. Intell. Rev.* 11 (1–5) (1997) 7–10.
- [37] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [38] L. Breiman, J. Friedman, R. Olshen, C. Stone, *Classification and Regression Trees*, Wadsworth & Brooks, Monterey, CA, 1984.
- [39] J.H. Friedman, Lazy decision trees, in: *Proceedings of the National Conference on Artificial Intelligence*, 1996, pp. 717–724.
- [40] A.K. Jain, Data clustering: 50 years beyond k-means, *Pattern Recognit. Lett.* 31 (8) (2010) 651–666.