Proc. of the 10th Int. Conf. on Mobile Robots and Competitions (ROBÓTICA2010), Leiria, Portugal, March 24th, 2010

Classification of FC Portugal Robotic Soccer Formations: A Comparative Study of Machine Learning Algorithms

Brígida Mónica Faria [#, & 1], Gladys Castillo*[2], Nuno Lau[#3], Luís Paulo Reis[$4]

[#] *DETI/UA – Departamento de Electrónica, Telecomunicações e Informática, Universidade de Aveiro, IEETA - Instituto de Engenharia Electrónica e Telemática de Aveiro, Aveiro, Portugal*

[&] *ESTSP/IPP – Escola Superior de Tecnologia de Saúde do Porto, Instituto Politécnico do Porto, Porto, Portugal*

[#] *DMAT/UA – Departamento de Matemática, Universidade de Aveiro, Aveiro, Portugal*

[$] *DEI/FEUP – Departamento de Engenharia Informática, Faculdade de Engenharia da Universidade do Porto, LIACC – Laboratório de Inteligência Artificial e Ciência de Computadores da Universidade do Porto, Porto Portugal*

[1]btf@estsp.ipp.pt, [2]gladys@ua.pt, [3]nunolau@ua.pt, [4]lpreis@fe.up.pt

***Abstract*— RoboCup is an international project aiming at promoting research in Artificial Intelligence and Robotics. The project includes several distinct leagues were teams composed by different types of real or simulated robots play soccer games following a set of pre-established rules. The simulated 2D league uses simulated robots encouraging research on artificial intelligence methodologies like high-level coordination and machine learning (ML) techniques. This paper presents a concrete application of several ML techniques in the identification of the opponent team and automatic detection of some of its characteristics and also on the classification of robotic soccer formations. The experimental tests performed, using three distinct datasets, enabled us to conclude that the Support Vector Machines (SVM) technique allows classifying complex data with higher accuracy than the k-Nearest Neighbor and Neural Networks. The very good accuracy revealed by the SVM technique, even with small data sets, enables the use this ML technique in real games for performing online opponent and formation classification.**

Keywords* – *Knowledge Discovery, Data Mining; Support Vector Machines; RoboCup Soccer; Simulation; Formations.

I. INTRODUCTION

RoboCup Simulation League has been one of the first competitions integrated on the RoboCup international project. The main goals of this league are concerned with developing the high-level decision and coordination modules of teams of robots [1]. The 2D simulation league has evolved over the years, but the principal architecture of the simulator is the same at it was firstly used in 1997 [1]. The *Soccerserver* is the simulator that creates a 2D virtual soccer field and the virtual players that are modeled as circles. This simulator implements the movement, stamina, kicking and refereeing models of the virtual world [2]. Another aspect that brings realism is the fact that the models in the simulator are taken both from real robots and from human like characteristics. The team of FC Portugal [2,3] has demonstrated very good results since its creation in 2000 and has won several European and World competitions [1]. The research focused development of the team is one of its main assets and still continues as every year new challenges are introduced. One concept that has been studied it is the usability of formations [4]. One important aspect is to be able to classify and predict the formations that are being used on games. Another important aspect is to identify the opponent team that FC Portugal is playing with and its characteristics in the first moments of the game. In this paper a comparative study of three techniques for classification is presented. The following techniques have been used: Support Vector Machines (SVM) [5]; Artificial Neural Network (ANN) and k-Nearest-Neighbor. The environment tool used for machine learning and data mining experiments was RapidMiner [6].

This paper is organized with an initial explanation of the RoboCup Competition with special relevance for the simulation leagues. Next, an explanation and description of the three algorithms is presented. After that the kind of measures used to compare the classifiers and the statistical hypothesis to compare the average performance of the classifiers are presented. Finally the experimental results are presented along with some conclusions and future work.

II. ROBOCUP SOCCER

RoboCup is an international cooperative project to promote Artificial Intelligence, Robotics and related fields. It is an attempt to promote artificial intelligence and robotics research by providing a standard problem where a wide range of technologies can be integrated and examined. The known goal of the RoboCup project is to create a soccer team with humanoid robots that can play and win to the world champion soccer team, by the year of 2050 [7]. In this project there are different leagues divided in two main groups: robotics and simulation. The first group involves physical robots with different sizes and different rules based on the competition that they integrate. The second one has the goal of, without the necessity to maintain any robot hardware, being able to research on artificial intelligence, coordination methodologies and team strategy. There is plenty of work performed and published concerning RoboCup [1][2][3][7] and it is interesting to conclude that this competition and its associated problems still maintain the initial activity and interest. More information about the distinct leagues can be found in [3][7], since the detailed description of these leagues is beyond the scope of this work. However a basic explanation of the simulation league is presented bellow.

A. Simulation league

There are three fields of simulation: 2D and 3D Simulation League and Mixed Reality with Eco-Be Citizen Robots. In the 2D Simulation League two teams of eleven autonomous agents (software programs) each play soccer in a two-dimensional virtual soccer stadium implemented by a central server, called *SoccerServer*. This server knows everything about the game, the current position of all players and the ball. The game further relies on the communication between the server and each agent. Each player receives from the server the relative and noisy input of his virtual sensors (visual, acoustic and physical perceptions), and may perform some basic commands like dashing, turning or kicking, in order to influence its environment [7]. Adding an extra dimension and more complex physics increases the simulation realism. In fact, the 3D Simulation League began with a spherical robot model. Nowadays it implements a humanoid model identical to the Nao Robot from Aldebaran. Finally the mixed reality was introduced in 2007 using the Eco-Be Robots from Citizen as a standard platform. Here the soccer game is on top of a virtual field with a virtual ball, using the concept of augmented reality. Fig. 1 shows examples of these three categories. The first picture is a frame of a 2D football game, the second an example of humanoids on a 3D field and the last one shows four Eco-Bes in a platform that is a soccer field with a virtual ball.

Figure 1. Three different simulation leagues (Adapted from [1])

There are many research teams from all over the world that participate in RoboCup simulation leagues. Portugal also participates in this type of leagues with the FC Portugal team (joint project from the Universities of Aveiro and Porto). In the next section a more detailed description about this Portuguese team can be found with emphasis to the 2D Simulation League.

B. FC Portugal Team

FC Portugal was created in 2000 and entered in that year at its first competition, the European championship at Amsterdam, The Netherlands. Since then FC Portugal has won five European and three World Championships [1] in several simulation leagues. There are certain unique characteristic only available in this team, such as its flexible strategy, capacity to play with different formations and dynamic positioning and role exchange. Preliminary work on knowledge discovery using RoboCup simulation league data [4] enabled to apply, to this type of data, several learning algorithms available on WEKA [8] such ZeroR, OneR, J48, Naive Bayes, k-Nearest Neighbor, PART, Multilayer Perceptron and Sequential Minimal Optimization (SMO). However this study only considered the data obtained by simplified simulation 2D games using a simplified version of the FC Portugal team as the object of study. The main simplification was the removal of the ability to perform dynamic positioning and role exchange from the team. One proposal of this work is to study the main differences between the data base inspired by [4] and a new, more challenging, data base with dynamic positioning and role exchange active. The work is also focused on the detection of the team opponent in the first moments of the game.

III. Knowledge Discovery and Data Mining

In Knowledge Discovery in Database (KDD) and Data Mining (DM) methodologies there are several phases to extract knowledge [9]. Initially it is important to identify the domain and application of DM, next it is necessary to select an appropriate set of data. When the data is chosen, a stage for performing cleaning operations and dealing with missing values is taken care. This phase is typically known as pre-processing. The transformation is another essential phase for reducing the data dimension, or choosing the attributes or even making the data discrete. Another step is the DM process that involves the definition of the task, the model and the learning algorithm. Evaluation of the model and interpretation of the results should help the next phase of decision by observing the extracted knowledge. The three machine algorithms discussed here are categorized as supervised, because the classes are known and the objective is to sign the new observation to the respective class using a specific function.

A. Support Vector Machines

Support Vector Machine is a technique based on statistical learning theory which works very well with high-dimensional data and avoids the curse of dimensionality problem [11]. The objective is to find the optimal separating hyperplane between two classes by maximizing the margin between the classes' closest points. There are several cases which should be study, one it is the linearly separable classes with class +1 and class -1. The problem can be interpreted as an optimization which

$$\min_{w} \frac{\|w\|^2}{2} \quad (1)$$

subject to the next Equation:

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 \text{ where } i = 1,2,..., N \quad (2)$$

where $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ is the hyperplane of a linear classifier which maximizes the margins and y_i represents the class, and x_i represents the input vector. The points on the boundaries are called support vectors.

There are solutions for multiclass problems and for the cases that are not linearly separable. These explanations can be found at [9][10][11]. To obtain the parameters most of the times it is used cross-validation or another scheme of experimental parameters [9][10].

B. Artificial Neural Network

Artificial Neural Network (ANN) is a mathematical/computational model that attempts to simulate the structure of biological neural systems. It consists of an interconnected group of artificial neurons and processes information using an approach of connection. In most cases an ANN is an adaptive system that changes its structure based on external or internal information that flows through the network during the learning phase. Fig. 2 shows an example of a simple ANN:

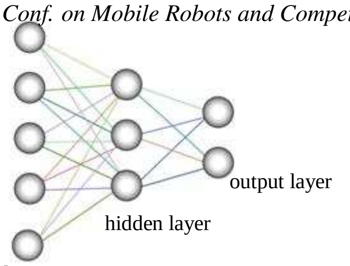

Figure 2. Simplified view of an ANN (Adapted from [12])

The neurons are typically identical units that are connected by links. The interconnections are used to send signals from one neuron to the other [13]. The concept of weights between nodes is also present since it is used for establishes the importance from one connection to the other. The network may contain several intermediary layers between its input and outputs layers. The intermediary layers called hidden layers and the nodes embedded in these layers are called hidden nodes. In a feed-forward neural network the nodes in one layer are connected only to the nodes in the next layer. The Perceptron is the simplest model since do not use any hidden layers. One of the most used models for classification using ANNs is the Multilayer Perceptron using the backpropagation algorithm in which ANNs have 3 or 4 layers. The latter was the model used in this study.The ANN model has several characteristics like the capability of handling redundant features since the weights are automatically learned during the learning phase. The weights for redundant features tend to be very small. The method called gradient descent [9] is used for learning the weights which often converge to some local minimum; however one way to overpass the local minimum is to add a momentum term [9] to the weight update formula. Another known characteristic is the consuming time for training an ANN, especially when the number of hidden nodes is large.

C. K-Nearest Neighbor

A Nearest Neighbor classifier represents each example as a data in a d-dimensional space, where d is the number of attributes. Given a test example it is computed the proximity to the rest data points in the training set, using a measure of similarity or dissimilarity, such as Euclidian measure or its generalization, the Minkowski distance metric, the Jaccard Coefficient or Cosine Similarity [9]. The k-nearest neighbor (k-NN) of a given example refers to the k points that are closest to the example. Some of the main points that characterized k-NN are the insertion in the category of lazy learners, since they do not require building a model and only make their predictions on local information. However classifying a test example is an expensive task because it is necessary to compute individually the proximity values between the test and training examples. An important decision about the proximity measure it is also necessary since the wrong choice can produce wrong predictions [9].

D. RapidMiner Environment

The RapidMiner is a software for all stages in Knowledge Discovery in Databases. It runs on every platform and operating system with the language Java, the KDD projects are modeled as trees operator which is extremely intuitive and can be saved as building blocks for later re-use. The internal XML representation ensures standardized interchange format of data mining experiments. Other interesting characteristics of RapidMiner are: simple scripting language allowing for automatic large-scale experiments; multi-layered data view concept ensuring efficient and transparent data handling. An additional property is that the machine learning library WEKA is fully integrated in RapidMiner [6].

The flexibility in using RapidMiner is another characteristic, since it has graphical user interface (GUI) for interactive prototyping; a command line mode (batch mode) for automated large-scale applications and Java application programming interface (API) to produce more programs. The initial version known as YALE (Yet Another Learning Environment) has been developed by the Artificial Intelligence Unit of University of Dortmund [6]. Today the core of RapidMiner is Open-Source and an edition for the Community is free of charge, however the Enterprise Edition needs a proprietary license. The recent version 4.5 brings more facilities like a new operator called “Script” [6] for professional analysis process design where built-in operators are not sufficient to achieve a desired task. The RapidMiner project is also characterized for giving quick responses to developer questions posted in its forum (rapid-forum [6]), since it is maintained by several full members. This reveals the activity and growing of this software allied to the attention given by the users and researchers on Data Mining.

IV. EXPERIMENTAL DEVELOPMENT

The comparative study of the three above mentioned algorithms involves the dataset produced by the positions of the players of the FC Portugal in 2D Simulation league. The performance measures are briefly described in this section together with the experimental settings and results.

A. Data set description

The dataset was produced with the x, y positions of eleven players of FC Portugal in 2D Simulation League in six distinct games with dynamic positioning and role exchange for the players. FC Portugal played two games against some known robotic soccer teams: Hellios, Brainstormers and NCL [7]. The attributes used for this study are the ball and players’ positions and the class is the formation that the team was playing with. The classification became a multi-class problem since the FC Portugal could play with ten different formations.

Table I displays the possible formations that the team could play and Fig. 3 presents an example of a formation (325).

TABLE I. FORMATIONS OF FC PORTUGAL - MULTI-CLASS PROBLEM

Classes	*One*	*Two*	*Three*	*Four*	*Five*	*Six*	*Seven*	*Eight*	*Nine*	*Ten*
Formation	**433**	**442**	**343**	**352**	**541**	**532**	**361**	**451**	**334**	**325**

Figure 3. FC Portugal team playing in 325

The coordinate x has the range of -52,5 and 52,5 and the coordinate y varies between -34,0 and 34,0 (corresponding to a typical real soccer field of 105x68m), where the center of the field is the origin of the referential [3]. The games were executed in Linux and the logs files are converted in text files with a simple application *getWState* [4] written in C++ for this purpose. The information that can be extracted from the games are the position and velocity of the ball and the eleven players of the two teams and other particular characteristics like stamina, kicks, head and body angles. In a previous work [4] it was discovered that the database with the center of mass of the FC Portugal team produces better results. Since the primordial objective of this work is to compare three different classifiers and obtain the best model, the variables corresponding to the center of mass were included on the databases. Therefore the final data set had the positions of the players, the position of the ball, the center of mass and the formation that FC Portugal was playing. Thus, the data base has 26 numerical and continuous attributes ($\mathbf{R}^{26}$) and one nominal attribute (10 formations options of FC Portugal). The first dataset (Database A) has 37943 examples with approximately 6000 cycles by game. There are differences on the number of examples since there is the possibility to have periods in the game that are stopped or others that are not counted but in which players are still moving and thus are included in the database. After an initial exploratory study it was verified that most of the samples that were incorrectly classified were near to each other and near (in time) the transition between two distinct formations. This fact revealed that the changing of the formation is not instantaneous. In fact, there is a period of time of adaptation to the new formation. In this period, players are moving from their positions in the previous formation to their positions in the new formation. Thus, although in the database the correct formation of the team is considered to be the new formation, the real formation is between the old and new formations during this transition stage. To enable performing different experiments, five cycles were removed from the transition period (where the correct formation is impossible to recognize) obtaining a second database (Database B) with 35660 examples.

B. Performance measures, comparision analysis and configuration

To be able to quantify and compare the performance of the three algorithms several measures were analyzed. Performance evaluation of a classification model is based on the counts of test records correctly and incorrectly predicted by the model. These counts are written in the confusion matrix. Although this matrix provides the information necessary to determine how well a classification model performs, it is significant to compare the performance of different models with just one single number. This can be done using accuracy (ratio between number of correct predictions and the total number of predictions) or error rate (ratio between number of wrong predictions and the total number of predictions). Besides that there are several methods for estimating the generalized error of a model during training, for finding a model with a manageable complexity and not susceptible to overfitting. The methods used for evaluate the performance of a classifier are: holdout method; random subsampling, cross-validation and bootstrap [9].

Another line of study is to answer to the question of the required dimension of the training set using different classifiers. The construction of the learning curve is an appropriate method and the procedure firstly divides the dataset into two parts and with an incremental ratio of 5% iteratively adds this percentage of examples into the training subsets and then calculates the performance values on the fixed test set. Recall and Precision are two more metrics employed for studying more precisely the distribution and prediction of classes and in cases where successful detection of one of the classes is considered more significant than detection of the others classes [9]. In other words Precision can be seen as a measure of exactness or fidelity and Recall as a measure of comprehensiveness. In this paper it is also used the 10-fold cross-validation to compare the performance of the three classifiers. The procedure initially begins with the division of the dataset into 10 equal-sized partitions. Then each classifier is applied to construct a model from 9 of the portions and test it on the remaining partition. This step is repeated 10 times, each time using a different partition as the test set. Since the objective is to compare the three classifiers it was followed the recommendation to acquire a bigger sample by implementing a scheme of 3x10 cross-validation in order to obtain a dimension sample of 30 examples and by the Central Limit Theorem perform a parametric test. The paired samples t test and ANOVA were performed with a significance level of 5%. This was compared to p value to obtain (or not) statistical evidence that the classifiers are different. By applying the t test and ANOVA the p values obtained were near to zero so less than 0,05. Thus, there are statistical evidences to affirm that the means of the results produced by the three classifiers are different. It is important to refer the settings and parameters used for applying the classifiers. The comparison of the classifiers was made using mostly the system default parameters. However in RapidMiner it is available the GridParameterOptimization operator for generating the best parameters for a particular task. This kind of study can be done in future work. Since the classification is a multiclass problem the RapidMiner incorporates the LibSVM developed by Chih Chang et Chih-Jen Lin [14] which supports multiclass learning and probability estimation based on Platt scaling for proper confidence values after applying the learned model on a classification data set. The operator supports the SVM types C-SVC and nu-SVC for classification tasks. The differences are basically over the parameters. The range of C is from zero to infinity but nu is always between [0, 1]. For experimental results the type of SVM used was C-SVC which applies one against one approach and the kernel used was RBF. The parameters introduced are registered in the table II:

TABLE II. PARAMETERS USED WITH LIBSVM

Parameters	***Values***
Kernel Type	RBF
Gamma	0
Tolerance of termination criterion	0,001
Cost parameter for C-SVC	0

To apply Artificial Neural Networks the W-Multilayer Perceptron operator, also available on WEKA, was chosen. This operator is characterized for producing a classifier that uses backpropagation to classify instances and were the network can be built by hand, created by an algorithm or both. The network can also be monitored and modified during training time. The nodes in this network are sigmoid except for when the class is numeric in which case the output nodes become unthresholded linear units. The parameters are in table III:

TABLE III. PARAMETERS USED WITH MULTILAYER PERCEPTRON

Parameters	***Values***
Learning rate for backpropagation	0,3
Momentum rate for backpropagation	0,2
Number of epochs of training	500
Hidden layers in the network	1
Hidden nodes in the network	18

The default parameter for the hidden nodes was calculated using information about the number of attributes and classes by the formula hidden nodes = [(attributes+classes)/2]. The simplest classifier Nearest Neighbors was performed with K=3 and the type of measure used was mixed Euclidean distance. In the next section the results of the comparison performed between the classifiers are discussed together with a more detailed study.

V. EXPERIMENTAL RESULTS

This section describes the results of several experimental tests. Applying support vector machine, neural network and k-nearest neighbor and compare the results conducted on the dataset with information about FC Portugal team. The experimental results were performed in RapidMiner Version 4.4 [6] in a Pentium dual-core processor T2330 (1.60 GHz, 533 MHz FSB L2 Cache) and 2 GB DDR2. The processing time was also measured and compared in the different experiences. The first experience was a simple project using the first dataset without the data pre-processing and using the method of 10-cross-validation for evaluating the accuracy. This experience was also repeated with the second dataset. The results of this experience are shown in table IV. It can be observed that the time consumed by the project that involves Neural Networks is the most expensive and the best results about accuracy are demonstrated by 3-Nearest Neighbor.

TABLE IV. ACCURACY AND TIME OF EXPERIENCE I

		Classifier		
		SVM	**NN**	**3-NN**
Data Base A	*Accuracy (%)*	95,77	80,99	99,78
	Time	48´21´´	5h18´16´´	1h7´3´´
Data Base B	*Accuracy (%)*	97,69	84,72	99,97
	Time	34´36´´	4h33´36´´	17´47´´

The next step consisted in use the second dataset and constructs the learning curves which are represented in Fig. 4.

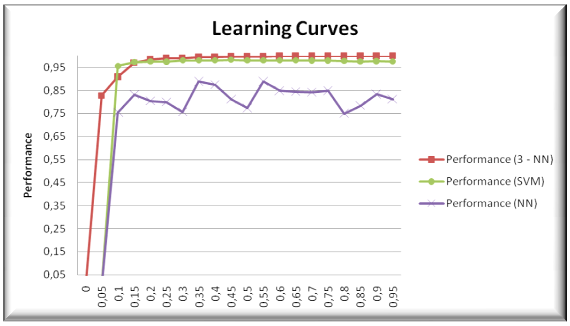

Figure 4. Relation between training set dimension and accurary

It is possible to obtain some conclusions about the learning curves. In fact with approximately 15% of the training set the accuracy is almost at their pick using SVM and 3-NN. The initial slope of the learning curve using SVM is more marked than the others two learning curves which indicates a more quickly phase of learning with a smaller training set. The ANN has a more oscillatory curve and do not pass the 90% of accuracy. This may be due to the reduced number of hidden nodes used that may limit the network learning abilities. Further experiments using distinct number of hidden nodes may be needed to further validate this approach. For testing the three classifiers with a different database and with different games were performed experimental work with a Database C with 2 games of the recent FC Portugal team. The objective of this experience is to study how well the previous classifiers (using database B as the training set) can predict the formations of FC Portugal team but in a different context of games and strategies.

TABLE V. ACCURACY AND TIME

	SVM	**NN**	**3-NN**
Accuracy (%)	50,14	50,13	45,90
Time ('')	8´14´´	40´19´´	7´55´´

In these conditions the best performance were produced by SVM with an accuracy of 50,14%. The neural networks also had a near performance with 50,13%, however using the quintuple in terms of consumed time. The analysis of the confusion matrixes reveals more information about in what FC Portugal formation the classifiers encountered more prediction difficulties. Observing the confusion matrixes the class six has a higher value of class recall on the three classifiers and the class nine represents the lowest results in recall and precision. Another particular outcome is the distribution of the predicting class when the true class is 361, 334 or 325. This reveals some difficult in distinguish these formations among others. The worst results are verified near the consecutive class with several examples incorrectly classified between the consecutive classes (for example, several examples of class 1 are incorrectly classified as being of class 2). The main explanation for this result seems to be related with the simple cleaning process applied to the database. In fact, only five cycles in the class transition were cleaned but analyzing the games afterwards lead us to the conclusion that the transition process may take about 20 or 30 cycles while players move from the old formation position to the new formation position. Thus, several examples in the transition between these classes were used to train the classifier but were incorrectly classified. This may lead, not only to the incorrect classification of the examples near the transition but also to the incorrect classification of examples in the rest of the class (since incorrect examples of that class were used in the training process). Thus, perhaps a more thoughtful cleaning process of the database should be performed. This process should be conducted taking in consideration the opinion of experts that may correctly classify the example near the transitions between formations, stating the exact point where the formation has completely changed.

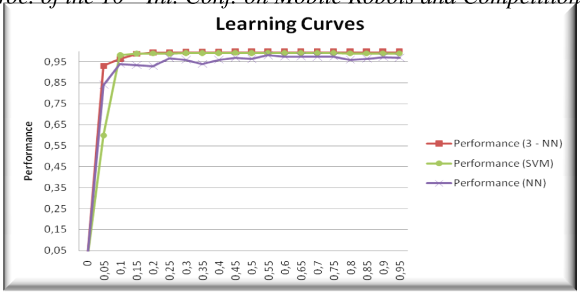

Figure 5. Relation between training set dimension and accurary for the Opponent detection experiment

A final experiment was performed to detect the opponent team playing against FC Portugal and its formation. The first results achieved were quite surprising since, using one game with each team as training set, it was very easy to correctly classify the other games. A very simple algorithm such as K-NN could achieve almost 100% of examples correctly classified in the first 20 cycles of the testing game. However, the justification for this result is also easy. The positions for the opponent players are typically the same, in each game, before the start of the game due to the fact that the players are directly positioned on their starting position. Thus, the test set composed by the start of the game is not very good to perform this analysis since it is very easy to classify. A test set taken from the middle of the game would be more interesting to use for this experiment. One should take in consideration that the final objective is to identify the team but also its characteristics that may change throughout the game. Thus a test set taken in the middle of each of the test games (cycles 500-3000 and cycles 3500-6000) was used for testing the classifier. The results achieved may be analyzed on Fig. 5. Analyzing the figure it is easy to see that the type of results achieved is similar to the results achieved for the formation classification task. However, the accuracy achieved is superior for this task than for the formation classification task. This seems to be due to the fact that in this task we were only classifying the examples in three classes and the three opponent teams use very different formations, consequently being very easy to identify. For other teams that use the same base code, and thus the same global formation this may be a lot more difficult.

VI. Conclusions and Future Work

The comparative study of SVM, NN and k-NN was conducted by three ways. The first was analyzing theoretical concepts behind the machine learning algorithms and the inspiration that supports each of the techniques. The k-NN is a simple technique which looks around the most similar examples to classify the newest example. The NN that is inspired in the properties of biological neurons. Finally the SVM has an interesting geometrical interpretation in order to find the best decision boundary. The experimental results achieved enable us to compare the classifiers models in study. Experiences were performed in order to obtain results about predictions of formations of FC Portugal team and their opponent teams. The conclusions obtained revealed that if a model is trained with certain games the 3-NN shows better results in predicting formations with those games. However in reality the training games are not the same from what we wish to predict. So, when is applied another test set with a different data set of games the results produced by SVM are in terms of accuracy the best. Another important point to be referred is the time consumed to implement and finalize the projects in RapidMiner. Applying cross-validation for all schemes the NN is the most expensive in terms of computational time consumed.

For future work it is important to test and apply other algorithms and test several others measures in order to confirm the results here obtained. Also, a more detailed cleaning in the instances should be made using the opinion of experts. In fact, in the tested datasets, there are some cycles that are similar to others because of game stopped times. Also the changing of formations periods should be taken in consideration more carefully. Finally further tests are needed in the opponent classification task, using a higher number of different opponents that use similar strategies and similar formations.

Acknowledgments

The authors would like to acknowledge to FCT – Portuguese Science and Technology Foundation (PhD Scholarship FCT/ SFRH / BD / 44541 / 2008), DETI/UA – Departamento de Electrónica, Telecomunicações e Informática and ESTSP/IPP – Escola Superior de Tecnologia de Saúde do Porto - IPP. This work was partially supported by the Portuguese National Foundation for Science and Technology FCT-PTDC/EIA/70695/2006 Project - “ACORD: Adaptive Coordination of Robot Teams”.

References

[1] Lau, N.; Reis, L. P., “FC Portugal-High-level Coordination Methodologies in Soccer Robotics”, Robotic Soccer, Book edited by Pedro Lima, Itech Education and Publishing, Vienna, Austria, pp. 167-192, December 2007, ISBN 978-3-902613-21-9.

[2] Reis, L. P. and Lau, N., “FC Portugal Team Description: RoboCup 2000 Simulation League Champion”, in Peter Stone, Tucker Balch and Gerhard Kraetzschmar, editors, RoboCup-2000: Robot Soccer World Cup IV, Springer LNAI, Vol. 2019, pp.29-40, Berlin, 2001, ISBN 3-540-42185-8.

[3] Reis, L. P.; Lau, N; Oliveira, E., “Situation Based Strategic Positioning for Coordinating a Team of Homogeneous Agents in Markus Hannebauer”, Jan Wendler and Enrico Pagello Editors, Balancing Reactivity and Social Deliberation in Multi-Agent System – From RoboCup to Real-World Applications, Springer LNAI, Vol. 2103, pp. 175-197, Berlin, 2001, ISBN 3-540-42327-3.

[4] Almeida, R.; Reis, L. P.; Jorge, A. M., “Analysis and Forecast of Team Formation in the Simulated Robotic Soccer Domain”. in L.S.Lopes, et. al. editors, 14th Portuguese Science Conf.. on Artificial Intelligence, EPIA'2009, Aveiro, LNAI 5816, Springer, pp. 239-250, October 12-15, 2009

[5] Vapnick, V., “The Nature of Statistical Learning Theory”. Springer-Verlag, 1995. ISBN 0-387-98780-0.

[6] Rapid|Miner, “RapidMiner Report the Future,” Available at [http://rapid-i.com/content/blogcategory/38/69/]. Consulted April 2009.

[7] Robocup, “Robocup 2009”, Available in [http://www.robocup2009.org/]. Consulted on December 2009.

[8] Eibe, Frank; Mark, Hall; Len, Trigg, “Weka 3: Data Mining Software in Java,” Available at [http://www.cs.waikato.ac.nz/~ml/weka/index.html]. Consulted January 2010.

[9] Tan, P.; Steinback, M.; Kumar, V., “Introduction to Data Mining”, Pearson Education, Inc, 2006. ISBN: 0321321367.

[10] Bennet, K.; Campbell, C., “Support Vector Machines: Hype or Halleluja?”, SIGKDD Explorations, Volume 2, Issue 2, ACM SIGKDD, December 2000.

[11] Anguita, D.; Boni, A.; Ridella, S.; Rivieccio, F.; Sterpi, D., “Theoretic and Pratical Model Selection Methods for Support Vector Classifiers”, Springer Verlag, Studies in Fuzziness and Soft Computing, Vol. 177, pp.159-179, 2005

[12] Kunzle, P., “Vehicle Control with Neural Networks”, Available at [http://www.gamedev.net/reference/programming/features/vehiclenn/]. Consulted on January 2010.

[13] Nimeh, S. A., Nappa, D., Wang, X., Nair, S., “A Comparison of Machine Learning Techniques for Phishing Detection”, APWG eCrime Researchers Summit, October 4-5, 2007, Pittsburgh, PA, USA.

[14] Chang, C., Lin, C., “LIBSVM : a library for support vector machines”, 2001. Software available at [http://www.csie.ntu.edu.tw/~cjlin/libsvm] Consulted on January 2010.