Finding roles of players in football using automatic PSO-Clustering algorithm

Iman Behravan: Department of Electrical Engineering, PhD student, University of Birjand, i.behravan@birjand.ac.ir

Seyed Hamid Zahiri: Department of Electrical Engineering, Faculty of Engineering, University of Birjand, hzahiri@birjand.ac.ir

Seyed Mohammad Razavi: Department of Electrical Engineering, Faculty of Engineering, University of Birjand, smrazavi@birjand.ac.ir

Roberto Trasarti: KDD lab, ISTI-CNR, Pisa, Italy, roberto.trasarti@isti.cnr.it

ABSTRACT

Recently professional team sport companies invest their resources to analyze their own and opponents' performance. So developing methods and algorithms for analyzing team sports has become one of the most popular topics among the data scientists. Analyzing football is hard because of the complexity of the game, number of events in each match and constant flow of circulating the ball. Finding roles of players in order to analyze the performance of a team or make a meaningful comparison between players is crucial. In this paper an automatic big data clustering method, based on a swarm intelligence algorithm, is proposed to automatically cluster the dataset of players' performance centers in different matches and extract different kinds of roles in football. The proposed method which is created using particle swarm optimization algorithm has 2 phases. In the first phase the algorithm searches the solution space to find the number of clusters and in the second phase it finds the positions of centroids. To show the effectiveness of the algorithm it is tested on 6 synthetic datasets and its performance is compared with two other conventional clustering methods. In the next step the algorithm is used to find clusters of a dataset containing 93000 objects which are the centers of players' performance in about 93000 matches in different European leagues.

KEY WORDS: big data clustering, particle swarm optimization, swarm intelligence, football analysis

1. INTRODUCTION

Automatic analysis of team sports such as soccer, baseball, basketball and etc. has become one of the interesting topics and recently professional team sport companies invest their resources to analyze their own team performance as well as the performance of their opponents in a league. Analyzing complex data generated by (semi-) automated technologies such as soccer logs [1, 2] provides valuable knowledge about the players, their performance and their roles for the coaches, sport analyzers, commentators and etc. Also these datasets can help in designing tactics and strategies for each match in a league. Analyzing football (soccer in USA), the most popular sport in the world, is difficult due to its complex nature and constant flow of circulating the ball during a match. Available big datasets generated by different companies and websites, like whoscored.com, provide good opportunities for data scientists to extract useful information from

these datasets and analyze football matches. These datasets contain the position of different events in the field (e.g., tackles, passes, shots, crosses, interceptions and etc.), generated by players, and their corresponding time stamps. Data mining and big data mining methods can be used to analyze and process these kind of datasets in order to find useful information about the performance of the players in each match and their influence on the final result of the match. Also problem of rating the performance of the soccer players can be solved using these methods. Clustering, the process of grouping data points in to different clusters based on their similarities, is widely used in different fields like biology, ecology, social science, marketing and sports analytics [3-6]. Many algorithms have been introduced for clustering such as *kmeans* [7], *fuzzy c-means* [8] and X*means* [9]. Clustering big datasets is beyond the ability of these conventional methods. So searching for efficient and accurate big data clustering algorithms is very important. Many real-world applications and problems can be formulized as optimization problems and solved using optimization algorithms. Heuristic optimization algorithms which can solve complex and discrete optimization problems, have become very popular among researchers. Swarm intelligence and evolutionary algorithms are two kinds of heuristics which basically search the solution space of the problem by a population of individuals while each individual is a potential solution of the problem. *Particle Swarm Optimization* [10] *Inclined Planes system Optimization* [11] *Ant Colony Optimization* [12] and *Grey Wolf Optimization* [13] are some of the well-known swarm intelligence algorithms and *Genetic algorithm* [14] is the most famous evolutionary algorithm. In this paper an effective clustering method, based on a swarm intelligence algorithm called Particle Swarm Optimization (PSO), is introduced. The proposed method is used to cluster a massive football dataset in order to find different players' roles in a football match. The paper is organized in the following manner: section 2 is a background about using data science in football analytics and a brief introduction about PSO. In section 3 the proposed method is completely explained. Section 4 contains the simulation results on synthetic datasets. Sections 5 and 6 are devoted to data description and extracting players' roles respectively and final section is conclusion and future works.

2. RELATED WORKS

In [15] Duch et al. introduced a metric (flow centrality) to evaluate performance of the players. Their metric is based on the passes result in shots. In fact, this metric is suitable to evaluate midfielders and forwards not defenders because defenders usually try to stop opponent's attacks. They used their methodology to rank players in Euro Cup 2008 and they reported that 8 players in the top 20 players found by their methodology were also in the list of top 20 players selected by UEFA. Brooks et al. [16] tried to rank players in La Liga based on a so called PSV (Pass Shots Value) metric. They used this metric to find important passes made by the players. In this metric passes result in a shot (considering their origin and destination) are considered as important passes. Again the drawback of this method is ignoring the other events in the game and ranking the players based on the passes made by them. In a more comprehensive research Pappalardo et al. [17] proposed a data-driven framework called PlayeRank to rank the players. They used data collected from the matches in four seasons of five prominent European leagues. In this framework which consists of five steps they considered the whole events in a match (passes, shots, tackles, free kicks and etc.) which makes their results more reliable and meaningful also for defenders. Actually, they

overcame previous works' drawbacks by considering the role of players. In another kind of analytics Stanojevic and Gyarmati [18] tried to find relation between the performance of the players and their market value. In another research, Muller and his colleagues introduced an approach to estimate value of players in transfer market [19]. In this research a new parameter free clustering algorithm is used to solve the problem of finding roles of players in a football match.

2.1. PARTICLE SWARM OPTIMIZATION ALGORITHM

Particle swarm optimization algorithm [10], which is suggested by Kennedy and Eberhard in 1995, is an iterative algorithm generated by mathematically modeling the behavior of different species in group like birds' flocking. This algorithm uses a population of individuals, called particles, to search the solution space. In fact, each individual or particle is a potential solution of the optimization problem. These particles search different areas of the solution space to find the optimum solution by cooperating with each other. Each particle has a position vector (the potential solution), velocity vector, which shows the direction of its movement, and a memory to save its best position from the beginning to the current iteration. In each iteration, particles change their position using the following equations:

$$v_{id}^{t+1}=\mathrm{w}.v_{id}^{t}+c_1.rand.(p_{best}^{d}-x_{id}^{t})+c_2.rand(p_{gbest}^{d}-x_{id}^{t}) \quad (1)$$

$$x_{id}^{t+1}=x_{id}^{t}+v_{id}^{t} \quad (2)$$

where, v_{id} is the *d*th dimension of the velocity of the *i*th particle, *x* denotes the position of the particle, *t* is the number of iteration, c_1 and c_2 are learning factors, *rand* is a positive random number between 0 and 1 under normal distribution, *w* is the inertia weight coefficient, p_{best} is the best position of the particle from the beginning to current iteration and p_{gbest} shows the position of the best particle in each iteration which has the best fitness amount in the population. According to these equations, particles move toward the position of the best particle in the population (p_{gbest}) if c_2 is high and c_1 is low. But if c_1 is high and c_2 is low, particles search around their best positions observed from the beginning to the current iteration (p_{best}). So choosing optimal amount for these two factors, designed to control the exploring and exploiting ability of the algorithm, is necessary. Based on different literatures [20, 21] fixing them at 2 will result in a good convergence but for complex problems this may cause premature convergence. To escape from local optimum point and search the solution space thoroughly, in our algorithm the amount of C_1 and C_2 are dynamically changed during the process. In the beginning iterations C_1 is high and C_2 is low and in the last iterations vice versa. Their amounts are changed exponentially during the process.

3. PROPOSED METHOD

3.1. PSO-CLUSTERING ALGORITHM

Solving an optimization problem requires suitable encoding of the search agents (particles in PSO) and designing a fitness function for the problem. Our algorithm, which is designed to solve clustering problem, searches the solution space to find the best possible centroids. So particles should contain the position of *k* centroids. This means that each particle is an array containing $k \times p$ elements where *k* is the number of clusters and p is the number of features in the dataset. In

figure 1 a particle with *n* centroids for a 2-D dataset is demonstrated. In this figure C_{ij} is *j*th dimension of the *i*th centroid.

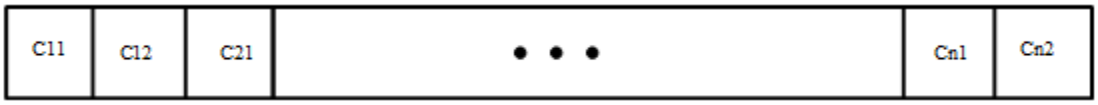

Figure 1- A particle containing n centroids for a 2-D dataset

To find the best set of centroids or in other words to find the best particle (possible solution) a function should be used to measure the quality of particles and since each particle is a possible solution of the clustering problem, this function should measure the quality of the partitioning proposed by each particle. Several indexes have been introduced up to now to measure the quality of the partitioning like *Silhouette index* [22], *Davies-Bouldin index* [23], *Dunn index* [24], *Calinski-Harabasz index* [25] and etc. In this research *Calinski-Harabasz index* has been used to evaluate the quality of each particle due to its low complexity in compare to *Silhouette index* and its effectiveness in finding the number of clusters. This index measures the quality using the following equations:

$$VRC_k = \frac{SS_B}{SS_W} \times \frac{(N-k)}{k-1} \tag{3}$$

$$SS_B = \sum_{i=1}^{k} n_i \|m_i - m\|^2 \tag{4}$$

$$SS_W = \sum_{i=1}^{k} \sum_{x \epsilon c_i} \|x - m_i\|^2 \tag{5}$$

In these equations, SS_B is the overall between-cluster variance, SS_W is the overall within-cluster variance, *k* is the number of clusters, *N* is the number of data points, m_i is the centroid of the *i*th cluster, m is the overall mean of the sample data, x is a data point, c_i is the *i*th cluster and $\|m_i - m\|$is the Euclidean distance between two vectors. Higher VRC means better partitioning. So in the minimization case, finding the minimum point of $f = \frac{1}{VRC}$, results in finding the best solution. The first step of the algorithm is generating a random population. In this step the particles should be positioned randomly in the solution space. For this purpose, instead of generating random numbers for the position of the particles, for each particle *k* samples from the dataset are randomly selected and their positions are considered as the centroids of the particle. In the next step each particle is evaluated using the *Calinski-Harabasz index*. After that the particles move in the space and change their positions using equations 1 and 2. Then for each particle a subset of samples is selected randomly from the dataset and each centroid is replaced with the position of the closest sample of this subset. Again evaluating the particle is performed in the next step. This procedure is repeated until the last iteration. Different steps of the algorithms are indicated in figure 2.

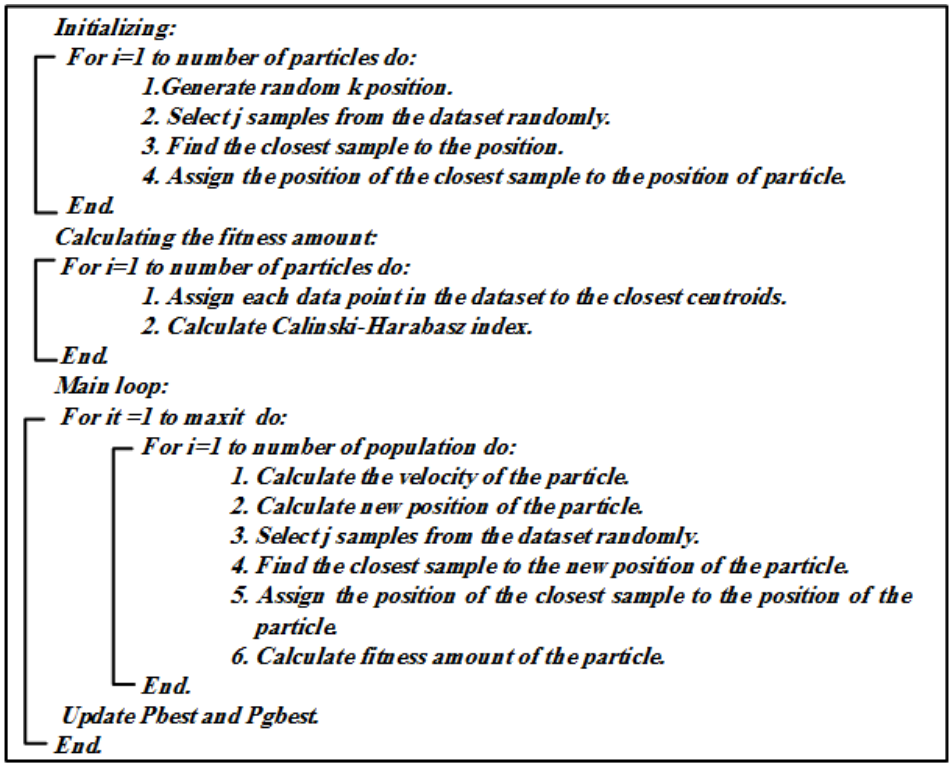

Figure 2- The procedure of PSO-Clustering algorithm

3.2. AUTOMATIC PSO-CLUSTERING ALGORITHM

One of the main drawbacks of conventional clustering methods such as *kmeans*, is their limitation in finding the number of clusters. Actually finding the number of clusters accurately is really necessary specially when dealing with big datasets. Although a version of *kmeans* (called *Xmeans*) is introduced which has the ability of finding the number of clusters, but its accuracy is not high enough which is also observed in our experiments. To overcome this limitation and make our algorithm able in finding the number of clusters accurately, another phase has been added to our algorithm to search the solution space for proper number of clusters. Generally, our automatic clustering algorithm consists of two phases. The first phase is designed to search for the proper number of clusters and in the second phase algorithm tries to find the best centroids' positions. In the first phase, a tree structure of PSO-Clustering algorithm is run. In the first node of the tree an integer random number for k is selected and PSO-Clustering is run to find k centroids. In the second node the algorithm is run with a new k which is the result of the following equation:

$$k_{new} = k_{old} \pm \varepsilon \tag{6}$$

Where ε is a random integer number. The best result, including k and the fitness amount of the best particle found by the algorithm, will be saved. Then from the third node to the end of the tree this procedure continues and the result of each node is compared to the best result found in the previous nodes. If the result (fitness amount) is better than the best result, then k will be updated. m trees are run in the first phase and for each tree a random population is generated and used for all of the nodes. In each node PSO-Clustering algorithm is run with low number of population

members and low number of iterations. Finally, in the second phase, PSO-Clustering algorithm is run with the best *k*, found in the previous phase, to find the position of the centroids. The pseudo code of the automatic PSO-Clustering algorithm is shown in figure 3.

```
Stage 1:
Inputs: number of trees, length of tree, number of population
For i=1 to number of trees do:
        1.k = Generate a random integer number
        2. beginning population = Generate a random population
        3.Current state.fitness = inf
        4. Current state.k=Ø
        for j=1 to length of tree do:
            if (j==1) do:
                current state.k=k
            end
            best fitness = PSOclustering(k , beginning population)
            if best fitness < current state.fitness
                current state.fitness=best fitness
                curent state.k=k
            end
            k = current state.k ± ε
        end
Bestk[i] = current state.k
End
Output: find the best solution which has the best fitness amount and its corresponding k
and its corresponding beginning population
Stage 2:
Final output = PSOclustering(best k, beginning population)
```

Figure 3- Pseudo code of the automatic PSO-Clustering algorithm

To evaluate the accuracy of our algorithm both in finding the number of clusters and in finding the position of clusters, our automatic PSO-Clustering algorithm is tested on 6 synthetic datasets. The corresponding results are reported in the next section.

4. SIMULATIONS AND EXPERIMENTAL RESULTS ON SYNTHETIC DATASETS

The proposed method is tested on 6 synthetic datasets [26] and its accuracy is calculated by measuring Rand index [27] and Normalized Mutual Information (NMI) index [28] to evaluate its effectiveness. Table 1 contains the characteristics of these datasets. Also our algorithm is compared with *kmeans* and *Xmeans*. Tables 2 to 4 demonstrate the average results for S, A and high dimensional datasets respectively, after several experiments. According to these tables, not only the accuracy of APSO-Clustering algorithm, in finding the number of clusters, is high, but also the NMI index is remarkable. This index shows the similarity of the shapes of the clusters found by clustering algorithm to the shapes of real clusters. In fact, just finding the number of clusters

accurately doesn't mean that the clusters are found correctly. High amount of NMI shows that the detected clusters are similar to the real clusters and the algorithm has clustered the data precisely. So higher NMI and accurate number of clusters in addition to higher rand index indicate the superiority of APSO-Clustering algorithm over the other two conventional clustering algorithms. Actually, these tables demonstrate the power of the proposed method in clustering datasets containing several clusters and also high dimensional datasets. In figures 4 to 6 the centroids found by the APSO-Clustering algorithm for one experiment on S1, A1 and dim064 are shown respectively. In these figures the blue circles are the data points and the grey stars are the centroids. In figure 5 the red star is the extra centroid found by the algorithm. Also in figures 7 to 9 centroids found by *Xmeans* algorithm are shown. These figures show that our algorithm has found the clusters accurately while, according to figures 7 and 8, *Xmeans* has failed to cluster the dataset properly. Figures 7 and 8 clearly show that *Xmeans* has merged close clusters and put their samples in one cluster while figures 4 and 5 indicate that all of the clusters are accurately found by APSO-Clustering algorithm.

Table 1- characteristics of the synthetic datasets

	Dataset	*Number of data points*	*Number of features*	*Number of clusters*
S datasets	S1	5000	2	15
	S2			
A datasets	A1	3000	2	20
	A2	5250	2	35
High dimensional datasets	Dim064	1024	64	16
	Dim1024	1024	1024	16

Table 2- Results for S datasets

Method	*Rand index*		*NMI*		*Number of detected clusters*	
	S1	*S2*	*S1*	*S2*	*S1*	*S2*
APSO-Clustering	0.9959	0.9911	0.9710	0.9413	16.25	15.75
Xmeans	0.9225	0.9353	0.8341	0.8091	8	9
kmeans	0.9901	0.9777	0.9489	0.8878	-	-

Table 3- Results for A datasets

Method	*Rand index*		*NMI*		*Number of detected clusters*	
	A1	*A2*	*A1*	*A2*	*A1*	*A2*
APSO-Clustering	0.9981	0.9975	0.9897	0.9752	20.25	38.33
Xmeans	0.862	0.9104	0.7208	0.7482	6	9
kmeans	0.9877	0.9924	0.9574	0.9586	-	-

Table 4-Results for high dimensional datasets

Method	*Rand index*		*NMI*		*Number of detected clusters*	
	Dim064	*Dim1024*	*Dim064*	*Dim1024*	*Dim064*	*Dim1024*
APSO-Clustering	0.999	0.999	0.9987	0.9988	17	17
Xmeans	0.9844	0.9922	0.9682	0.9843	14	15
kmeans	0.9984	1	0.9966	1	-	-

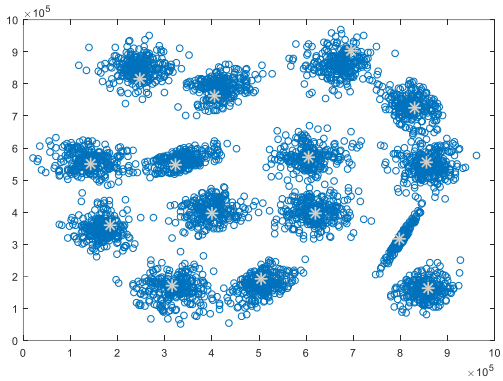

Figure 4- Centroids of S1 dataset found by APSO-Clustering algorithm

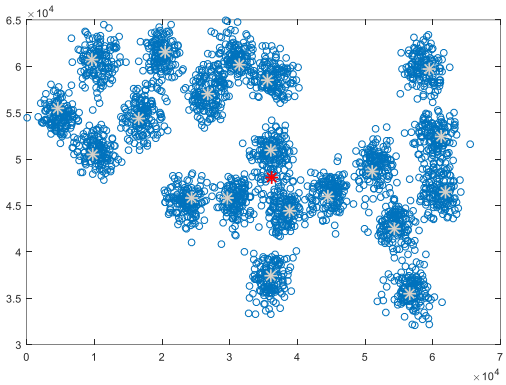

Figure 5- Centroids of A1 dataset found by APSO-Clustering algorithm

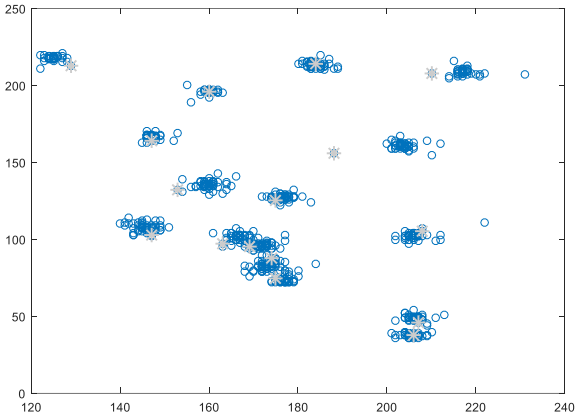

Figure 6- Centroids of dim064 dataset found by APSO-Clustering algorithm

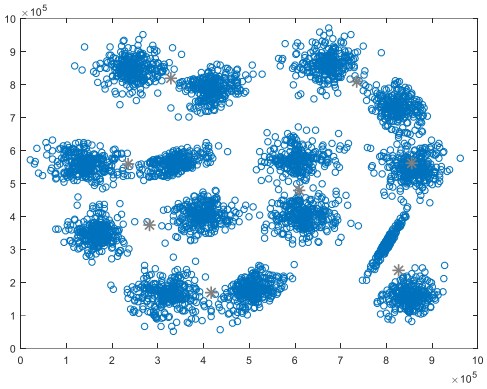

Figure 7- Centroids of S1 dataset found by *Xmeans* algorithm

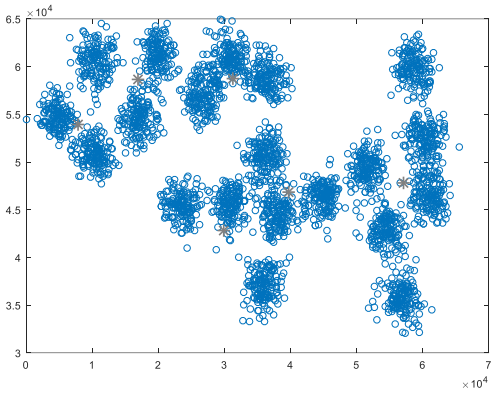

Figure 8- Centroids of A1 dataset found by *Xmeans* algorithm

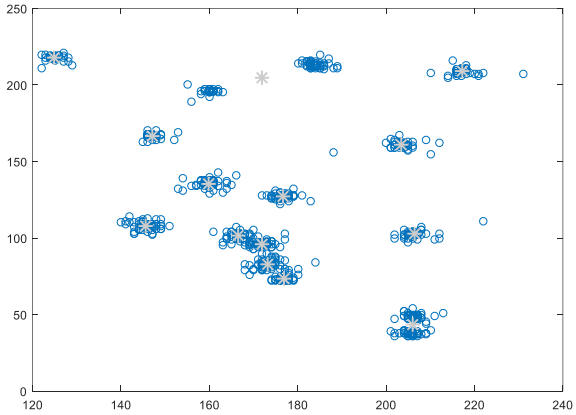

Figure 9- Centroids of dim064 dataset found by *Xmeans* algorithm

These experiments are performed to evaluate the performance of the proposed algorithm on different datasets (not necessary big datasets) with different characteristics. According to the tables and figures, APSO-Clustering's performance on high-dimensional datasets and datasets with high number of clusters is remarkable. In the next section the result of testing the algorithm on a real big dataset is reported.

5. DATA DESCRIPTION

The data which is used in this research is extracted from whoscored.com website. This dataset contains 93000 objects which are related to the areas different players touched the ball in almost 93000 football matches. Actually in a football match each player, based on his role, touches the ball in different areas of the field several times and each object in this dataset is the center of the area in which a specified player touched the ball in a specified match. The dataset contains the centers of the areas in which 2005 different players touched the ball in 93000 matches.

6. EXTRACTING PLAYERS' ROLES

Different players have different tasks based on their roles in a football match. For example, a defender should stop the opponent's attacks and prevent them to score while a forward should try to score and use the opportunities provided for him/her. Usually a defender makes more tackles and interceptions than a forward while a forward makes more shots and passes in a game. So types of events created by different players in a match are correlated to their roles. This means that comparing players without considering their roles is meaningless. Therefore, it is necessary to use an accurate method for extracting different roles of players. APSO-Clustering algorithm found 8 different clusters which indicate 8 different roles in a football match. Figure 10 shows the output of the algorithm including samples of the dataset (blue circles) and detected centroids (grey stars). Also in figure 11 the equivalent role for each cluster is shown. Table 5 contains the name of each role. These names are selected from Pappalardo and his colleagues' paper [17].

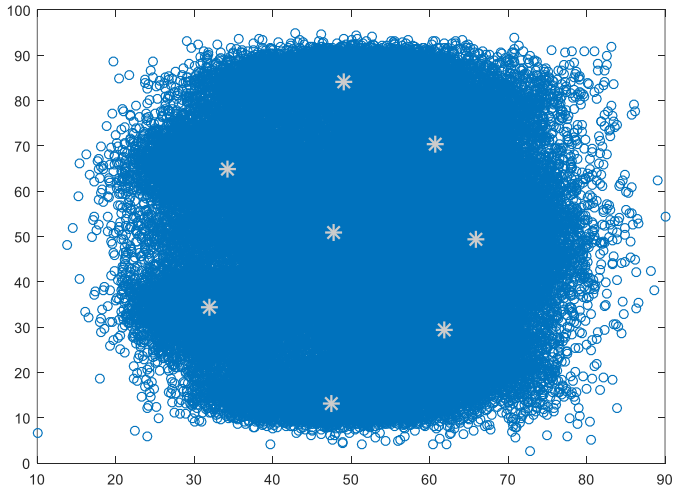

Figure 10- output of APSO-Clustering algorithm for football data

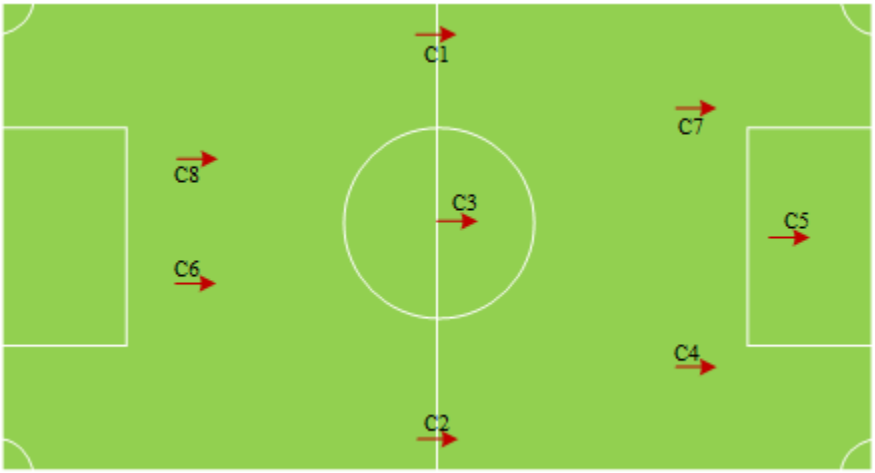

Figure 11- Eight different roles for football players

Table 5- Name of 8 different roles in football

Role	*name*
C1	Left fielder, left wing, left back
C2	Right fielder, right wing, right back
C3	Central midfielder, internal midfielder
C4	Right wing, right forward
C5	Central forward
C6	Right back
C7	Left wing, left forward
C8	Left back, left central back

Although each player has a specified role in each match and his tasks are defined based on the role, he sometimes touches the ball in other areas of the field or creates some events far from his main area. This really depends on the flow of the game. For example, for a team losing the match and needing just one goal to equalize, sometimes also defenders participate in attacks to score. Also some players have the ability to play in more than one area and more than one role in different matches or even in one match. This means that although each player, depending on his skills, is asked to play in a specified area, but no one expects not to see him in the other areas of the field. Extracting frequency of touching the ball, participating in challenges or creating events in different areas are important for the coaches to find movement patterns of the players and the team, design effective strategies for each match, select the best players based on the designed strategies and make the team ready for every situation. To analyze better, for each player in the dataset, we extracted the normalized frequency of touching the ball in 8 different areas. So we extracted a new dataset including 2005 objects and 8 features showing the frequency of touching the ball by the players in the corresponding area in different matches. In the next step again we clustered this dataset using our APSO-Clustering algorithm. The algorithm found 8 centroids which are demonstrated in table 6. Each centroid contains 8 elements showing touching frequency of the ball by different players in different areas of the field. In each centroid there is one dominant element (bigger than 0.5) indicating that most of the players belong to that cluster, play in that corresponding area. For example, the players belong to the first cluster, touches the ball (on average) 78 percent in the left side of the field (C1). Actually they mostly play as left fielder, left wing or left back (according to table 5). Also 15 percent of the events created by these players happened in C8 area where they had to play as a left defender. In other words, these players' main role is left fielder or left wing and depending on the situation of the match they also played the role of a left defender (or left back). Actually they have participated in defense while their main role is attacking and creating opportunities for the forwards. Figure 12 shows this concept. Table 7 shows the performance of three players of the first cluster. This table indicates how many times each of these players touched the ball in different areas.

Table 6- Eight centroids found from ball touching frequency data

roles / centroids	*C1*	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*	*C8*
1	0.788	0.022	0	0.011	0.022	0	0	0.155
2	0.176	0	0	0	0.117	0.117	0	0.588
3	0	0	0.914	0	0.085	0	0	0
4	0.058	0	0.176	0	0.588	0	0	0.176
5	0	0.040	0	0.163	0	0	0.795	0
6	0	0.782	0	0.102	0	0	0.115	0
7	0	0	0	0.698	0	0	0.301	0
8	0	0	0.047	0.023	0	0.88	0.023	0.023

Table 7- Frequency of touching the ball in different areas by three players of the first cluster

Player	*C1*	*C2*	*C3*	*C4*	*C5*	*C6*	*C7*	*C8*
1	69	13	0	0	2	0	0	13
2	29	2	0	0	1	0	0	1
3	16	5	1	0	2	3	0	10

According to table 7, the center of performance of the first player was C1 in 69 matches, C2 and C8 in 13 matches and C5 in 2 matches. Figure 13 shows the box plot of the players belong to the first cluster. This figure emphasizes that although some of the players of this cluster have played in C2, C5 and C8 areas but these players often played in C1 area as left fielder. The second centroid shows similar information. This cluster contains players who have touched the ball in C8 area 58 percent of the times. Also 17 percent of the events, they created, were in C1 area. This shows that these players often play as defenders while they have played in C1 and C5 areas 17 percent and 11 percent respectively. Their roles are shown in figure 14 and the corresponding box plot is shown in figure 15. According to this figure the distribution of the players who have played in C8 area is higher than other areas and after that C1 is the most frequent area in which they have played. Also these players have played in C3, C5 and C6 areas in some matches. Figure 16 shows the roles of players mostly played as central or internal midfielder. 91 percent of their events are created in C3 area and in 8 percent of the times they touched the ball in C5 area. These players usually try to send passes to the forwards or the wingers and make good opportunities for them or participate in defense and help the defenders. Their play is focused on the central part of the field. Figure 17 shows distribution of touching the ball in different areas by these players. This figure clearly shows the stability of the central midfielders. In compare to the other roles they are the most stable players. Probably they are not good choices for the other roles or other players don't have the ability to play in their area and perform their tasks successfully. In other words, this analysis shows the importance of the role of central fielders. Usually they don't change their position and their role is really important both in defending and attacking. Similar information is shown in figures 18 to 27 for the other clusters. Figures 26 and 27 show the importance of players playing in C6 area. According to figure 27, most of the players in cluster 8, are right back defenders. Usually they don't play in the other areas, on the other hand just other kind of defenders, who have mostly played in C8 area, have played their role in some matches. This means that these players have a

key role in defending and stopping opponent's attacks. Also analyzing figures 20, 21, 24 and 25, we can infer that wingers are most important players in attack. The box plots clearly show that they usually play in their own position. According to these figures, right wingers have played the role of left wingers in 30 percent of the matches and on the other hand, left wingers have played the role of right wingers in 16 percent of the matches. These statistics indicate that wingers (left or right) can change the side of their play, but it is not common to see them playing another role.

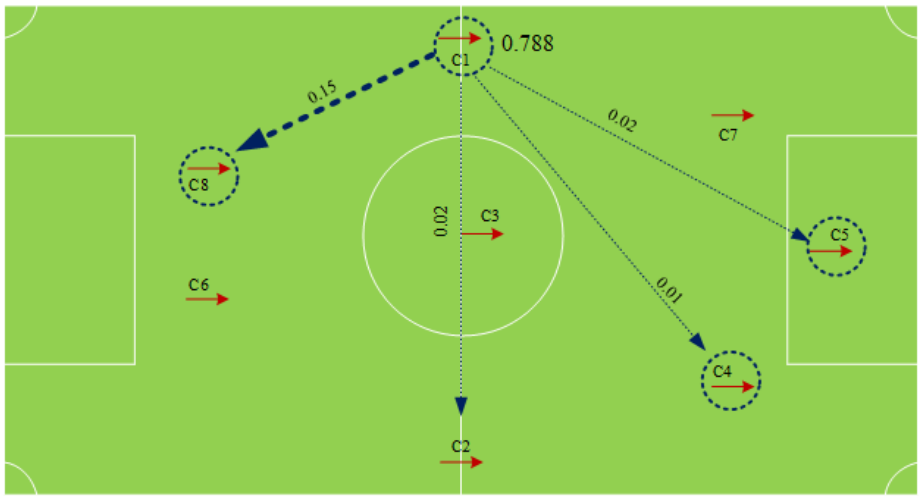

Figure 12- Different areas of touching the ball by left fielders, left wingers or left backs

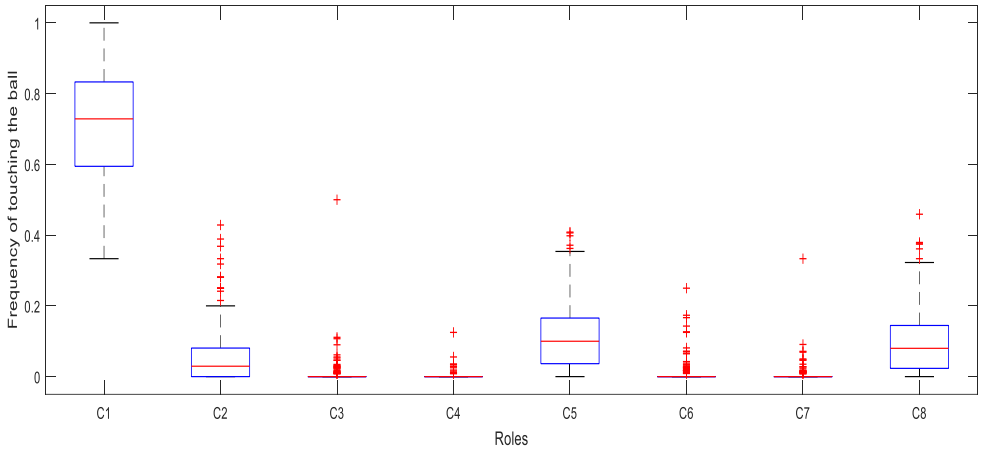

Figure 13- Box plot of the players belong to the first cluster

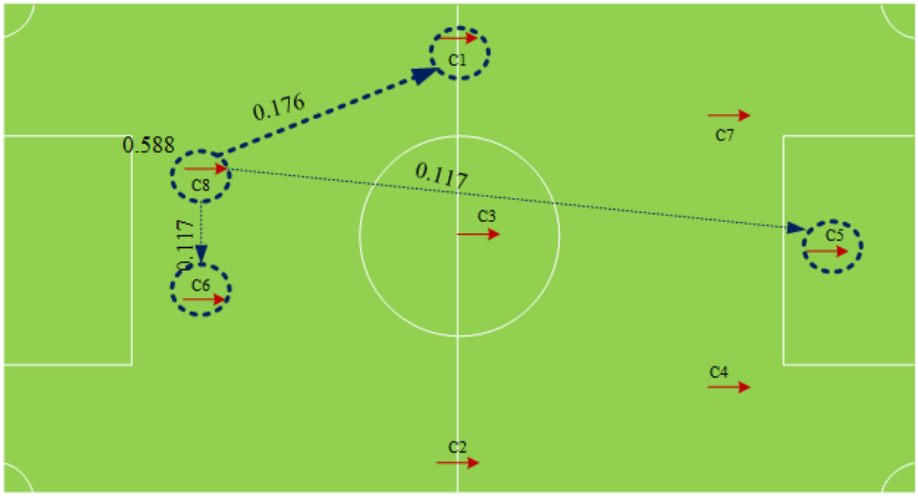

Figure 14- The areas of touching the ball by left back players .

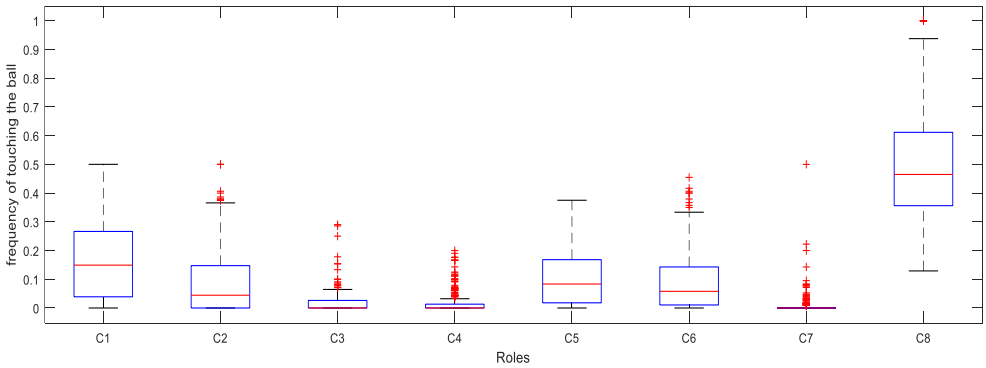

Figure 15- Distribution of touching the ball in 8 areas by players of the second cluster

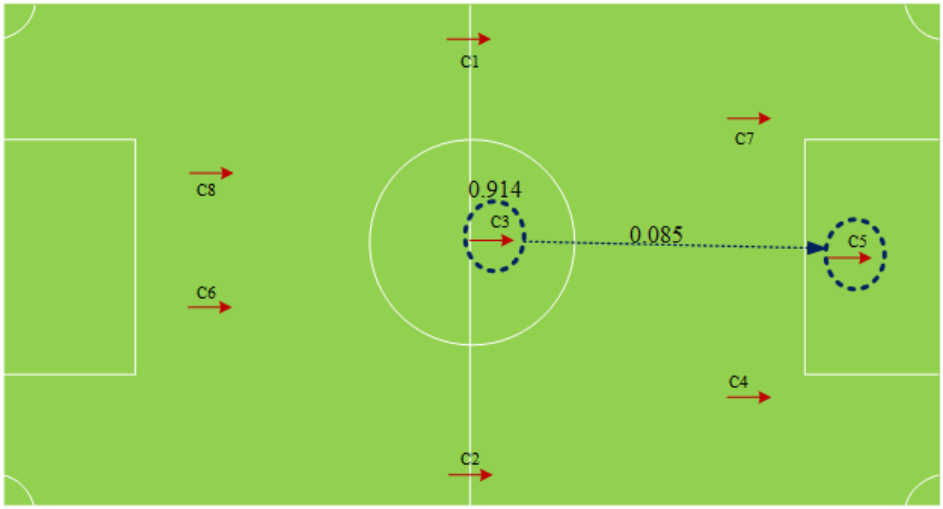

Figure 16- Central fielders touches the ball in C3 area and C5 area 91 and 8 percent of the times respectively.

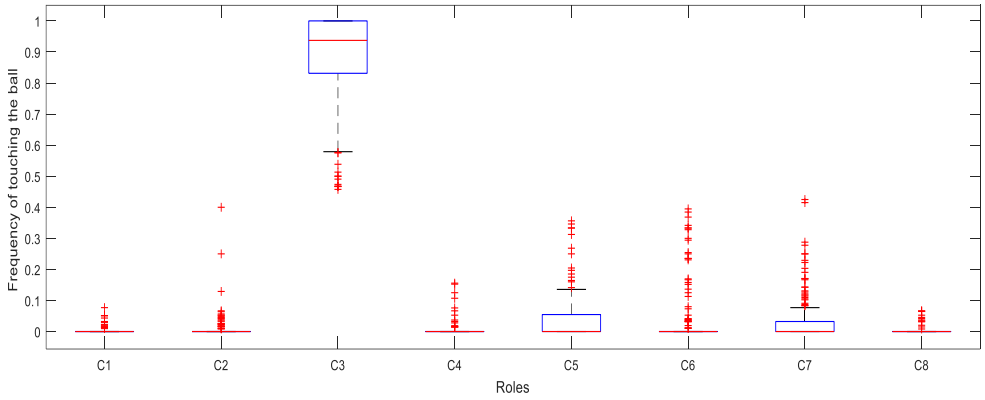

Figure 17- Distribution of touching the ball in different areas by central midfielders

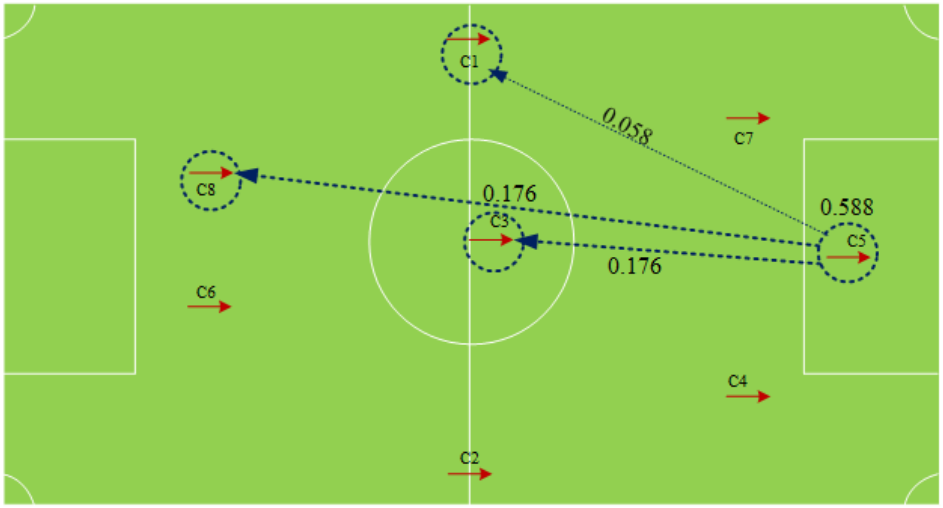

Figure 18- Different roles played by central forwards.

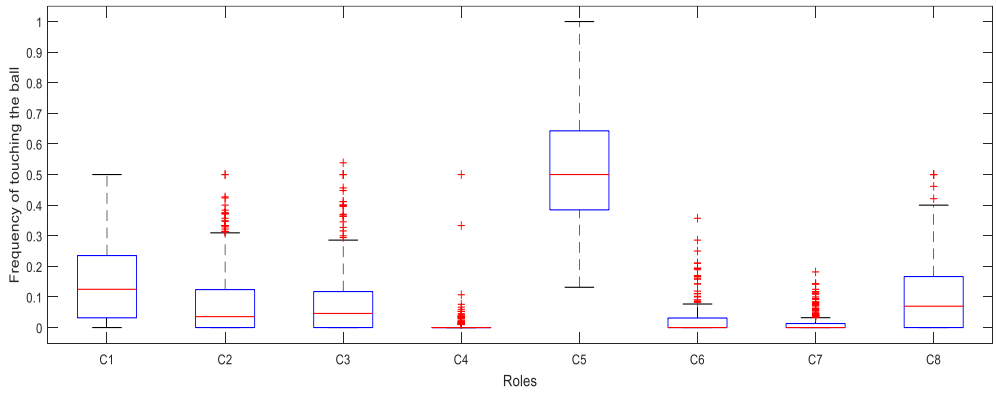

Figure 19- Distribution of touching the ball by central forwards.

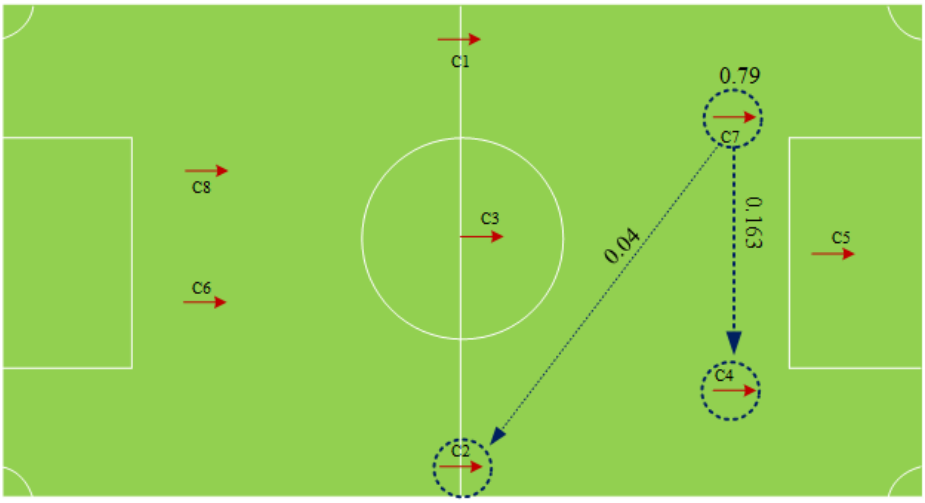

Figure 20- Left wingers' positions in different matches.

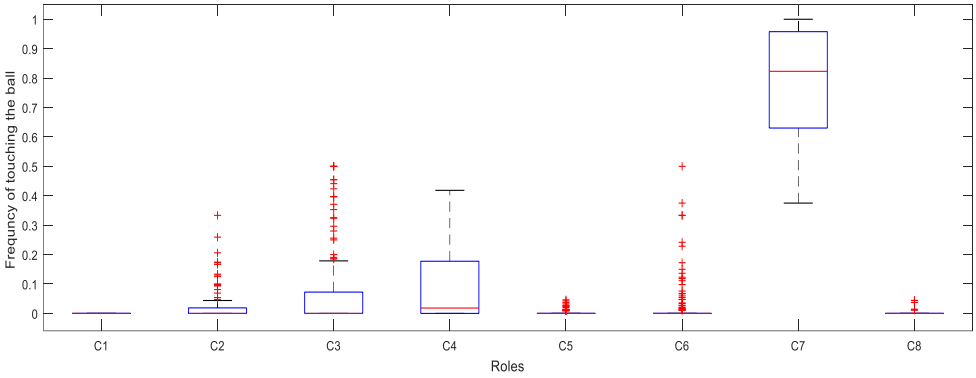

Figure 21- Different roles played by left wingers

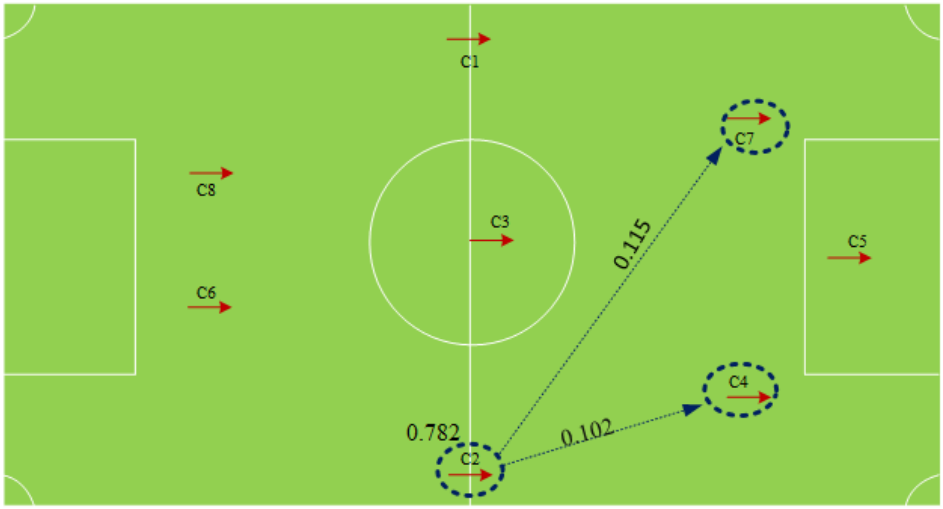

Figure 22- Right Fielders touched the ball in C7 and C4 areas 11 and 10 percent respectively.

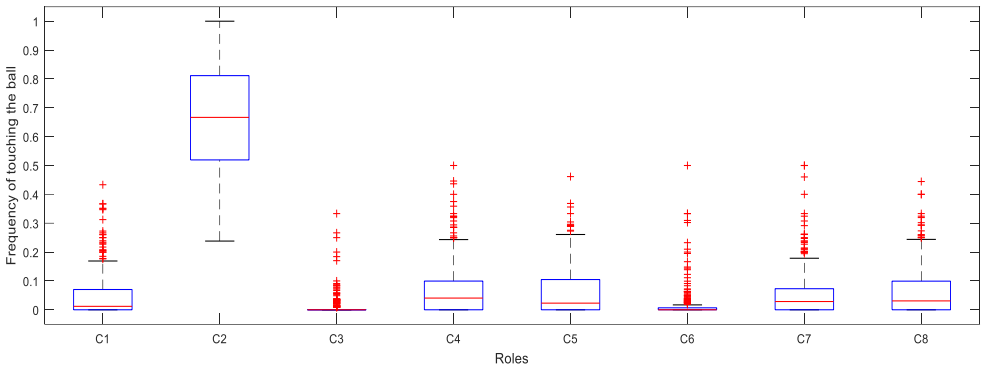

Figure 23- Distribution of right fielders' centers of performance

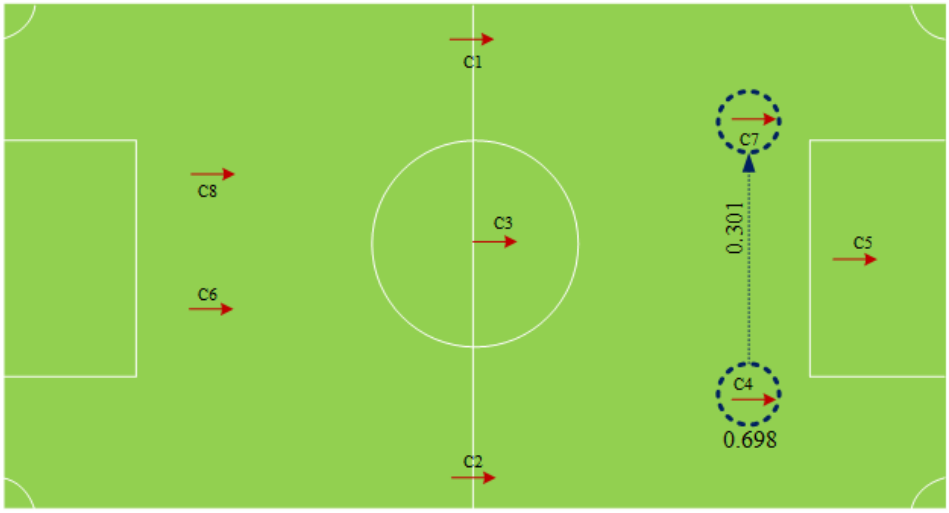

Figure 24- Right wingers played as left wingers in 30 percent of the matches.

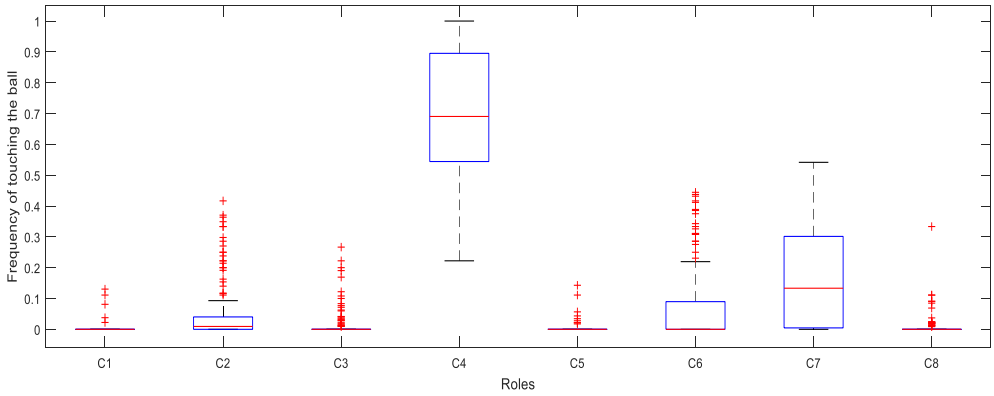

Figure 25- Distribution of different roles played by right wingers.

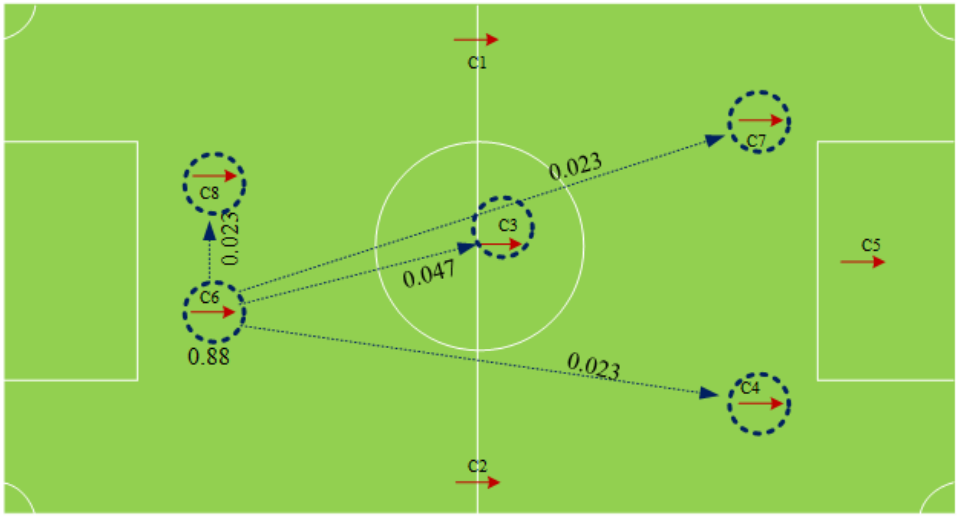

Figure 26- Different positions of central defenders in different matches.

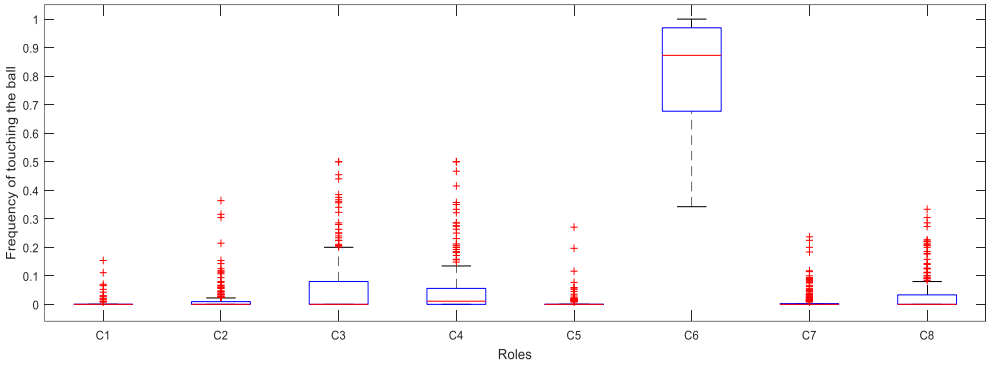

Figure 27- Distribution of different roles played by central defenders.

So these results show that, considering three major roles for players (defenders, midfielders and forwards), the most important and effective roles are right back, central midfielder and wingers. This indicates that having top players in these roles is necessary for a team trying to win the title. Figure 28 shows the data points of the second dataset after dimension reduction by a

multidimensional scaling algorithm (CMD). In this figure each cluster is shown using a specified color.

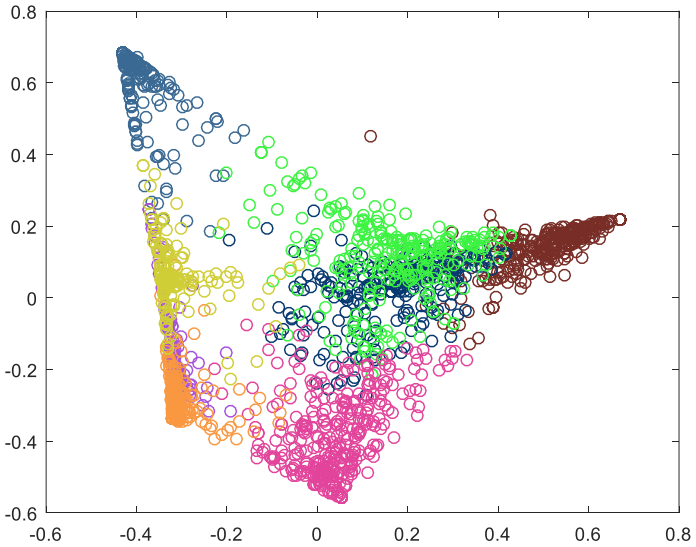

Figure 28- Different clusters of the second dataset after reducing the number of dimensions by CMD.

7. CONCLUSION AND FUTURE WORKS

In this paper a new clustering algorithm called, APSO-Clustering, is used to cluster a football dataset for finding different roles of players in football. The dataset consists of 93000 objects showing the center of players' performance in different matches. The algorithm found eight distinct clusters indicating eight roles in a team. In the next step the players who have switched their positions in different matches, have been found. In fact, for each player centers of the areas he played in each game are found and a new dataset including 2005 objects (players) and eight features showing the frequency of playing in each role, has been created. Interesting results which are achieved by clustering this dataset, indicate that playing more than one role in a match or in different matches depends on the player's skills, his main role and the situation of the match. Also results reveal three key roles in each team: right back, central midfielder and wingers. Based on the detected clusters, the players who play these roles usually don't change their positions which shows the importance of these roles. In addition to that, generally the results show the power and effectiveness of APSO-Clustering algorithm in clustering massive datasets. The proposed algorithm not only finds the position of the centroids accurately, but also it finds the number of clusters precisely which is very important specially for big data clustering. For future works, it is good to use this algorithm to analyze the performance of each team in a season and make a good comparison between the players based on their roles. Also using another swarm intelligence algorithm to cluster these kind of datasets and compare the results with PSO is interesting.

REFRENCES

[1] J. Gudmundsson, and M. Horton, "Spatio-temporal analysis of team sports," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 22, 2017.

[2] R. Rein, and D. Memmert, “Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science,” *SpringerPlus,* vol. 5, no. 1, pp. 1410, 2016.
[3] D. Risso, L. Purvis, R. Fletcher *et al.*, “clusterExperiment and RSEC: A Bioconductor package and framework for clustering of single-cell and other large gene expression datasets,” *bioRxiv,* pp. 280545, 2018.
[4] B. Jaillard, C. Richon, P. Deleporte *et al.*, “An a posteriori species clustering for quantifying the effects of species interactions on ecosystem functioning,” *Methods in Ecology and Evolution,* vol. 9, no. 3, pp. 704-715, 2018.
[5] D. Müllensiefen, C. Hennig, and H. Howells, “Using clustering of rankings to explain brand preferences with personality and socio-demographic variables,” *Journal of Applied Statistics,* vol. 45, no. 6, pp. 1009-1029, 2018.
[6] L. Sha, P. Lucey, Y. Yue *et al.*, “Interactive Sports Analytics: An Intelligent Interface for Utilizing Trajectories for Interactive Sports Play Retrieval and Analytics,” *ACM Transactions on Computer-Human Interaction (TOCHI),* vol. 25, no. 2, pp. 13, 2018.
[7] D. S. Rosenberg, “k-Means Clustering,” 2018.
[8] M. J. Rezaee, M. Jozmaleki, and M. Valipour, “Integrating dynamic fuzzy C-means, data envelopment analysis and artificial neural network to online prediction performance of companies in stock exchange,” *Physica A: Statistical Mechanics and its Applications,* vol. 489, pp. 78-93, 2018.
[9] D. Pelleg, and A. W. Moore, “X-means: Extending K-means with Efficient Estimation of the Number of Clusters,” in Proceedings of the Seventeenth International Conference on Machine Learning, 2000, pp. 727-734.
[10] J. Kennedy, "Particle swarm optimization," *Encyclopedia of machine learning*, pp. 760-766: Springer, 2011.
[11] M. H. Mozaffari, H. Abdy, and S. H. Zahiri, “IPO: an inclined planes system optimization algorithm,” *Computing and Informatics,* vol. 35, no. 1, pp. 222-240, 2016.
[12] N. M. Al Salami, “Ant colony optimization algorithm,” *UbiCC Journal,* vol. 4, no. 3, pp. 823-826, 2009.
[13] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in engineering software,* vol. 69, pp. 46-61, 2014.
[14] D. Whitley, “A genetic algorithm tutorial,” *Statistics and computing,* vol. 4, no. 2, pp. 65-85, 1994.
[15] J. Duch, J. S. Waitzman, and L. A. N. Amaral, “Quantifying the performance of individual players in a team activity,” *PloS one,* vol. 5, no. 6, pp. e10937, 2010.
[16] J. Brooks, M. Kerr, and J. Guttag, "Developing a data-driven player ranking in soccer using predictive model weights." pp. 49-55.
[17] L. Pappalardo, P. Cintia, P. Ferragina *et al.*, “PlayeRank: Multi-dimensional and role-aware rating of soccer player performance,” *CoRR,* vol. abs/1802.04987, /, 2018.
[18] R. Stanojevic, and L. Gyarmati, "Towards Data-Driven Football Player Assessment." pp. 167-172.
[19] O. Müller, A. Simons, and M. Weinmann, “Beyond crowd judgments: Data-driven estimation of market value in association football,” *European Journal of Operational Research,* vol. 263, no. 2, pp. 611-624, 2017.
[20] A. Lazinica, *Particle swarm optimization*: InTech Kirchengasse, 2009.
[21] Y. Shi, "Particle swarm optimization: developments, applications and resources." pp. 81-86.
[22] A. Starczewski, and A. Krzyżak, "Performance Evaluation of the Silhouette Index," *Artificial Intelligence and Soft Computing.* pp. 49-58.
[23] U. Maulik, and S. Bandyopadhyay, “Performance evaluation of some clustering algorithms and validity indices,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 24, no. 12, pp. 1650-1654, 2002.
[24] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, “Validity index for crisp and fuzzy clusters,” *Pattern recognition,* vol. 37, no. 3, pp. 487-501, 2004.

[25] T. Caliński, and J. Harabasz, “A dendrite method for cluster analysis,” *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1-27, 1974.
[26] School of Computing University of Eastern Finland. "clustering basic benchmarks," june 10, 2018; https://cs.joensuu.fi/sipu/datasets/.
[27] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846-850, 1971.
[28] A. F. McDaid, D. Greene, and N. Hurley, “Normalized mutual information to evaluate overlapping community finding algorithms,” *arXiv preprint arXiv:1110.2515*, 2011.