Faculty of Engineering and Architecture

Academic year 2016-2017

Soccer Network Analysis

Network Modelling and Design

't Jampens Cedric
Beernaert Simon
Casteleyn Louis

Contents

1 Introduction

The goal of the project is the analysis of the passing network of a soccer game. All events of official games are logged into JSON files and each of these can be identified through an event id. These data files have been downloaded from the specific webpage on `jsfiddle.net` for team formations[1] and game data[2]. The goal is to parse the information needed for this analysis from JSON files into Python. To link these event id's to real information, the 'Feed specification document (F24)' was used as documentation. It gives the specifics of an event id, but also provides structures to identify the formation, the starting eleven players, the substitutes, timestamps etc. The analysis consists of two parts: network visualization and analysis of the metrics. The metrics were based on the course contents and the article 'A Network Theory Analysis of Football Strategies' by Javier López Peña and Hugo Touchette (2012). The visualization was done in Gephi and Inkscape. The metrics were calculated in Python, some of them using the package `NetworkX`[3].

Another goal is to make the program flexible in handling the visualization of different games as input. This means that the coordinates of the formations have to be linked to the formations in the given JSON file. This has been done by the creation of a new JSON file that contains the formation id's and the coordinates of every possible player position with the aid of the documentation mentioned in the previous paragraph. It results in the program being able to correctly display the team formations in Gephi for every game as input.

The following metrics were chosen to analyze the network: vertex degree, vertex indegree, vertex outdegree, eccentricity, radius, diameter, closeness centrality, betweenness centrality, clustering and pagerank centrality. These metrics can be used in combination with the visualization of the network to analyze the players form, the teams flow, the most important players and help predict the possible outcome. However, as history has proven, statistics don't give full disclosure about the outcome of a game. This unpredictability can be seen as the beauty of the game and the reason why the sport still enjoys the popularity it has today.

[1] team formations link: https://jsfiddle.net/otb2epvu/
[2] game data link: https://jsfiddle.net/wkcewpfz/
[3] http://networkx.readthedocs.io/en/networkx-1.10/

2 Metrics

Before we can calculate any meaningful value per player, a network is required. We set up a network where the vertices are the eleven starting players and the edges are based on the passing. An adjacency matrix is needed to create a graph. This matrix is also used in different metrics in this section. To construct this matrix, we used the amount of successful passes between two players and inverted it to become the cost to travel through the network. This results in a matrix that displays how difficult it is to pass a ball between two players. In the Python program this matrix is referred to as the 'distance matrix' or `distance_matrix`.
As said in the introduction, some metrics where calculated by algorithms we based on a paper and other where invoked through the NetworkX Python package. This package is specifically written for the manipulation of complex networks.

Note: Below, all the used metrics and their calculation will be explained. When a new variable enters a formula, this will be explained, otherwise the variable has the same meaning of previous formula where it was used.

1. **Player degree in/out**
   The vertex or player degree out in our network is the sum of the passes made by this player. The degree out is a measure of how much the player gives a pass to other players and therefore how much he is serving other players. Logically, the metric 'player degree in' is the sum of the ingoing passes to one particular player and gives an indication of which player receives a lot of passes and is accessible.

$$vertexin_a = \sum_{b=1}^{n} A_{ba}$$

$$vertexout_a = \sum_{b=1}^{n} A_{ab}$$

   With

   A_{ab}: the adjacency matrix where the weights of the edges are the number of passes from a to b.

   a: the player that is reviewed.

   b,k: respectively the source and target player.

2. **Total player degree**
   When summing player degree in and player degree out, the total player degree is obtained. The total player degree gives an idea of how busy a player is in the game or how active this player is.

$$vertexdegree_a = \sum_{b=1}^{n} (A_{ba} + A_{ab})$$

3. **Eccentricity**
   Eccentricity is the maximum distance from a player to all other players. Players with high eccentricity within the team will be more influenced by other players' actions or will influence the others themselves. To calculate the eccentricity of the players we used the function

$$eccentricity_a = \texttt{nx.eccentricity(Graph)}$$

   of NetworkX. Where `Graph` is $\mathrm{A_{ab}}$.

4. **Radius**
   The radius can be explained as the minimum eccentricity of the team. We also used NetworkX to obtain the teams radius:

$$radius_a = \texttt{nx.radius(Graph)}$$

5. **Diameter**
   In contrast to the radius, the diameter is the maximum eccentricity. The NetworkX package calculated this metric as well.

$$diameter_a = \texttt{nx.diameter(Graph)}$$

6. **Closeness centrality**
   Centrality values the relevance or popularity of a player. This first centrality measure, the closeness, expresses how easy it is to reach a fellow teammate. This value denotes how close a player was to all the other players. Incoming and outgoing passes are treated equally.

$$Closeness_a = 2\frac{n-1}{\sum_{b \neq a} A_{ab} + \sum_{b \neq a} A_{ba}}$$

7. **Betweenness centrality**
   This second centrality measure displays the importance in ball-flow of a player between other players; how relevant is the player for connecting two others. In fact the betweenness is the fraction of shortest paths where the particular player is involved. The removal of a player with a low betweenness centrality from the team won't effect the team.

$$Betweenness_a = \frac{1}{(n-1)(n-2)} \sum_{a \neq b \neq k} \frac{n_{bk}^a}{g_{bk}}$$

   With

   $n_{bk}{}^a$: the number of shortest paths where player a is involved.
   g_{bk}: the total number of shortest paths.

8. **Clustering**
   This metric indicates how well a player, as the name suggests, clusters together with the team or how likely it is that they will form a clique.

$$Clust_a = \frac{1}{vertexout_a(vertexout_a - 1)} \sum_{b,k} \frac{\sqrt[3]{A_{ab}A_{kb}A_{ka}}}{max(A)}$$

9. **Pagerank**
   If a player gets a lot of passes from other popular players, he will be popular as well. Based on this assumption, Pagerank creates a probability for each player that this player will be involved in the game after a number of passes. This implies a simultaneous calculation for all the team members. To calculate this complex metric, NetworkX was used.

$$PRcentrality_a =$$

   `nx.pagerank(Graph,alpha,max_iter,tol,weight,nstart,dangling)`

   Apart from `Graph`, this function requires parameters. We decided to use for `alpha` and `tol` (tolerance) the default values of 0.85 and 1.0e-06. The calculate the eigenvalues the solver asks a maximum numbers of iterations (`max_iter`), we settled at 100. We indicated that our graph was a weighted graph by setting `weight` to 'weight'. Further, we left `dangling` and `nstart` untouched.

3 Python Model

3.1 JSON files

First of all, the JSON files from both teams, the game data, event id's and team formation coordinates are being loaded into the program. The model is dynamically which means that every game can be analyzed with this tool if the appropriate data is available on the jsfiddle.net website. In our example, the game between Real Madrid and FC Barcelona of 2015[4] have been used as a try out.

3.2 Program

The chore of this program is based on the chore of the structure of the game data. Each event during the game has been listed chronologically and each event has a certain ID which corresponds to a certain action that happened within the game. Therefore, a for-loop iterates from the start event until the last event. Note that each event has some useful data included such as: type of event, id of event, which team and player was involved in the event, time stamp, etc. As different actions during the game have different event id's, specific events can be filtered and the data within these events can be used for analyzing the data. Within a specific event, a distinction has been made to which team the analyzed event belongs to. This if-statements differentiate team 1 from team 2.

3.2.1 Starting information

A first important event we have to extract, is to gather the general information of both teams at the beginning of a game. This event id is equal to '34' and includes useful attributes such as the team selected, team formation and the starting eleven. Using these player id's can get us their corresponding player names and shirt numbers. According to the team formation, corresponding x- and y-coordinates are being extracted to draw the player's network of each team.

3.2.2 Passing matrix

The weights of the players' network correspond to the amount of successful passes one gives to another. The id of a pass is equal to '1' and is successful if the outcome is equal to '1'. If the next event includes a player from the same team, then a pass from player 'i' to player 'j' can be registered and stored in a 11x11 pass matrix. Unsuccessful passes and errors within the JSON file game data have been left out. Later on to calculate the distance between

[4] http://nl.soccerway.com/matches/2015/11/21/spain/primera-division/real-madrid-club-de-futbol/futbol-club-barcelona/2086376/

two players, the inverse of each amount of passes between two players will be calculated.

3.2.3 Calculating metrics

If a substitution occurs, the metrics for the current time window pass matrix of the team playing until the substitution will be calculated. A sub is being characterized by event id '18' for an outgoing player and event id '19' for an incoming player from the same team. If the referee blows his whistle and ends the game, the metrics from both teams also will be calculated. To determine the time window of a pass matrix, a time stamp of the substitution is being taken to determine the time period a certain eleven players of a team has been playing together.
Hereby, the distance between each player of a team will be calculated by inverting the amount of passes between two players. The more passes given between player 'i' and 'j', the shorter the distance d_{ij} will be. Thereafter, the shortest distance between any two players within the team will be calculated using the Dijkstra algorithm from the NetworkX library. This results in a dictionary with all target and source nodes with corresponding weights. This dictionary is then converted into a shortest distance matrix to ease further calculations. From this graph, the incoming, outgoing and total vertex degree of each player can be calculated. Furthermore, the radius, diameter and eccentricity of the players within the team will show us how many players maximal are needed to go from on outer player to another outer player of the network. Thereafter, the clustering coefficient of each player and the average clustering of the analyzed team are being calculated. Finally, the closeness-, betweenness- and pagerank centrality can be determined using adapted formula's found in the paper.

At last, all metrics are being placed in matrix, which is then exported to a CSV file to use in the report and presentation. Besides, the nodes with corresponding coordinates and edges with corresponding weights are also being exported to use in Gephi for network visualization. The final step exists of emptying the pass matrix for another time window.

4 Network visualization

4.1 CSV files

The visualization of the passing matrix is done in Gephi. It allows us to use CSV files as input. These CSV files are written at the end of the code for each team and each time stamp. 2 types of CSV files are written: nodes and edges.

4.1.1 teamx_minute_y_nodes.csv

The output of the CSV files with nodes is an array of the following format: `[['Id', 'Label', 'x-coordinate', 'y-coordinate']]`. The 'Id' is the id of the node, the 'Label' is the name of the node and thus the name of the player. Following are the coordinates of the type: double. This is required for the plug-in in Gephi that is used. This plug-in is called GeoLayout and allows you to make a network with fixed coordinates for every node [5]. For right visualization of the names of the player the data type 'windows-1258' was chosen instead of 'UTF-8' to include the accents on the names of several Spanish/Portuguese players. The coordinates are loaded from another JSON file that was created by us based on the documentation of the formations. An excerpt of the particular JSON file can be seen in Figure E. The coordinates are calculated on a [120x120] grid. This values are chosen to be able to give the coordinates integer values and facilitate allocation of positions.

4.1.2 teamx_minute_y_edges.csv

The edges are written in an analog way and the output is an array of the following format: `[['Source', 'Target', 'Weight']]`. 'Source' represents the id of the source node and 'Target' represents the id of the target node. 'Weight' represents the amount of passes given from the source to the target in that time stamp.

4.2 Gephi and Inkscape

Once these CSV files are opened in Gephi, GeoLayout is used to map the coordinates, it is important that these coordinates are doubles. Gephi stores these files as a SVG data type. Gephi allows you to filter certain nodes and calculate some metrics. However, we chose to calculate the metrics in Python. Unfortunately, Gephi is not that flexible in a nice visualization. The SVG files have been transferred to Inkscape, where some things like the name labels are adjusted. Then the files are exported to PNG.

[5] https://gephi.org/tutorials/gephi-tutorial-layouts.pdf

4.3 Passing networks

For the example game between Real Madrid and Barcelona, the constructed networks are respectively given by Appendix A and Appendix B.

5 Analysis and conclusion

5.1 Analysis of network

As an example to analyze the data generated by our program, the network of Barcelona between minute 27-56 will be analyzed because herein, they scored two goals and thus created useful chances via successful passes.
The passing network can be seen in figure 2(b). The density of the directed edge is related to the amount of passes given form player 'i' to player 'j'. The darker the arrow, the more successful passes that are given between these two players. If the network is analyzed, the attacks from Barcelona start most of the time via the left or right back players and pass directly to their corresponding left or right wing or attacker. Moreover, The central midfielder connects the two sides of the team and can rotate the game-play. In general we expect that defenders give more outgoing passes than they receive and attackers receive more passes than they give.

5.2 Analysis of metrics

For the same time window and team, the table with calculated metrics will be analyzed. This table 5 can be found in Appendix D: Metrics Barcelona.

5.2.1 Vertex degree

At first, the total player degree is related to the position of a player. The highest degree has been realised by midfielder Busquets (17), which means that he reaches and get reached by the most amount of different players. In general, all players tend to play well together such as Iniesta (15), Neymar (15), Sergi Roberto (15) and Daniel Alves (15) who also have a high total degree. The lowest degree can be found at the goalkeeper Claudio Bravo (7), due to the fact that he passes most of the team with the defenders in order to build an attack. Also the central forward Suarez (11) tend to have a low degree as he is more isolated in the front of the game and get crucial passes from the midfield and wings in order to create a shotting chance. No specific trend between incoming degree and outgoing degree between different positions can be made using this time window.

5.2.2 Eccentricity

All players tend to have an eccentricity of 2, which is the diameter of the network. Except for the central midfielder Busquets who has an eccentricity of 1, which is the radius of the network. This means that he is the central point of the team, which is his job as central midfielder. These low values

mean that all players can be reached by first passing to Busquets, making him an important connection within the network.

5.2.3 Closeness centrality

The highest closeness centrality values are created by Jordi Alba (2.27), Sergio Busquets (2.14), Jeremy Matthieu (2.21), Ivan Rakitic (2.11) and Andres Iniesta (2.23). These players have a small average distance towards other players, indicating that they are well-connected players in the team. The type of players with the highest values are mainly midfielders and partly defenders, the attackers: Suarez (1.66), Sergi Roberto (1.67) and Neymar (1.75) combined with the goalkeeper: Bravo (1.79) tend to have a lower value and thus are worse-connected players within the team. This also means that the opponent is defending well in the areas where the forward players want to receive the ball and give a crucial pass to create a chance on target.

5.2.4 Betweenness centrality

Here, the player which is included in the most shortest paths between other players is Rakitic (0.222) followed by Mathieu (0.144) and Busquets (0.111). So during this time window, the ball-flow between other players depends the most on these three players, and it would have the highest impact by removing Rakitic from the game. The lowest values can be found at all three attackers (all ¡ 0.1) and at the goalkeeper (0), meaning that they are more the begin- or end-station of a series of successful passes and thus have less impact by removing them from the game in order to maintain a good ball-flow. However, as attackers are important to score goals, their importance can not be overlooked.

5.2.5 Clustering coefficient

This is a measure on how tightly players interact in a team. The highest value can be found at Jordi Alba (0.276) followed by Iniesta (0.172) and Suarez (0.172). The lowest value is from Pique (0.08) and Daniel Alves (0.094), which means they did not interact well within the team compared to others.

5.2.6 Pagerank

This heuristic represents the probability that a player will have the ball after a number of passes has been made. Which indicated how popular the player is. The most popular player in this time window is Mathieu (0.115) followed by Busquets (0.109) and Alba (0.107). The least popular players are Bravo (0.072) and Suarez (0.073) as they play at the outside of the network.

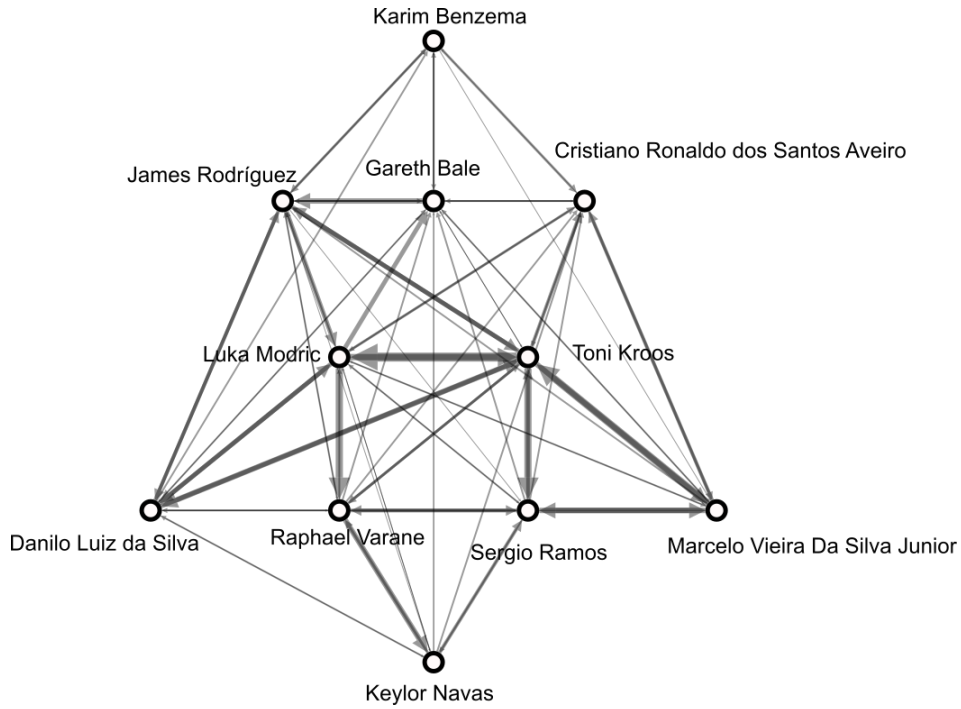
5.3 Conclusion and future work

The visualization of the networks in Appendix A and B offers the spectator a more general and non-mathematical approach to analyze the passes in a game. As can be seen in Figure 2, Barcelona exploits its flanks more than Real Madrid in Figure 1. They use key players like Rakitic and Iniesta and later on Messi to reach the forwards. Real Madrid however focuses on keeping the ball more on the central midfield between Modric and Kroos. This means their flanks are less protected and with the tactics of Barcelona, this might not have been the best idea. The score verifies this (0-4). A more precise and profound analysis can be done by combining the passing network with the metrics. This gives a more total image of the game and the most important players.

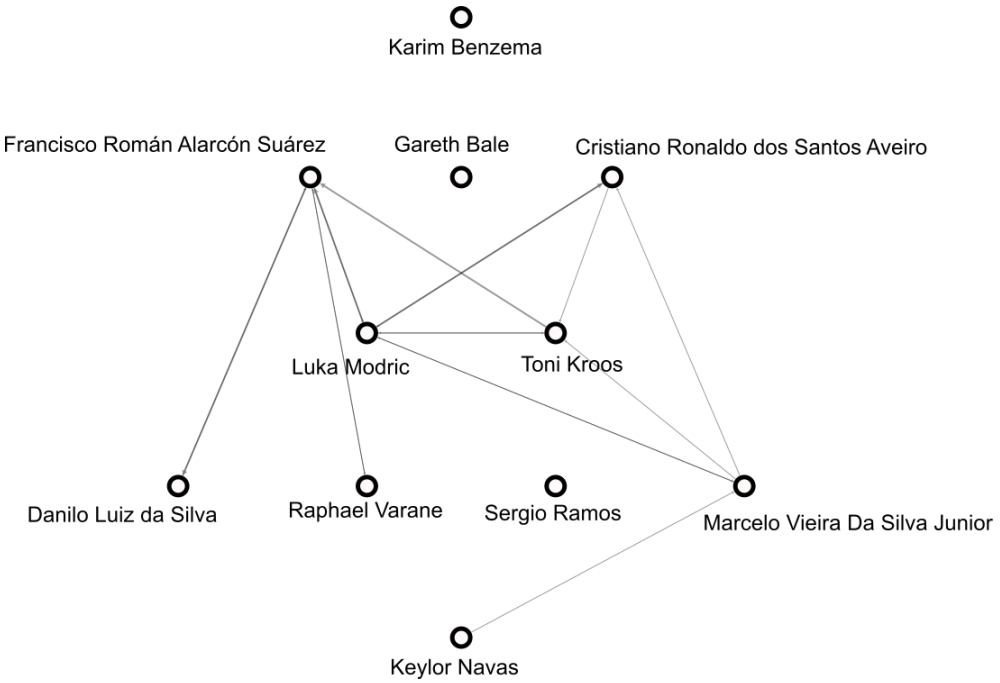
Passing structures are just one aspect of the enormous amount of information that is available about soccer games. The analysis of a soccer network can go way further than described in this project. However, due to limited time, only a selection of information was gathered and analyzed: we made it possible to load formations and linked coordinates to the these for structured visualization, the most important metrics were calculated and the passing network was analyzed. A professional analysis could go way further. The shooting matrix could be combined with the passing network for visualization, a dynamic network could be created that adds 1 pass to the network every time a pass is logged, number of shots on target, shot effectiveness... The main concern is parsing the information into Python and filter the information that is needed. This already has been done in the project. It means that extensions can be done in the same way, expanding the code that already has been written.

Personally, we have experienced this project as interesting and fun at the same time. Not only did it give us an introduction to Python, it also helped us see the practical use of the metrics that were seen in the course and the power of visualization. On the other hand, since we all are soccer players, the subject of the project was interesting to us. Finally, it gave us insight about the amount of data that is gathered every single game and how it can be used.

A Passing network Real Madrid

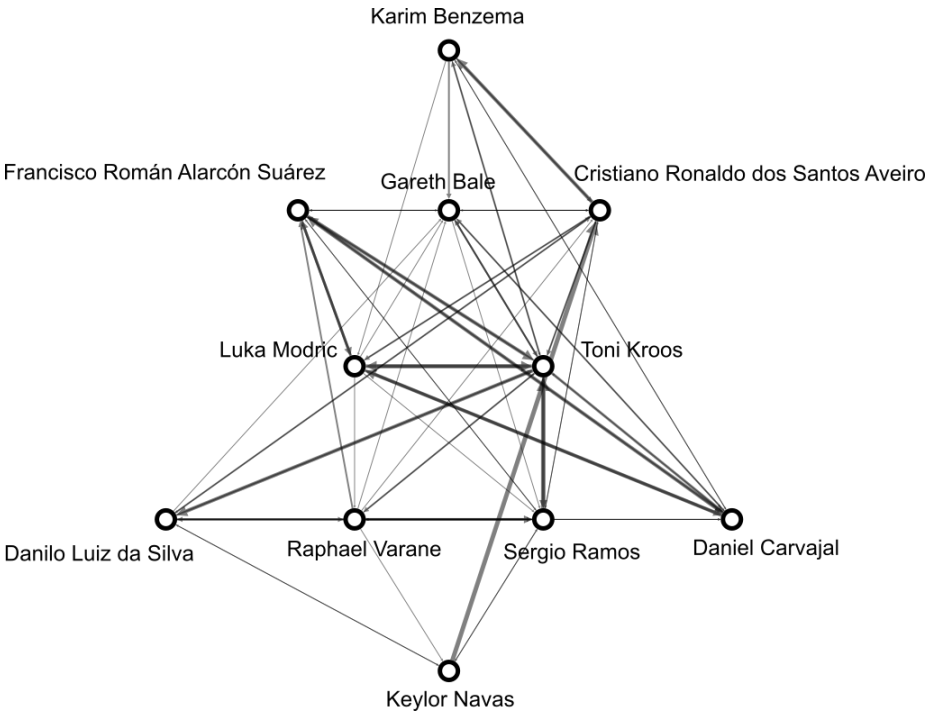


(a) Timestamp (min): 0 - 54



(b) Timestamp (min): 54 - 58

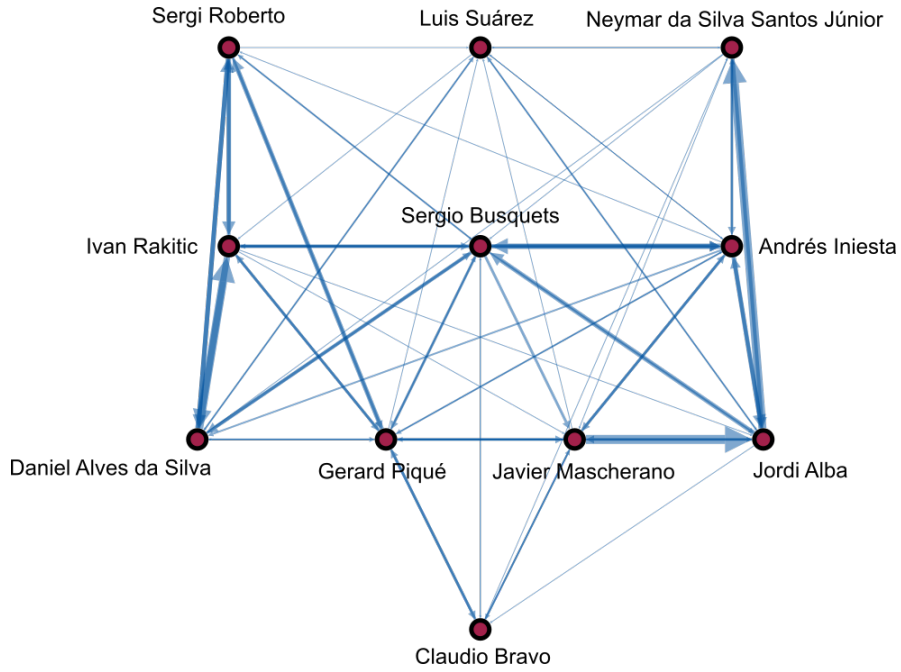
Figure 1: Passing network for Real Madrid



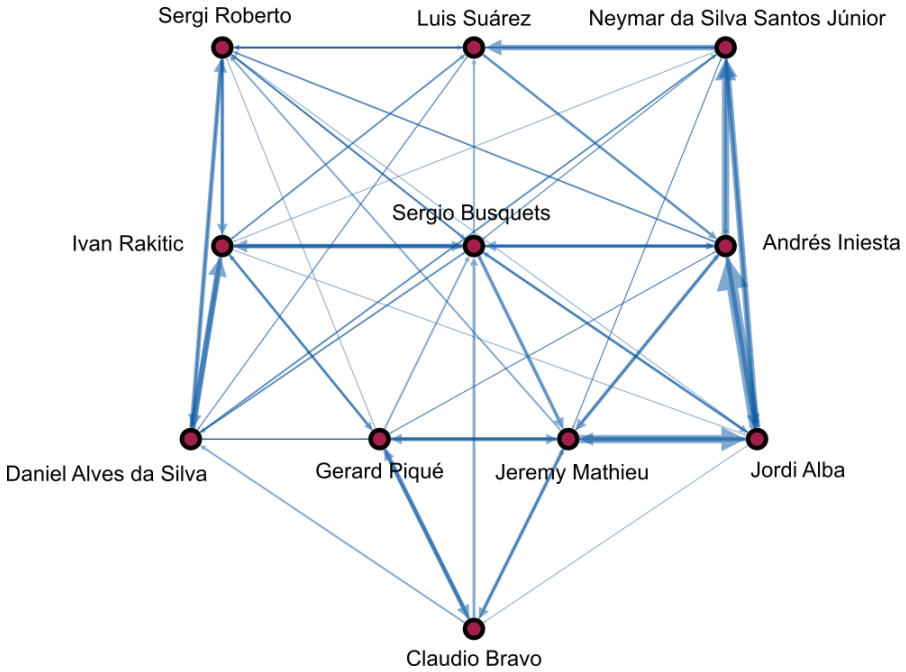
(c) Timestamp (min): 58 - 91

Figure 1: Passing network for Real Madrid

B Passing network Barcelona

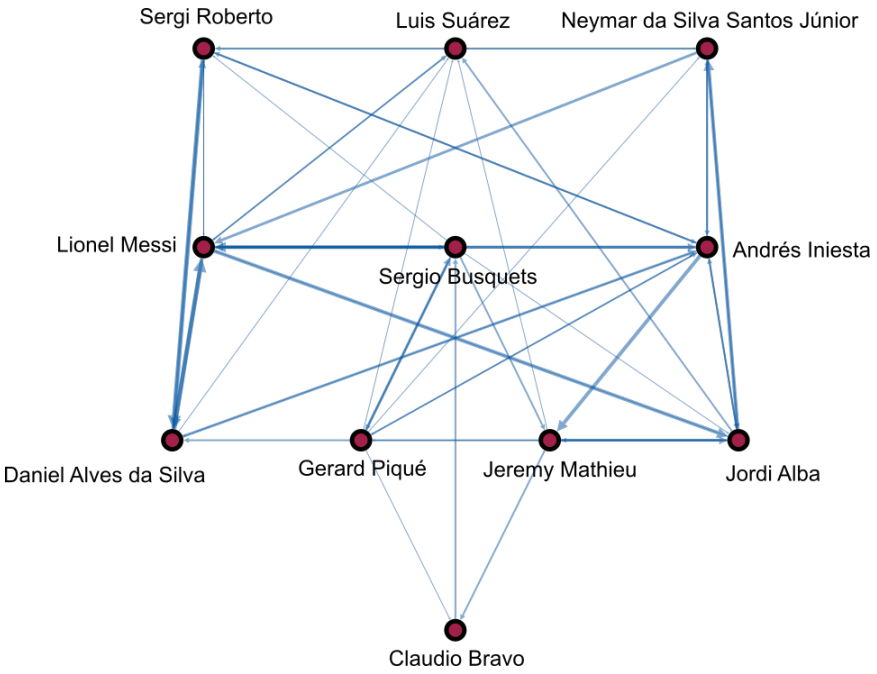


(a) Timestamp (min): 0 - 27



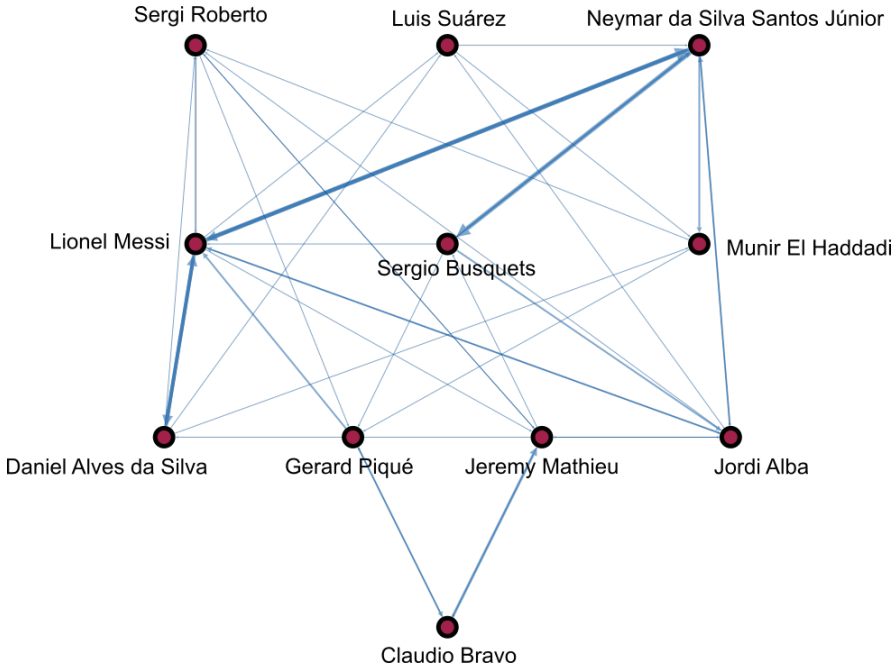
(b) Timestamp (min): 27 - 56

Figure 2: Passing network for Barcelona



(c) Timestamp (min): 56 - 76

Figure 2: Passing network for Barcelona



(d) Timestamp (min): 76 - 91

Figure 2: Passing network for Barcelona

C Metrics Real Madrid

Shirtnr	Player	deg_{in}	deg_{out}	deg_{tot}	ϵ	c_C	c_B	cc	rank
1	Keylor Navas	4	8	12	2	1.791	0.0	0.081	0.064
23	Danilo Luiz da Silva	7	8	15	2	2.597	0.0	0.178	0.089
12	Marcelo Vieira Da Silva Junior	6	7	13	2	2.554	0.056	0.183	0.086
8	Toni Kroos	8	8	16	2	3.3	0.244	0.277	0.125
5	Raphael Varane	6	8	14	2	2.451	0.144	0.176	0.107
4	Sergio Ramos	7	7	14	2	2.367	0.033	0.226	0.09
10	James Rodrguez	9	8	17	1	2.692	0.244	0.209	0.105
19	Luka Modric	8	9	17	2	2.878	0.133	0.203	0.122
9	Karim Benzema	5	4	9	2	1.49	0.0	0.189	0.047
11	Gareth Bale	10	5	15	1	1.98	0.0	0.335	0.091
7	Cristiano Ronaldo dos Santos Aveiro	8	6	14	2	1.739	0.0	0.249	0.074

Table 1: Timestamp (min): 0 - 54

Shirtnr	Player	deg_{in}	deg_{out}	deg_{tot}	ϵ	c_C	c_B	cc	rank
1	Keylor Navas	0	1	1	NaN	0.0	0.0	0	0.048
23	Danilo Luiz da Silva	1	1	2	NaN	0.667	0.0	0	0.072
12	Marcelo Vieira Da Silva Junior	2	3	5	NaN	0.545	0.0	0.188	0.141
8	Toni Kroos	3	2	5	NaN	0.6	0.0	0.897	0.158
5	Raphael Varane	1	1	2	NaN	0.522	0.0	0	0.045
4	Sergio Ramos	0	0	0	NaN	0.0	0.0	0	0.018
22	Francisco Romn Alarcn Surez	4	3	7	NaN	0.923	0.133	0.105	0.225
19	Luka Modric	4	4	8	NaN	1.0	0.178	0.202	0.158
9	Karim Benzema	0	0	0	NaN	0.0	0.0	0	0.018
11	Gareth Bale	0	0	0	NaN	0.0	0.0	0	0.018
7	Cristiano Ronaldo dos Santos Aveiro	2	2	4	NaN	0.6	0.0	0.815	0.101

Table 2: Timestamp (min): 54 - 58

Shirtnr	Player	deg_{in}	deg_{out}	deg_{tot}	ϵ	c_C	c_B	cc	rank
1	Keylor Navas	3	3	6	2	0.715	0.0	0.318	0.059
23	Danilo Luiz da Silva	4	7	11	2	1.421	0.0	0.124	0.081
2	Daniel Carvajal	7	6	13	2	1.337	0.0	0.286	0.081
8	Toni Kroos	9	9	18	2	1.91	0.578	0.238	0.161
5	Raphael Varane	5	7	12	2	0.816	0.0	0.172	0.076
4	Sergio Ramos	8	6	14	2	1.554	0.089	0.291	0.086
22	Francisco Romn Alarcn Surez	7	6	13	2	1.382	0.0	0.342	0.092
19	Luka Modric	5	7	12	2	1.584	0.056	0.223	0.103
9	Karim Benzema	3	5	8	2	0.939	0.0	0.176	0.064
11	Gareth Bale	8	5	13	2	1.027	0.0	0.413	0.076
7	Cristiano Ronaldo dos Santos Aveiro	9	7	16	2	1.179	0.122	0.223	0.12

Table 3: Timestamp (min): 58 - 91

D Metrics Barcelona

Shirtnr	Player	deg_{in}	deg_{out}	deg_{tot}	ϵ	c_C	c_B	cc	rank
13	Claudio Bravo	7	7	14	2	1.31	0.0	0.042	0.047
6	Daniel Alves da Silva	21	22	43	2	1.86	0.056	0.015	0.105
18	Jordi Alba	25	25	50	2	1.94	0.156	0.012	0.123
5	Sergio Busquets	22	21	43	2	1.99	0.122	0.022	0.108
3	Gerard Piqu	19	21	40	2	1.8	0.1	0.022	0.095
14	Javier Mascherano	17	18	35	2	1.77	0.056	0.019	0.114
4	Ivan Rakitic	21	23	44	2	1.81	0.056	0.015	0.105
8	Andrs Iniesta	20	23	43	2	2.07	0.144	0.018	0.113
9	Luis Surez	9	5	14	2	0.89	0.0	0.141	0.047
20	Sergi Roberto	17	15	32	2	1.65	0.022	0.025	0.08
11	Neymar da Silva Santos Jnior	13	11	24	2	1.52	0.0	0.028	0.064

Table 4: Timestamp (min): 0 - 27

Shirtnr	Player	deg_{in}	deg_{out}	deg_{tot}	ϵ	c_C	c_B	cc	rank
13	Claudio Bravo	3	4	7	2	1.79	0.0	0.132	0.072
6	Daniel Alves da Silva	7	8	15	2	1.53	0.0	0.094	0.077
18	Jordi Alba	7	5	12	2	2.27	0.1	0.276	0.107
5	Sergio Busquets	9	8	17	1	2.14	0.111	0.144	0.109
3	Gerard Piqu	5	7	12	2	2.09	0.089	0.08	0.081
24	Jeremy Mathieu	7	7	14	2	2.21	0.144	0.142	0.115
4	Ivan Rakitic	6	7	13	2	2.11	0.222	0.131	0.101
8	Andrs Iniesta	8	7	15	2	2.23	0.067	0.172	0.103
9	Luis Surez	6	5	11	2	1.66	0.022	0.173	0.073
20	Sergi Roberto	8	7	15	2	1.67	0.067	0.15	0.081
11	Neymar da Silva Santos Jnior	7	8	15	2	1.75	0.044	0.111	0.081

Table 5: Timestamp (min): 27 - 56

Shirtnr	Player	deg_{in}	deg_{out}	deg_{tot}	ϵ	c_C	c_B	cc	rank
13	Claudio Bravo	2	1	3	3	0.88	0.0	0	0.049
6	Daniel Alves da Silva	4	4	8	3	1.36	0.078	0.399	0.099
18	Jordi Alba	5	6	11	2	1.49	0.089	0.172	0.121
5	Sergio Busquets	3	5	8	2	1.13	0.078	0.033	0.092
3	Gerard Piqu	3	5	8	2	0.94	0.022	0.035	0.071
24	Jeremy Mathieu	3	4	7	2	1.3	0.1	0.158	0.105
10	Lionel Messi	7	6	13	2	1.78	0.4	0.244	0.133
8	Andrs Iniesta	6	7	13	2	1.52	0.133	0.168	0.113
9	Luis Surez	5	1	6	2	0.68	0.0	0	0.054
20	Sergi Roberto	5	4	9	2	1.24	0.0	0.343	0.086
11	Neymar da Silva Santos Jnior	4	4	8	2	1.29	0.0	0.267	0.079

Table 6: Timestamp (min): 56 - 76

Shirtnr	Player	deg_{in}	deg_{out}	deg_{tot}	ϵ	c_C	c_B	cc	rank
13	Claudio Bravo	2	2	4	3	0.59	0.022	0.0	0.046
6	Daniel Alves da Silva	2	4	6	2	0.84	0.0	0.07	0.092
18	Jordi Alba	4	5	9	2	0.72	0.0	0.117	0.082
5	Sergio Busquets	2	4	6	2	0.96	0.133	0.102	0.109
3	Gerard Piqu	4	2	6	2	0.66	0.067	0.252	0.088
24	Jeremy Mathieu	4	5	9	2	0.76	0.089	0.046	0.075
10	Lionel Messi	6	6	12	2	0.99	0.167	0.104	0.161
17	Munir El Haddadi	3	2	5	2	0.9	0.2	0.1	0.073
9	Luis Surez	3	2	5	3	0.5	0.0	0.397	0.062
20	Sergi Roberto	5	3	8	2	0.67	0.0	0.195	0.074
11	Neymar da Silva Santos Jnior	4	4	8	3	1.04	0.189	0.151	0.138

Table 7: Timestamp (min): 76 - 91

E formations.JSON


```
{"bin_ver": 7,
"Team formation": {
                "1": {
                "id": "2",
                "formation": "442",
                "pos1": {
                        "x": "60",
                        "y": "15"
                },
                "pos2": {
                        "x": "15",
                        "y": "45"
                },
                "pos3": {
                        "x": "105",
                        "y": "45"
                },
                "pos4": {
                        "x": "45",
                        "y": "75"
```


Figure 3: Head of created JSON file to parse formations into Python