Analysis Cooperation Structure of Network and Making Teaming Strategies
——based on Sliding Window Model

Teamwork can solve complex problems that may be difficult for individuals to handle, so improving team effectiveness is meaningful. We analyzes the typical *cooperation structure* from the ball passing network, and establishes the Sliding Window Model to identify the cooperation mode suitable for the Huskies, and then puts forward the suggestions of strategic adjustment.

Firstly, we set up a directed ball flow network with the soccer players as the node and the passing as the arc. The we select the structure of all passes larger than 5 from network as **cooperation structure** and set the feature vector for it. According to the feature vector value, we use the average-linkage method of hierarchical clustering to cluster the path, and use the main component analysis method to reduce the dimension. Eventually, we extract six kinds of typical strategy patterns from the network and analyze their structural characteristics.

Secondly, we choose the *average node connectivity*, *transitivity* and other indicators to measure teamwork. We establishes a **Sliding Window Model** to describe the changes of the performance indicators during the match, which sets a 900-second (1/3-half-time) window to slide. The window is a time-sharing subplot of the match, which can precisely capture the configuration and dynamic strategy of the team at that time sliding the position of the highlight time period .

Thirdly, after analyzing the structure that the Huskies often used in highlight performance., we find that the Huskies is more inclined to use defensive counterattack strategy. The Huskies usually performs better in the middle-front and back courts, but is very weak in the middle-back courts. Therefore, we recommend that the Huskies carries forward the long-pass punching mode and use the 4-4-2 formation.

Finally, we find that enhancing the flexibility of the team structure and establishing a local coordination structure are beneficial to improve the team's efficiency. It is necessary to refine and analyze the main mode of team contact. When the overall indicators are too difficult to explain in detail, the idea of sliding window may be useful. If we can capture more information about the characteristics of successful teams, we may be able to build a more generalized model for measuring team performance.

Keywords: Network, Cooperation Structure, Sliding Window Model, Teaming Strategies

Contents

1 Introduction

1.1 Background

Soccer is a complex system based on people's general understanding. The overall performance of a team does not equal to the sum of all players' skills. The victory of a game calls for suitable strategies and flawless cooperation. Both attack and defense subsystems are relatively independent in structure, functions, and behaviors. They are interrelated and restrict each other. Meanwhile, soccer match is also an open system interacting with outside. A variety of factors influence the outcome of a match.

1.2 Restatement of the Problem

According to the competition data of this season, our home soccer team Huskie did not behave so well. We are invited by their coach to explore how their interactions lead to final results. Various types of team style and strategy may affect team's performance greatly, sometimes even be a constraint. Thus, we will also offer suggestions based on past defect to improve Huskie's outcome next season.

Problem 1 requires us to create a network in accordance with passing data. We use nodes and edges to represent players and the ball passing routes respectively. The number of passing routes reflects the player's importance in a match. Moreover, the direction of arrows shows a player's function such as forward and center. We identify the network's pattern both in space and time dimension.

Problem 2 requires us to explore indicators of successful teams in addition to scores and wins. We briefly summarize several indexes to describe Huskie's performance despite the final result. All indexes are come from full events data. There are thirty-eight matches in total. We only use half of them to set up a model. And the next half is used to evaluate the model's universality.

Problem 3 requires us to give Huskies' coach some suggestions. We analyze the network of problem 1 and consider all the success indicators. There are mainly three kinds of team style, which is defensive, offensive and balanced one. We will offer structural tactics depending on opponents' level. When the team is competing against low-level opponents, athletes often have more space and time to apply techniques because of less confrontation or even none. Since players can fulfill observation and judgement timely, the success rate of a technique is relatively high. On the contrary, high-level opponents oppress our players in space and time. Their strong defense ability causes the declination of our technique success rate.

Problem 4 requires us to widen the use of our model.   With the specialization process becomes popular, team work gains great popularity in the society. Since we have functions to evaluate the effect of a collaborative working system, it is easy to pick out the best combination. We generalize the characteristic of a large number of excellent teams; then give a model of team work which can apply universally. We need extra information if put into common use.

1.3 Core Tasks at hand

- Design a network for passing activities.
- Identify elements that can lead to success in addition to points.
- Offer some advice to coach on the strategy improvement of next season.

- Summarize our results and think further about team effort in other fields.

2 Assumptions

- All teams can play in the competition with all their strength, regardless of the weather conditions, good luck and other factors may have an impact on the outcome of the game.
- The skill level of the athletes playing football remains the same.
- The match results are absolutely fair, regardless of false balls, black whistles and other circumstances.
- The players can fully implement the coach's tactics and strategy.

3 Model Developments

Based on the passing data between players, we establish a directed ball flow network. We use circle node to represent players. The size of nodes is proportional to the degree of nodes. The arcs between two nodes link two players together, which width depends on the successful passes. We can intuitively understand the cooperation structure of Huskies by drawing this network. Also, we can make a preliminary qualitative analysis on the basis of it. On the left of Figure 1, it is the passing network between 14 players of the first match, which shows the closeness between different players. In this way, we can find the core player in team cooperation with simple analysis based on the degree of the nodes and the weight of the edges. For instance, Huskies_M1 in this match takes great responsibility in passing. On the right of Figure 1, it is the passing network between all players during the season. Compared with outer nodes, links between 14 inner nodes are closer. That is to say, a section of Huskies' 30 players participate in more team cooperation and contribute more in the match.
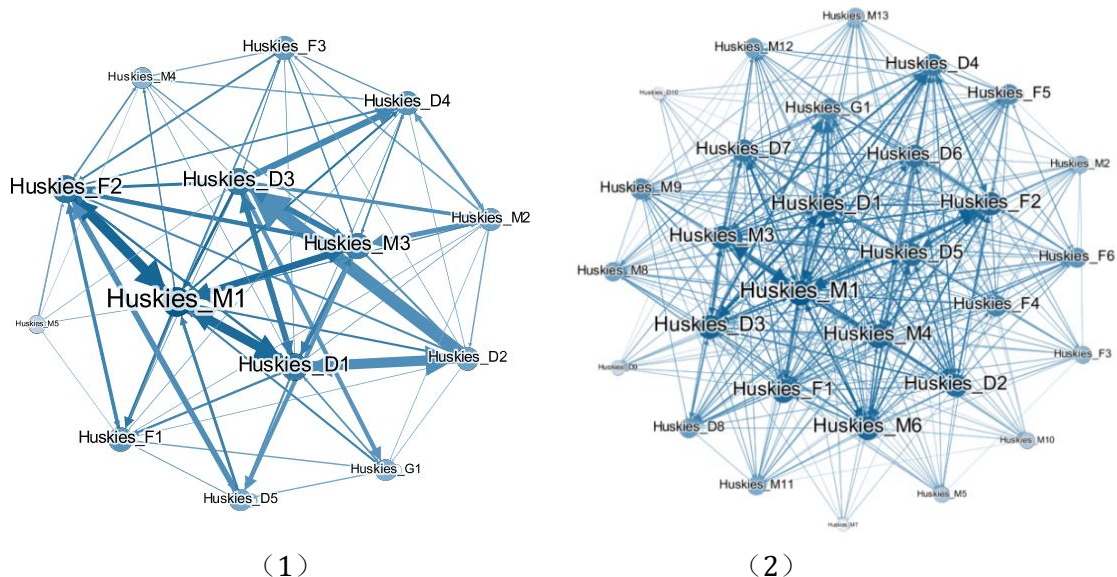

（1）   （2）

Figure 1: The ball flow network

3.1 From global to local: Identify Cooperation Structure

To depict the passing network between players, we need to extract possibly valuable accesses[1] for statistics to analyze the feature of passing event.

Now we must define a professional concept cooperation structure. It represents a short-time cooperation tactics between several players, which can be reflected by continuous soccer ball passing access. Unlike overall strategies in a match, cooperation structure more emphasizes the effect on the micro level.

Every cooperation structure is regarded as a unit. Continuous passes per unit is a significant standard describing structure characteristic. More passes will result in the lack of cooperation structure. If we choose fewer passes as initial screening conditions for cooperation structure, although we can get enough data, it cannot reflect obvious features. Considering the actual situation, we select all cooperation structure that passes is larger than 5 for quantitative process.

3.1.1 Selecting Accesses and Feature Vector Calculation

We find the feature vector that can describe the property of a cooperation structure.

$$\left[\bar{X}, \bar{Y}, avg\left(\sqrt{\Delta x_i^2 + \Delta y_i^2}\right), \rho, N, \bar{T} \right]$$

The definition of variables is as follows[2]:

Variable Name	Explanation
$\bar{X}$	the arithmetic center's horizontal coordinate, reflects the position of cooperation structure
$\bar{Y}$	the arithmetic center's vertical coordinate, reflects the position of cooperation structure
$avg\left(\sqrt{\Delta x_i^2 + \Delta y_i^2}\right)$	average passing distance, directly describes the size of cooperation structure
ρ	passes, reflects the scale and complexity of cooperation structure
N	number of passer, reflects the scale and complexity of cooperation structure
$\bar{T}$	average passing time[3], reflect the tempo of cooperation structure

We select 1352 accesses to represent typical cooperation structure. Since the change of field, it is easy to misjudge the average passing time. We get rid of seven abnormal accesses data and get 1345 accesses.

---

[1] When choosing the soccer ball movement track, we believe that continuous passing can reflect how players cooperate with each other. At first, we only plan to use passing events of the season. However, players' actions are changeable and coordinates of the ball are dispersed. Hence we are not able to get a complete ball track. Therefore, we finally decided to use full events for cooperation structure screening.

[2] There is no need to normalize coordinates as absolute location. We use the relative coordinates given by problem. It is more persuasive to design cooperation structure.

[3] Average passing time equals to the difference between the start time and the end time divided by the passing times.

3.1.2 Cluster and Dimension reduction

When analyzing the cooperation structure, we use hierarchical clustering method. It has perfect visual and explaining effect, but requires extremely high time complexity O(m2logm) and space complexity O(m2). Considering the amount of data is not so large, it is feasible to use hierarchical clustering. Besides, it can directly solve the K value for the convenience of cooperation structure classification.

- **Average-Linkage**: It calculates the distance between each data point and all the other ones in two cooperation structures. Then use the average as the distance between data of two cooperation structures. The disadvantage is enormous calculation amount. However, its result simulates the real soccer match better.
- **Distance Measurement Method**: Use Euclidean (two-norm) to calculate distance between points.

Calculate the distance between two clusters:

$$d(u, v) = \sum_{ij} \frac{d(u[i], v[j])}{(|u| * |v|)}$$

We get a clustermap by calculation. Number on the right side is the order number of each cooperation structure. Feature vectors are at the bottom of Figure 2. We used D and P for average passing distance and average number of passers respectively. Other variables' meaning is the same as figure 1 above. Cluster tree is on the left and upper part of the figure. The left one represents cooperation structure's cluster and the upper one represents feature vectors' cluster.
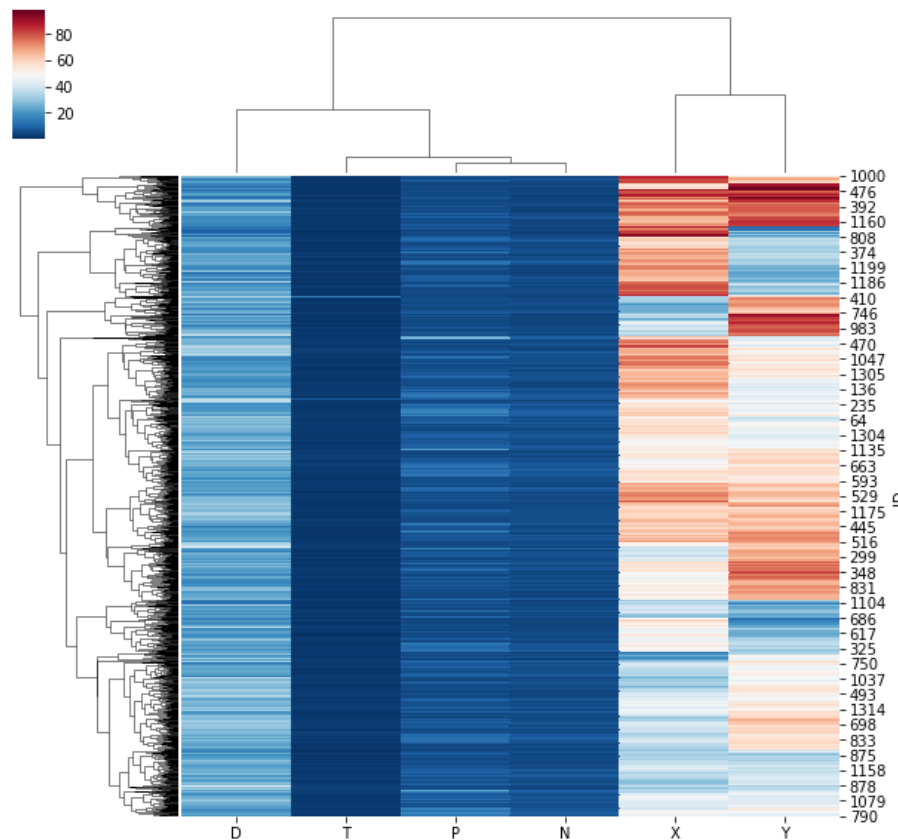

Figure 2: Clustermap

Since the data dimension is over three, it limits the visualization process of cluster results. We use principal component analysis to reduce the dimension. Utilizing two principal components with 95% variance, we draw the figure below:
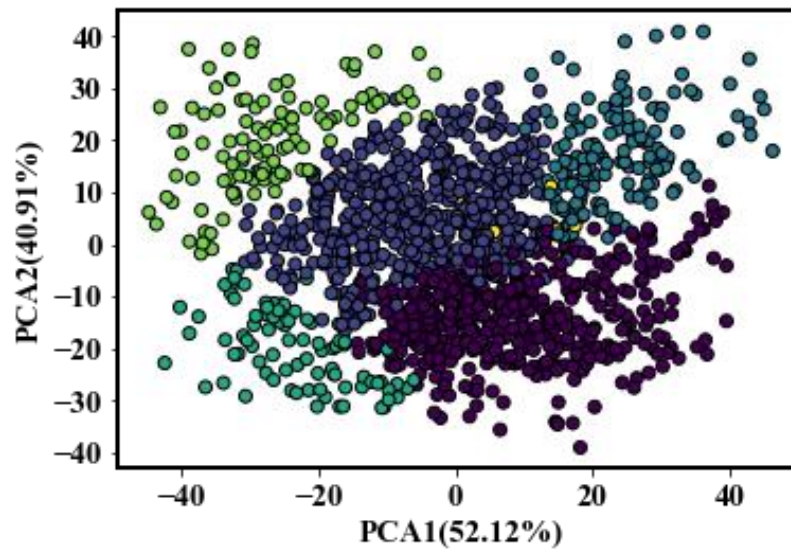

Figure 3: PCA

Link figure 2 and 3 together, our final cluster result is as following:
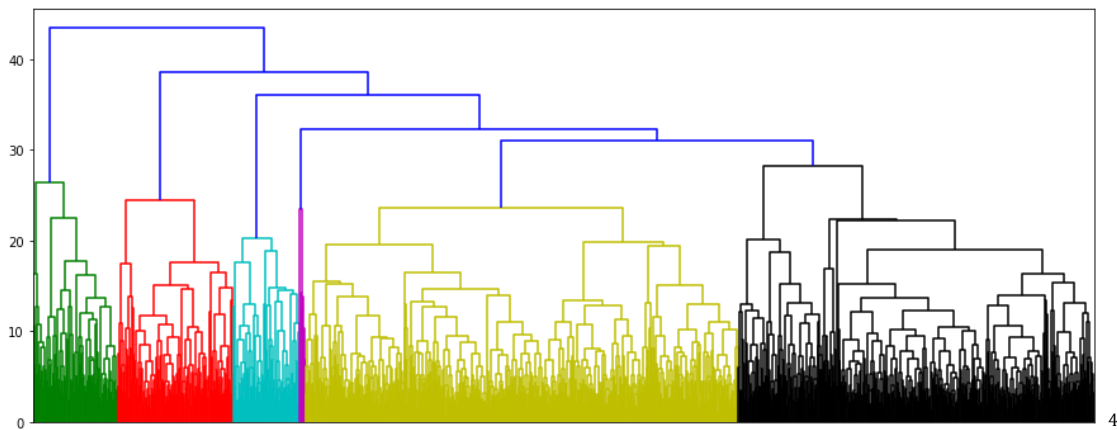

Figure 4: Cluster result

We summarize unique cooperation structure after analyzing their features. In figure 5, every circle's color represents the same cooperation structure as figure 4.

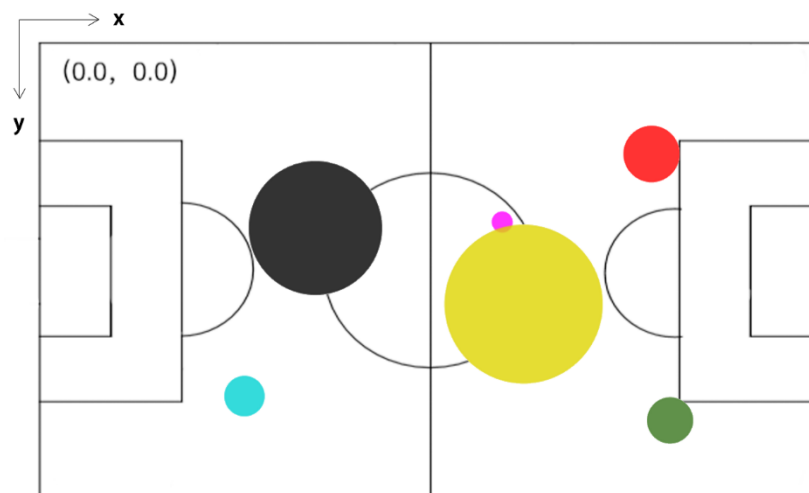4 We omit the ID below because of cluster density. It is the same as

Figure 5: The distribution of cooperation structure

Its location means where the average place a cooperation structure located. The size of a circle stands for the appearing times of a cooperation structure.

3.2 The Features of Different Cooperation Structures

The figure below shows the feature vector's value of six cooperation structure. We can directly generalize some characteristic to use in real soccer games. For instance, the average passes and average number of passers of the first cooperation structure are both small. Moreover, it costs the least average time to pass. Thus, we can infer that this cooperation structure is suitable for fast and short passes. It often appears near the opponent's goal and requires players' strong explosiveness.
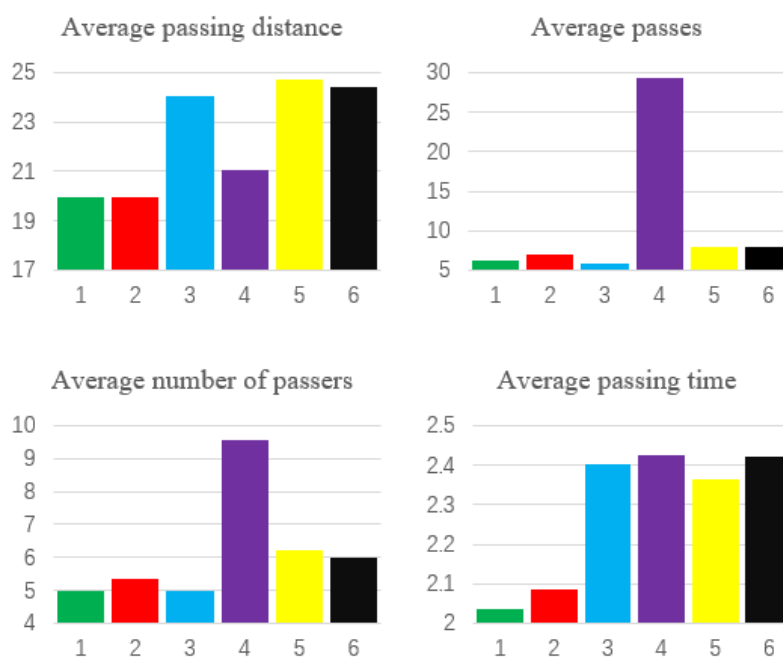

Figure 6: The analysis of cooperation structure

Combine all the analysis above, we get six excellent cooperation structures. Nodes and edges represent players and accesses between them respectively.

- **Winger Machine Gun**

  This cooperation structure is not outstanding in average passing distance and average number of passers. However, it costs a really short average passing time. We can infer that players are required to run very fast. The location is exactly beside opponent's goal. If Huskies can break through their defense system effectively, it is easy to seek shot opportunities.
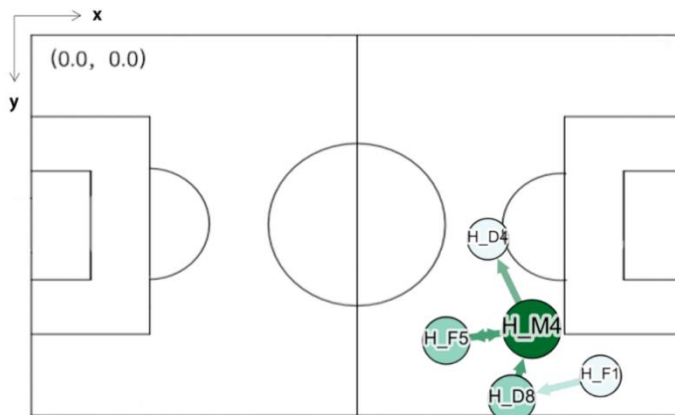
Match ID	30
Team ID	Huskies
Event Time5	2H 2017.65

- **Sidewalk Tiger**

This kind of cooperation structure is similar to the above one except for location. They are both offensive cooperation structure. The core player attacks from the left sidewalk.
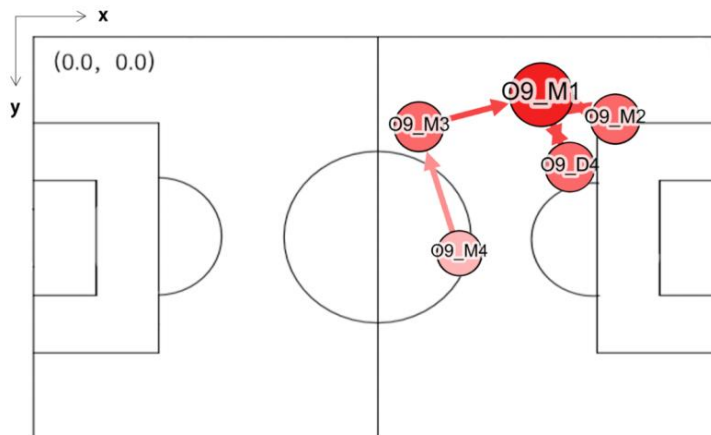
Match ID	9
Team ID	Opponent 9
Event Time	2H 2077.539009

- **Defensive Shield**

When opponents are very close to Huskies' goal, players can turn from a offensive state to a defensive one. This cooperation structure requires long average passing distance and time. That is to say, rear guard may be passing in the backfield to adjust the tempo. Or it includes passing back to slow down opponents' fierce offend.

---

5 On the right side of Figure 8 is the information table of this kind of cooperation structure.
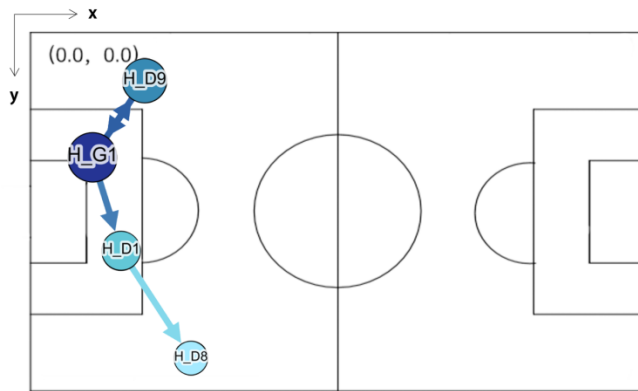
Match ID	27
Team ID	Huskies
Event	1H
Time	1991.672292

- **Absolute Suppress**

    This cooperation structure is relatively special compared with others. In full events data, it only appears seven times. A star player with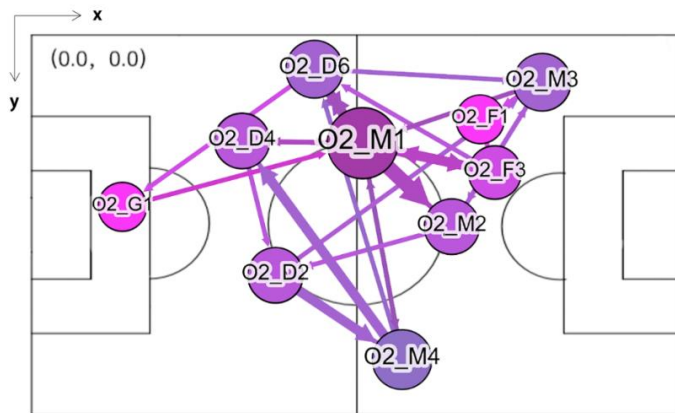 strong ability is the core. The whole structure almost involves all players and their action range is really wide. It only happens when one side wholly control the ball.

Match ID	32
Team ID	Opponent 2
Event Time	1H 883.854199

- **Midfield Overall Planning**

This cooperation structure can adjust the tempo of a team, and is used most frequently. Its average passing distance and average passing time are both large, but smaller than Absolute Suppress. It usually appears when players are seeking for offending opportunities.
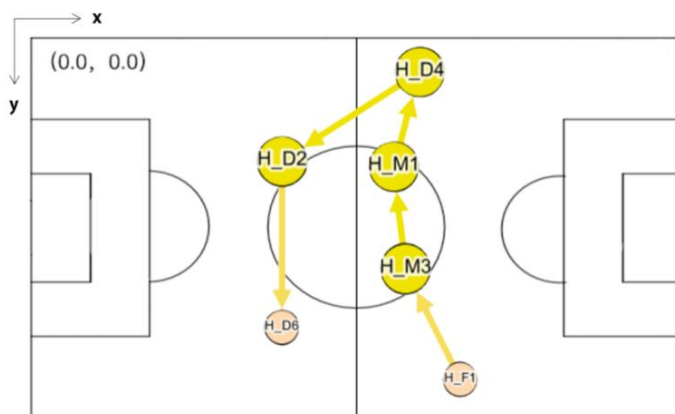
Match ID	1
Team ID	Huskies
Event Time	1H 286.49734

- **Backfield Location Adjustment**

It is almost the same as *Midfield Overall Planning* except for the location. Because opponents' location is very close to Huskies' goal in this cooperation structure, it is im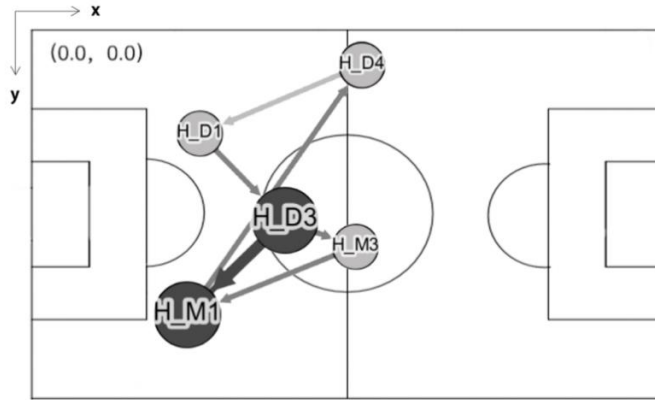portant to choose players with high-level defense ability. All players should be very careful about opponents' offensive actions in this process

Match ID	37
Team ID	Huskies
Event Time	1H 1808.786016

4 Performance Evaluation & Structure Characterization

4.1 Selection of performance indicators

4.1.1 Average Node Connectivity

Considering the concept of the *average node connectivity* of a graph, we define it to be the average, over all pairs of vertices, of the maximum number of internally disjoint paths connecting these vertices.[6] It is obvious fact that the connectivity $k(G)$ is equal to $Min\{kG(u,v): u, v \in V(G)\}$. When the order of G is n, then the average node connectivity of G, denoted $\bar{k}(G)$, can be described as

$$\bar{k}(G) = \frac{\sum_{u,v} k_G(u,v)}{\binom{n}{2}}$$

Edmonds-Karp algorithm is used to calculate the maximum flow between a pair of nodes. Average node connectivity reflects coordination between players, and if each member of a team has a high level of participation, then the overall level of connectivity is high. It can reflect whether the players are widely involved or not, and the value is lower if some players are playing alone. This also shows the comparison of players' contribution. If the whole team only passes the ball to a couple of star players to fight, rather than play together, the value is low.

---

[6] Beineke, L., O. Oellermann, and R. Pippert (2002). The average connectivity of a graph. Discrete mathematics 252(1-3), 31-45.

4.1.2 Transitivity

Football games are through clever tactical cooperation to get more offensive opportunities, and the ever-changing tactical cooperation is always inseparable from the cooperation of three people. Because in soccer, if we want to form an effective attack, is often to form a formation in somewhere of the field with the majority against the minority, and in defense also to form such a formation. According to the statistics of all the goals scored in the 15th World Cup in the United States, nearly 60% of the goals were scored locally through the cooperation of three players. Here are some typical triadic configurations, in which nodes represent players and arcs represent passing paths.
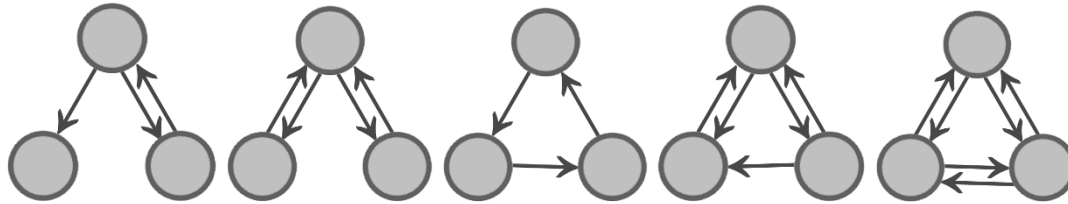
Typical Triadic Configuration

Such a triadic configuration is often represented as triangles and triads (two edges with a shared vertex) in the diagram. So we define *transitivity* as

$$Transitivity\ =\ \#triangles\ +\ \#triads$$

In this equation, $\#triangles$ the number of triangles in the passing network at the current moment, and $\#triads$ represents the number of triads. As we see in our previous analysis, the higher the value, the more transitive and flexible the team is.

4.2 Sliding Window Model

Performance indicator is often valuable in a whole soccer game. In addition to wins and goals, we propose another two indicators.

It is troublesome to find key details from the overall indicator while partial indicators can solve the puzzle. Since the match result also depends on opponents' ability level largely, we cannot evaluate Huskies' concrete performance through goals effectively. Actually we are more concerned about how indicators change over time. From the changes we can discover a team's real-time match state. If performance indicator changes fiercely in a certain time period, we can locate what happened by full events data. After exploring the structure and adjustment, we can select some configuration patterns full of team characteristics.

The indicators we propose both meet the requirement of partial analysis.

We use the Sliding Window Model to judge partial performance. The window diagram is as follows:
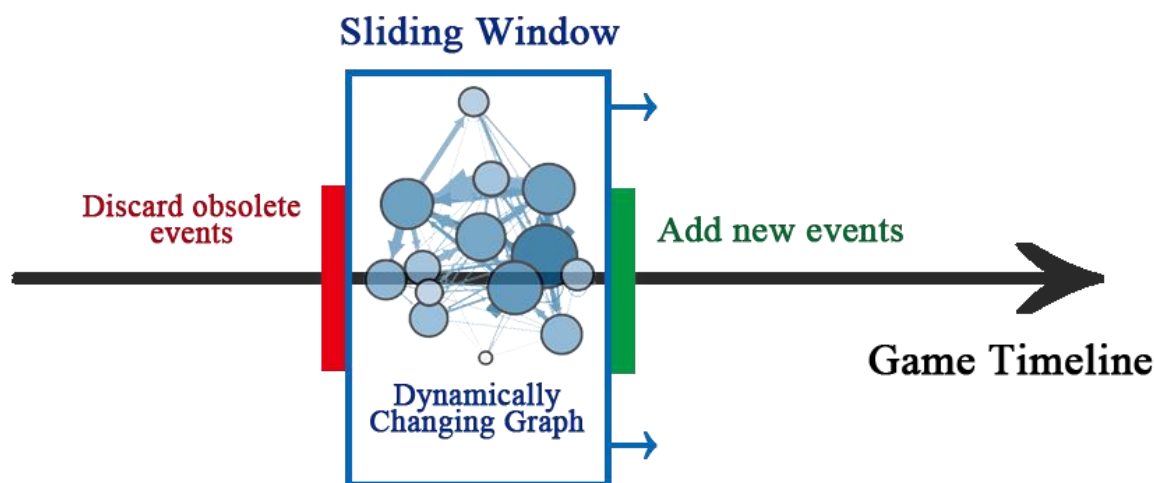
To start with, we make a metaphor to explain the concept of *sliding window*. When we take a distant view photograph, it is often too large to see the details clearly. Then the most common method is sliding from left to right on the screen.

The sliding window concept is the same. As the window slides on the game timeline, the right side adds new passing events continuously. Meanwhile, the left side discards some obsolete passing events to obtain the window's size[7]. These events directly influence how the directed graph looks like in the window. Since different graph types have different performance, we can find out its changing process over time.

It is obvious that the size of sliding window t is a significant parameter. Too large size goes against the concept of partial indicator while smaller ones cannot clarify the situation clearly. Standard soccer game lasts about 5400 seconds (90 minutes). Average soccer player often get into competition state after 15 minutes. Considering about the objective fact, we choose 900 second (one third of Halftime, 15 minutes) as sliding window size.

Based on this model, time between 0 and t is the process of network establishment, so there is no data generation. From time t to the end of match[8], it generates data consistently.

Next we utilize the sliding window model to capture team's highlight moment, which is helpful to locate team structure and configuration.

4.2.1 Average Node Connectivity

Match ID	Opponent ID	Result[9]
15	15	2:0

---

[7] Ideal sliding window does not change in size. Since the database is made up with time points, it may change slightly around the specified size.

[8] Since the database is made up with time points, this sentence means every event larger than t generates data.

[9] The result format: Huskies: Opponent. It is the same hereinafter.
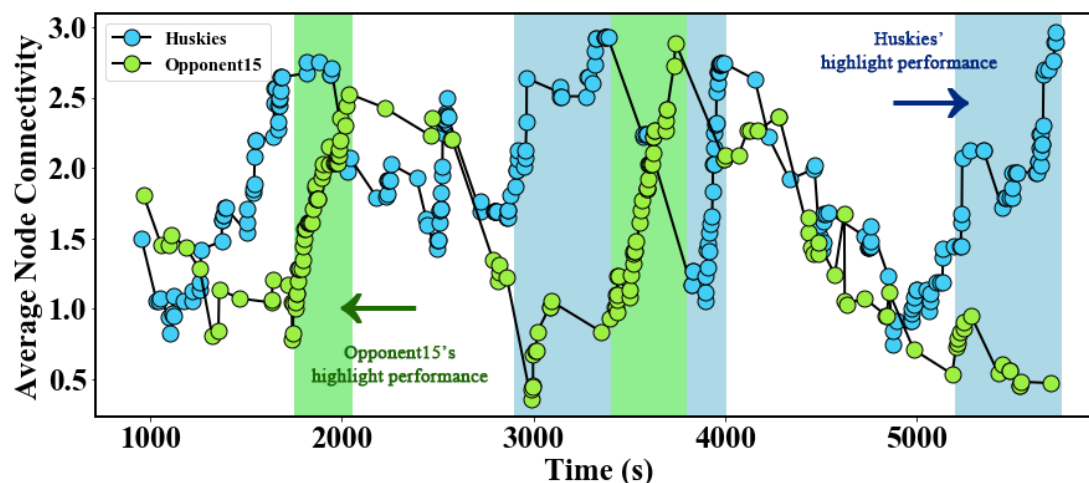
The Average Node Connectivity represents every player's involvement degree at the moment. The figure above shows that Huskies' highlight performance time is obviously more than the opponent. Opponent 15's highlight time does not last so long while Huskies' connectivity almost above average level. We can get empirical support from the result 2:0.

4.2.2 Transitivity

We use transitivity to depict the players' flexibility. High transitivity implies perhaps excellent cooperation happens at the moment.

The figure below shows a match between Huskie and Opponent 4. Over the match process, almost all Huskies' transitivity value is smaller than the opponent. Moreover, opponent 4's highlight time is longer than Huskies at a large scale. Thus, it is easy to explain why the result is so cruel.

Failure result can give a lesson to Huskie. We can locate according to the happening time and collect opponent 4's highlight cooperation structure in order to prepare for the next match.

Match ID	Opponent ID	Result
23	4	0:4
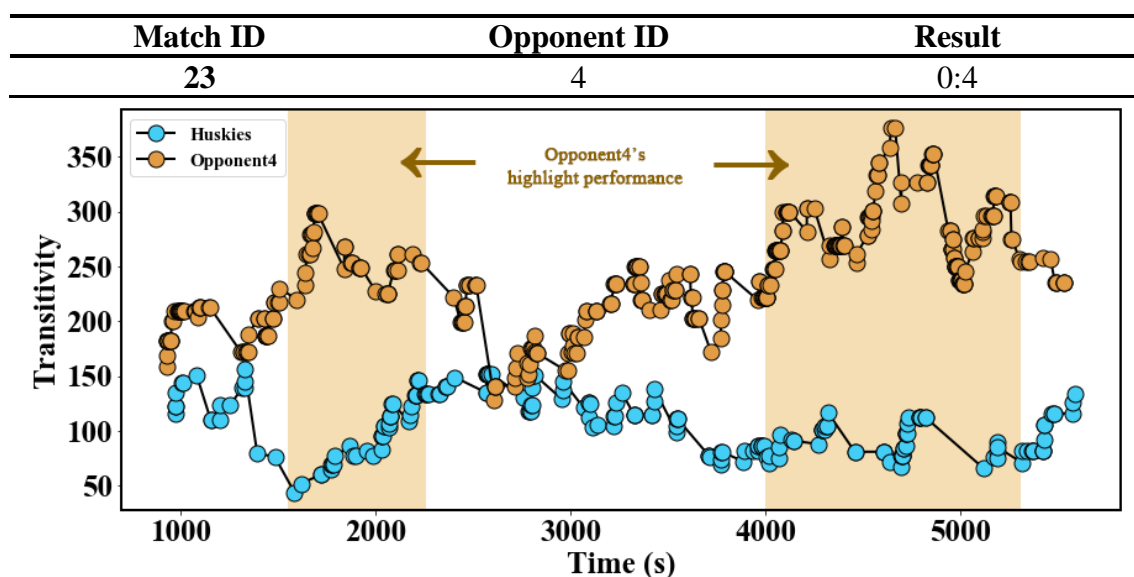
5. Characteristic Analysis and Structural Advice

5.1 Universality Analysis Based on Performance

In this part we will analyze Huskies' characteristics and offer some advice on structure.

The figure on the right side shows the difference between Huskies and other opponents on the use of cooperation structure. The two green bars are above horizontal axis, showing the high using frequency. Tactics three and five represents Midfield Overall Planning and Defensive Shield respectively. Thus, we can infer that Huskies often use defensive counterattack strategy in this season. In professional soccer pattern, this is called to be *Long Ball*. Players are waiting for opportunities to break through opponents' goal. When the time is ripe for an easy long ball, forward can control the ball and shot.

After exploring Huskies' common use cooperation structure, we need to find the most suitable one. This figure shows the difference between Huskies and other opponents on the use of cooperation structure during the highlight period.

This figure has obvious result and is more helpful to work out overall strategies. We can infer that Huskies do well in structure two, three and five, which means Sidewalk Tiger, Defensive Shield and Midfield Overall Planning respectively. When Huskies uses them, highlight performance appears more easily.

Looking back on the figure below, we discover that Huskies is more proficient in backfield and mid-forth field while really weak in mid-back field. Hence Huskies is right to use counter attack and long ball this season.

Structure Frequency

5.2 Our Advice

- Reinforce to develop the skill of counter attack and long ball in order to produce the best possible results.
- From the Highlight Performance figure, *Sidewalk Tiger*'s performance is also slightly higher than the average of whole season. However, the using times of this cooperation structure is smaller than the average. Thus, we think it is possible to make a break-through in sidewalk attack.
- Squad: $4 - 4 - 2$[10]

Highlight Structure Frequency

---
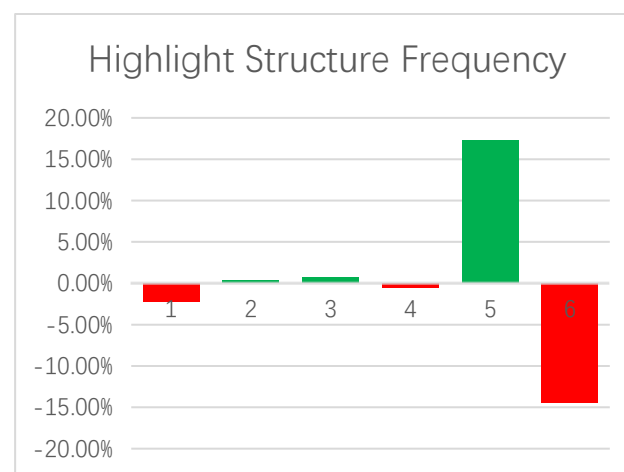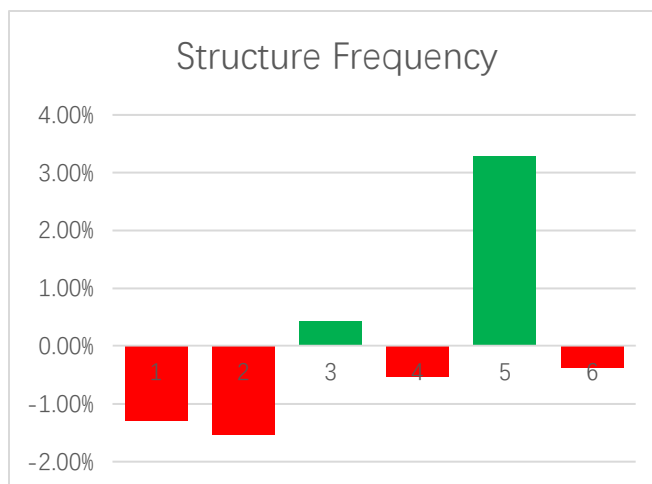
[10] We use PageRank to calculate the participated player.
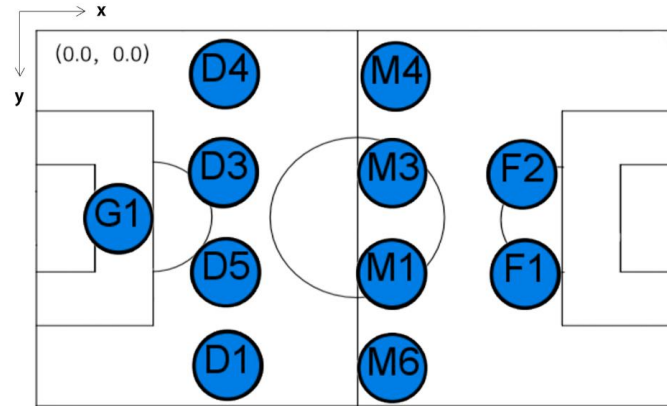
- **A Warning on Special Opponent & cooperation structure:**

In our cooperation structure, Absolute Suppress is a really special one. Its passing times is very large and with a wide passing range. Almost all players involve in this structure, which represents a kind of champion grace. Huskies' ability cannot meet this structure's requirement. However, opponents using Absolute Suppress can be extremely cruel to Huskies. Only in the match 16, Huskies end in tie. Another three matches are all bitter failure. Especially in the match 32 (indicator figures are as follows), opponent 2 create four times of this structure and Huskies' tempo is wholly destroyed. Huskies' Average Node Connectivity and Transitivity are both extremely low. On the contrary, opponent 2's indicators are very high, which means they wholly control the ball. Therefore, coaches should pay full attention to precaution opponents' Absolute Suppress structure.
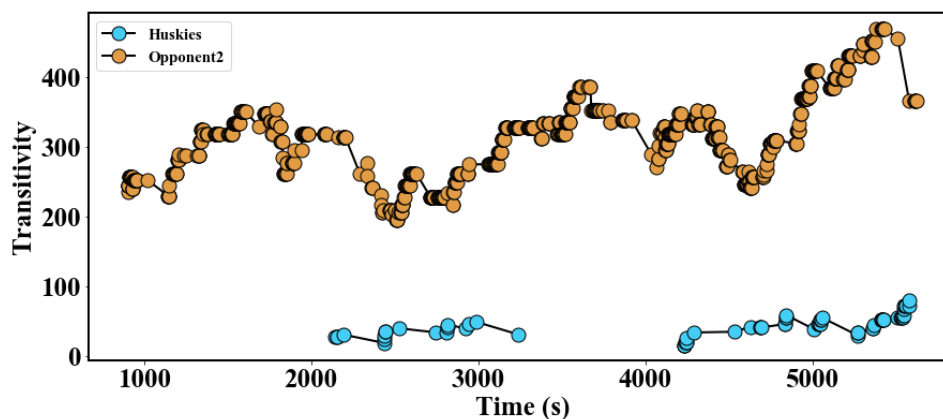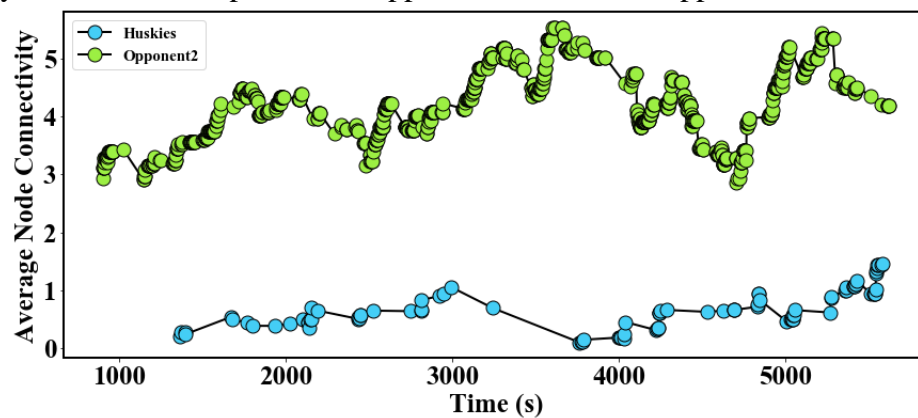
6. how to design more effective teams

6.1 Key elements of designing effective teams

If we want to design an effective team, the first step is expected to be finding the basic pattern of team cooperation. The collaboration between team members may be seem messy at first glance , which is difficult to find useful messages. So it is important to find the typical features of cooperation among team members. Like the football team analyzed in this article, the players have a variety of actions on the field, but passing is the most common and typical interaction among the players, so we can first analyze information about passing. By the same token, other teams can be treated it in a similar way.

For example, a research group whose main goal is writing paper of high quality , its members may have multiple communication each day, but the most typical cooperation of them is to share the task and progress of the paper. Therefore, in order to improve the team effectiveness of the research team, it is necessary to first count the task documents they shared during the study period and further study their collaborative characteristics.

When thinking about team strategy, in addition to considering the overall changes in parameter indicators, using windows to capture changes in indicators at each moment may give more enlightening conclusions. Windows are useful for small-minded, and can be more accurately identified for the team-friendly construction patterns. Making good use of this model and its ideology is beneficial to our team efficiency.

6.2 Other aspects of teamwork we want to know

If we can capture the reaction and characteristics of teamwork after external interference, a more comprehensive model of team performance may be established. Besides, to enhance the measurement accuracy of performance indicators, more information about successful teams' characteristics is required (e.g. If there is more information on goalscoring, we can identify more features about paths of success and then let people follow suit).

7 Strengths and Weaknesses

In future work, we would like to address the weaknesses of our model and improve on the computational efficiency of our simulations. This model had a number of strengths and weaknesses.

7.1 Strength

- **Our model innovates in designing Sliding Window Model.**
  Like distant view photograph, soccer teams can analyze their performance of each time period. Then design strategies of macro level based on micro data.
- **Our model is robust and with great generalization ability.**
  The parameters are flexible, which means the model can capture differen fields' data characteristics other than soccer game.

- **Our model has excellent visualization figures.**
  To identify the typical cooperation structure, we draw six figures on the basis of real data. They reflect the tactics intuitively.

7.2 Weakness

- **The arithmetic is large in time complexity.**
  Since the figure reconstitutes partly at every time point, the model should be used cautiously when data amount is enormous.
- Slight distortion in the processing of sliding windows
  Since the data set is a point in time, a slight change in window size occurs.

8   References

[1]Duch J, Waitzman J S, Amaral L A N. Quantifying the performance of individual players in a team activity[J]. PloS one, 2010, 5(6).

[2]GÜRSAKAL N, YILMAZ F M, ÇOBANOĞLU H O, et al. Network motifs in football[J]. Turkish Journal of Sport and Exercise, 2018, 20(3): 263-272.

[3]Buldu, Javier & Busquets, J. & Echegoyen, Ignacio & Seirul.lo, F.. (2019). Defining a historic football team: Using Network Science to analyze Guardiola's F.C. Barcelona. Scientific Reports. 9. 10.1038/s41598-019-49969-2.

[4]Cintia P , Rinzivillo S , Pappalardo L . A network-based approach to evaluate the performance of football teams[C]// Machine Learning and Data Mining for Sports Analytics workshop (MLSA'15), ECML/PKDD conference 2015. 2015.

[5]Lloyd Smith, Bret Lipscomb, and Adam Simkins. 2007. Data mining in sports: predicting Cy Young award winners. J. Comput. Sci. Coll. 22, 4 (April 2007), 115–121.

[6]Clemente, F. M., Couceiro, M. S., Martins, F. M. L., & Mendes, R. S. (2015). Using Network Metrics in Soccer: A Macro-Analysis. Journal of Human Kinetics, 45(1), 123–134.

9 Appendices

9.1 Codes

```python
import networkx as nx
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import matplotlib.patches as mpathes

def InitG(data):
    G = nx.DiGraph()
    ET = np.array(data['EventTime'])
    i=0
    while(ET[i] < 900):
        i+=1
    length = i                          #未来的 front

    Weight = np.ones(length, dtype=int)
    Ori = np.array(data['OriginPlayerID'])
    Des = np.array(data['DestinationPlayerID'])
    for j in range(0,length):
        if(G.has_node(Ori[j])&(G.has_node(Des[j]))):
            if(G.has_edge(Ori[j],Des[j])):
                G[Ori[j]][Des[j]]['weight']+=1
            else:
                G.add_edge(Ori[j],Des[j],weight=Weight[j])
        else:
            G.add_edge(Ori[j],Des[j],weight=Weight[j]) #边的起点，终点，权重
#       #pos = nx.circular_layout(G)
#       #print(G.edges(data=True))
#
#nx.draw_networkx_edges(G,pos,with_labels=True,edge_color='black',alpha=1,font_size=10,width=
[float(v['weight']*0.5) for (r,c,v) in G.edges(data=True)])
#       #nx.draw_networkx_nodes(G,pos,with_labels=True)
#       nx.draw(G,pos,alpha = 0.8,with_labels=True,font_size=10,width=[float(v['weight']*1) for
(r,c,v) in G.edges(data=True)])

#       plt.show()
    data = data[(data['MatchPeriod']=='1H')]
return G,length,np.array(data['EventTime'])[len(data['EventTime'])-1]
```

```
def Calc1(String1,String2,MatchID):
    data = pd.read_excel('gogofootball.xlsx',sheet_name='Sheet1')
    data = data[(data['TeamID']==String1)&(data['MatchID']==MatchID)]

    G,edgeNum,last1H = InitG(data)
    # 时间数据处理
    data['EventTime'][(data['MatchPeriod']=='2H')]+=last1H

    front = edgeNum                # 前指针
    rear =0                        # 后指针
    zongchang = len(data['TeamID'])

    #Degree = []                    # 图的密度
    #Clustering = []                # 图或网络中节点的聚类系数
    Transitivity = []              # 图或网络的传递性
    T1 = []                        # 时间轴
    Triangles = []
    Triads = []
    Anc = []

    ET = np.array(data['EventTime'])
    Ori = np.array(data['OriginPlayerID'])
    Des = np.array(data['DestinationPlayerID'])

    while(front<zongchang):
        t = ET[front]
        T1.append(t)

        # 删东西
        while(ET[front] - ET[rear] > 900):
            if(ET[front] - ET[rear] < 900):
                break
            if(G[Ori[rear]][Des[rear]]['weight']>1):
                G[Ori[rear]][Des[rear]]['weight']-=1
            else:
                G.remove_edge(Ori[rear],Des[rear])
                if(G.degree(Ori[rear])==0): G.remove_node(Ori[rear])
                if(G.degree(Des[rear])==0): G.remove_node(Des[rear])
            rear+=1

        # 加东西
        if(G.has_node(Ori[front])&(G.has_node(Des[front]))):
            if(G.has_edge(Ori[front],Des[front])):
                G[Ori[front]][Des[front]]['weight']+=1
            else:
                G.add_edge(Ori[front],Des[front],weight=1)
        else:
            G.add_edge(Ori[front],Des[front],weight=1)
```

```
        # 计算并附加
        #Degree.append(nx.degree(G))
        #Clustering.append(nx.clustering(G))


#            tri = np.sum(list(nx.triangles(G).values()))/3
#            triads = (tri*3)/nx.transitivity(G)
#            Transitivity.append(tri + triads)

        Anc.append(nx.average_node_connectivity(G))
        # 步长
        front+=1
#    return T1,Transitivity
return T1,Anc






def Calc2(String1,String2,MatchID):
    data = pd.read_excel('gogofootball.xlsx',sheet_name='Sheet1')
    data = data[(data['TeamID']==String2)&(data['MatchID']==MatchID)]

    G,edgeNum,last1H = InitG(data)
    # 时间数据处理
    data['EventTime'][(data['MatchPeriod']=='2H')]+=last1H

    front = edgeNum            # 前指针
    rear =0                    # 后指针
    zongchang = len(data['TeamID'])

#      Degree = []                    # 图的密度
#      Clustering = []            # 图或网络中节点的聚类系数
    Transitivity2 = []          # 图或网络的传递性
    T2 = []                     # 时间轴
    Triangles = []
    Triads = []
    Anc = []

    ET = np.array(data['EventTime'])
    Ori = np.array(data['OriginPlayerID'])
    Des = np.array(data['DestinationPlayerID'])

    while(front<zongchang):
```

```
        t = ET[front]
        T2.append(t)

        # 删东西
        while(ET[front] - ET[rear] > 900):
            if(ET[front] - ET[rear] < 900):
                break
            if(G[Ori[rear]][Des[rear]]['weight']>1):
                G[Ori[rear]][Des[rear]]['weight']-=1
            else:
                G.remove_edge(Ori[rear],Des[rear])
                if(G.degree(Ori[rear])==0): G.remove_node(Ori[rear])
                if(G.degree(Des[rear])==0): G.remove_node(Des[rear])
            rear+=1

        # 加东西
        if(G.has_node(Ori[front])&(G.has_node(Des[front]))):
            if(G.has_edge(Ori[front],Des[front])):
                G[Ori[front]][Des[front]]['weight']+=1
            else:
                G.add_edge(Ori[front],Des[front],weight=1)
        else:
            G.add_edge(Ori[front],Des[front],weight=1)

        # 计算并附加
        #Degree.append(nx.degree(G))
        #Clustering.append(nx.clustering(G))


#           tri = np.sum(list(nx.triangles(G).values()))/3
#           triads = (tri*3)/nx.transitivity(G)
#           Transitivity2.append(tri + triads)


        Anc.append(nx.average_node_connectivity(G))
        # 步长
        front+=1
#       return T2,Transitivity2
return T2,Anc




def Calc(String1,String2,MatchID):
#       T1,Transitivity = Calc1(String1,String2,MatchID)
#       T2,Transitivity2 = Calc2(String1,String2,MatchID)
```

```
        T1,Anc1 = Calc1(String1,String2,MatchID)
        T2,Anc2 = Calc2(String1,String2,MatchID)
#       return T1,T2,Transitivity,Transitivity2
return T1,T2,Anc1,Anc2




# Main Program
String1 = 'Huskies'
String2 = 'Opponent18'
MatchID = 18
#T1,T2,Transitivity,Transitivity2 = Calc(String1,String2,MatchID)
T1,T2,Anc1,Anc2 = Calc(String1,String2,MatchID)

plt.figure(figsize=(15, 6))
# 线条颜色 black, 线宽 2, 标记大小 13, 标记填充颜色从网上找 16 进制好看的颜色
plt.plot(T1, Anc1, '-o', color='black', markersize=13, markerfacecolor='#44cef6', linewidth=2,
label=String1)
# plt.plot(T2, Anc2, '-o', color='black', markersize=13, markerfacecolor='#e29c45', linewidth=2,
label=String2)
plt.plot(T2, Anc2, '-o', color='black', markersize=13, markerfacecolor='#9cef43', linewidth=2,
label=String2)

font = {'family': 'Times New Roman', 'weight': 'normal', 'size': 15}
plt.legend(prop=font)

# 字体设置: 字体名称 Times New Roman, 字体大小 34
font_format = {'family':'Times New Roman', 'size':25}
plt.xlabel('Time (s)', font_format)

#plt.ylabel('Transitivity', font_format)
plt.ylabel('Average Node Connectivity', font_format)
# 设置坐标轴 x 范围 0~3*pi, y 范围-1.2~1.2

#plt.axis([0, 3*np.pi, -1.2, 1.2])

# 横纵坐标上的字体大小与类型(不是 xlabel, 是 xticks)
plt.xticks(fontproperties='Times New Roman', size=25)
plt.yticks(fontproperties='Times New Roman', size=25)
# 整个图像与展示框的相对位置
plt.subplots_adjust(left=0.19,right=0.94, bottom=0.13)
# 调整上下左右四个边框的线宽为 2
ax=plt.gca()
ax.spines['bottom'].set_linewidth(2)
ax.spines['left'].set_linewidth(2)
```

```
ax.spines['right'].set_linewidth(2)
ax.spines['top'].set_linewidth(2)


# rect1 = mpathes.Rectangle((1550,0),700,400,color='wheat')
# ax.add_patch(rect1)

# # plt.text(1800,350,'Opponent4's highlight
performance',fontsize=15,verticalalignment="center",horizontalalignment="center")
# rect2 = mpathes.Rectangle((4000,0),1300,400,color='wheat')
# ax.add_patch(rect2)


# rect1 = mpathes.Rectangle((1750,0),300,3.1,color='lightgreen')
# ax.add_patch(rect1)
# rect2 = mpathes.Rectangle((2900,0),500,3.1,color='lightblue')
# ax.add_patch(rect2)
# rect3 = mpathes.Rectangle((3400,0),400,3.1,color='lightgreen')
# ax.add_patch(rect3)
# rect4 = mpathes.Rectangle((3800,0),200,3.1,color='lightblue')
# ax.add_patch(rect4)
# rect5 = mpathes.Rectangle((5200,0),550,3.1,color='lightblue')
# ax.add_patch(rect5)

plt.show()
```

_____cluster↓ _____

```
#  找到传球路径，提取特征值，保存为 Excel
import networkx as nx
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

MID = []
TEAM = []
ET = []
X = []
Y = []
D = []
P = []
N = []
T = []

HOWLONGISQUEQUE = 5
```

```python
data = pd.read_excel('full.xlsx',sheet_name='Sheet1')

MatchID = np.array(data['MatchID'])
TeamID = np.array(data['TeamID'])
EventTime = np.array(data['EventTime'])
EventType = np.array(data['EventType'])
OriginPlayerID = np.array(data['OriginPlayerID'])
DestinationPlayerID = np.array(data['DestinationPlayerID'])
Ox = np.array(data['EventOrigin_x'])
Oy = np.array(data['EventOrigin_y'])
Dx = np.array(data['EventDestination_x'])
Dy = np.array(data['EventDestination_y'])

i = 0
while(i < len(MatchID) - HOWLONGISQUEQUE):
    #找到第一个符合条件的（PASS）
    if(EventType[i]!='Pass'):
        i+=1
        continue
    flag = True
    #筛选连续出现 HOWLONGISQUEQUE 次的
    for j in range(1,HOWLONGISQUEQUE):
        if(MatchID[i]!=MatchID[i+j] or TeamID[i]!=TeamID[i+j] or EventType[i]!=EventType[i+j]
or str(DestinationPlayerID[i+j])=='nan'):
            flag = False
            break

    if(flag==False):
        i+=1
        continue
    else:
        #此时 i 是路径的开始，找到路径末尾。
        for k in range(HOWLONGISQUEQUE-1,200):
            if(MatchID[i]!=MatchID[i+k] or TeamID[i]!=TeamID[i+k] or
EventType[i]!=EventType[i+k] or str(DestinationPlayerID[i+k])=='nan'):
                move = k
                break
        #此时 i 是路径的开始，i+move 是路径的结尾+1，move 是路径的长度。
        #print(i,i+move,move)
        MID.append(MatchID[i])
        TEAM.append(TeamID[i])
        ET.append(EventTime[i])
        P.append(move)
        T.append((EventTime[i+move-1]-EventTime[i])/move)
        sumX = sumY = sumD = 0
        Peo = set()
        for l in range(i,i+move):
            sumX += Ox[l]+Dx[l]
            sumY += Oy[l]+Dy[l]
```

```
            sumD += ((Dy[l] - Oy[l])**2 + (Dx[l] - Ox[l])**2)**0.5
            Peo.add(OriginPlayerID[l])
            Peo.add(DestinationPlayerID[l])
#           if(np.nan in Peo):
#               Peo.remove(np.nan)
        sumX+=Ox[i]
        sumY+=Dy[i+move-1]
        X.append(sumX/(move*2))
        Y.append(sumY/(move*2))
        D.append(sumD/move)
        N.append(len(list(Peo)))

        i+=move

#print(MID,TEAM,ET,X,Y,D,P,N,T)
#print(N)
#arr1 = np.array(MID,TEAM,ET,X,Y,D,P,N,T)
# columns=['MID','TEAM','ET','X','Y','D','P','N','T'],
temp = pd.DataFrame(data=[MID,TEAM,ET,X,Y,D,P,N,T])
temp.to_excel('xzz.xlsx')
```

```
import pandas as pd
import seaborn as sns    #用于绘制热图的工具包
from scipy.cluster import hierarchy    #用于进行层次聚类，话层次聚类图的工具包
from scipy import cluster
import matplotlib.pyplot as plt
from sklearn import decomposition as skldec #用于主成分分析降维的包

df = pd.read_excel("xzz - test.xlsx",index_col=0)
Z = hierarchy.linkage(df, method ='average',metric='euclidean')
label = cluster.hierarchy.cut_tree(Z,height=30)
label = label.reshape(label.size)
label
#lable = np.array(label)
temp = pd.DataFrame(data=[label])
temp.to_excel('label.xlsx')
```

```python
import pandas as pd
import seaborn as sns    #用于绘制热图的工具包
from scipy.cluster import hierarchy    #用于进行层次聚类，话层次聚类图的工具包
from scipy import cluster
import matplotlib.pyplot as plt
from sklearn import decomposition as skldec #用于主成分分析降维的包

plt.figure(figsize=(15, 6))
df = pd.read_excel("xzz - test.xlsx",index_col=0)
Z = hierarchy.linkage(df, method ='average',metric='euclidean')

hierarchy.dendrogram(Z,labels = df.index)


plt.show()

label = cluster.hierarchy.cut_tree(Z,height=30)
label = label.reshape(label.size)
```

```python
#根据两个最大的主成分进行绘图
pca = skldec.PCA(n_components = 0.95)       #选择方差 95%的占比
pca.fit(df)     #主城分析时每一行是一个输入数据
result = pca.transform(df)    #计算结果
plt.figure()    #新建一张图进行绘制
plt.scatter(result[:, 0], result[:, 1], c=label, edgecolor='k') #绘制两个主成分组成坐标的散点图

# for i in range(result[:,0].size):
#        plt.text(result[i,0],result[i,1],df.index[i])        #在每个点边上绘制数据名称

x_label = 'PCA1(%s%%)' % round((pca.explained_variance_ratio_[0]*100.0),2)    #x 轴标签字符串
y_label = 'PCA2(%s%%)' % round((pca.explained_variance_ratio_[1]*100.0),2)    #y 轴标签字符串


# 字体设置: 字体名称 Times New Roman, 字体大小 34
font_format = {'family':'Times New Roman', 'size':15}
plt.xlabel(x_label, font_format)       #绘制 x 轴标签
plt.ylabel(y_label, font_format)       #绘制 y 轴标签
# 设置坐标轴 x 范围 0~3*pi, y 范围-1.2~1.2
```

```
#plt.axis([0, 3*np.pi, -1.2, 1.2])

# 横纵坐标上的字体大小与类型(不是 xlabel, 是 xticks)
plt.xticks(fontproperties='Times New Roman', size=15)
plt.yticks(fontproperties='Times New Roman', size=15)
# 整个图像与展示框的相对位置
plt.subplots_adjust(left=0.19,right=0.94, bottom=0.13)
# 调整上下左右四个边框的线宽为 2
ax=plt.gca()
ax.spines['bottom'].set_linewidth(2)
ax.spines['left'].set_linewidth(2)
ax.spines['right'].set_linewidth(2)
ax.spines['top'].set_linewidth(2)
plt.show()
```

```
sns.clustermap(df,method ='average',metric='euclidean',cmap='RdBu_r')
```