For office use only
T1 ________________
T2 ________________
T3 ________________
T4 ________________

Team Control Number
2001675

Problem Chosen
D

For office use only
F1 ________________
F2 ________________
F3 ________________
F4 ________________

2020
MCM/ICM
Summary Sheet

How Does Your Team Perform?

Summary

Keywords: keyword1; keyword2; keyword3

Contents

1 Introduction

1.1 Background

Instead of having only one capable talent, today's companies or leaders tend to focus on constructing a team with people specialized in a different area in order to address more problems in face of various kinds of challenges. Great interests have been drawn on how to form a competitive team with more scientific, creative and physical capabilities to deal with various issues. Noticing that both individual contribution and team coordination are indispensable in forming a successful group.

Focusing on team sports, a highly competitive team must not only have talented individual players but also exceed others in many aspects involving with teamwork. Each team member needs to have trust, enthusiasm, ambition and motivation towards their teammates and the game [1]. A team with good teamwork, such as diverse playing types, coordination, each player's contribution, and other team level properties like adaptability, flexibility, tempo and flow may have a higher possibility to win. With good teamwork, we can make sure that the entire outcome is greater than the sum of all individuals.

Now we are asked to help analyze the team the Huskies based on their performance data last season in order to offer some strategies of how to improve in the next season.

1.2 Problem Analysis

The problem asks us to complete the following tasks:

- First only study the passing event of the the Huskiess and construct a network indicating the passes inside the team in last season, with each node representing an individual and the links as the passes between the teammates. Then try to identify some specific network patterns like certain two nodes or three nodes formation with high passing density. Remember that some other network properties and structural indicators need to be taken into account as well. We are also required to explore different scales of the passes on both micro and macro levels.
- Creating a teamwork assessment model leading to the overall performance of the team. This model should take various indicators covering the structural, configurational and dynamic aspects of teamwork including the performance indicators and some team level processes. We also need to be able to use this model to decide whether our optimized strategies from the next part is widely useful or differs by different actions the opponents take.
- Based on the former teamwork model and simulating the match, create a model that can help the coach decide what strategies should be used in the next season's match.
- Finally, we need to generalize our findings of the the Huskiess from previous models into designing effective teams and consider other aspects that are also essential for developing a generalized model of team performance.

1.3 Literature Review

Team and teamwork have attracted many researchers to deal with various issues inside since team, serves as a group of people with different characters and skills, are more likely to lead to success. Among all the sports, soccer stands out as a perfect group sport to study team and teamwork. Various work has been done on this subject.

Pappalardo [2] [2] has created an open collection containing all the events and their timestamps of seven excellent soccer competitions in a season, combining the spatial and temporal dimensions. From the analysis of the data set, we inspect that team performance are associated with connectivity between teammates. This clearly shows that teamwork act as an important influencing factor. The data set can also compare the performance of different players by their flow centrality and playback. Another journal [3] that focuses on how to assess a certain football team F.C.Barcelona constructs the passing network and calculates the changes in the properties. And by the work done by Cientia and his co-workers [4], we know there is a positive relation between team's average passes and goals, goal attempts and points. However, the rank of the teams is not always consistent with the number of passes. Therefore, here, an interesting parameter introduced in this journal is called pezzali score which equals to $\frac{goals(team)}{attempts(team)}\frac{attempts(opponent)}{goals(opponents)}$. It indicates the team is effective in both defense and attack when the value is high. Only when combined this value with the passes can we get a quite precise prediction of the overall performance. Meanwhile, some researchers use some new motif analysis method in network to show the most essential part of the passing network, leading to which part of the motif should we pay attention to if we want to weaken the opponents. [5]

1.4 Our Work

Based on the thorough analysis of the problem, we divide our work into four parts.

- First, construct the network for the passing event among all the players in each match by considering each player as a single node and the passes between pairwise players as the link between them. We present the network graph of passes of match 1 and 2, and by comparison of the second match is more competitive. Next, we introduce the network motifs and plot the pass network of each match with corresponding motifs. For structural indicators, we filter the edges larger than 5 to be the important edges. Edges with the different thresholds of the first match are also shown to present important dyadic and triadic configurations and other formations of the team. Among which, the core of the team is a triadic configuration formed by M1, D1 and D2. What's more, we explore four combinations of the micro, macro team and time, and plot the corresponding graph.
- In this part, we first identify all the performance indicators and team level processes that reflect teamwork and quantify all these factors including diverse playing types, coordination, each player's contribution, adaptability, flexibility, tempo and flow by the data given. Then we link successful teamwork with the score difference between the Huskies and the opponent in each match. By considering adaptability as constants that taking its values only depends on which side the team is playing and inputting the data of each match of the other six indicators and score difference, we construct the Neural Network Teamwork Assessment Model, with really small MSE of testing data which is only $9.43910 * 10^{-12}$, indicating the great predict capability of our neural network.
- In the third part, we use the Markov Decision process to simulate a match. Parameters include states, actions and their conditional probability. We define the states of a match with the regard of time, the position of the ball and whether the player has the ball. The reduction of parameters includes classification of 4 events which concludes all possible events in a match and allocation of players by their origin. Then we define the reward by using the teamwork performance value of the former model and all the necessary parameters will be shown in the passage. Next we use the optimal policy generated by MDP (shot more when $d > 90$ and duel more when$70 < d < 80$) with the optimal value 7.5833 to compare with the opponents' teamwork performance value. We can infer that this policy is not universally applicable.
- At last, we generalize the findings of our model of the the Huskies to all the teams. We also offer some applicable approaches to improve team effectiveness basing on our work and some other factors.

2 Assumptions and Justifications

- One person can belong to multiple groups.

- Assume the goal to be shot by the the Huskies is at x=100m.
- Each event takes a constant time 3.72s to complete, since the full event data indicates there is a 99% CI=3.72± 0.03.
- The ball will be initialised from x=40 in the process of Markov Decision. x=0 is our goal while x=100 is the goal of us.
- The influences of 'Home' and 'Away' are considered as constant in the teamwork performance model.
- Other indicators that could possibly influence teamwork except those we state are not considered in our Neural Network Teamwork Performance Model.

3 Symbols and Definitions

Notations

Symbol	Definition
DT_i',$i\in\mathbb{N}$	Processed indicator of Diversity in types of play in the ith match
PA_i',$i\in\mathbb{N}$	Processed indicator of coordination among the players in the ith match
S_{C_i}',$i\in\mathbb{N}$	Processed indicator of the contribution of each player in the ith match
A_i',$i\in\mathbb{N}$	Processed indicator of adaptability in the ith match
FL_i',$i\in\mathbb{N}$	Processed indicator of flexibility in the ith match
T_i',$i\in\mathbb{N}$	Processed indicator of Tempo in the ith match
fl_i',$i\in\mathbb{N}$	Processed indicator of flow of the team in the ith match
TPV_i',$i\in\mathbb{N}$	Processed team performance value in the ith match
ST_j,$j\in\mathbb{N}$	The jth state of a match
i	Whether the ball is with our player
d	The x coordinate of the ball
t	Time
DD	Defenders
FF	Forwarders
MM	Midfield

4 Network Construction

4.1 Network Formation

To illustrate the network representing the passes between players, we reference the first 4 matches played by the the Huskies in passingevents.xlsx to [6] and the result is as follows.

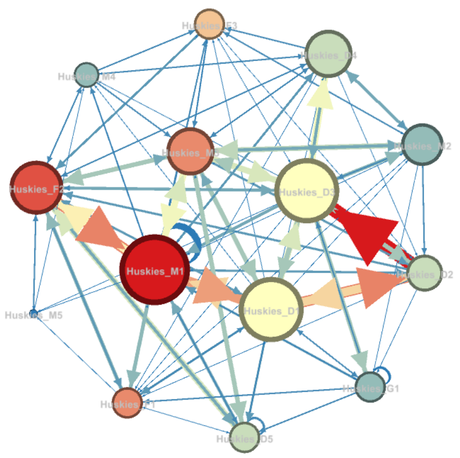


Figure 1: the the Huskies match 1 all passing events among all players

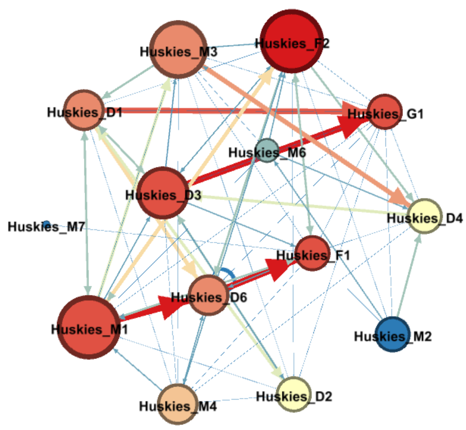


Figure 2: the the Huskies match 2 all passing events among all players

The network graph is generated by Gephi, as an Open Graph Viz Platform [7]. In Gephi define the hotter the color, the more incoming passes of the player, the larger the size of the node, the more outgoing passes from this node. In terms of edges, the redder the edges and the wider the edges represent the number of edges from these two nodes within this match. The maximum and minimum circle and edges of the network are defined as 80 and 10 respectively. Additionally, the layout method is denoted as Fruchterman Reingold, which is a force-directed layout algorithm [8].

Based on the network comparison between the first and second match, it is clear that the second match is more competitive than the first match. For the first match, it is worth noticing that the M1 is the most player who receives the most passes as 51 passes and D1 gives out most of the goals as 52 goals. Correspondingly, the reddest circle is M1 and the largest circle is D1. In terms of edges, the largest edge is from D2 to D3, as 18 and it is worth noticing that the reddest edge is from D2 to D3.

In the second match, D6, F1, F2, and G1 are among the hottest players who received from other players, which are 22, 20, 20 and 21 respectively. For outgoing edges, M1 is the biggest circle in the network with outgoing passes as 29. In terms of edges, the number of passes within the most frequent edges in the network is 8, which is represented in the M1 to D6, D6 to F1 and D3 to G1.

The network patterns and structural indicators are defined as follows. In terms of network patterns, such as dyadic and triadic configurations and team formations. According to Necmi, motifs are essential for analyzing the football pass network [5]. The network motif is defined as follows[5].

Algorithm 1: Find motifs

Result: Network Motifs
find n-node subgraphs in a real network
find all -node subgraphs in a set of randomized graphs with same in and out-degree distributions
for *each subgraph* **do**
    Z=$\frac{N_{real-N_{rand}}}{\sigma_{rand}}$
end
Set subgraphs with high Z scores as network motifs.

Apart from that, we find motifs for the network 1 (Passing events of the the Huskies for match 1) and network 2 (Passing events of the Huskies for match 2) correspondingly. Based on the codes in appendix 1, we plot the pass network for match one and two with the corresponding motifs.

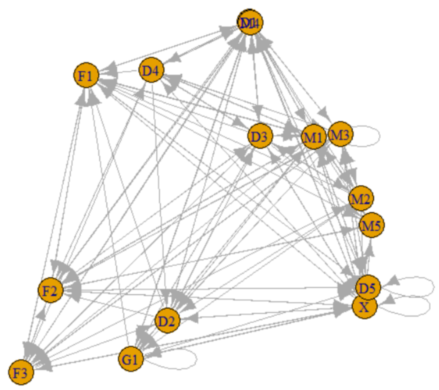


Figure 3: the the Huskies match 1 all passing events among all players

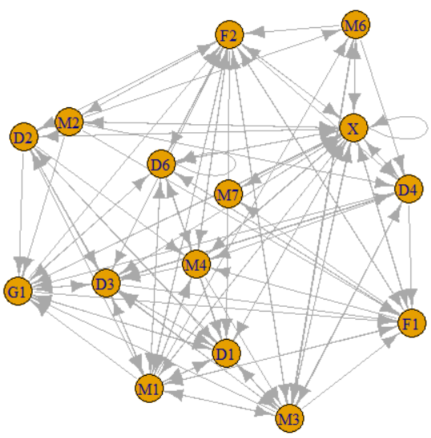


Figure 4: the the Huskies match 2 all passing events among all players

The corresponding motifs are as follows. The first list is

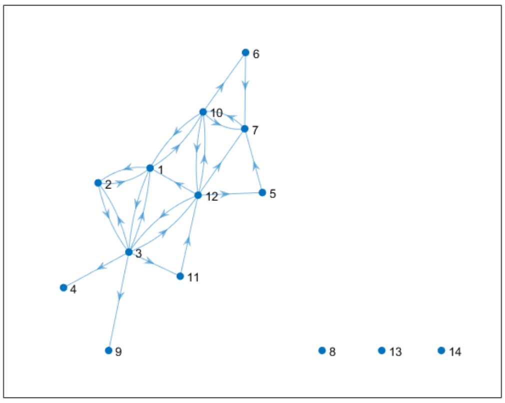
```
[NA NA  2 NA 13 19  4 11  9 43 70  0 15 13 94 86]
```

The second motif is

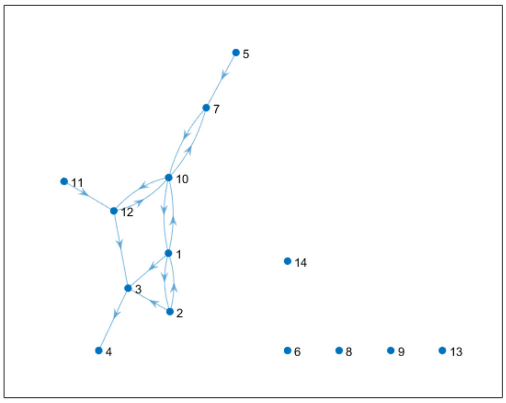
```
[NA NA 11 NA 18 29  9  9  5 30 57  5 16 10 68 46]
```

4.2 Network Properties and Structural indicators

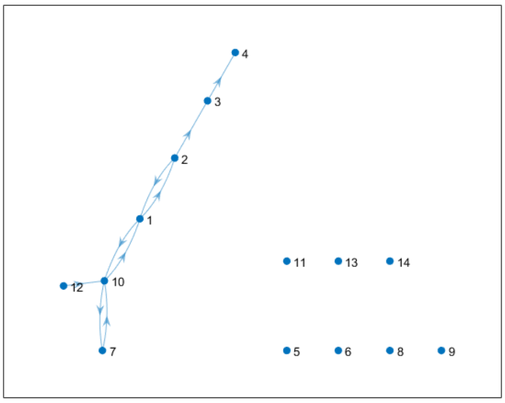
In terms of network properties, we consider strongly connected components in the direct graph as other structural indicators and network properties across the games, however, it was found out that in the game, there is only one strongly connected component which is formed by the whole graph. Therefore, we filter the edges to be larger than five as the important edges for the network.



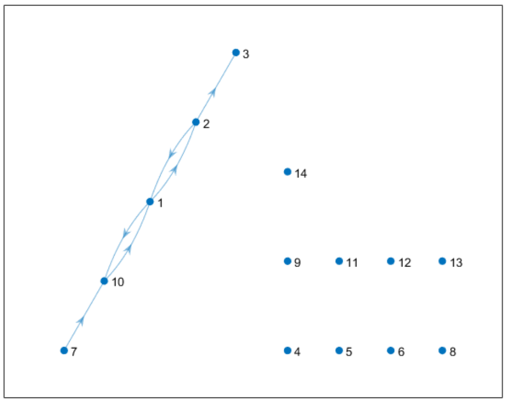
(a) Edges greater or equal than five



(b) Edges greater or equal to seven



(c) Edges greater or equal to nine



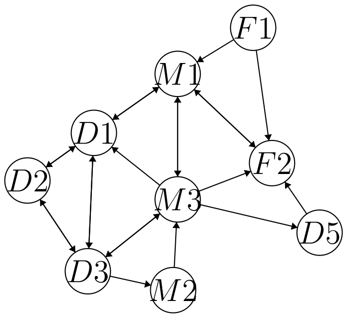
(d) Edges greater or equal to eleven

Figure 5: Filter the network with threshold as 5,7,9,11

Apart from that, in order to show the information of the essential network, we set the thresholds of the edge to represent the match information of the first match. The corresponding information is as follows.

- Edges greater or equal to 5:

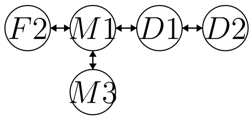
  The strongly connected components are 7,10,12,1,2, which are F2,D1, D2 M3 and M1. These players form a four configuration, which is as follows.



Since there are too many edges in this strongly connected network, we further filter the network with a threshold of 7.

- Edges greater or equal to 7:

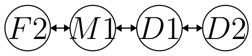
  The strongly connected components are 7,10,12,1,2, which are F2,D1, D2 M3 and M1. These players form a four configuration, which is as follows.



  It can be seen as a triadic configuration of F2, M1 and M3 and two dyadic configurations of D1 and D2 and M1 and D1.

- Edges greater or equal to 9:

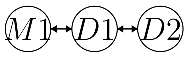
  The strongly connected components are 7,10,1,2, which are F2,D1, D2 and M1. These players form a four configuration, which is as follows.



  It can be described as three dyadic configurations, which are F2 and M1, M1 and D1, and D1 and D2.

- Edges greater or equal to 11:

  The strongly connected components are 10,1,2, which are D1, D2 and M1. These players form a triadic configuration, which is as follows.

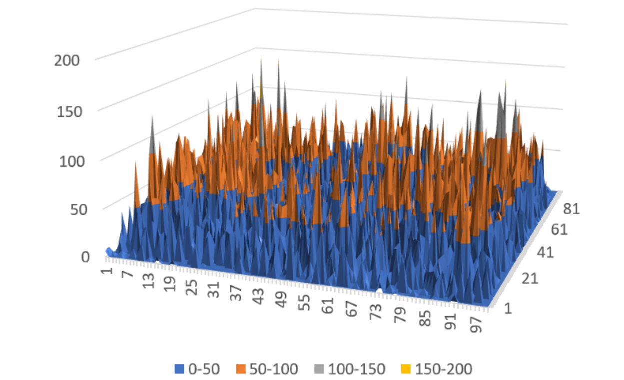


In conclusion, the core of the team is the triadic configuration, formed by M1, D1 and D2. dyadic configurations between F2 and M1, M1 and D1, and D1 with D2, also contributes strongly connected essential components in the network.

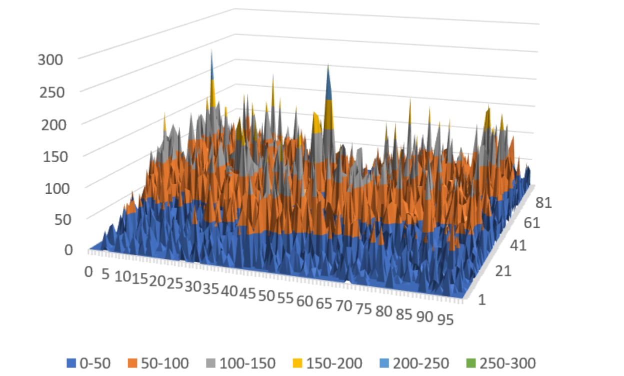
4.3 Across multiple scales

4.3.1 macro team macro time

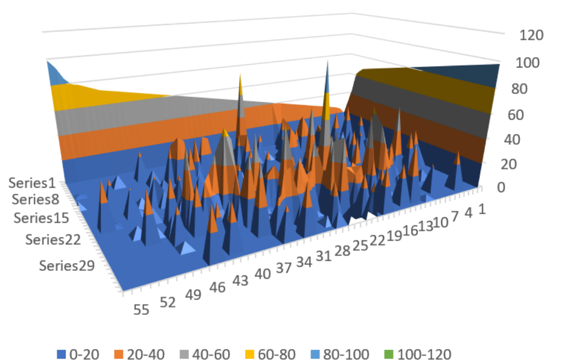
This is denoted as the team's whole duels, passes, shots, fouls distributions in the stadium across the whole season.



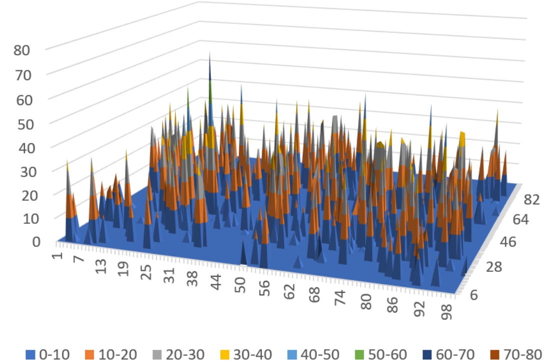
(a) Dueles' distributions of the Huskies team for the whole season



(b) Passes' distributions of the Huskies team for the whole season



(c) Shots' distributions of the Huskies team for the whole season



(d) Fouls' distributions of the Huskies team for the whole season

Figure 6: Team's whole duels, passes, shots, fouls distributions across the whole season

It is worth noticing that duel and pass occur generally randomly with slightly higher peaks near the opponents' goal in the stadium. However, the scale of passes which is 300 is higher than that of a duel, which is around 200. In terms of shots, the top position for the shot is (88,46), which is close to the goal, but not in the front of the goal. Finally, the distribution of the foul is also generally random, but slightly higher when near the opponent's goal.

4.3.2 macro team micro time

The indicator of macro team and micro time is denoted as the number of duels, goals, passes, fouls along with one game of team the Huskies (Figure 7).

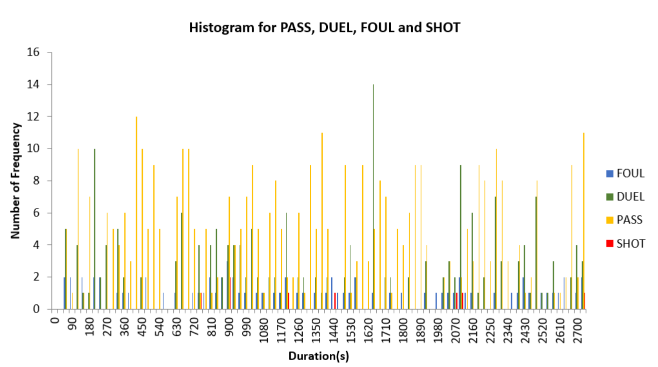


Figure 7: Histogram for PASS, DUEL, FOUL for the Huskies' first-half match

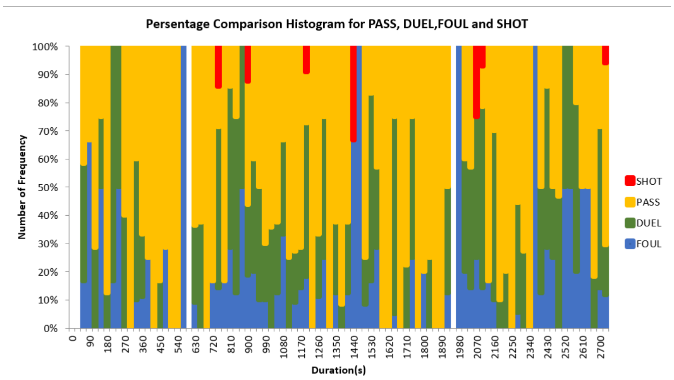


Figure 8: Percentage Comparison Histogram for PASS, DUEL, FOUL and SHOT

Based on figure 8, Based on figure 8, there is an interesting observation that before each goal, the percentage of the pass is increasing, which denotes the importance of passes for goals. In terms of defending perspectives, according to Pezzali,*It's the harsh rule of the goals: you play a great game but if you don't have a good defense, the opponent scores and then wins* [4]. Therefore, passes are not only important for attacking but also defending. Thus, the number of passes is the core indicator for success of matches.

4.3.3 micro player micro time

Since it is a micro player scale, indicating it is the pairwise behavior, we denote the number of passes, and duels representing the micro players' performance along with scale minutes of the first half match of the whole season. It is referred to as follows (Figure 9).

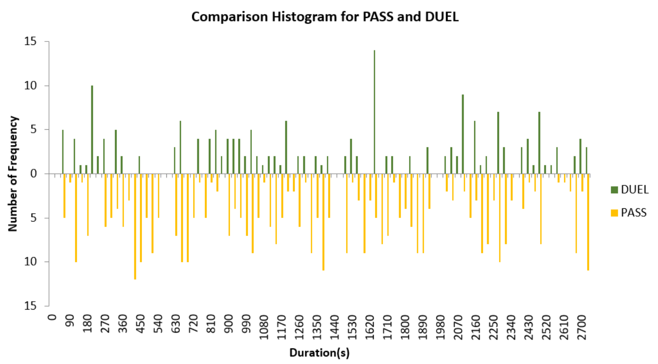


Figure 9: Comparison Histogram for PASS and DUEL

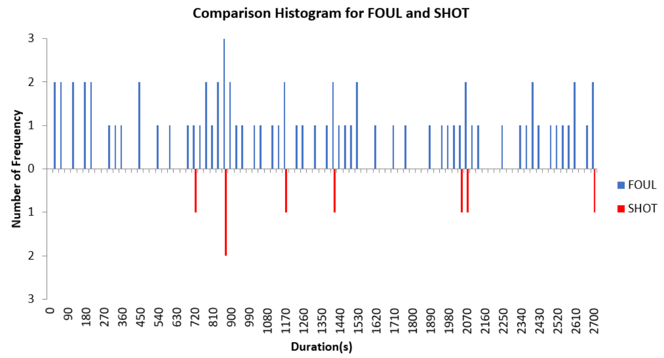


Figure 10: Comparison Histogram for FOUL and SHOT

It is worth noticing that the number of passes in one match is showing a periodic pattern, indicating the rhythm of one match. The number of duels does not show a self-repeating pattern but violates intensively depending on the situation of the game.

Additionally, figure 10 indicates the comparison between Foul and Shot, it is observed that the number of fouls is influenced by the distribution of goals within the match. It is common that if there is a change of goal, the match intensity is increasing, and a higher chance of foul correspondingly..

4.3.4 micro player macro time

Micro player's performance along the whole season is denoted as the total number of passes between each player, along with the whole 38 matches (Figure 11). The configuration and layout follow the configuration of the network formation part.

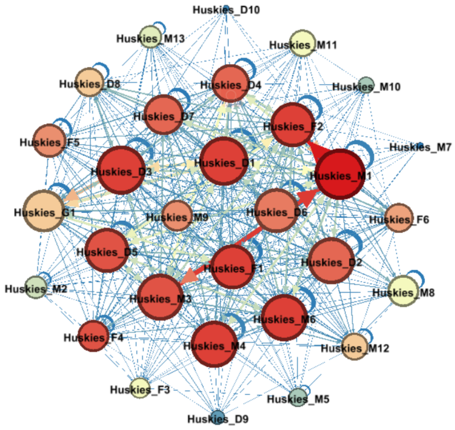


Figure 11: Total number of passes among players for the whole season

Based on this figure, the most played pass is from M1 to F2, and M1,F2,D1,M4 are among the players who pass the most. Even though M4, D1,D3 received many passes, there is not a specific destination player to pass, but pass randomly to the near player. In terms of M1, F2, D6, M3, there is a specific passing pattern hidden there, which is $M_3 \leftarrow D_6 \rightarrow M_1$

5 Teamwork Performance Indicators

It is quite clear that teamwork plays an essential role in the overall performance of the team. Not only should the players have their own capability, but they also need to coordinate with each other and stimulate teammates' as well as their own maximum potential. After having created a network for ball passing in the last task, we are required to present how the teamwork will influence the final result of each match in this task. Teamwork can also be affected by various indicators, including many different performance indicators and some processes on team level.

In this question, we first choose some essential indicators and quantify them by using the given data set, then we create a model with a neural network which is able to output the prediction of the result considering our teamwork. This will help us evaluate our optimal strategies generated by the Markov decision process in the next section. The teamwork model also leads to the clarification of whether our strategies is universally useful or will be influenced by what actions or strategies the opponent takes. Since different indicators have shown different aspects of the teamwork, our model can certainly show what kind of roles teamwork plays in the match in different ways.

5.1 Model preparation

5.1.1 Performance indicators

First, in order to construct the neural network teamwork assessment model, we need to identify the performance indicators that could possibly show the teamwork of the team, using the data given to quantify all the indicators.

- Diversity in types of plays (DT_i)

  The number of types of different formation the team adopts in the match shows how many different roles each person inside the team can act as which indicates the teamwork level. Limited amount of formations may indicate poor teamwork, possibly leading to bad respond to opponent's strategies. Here we use the changes of formation the Huskie have made in each match last season to represent this factor, denoted as DT_i.

- Coordination among the players (PA_i)

Coordination has been viewed as an important indicator of teamwork since the cooperation among players can directly reflect how well they can work as a whole, known as teamwork. We use the passing rate, that is the success passes divided by all the passes the team try to implement [9], denoted by PA_i.

- The contributions of each player (S_{C_i})

  Having a star player may be a great pleasure for the coach, but what if the rest of them are just useless? Maybe they cannot even catch a good ball of the star player, leading to the bad performance of the whole team. Therefore, we need to make sure that the performance of all the players are quite similar.

  We first define the contribution of the team member as a simple combination of all the contributions this person have made in three event that could lead to the success of the team: pass, duel and shot[10], with the equation written as :

$$C_i = \frac{P_i}{\sum_{i=1}^{n} P_i} + \frac{D_i}{\sum_{i=1}^{n} D_i} + \frac{S_i}{\sum_{i=1}^{n} S_i}$$

  Here, P_i, D_i and S_i indicates the number of passes, duels and shots the i-th teammate has made in a certain match. C_i indicates the total contribution of this person. Next by calculating the standard deviation of all teammates' contribution in the i-th game S_{C_i}, we can obtain the trend of the difference of players' performance in each game during the last season. The smaller the value is in each game, the better the teamwork.

5.1.2 Team level processes

Some team level processes and properties also serves as important factors of teamwork. We take four properties into account here.

- Adaptability (A_i)

  How well can the team adapts to different situation? On this aspect, we can think of the environment and other external factors, here, we take it as the sides the team plays at. However, there are only two different states: home or away, so we take this indicator out from the other factors that we will train the neural network with and simply treat it as a constant depending on which side the Huskie plays in this match. We calculate the corresponding probability of win(1), tie(0) and loose(-1) of the team plays at home and away, which will lead to the expectation of playing at each side. Using their normalized values as the constants that influence the overall performance of the team. We denoted the constant of adaptability for home side as AH_i=-1.32812, AA_i=1.848132, these two are generally referred to as A_i.

- Flexibility (FL_i)

  We use the number of substitutions FL_i in each game as the indicator for the team's flexibility. Although this factor remains the same number 3 in every match, we still take this factor as an important indicator of teamwork.

- Tempo (T_i)

  The tempo of the match, which can be represented as the number of all passes and duels in the whole game shows that how fast the game is going, denoted as T_i. If the value of the tempo is big, we can say that the team is well-coordinated.

- Flow of the team (fl_i)

  According to [11], flow refers to a specific state in which people are so absorbed in the event that ignores other things. A team with great flow may indicate that they have a good state of mood and no need worrying about the outside interrupting factors. Here, we take the number of all events to show the flow since it is greatly impacted by the involvement of the teammates [11], denoted as fl_i.

5.1.3 Data pre-processing and data set augument

After having collected all the data of every indicator, we do some data processing in order to normalize the data and minimizing the error. We first expand the data of the 6 indicators that we will input into the neural network by 100, and normalize all the data, the processed data are denoted as DT_i',PA_i', S_{C_i}', FL_i', T_i' and fl_i', and the value are processed by $TPV_i' = 10*\lg(\exp(TPV_i - A_i)+1)$ in order to balance the team performance value TPV_i with the indicators and minimize the error. We take the processed data of match 1 as an example. The other processed data are similar.

DT'_i	PA'_i	S'$_{C\_i}$	FL'_i	T'_i	fl'_i	A_i	TPV$_i$
1.52439	3.560127	3.731216	2.631579	3.046439	2.753081	5.745464	-1.32182

What is more, in order to generate more data for the neural network to have a better performance in machine learning, we use seasonal time series in matlab to enlarge our data set into 380 sets of data.

5.2 The Neural Network Teamwork Assessment Model

Considering all seven indicators(including the adaptability that we will only treat as a constant), we consider the inner work as the following equation:

$$A * [DT_i' \quad PA_i' \quad S_{C_i}' \quad FL_i' \quad T_i' \quad fl_i']^{\mathrm{T}} + A_i = TPV_i' \tag{1}$$

Here A denotes the coefficient matrix, however, in neural network, we cannot get the exact matrix, this is just for symbolic usage.

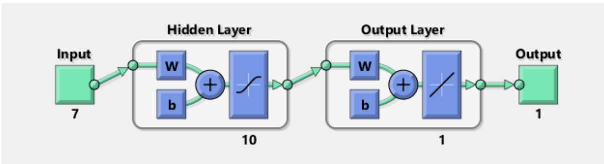


Figure 12: The structure of the neural network

We input all the processed data into our neural network which has 6 input nodes, 10 hidden layers and 1 output node. The method inside the neural network is Bayesian Regularization. The structure of our neural network is as above.

By the regression figures of our training set, test set and all set, we know that the pre-processed and augumented data is highly reliable since R is extremely close to 1. We can also say that our model may produce a great prediction of the game result with a very small error as well as avoiding over-fitting. It is also inferred from the graph that the best performance is $7.7707 * 10^{-12}$ at epoch 1000.

	samples	MSE	R
Training	266	$9.10688 * 10^{-12}$	$9.99999 * 10^{-1}$
Validation	57	0	0
Testing	57	$9.43910 * 10^{-12}$	$9.99999 * 10^{-1}$

The Neural Network Teamwork Assessment Model provide a feasible method to quantify how the level of teamwork will lead to match result based on the relation of seven teamwork indicators. Among the indicators, DT_i, FL_i are considered as on the structural level, FL_i can be also viewed as on the configurational level, while on the dynamic level we focusd on PA_i and T_i. In the next section, we use this model to predict the game result of our optimal strategies and decide whether our strategy is widely applicable or greatly influenced by opponent's strategy.

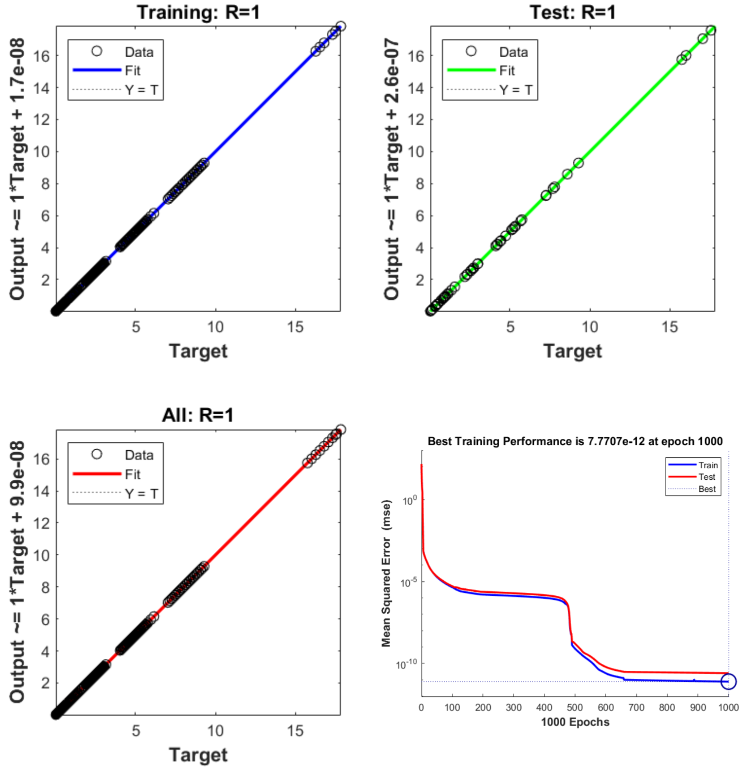


Figure 13: The graphical result of the neural network

6 Effective Structural Strategies

Referencing the quantified indicators derived from the teamwork performance model we want to build a dynamic decision process that can both identify team action under different circumstances and optimize teamwork performance. Therefore, we use Markov Decision Process to describe the actions leading to a change in each event. We also consider the probability and reward of each action regarding the teamwork performance value to obtain an optimal teamwork performance.

6.1 Markov Decision Process

6.1.1 Parameters Identification

In a standard Markov Decision Process, all possible states, actions, and conditional probabilities. The states in this process are considered as a reduction of a different situation in a real match. Then, we use the idea of the teamwork performance model and combine some indicators related to different actions. After the definition of states and classification of actions, we compute the conditional probability and each reward.

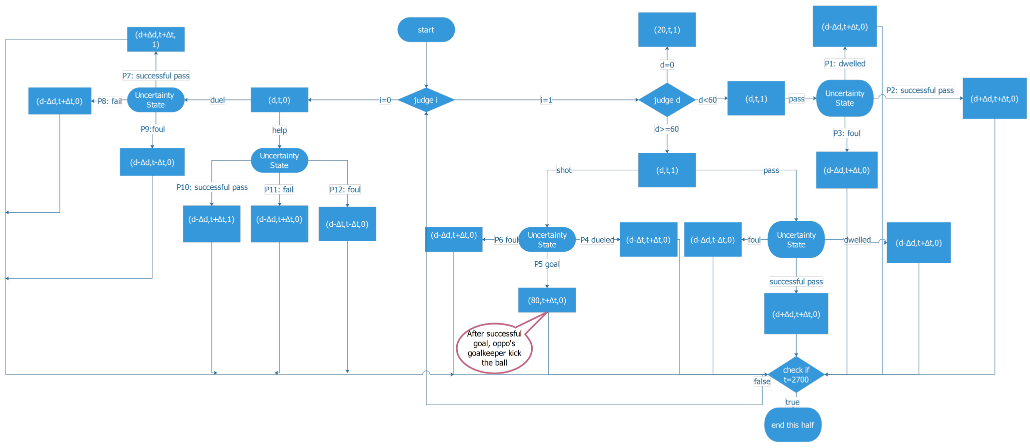


Figure 14: Markov Decision Process for the Huskies' team member

Following this instruction, we compute our parameters.

- States (ST_j)
  we use ST_j , $j \in \mathbb{N}$ to represent the different states in a match. To simulate a real match, we first consider whether a player is in control of the ball. In this case, we define i as the indicator clarifying whether the ball is with our team play.
  $$i = 0, or, i = 1$$
  Focusing on the action of players and the state of the ball, We also consider the position d which represents the x coordinates of the ball. The changes of d is defined by Δd, and each step of actions could probably result in a Δd to d. Indeed d could be different depending on different plays and situations. However, to obtain the optimization we define Δd as the best optimal passing distance 20 [4]. Since the total distance of the field according to our data set is 100, and we set the beginning point in the middle of the field where x=50, therefore, all possible situations for d are
  $$D = \{d = 0, d = 20, d = 40, d = 60, d = 80, d = 100\}$$
  To identify different time periods after one action in a match, we define Δt as the time lag sacrificed for an action. The definition of each time lag is referenced by the period between each event in the data set. We use statistical methods to evaluate all the time lags and their value in 38 matches to estimate the most possible value of a time lag. The result is
  $$\Delta t = 3.72$$
  (s) with a 90% confidence interval [3.69, 3.75]. Therefore, we conclude the state ST_j as a set of (i,t,d).
  $$ST_j = (i, t, d), i = 0, 1, t \in [0, 5400], d \in D, \Delta t = 3.72, \Delta d = 20, d_0 = 50 \tag{2}$$

- Classification of Actions
  To reduce the situation, we divide all the events into 4 types: *Pass, Duel, Foul and shot*. Using this division, we classify all the event types and events sub-types in the data set.

Classification	Type of events
Pass	Pass
	Touch
Duel	Duel
	Clearance
	Save Attempt
Foul	Foul
	Free kick
	Goalkeeper leaving line
	Ball out of the field
	Offside
Shot	Shot

- Conditional Probability and Rewards(P_n)

  We also divide the players by their origins to access a more clearly and reliably optimal strategies for the coach. So we label them as 4 groups of players $FF : forward$, $DD : defense$, $MM : midfield$ and $GG : goalkeeper$. Each action of different groups of players lead to some result and the match will switch to next state. We use the first letter of the name of the actions to represent them as elements in action set ACT, and we use P_n to represent the different possible results derived by the action and compute their conditional probability.

$$ACT = P, D, F, S$$

$$P_n = \{P_n | n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

  The follow table is a result example for conditional probability for Defenders

Meanings	P_n	Value
$P((0, d - \Delta d, t + t) \| P, (1, d, t))$	P_1	0.245645
$P((0, d + \Delta d, t + \Delta t) \| P, (1, d, t))$	P_2	0.741202
$P((0, d - \Delta d, t) \| P, (1, d, t))$	P_3	0.013153
$P((0, d - \Delta d, t + t) \| S, (1, d, t))$	P_4	0.967511
$P((0, 80, t + \Delta t) \| S, (1, d, t))$	P_5	0.019336
$P((0, d - \Delta d, t) \| S, (1, d, t))$	P_6	0.013153
$P((1, d + \Delta d, t + t) \| S, (0, d, t))$	P_7	0.259356
$P((0, d + \Delta d, t + \Delta t) \| S, (0, d, t))$	P_8	0.0.737188
$P((0, d - \Delta d, t) \| S, (0, d, t))$	P_9	0.013153
$P((1, d - \Delta d, t + t) \| S, (0, d, t))$	P_{10}	0.741202
$P((0, d + \Delta d, t + \Delta t) \| S, (0, d, t))$	P_{11}	0.255394
$P((0, d - \Delta d, t) \| S, (0, d, t))$	P_{12}	0.013153

  Table 1: Conditional Probability for Defenders

  By the same methods, we derived the conditional probability of other groups of players, which will be shown in the next.

- Reward

  To obtain an optimal teamwork performance, we relate the rewards of each action in the process of Markov Decision to our teamwork performance value in the former value. Considering our classification of action, we analyze the statistical relationship between the number of each action in each match and the teamwork performance value. A multiple regression analysis with Pearson correlation validation is applied to this process. Here, we use the same enlarges database of 380 matches to do this analysis. The full result combined with conditional probabilities and rewards with regard to different groups of players is shown below in (table2) The energy cost here is a constant b_0 derived by the constant value in regression analysis. We a modify to the constant to enable it as the energy reduction for each action by

$$EnergyCost = \frac{b_0}{N(events)}$$

P	P_1	P_2	P_3	P_4	P_5	P_6
DD	0.2456	0.7412	0.0132	0.9675	0.0193	0.0132
FF	0.3064	0.6572	0.0364	0.8970	0.0666	0.0364
MM	0.2445	0.7335	0.0220	0.9264	0.0516	0.0220
Rewards	0.0148	0.0069	-0.0102	0.0148	2.7170	-0.0102
EnergyCost	-0.0039	-0.0039	-0.0039	-0.0039	-0.0039	-0.0039
P	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}
DD	0.2594	0.7372	0.0132	0.7412	0.2554	0.0132
FF	0.2533	0.7372	0.0364	0.6572	0.3303	0.0364
MM	0.2570	0.7372	0.0220	0.7335	0.2607	0.0220
Rewards	0.0148	0.0000	-0.0102	0.0069	0.0000	-0.0102
EnergyCost	-0.0039	-0.0039	-0.0039	-0.0039	-0.0039	-0.0039

Table 2: Conditional Probabilities and Rewards of Groups for Markov Decision

6.2 Reinforcement Learning

Facing the randomness of the environment, scientists apply reinforcement learning to update the computer's understanding of the environment by iteration, with is reinforcement. UNITY and Google set up a multi agent system and implemented a reinforcement learning algorithm illustrated as follows[12],[13].

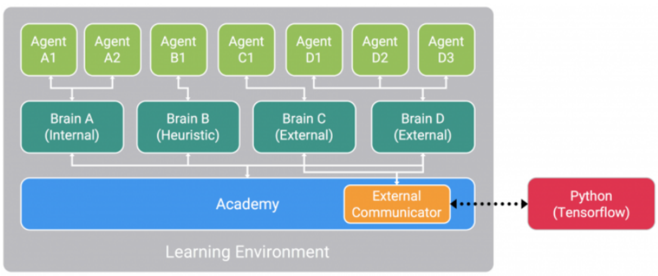


Figure 15: Unity Soccer Reinforcement Learning Architecture

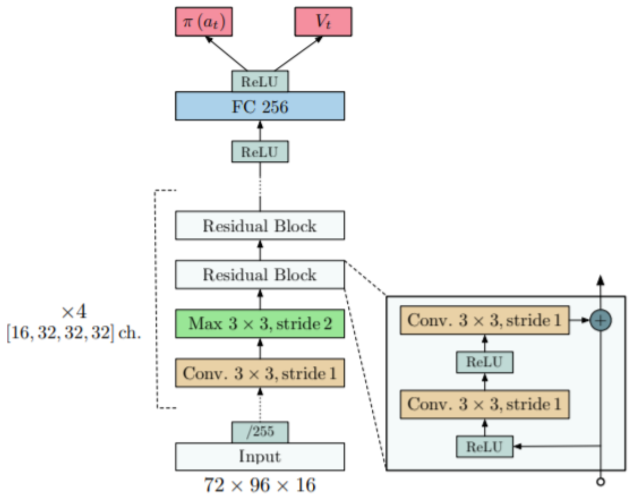


Figure 16: Google's Soccer Reinforcement Learning Architecture

6.3 How to Change Policy?

According to Robin, we can define policy evaluation function to approximate the states of the information to simulate the true environment[14]. The algorithm for defining the policy evaluation function is illustrated as follows. Furthermore, the illustrative graph is presented to further demonstrate the meaning of policy evaluation.

Comparing the opponents' predicting teamwork performance value(TPV) derived by the neural network model, we find some competitive teams such as OPPONENT12 and OPPONENT 17. The TPV of the different opponents is in the table below(table3). We omitted the teams whose TPV is less than 0, since we believe those teams are not so competitive. Teams in the yellow region should be considered further since our strategies are universally effective only except these teams.

ID	12	17	16	15	14	11	2	8	1	6
TPV	27.8847	24.9251	20.0592	15.9658	6.6449	5.2479	3.4025	1.1031	0.6823	0.6219

Table 3: Some Competitive Teams

For Opponent 12, 16 and 17, all of their team level process reflect the characteristics of fast tempo. Opponent 12 focuses more on the drive and performance of star players, which is very special. Opponent 16 as the team with most successful pass is more talented for passing accuracy, while Opponent 17 is more flexible for the switch of formation. Also we need to be careful of Opponent 15 (in the pink region), since its TPV is very closed to our optimal result.

Therefore, base on the analysis above, we propose the following strategies:

- Universally Effective Strategies

  Based on the value iteration and Q learning for this markov decision process, we find the optimal policy as follows to shot at after 90.

- For Special Opponents

  For Opponent 12, since its star player effect is very significant, we suggest to try to interrupt the collaboration between their star players. For example, cut the largest flow of their team network. For Opponent 16, we advise that the the Huskies could either improve the passing accuracy of themselves or try to seduce them to make longer passing to reduce their accuracy. For Opponent 17, we believe the the Huskies had better explore this team in advance and try to accumulate experience from each competition.

- How Does Network Performance Change? From our strategies, we find that the we need to perform more duels from x=70 to x=100, which leads to a radical type of formation. Also, the strikers should appear more at the position which is very near to the goal of us. For defenders and midfield, they need to derive more successful pass around x=60.

7 Generalized Successful Teamwork

7.1 Generalizing the findings to form a Successful Teamwork

From our above work, we know that teamwork acts as a critical part of forming a highly effective team. Apart from a soccer match, our findings are also suitable for other team level sports or even any work done by a group of people.

- Always prepare a backup method and always try to think more

  The diversity in types of plays can be viewed as the mode that the team adopts to solve the problem, it can also be seen as flexibility. In other words, whether they have different methods when one of them does not go well is essential for winning. A successful team should have several backup methods and think in different ways, making sure they can have a different choice when this way is not working.

- Make rules to maintain good cooperation

  Coordination also matters anywhere since good cooperation can help maximize overall efficiency. It can also help stimulate the potential of each individual. The leader can make some rules to punish those who don't want to work with others or ruin the harmonious atmosphere.

- Making sure everyone is making contribution

  Although it doesn't seem very important, making sure all members are making contributions can help improve the relationship and mood of the team, also hence the flow of the team since the contribution of each member can influence the self-assessment, which will directly link to the mood of the team. Assigning each teammate with work depending on their capability and skills, but also let them contribute to the team in their own way..

- Train the team in the different situation
  Whether a team can adjust itself to the various situations and environments shows its capability to act effectively without bothered by other external factors. Leaders can train the team in a different environment so that they may tend to adapt quickly when facing a new environment.
- Keep a stably increasing speed
  The tempo of the team does not have to be extremely quick, but keeping a stably increasing speed is quite important. The team can first think thoroughly and consider various issues, then try to speed up without making devastating mistakes.
- The flow of the team is considered as the total mood or spirit, which can be affected by sponsors, money. Reducing the worries team member could have can help them perform stably.

7.2 Other Factors

Besides all the factors found in our work, there are some other indicators that could possibly influence the effectiveness of the team.

- Whether the team has clear and strong aims
  A strong and feasible aim will act as a motive that could inspire the team and raise the spirit.
- The external support. The team should have sufficient external support such as money, sponsor or insurance to exclude the influence external factors could have on members' mental state.
- Personal factors. If some team members have personal issues, it may influence his own performance in the team, leading to some influence on the entire team.
- enthusiasm for the work they are implementing.

8 Sensitivity Analysis

The core step of this research is to enlarge the data set. It is used not only for the neural net work in the construction of the Teamwork Performance model but also significantly influence the Markov Decision Process. We use Seasonal Time series to enlarge the database, and we want it to be random and convincing. Therefore, we need to judge if the enlarged data set is sensitive to time. By error analysis, we evaluate the standard error between the initial values and the predicted values(table4).

Name	PlayTypes	SuccessPass	Contribution	Substitution	Tempo	Flow
RealValue	2.25	2.65	3.54	2.63	2.53	2.82
Enlarged	2.38	2.60	2.73	2.63	2.71	2.44
StandardError	6.14%	-1.91%	-22.78%	0.00%	7.06%	-13.57%

Table 4: Standard Error Analysis

We find the distribution of contribution is quite sensitive to the time, but it is still under our control. Therefore, we can conclude that our data set can be universally utilized in our models.

9 Strengths and Weaknesses

9.1 strengths

- The teamwork performance model is comprehensive, multi-dimensional
- The data set has been pre-processed and augmented properly, leading to a pretty good performance of the neural network.
- The construct of the Markov process is vivid and the parameters are statistically convincing.

9.2 weaknesses

- In this research, we use teamwork performance value to indicate and estimate a real situation performance. However, the score of a football match is pretty random actually.
- In the process of Markov Decision Δd=20 is too ideal for a real match, as well as Δt
- The evaluation taken for adaptability is limited and non-dynamic to micro match events.
- The strategies do not include a good substitution or the progress of a player.

10 Conclusions

10.1 Summary

In this article, we first focus on the passing event of the Huskie, exploring various scales and identifying particular team formations including the core of the team: a triadic configuration formed by M1, D1 and D2. Then we implement the Neural Network Teamwork Performance Model by the data of seven influencing indicators. Aiming at providing the team coach with an optimal strategy, we construct Markov Decision Process to generate the best teamwork performance value: 7.5833, and the optimal policy is try to shot more when $d > 90$ and duel more when $70 < d < 80$. By inputting 19 opponents' data into the Teamwork model we have constructed before and compare with our optimal TPV, we get that the policy's applicability depends on opponents' strategy. In order to make our policy more useful, we suggest the team to take flexible action, and can do several passes before shoting. In the end, we generalized our work to all the team and offer some practical suggestions based on our finding and some other factors.

10.2 Future Work

In the future, we may use the network flow algorithm to find the max flow and min cut of the network, which is essential for determining the most important position of the stadium. Furthermore, we may use the Cellular automaton algorithm to simulate the real dynamic perspective of the game.

References

[1] "The importance of teamwork in sports | the stanford daily," https://www.stanforddaily.com/2011/11/09/the-importance-of-teamwork-in-sports/, (Accessed on 02/17/2020).

[2] "A public data set of spatio-temporal match events in soccer competitions | scientific data," https://www.nature.com/articles/s41597-019-0247-7, (Accessed on 02/17/2020).

[3] "Defining a historic football team: Using network science to analyze guardiola's f.c. barcelona | scientific reports," https://www.nature.com/articles/s41598-019-49969-2, (Accessed on 02/17/2020).

[4] P. Cintia, F. Giannotti, L. Pappalardo, D. Pedreschi, and M. Malvaldi, "The harsh rule of the goals: Data-driven performance indicators for football teams," *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, 2015.

[5] N. GÜRSAKAL, F. M. YILMAZ, H. O. ÇOBANOĞLU, and S. ÇAĞLIYOR, "Network motifs in football," *Turkish Journal of Sport and Exercise*, vol. 20, no. 3, pp. 263–272, 2018.

[6] "2020 mcm/icm problems," https://www.comap.com/undergraduate/contests/mcm/contests/2020/problems/, (Accessed on 02/16/2020).

[7] "Gephi - the open graph viz platform," https://gephi.org/, (Accessed on 02/16/2020).

[8] "Fruchterman reingold · gephi/gephi wiki," https://github.com/gephi/gephi/wiki/Fruchterman-Reingold, (Accessed on 02/17/2020).

[9] "Quantifying the performance of individual players in a team activity," https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0010937, (Accessed on 02/17/2020).

[10] "Contribution ratings: How to assess the impact of individual player actions? - scisports," https://www.scisports.com/contribution-ratings-how-to-assess-the-impact-of-individual-player-actions/, (Accessed on 02/17/2020).

[11] "Flow and performance: A study among talented dutch soccer players - sciencedirect," https://www.sciencedirect.com/science/article/pii/S1469029211000215, (Accessed on 02/16/2020).

[12] "Unity ml agents & reinforcement learning – christine payne," http://christinemcleavey.com/world-cup-reinforcement-learning/, (Accessed on 02/17/2020).

[13] K. Kurach, A. Raichuk, P. Stańczyk, M. Zając, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet *et al.*, "Google research football: A novel reinforcement learning environment," *arXiv preprint arXiv:1907.11180*, 2019.

[14] "Cs221: Artificial intelligence: Principles and techniques," https://stanford-cs221.github.io/autumn2019/, (Accessed on 02/18/2020).

Appendices

Appendix A Passing Events of the Huskies in matches 1 and 2

Here is the passing events for match 1 and 2. The first column represents the source of the pass and the first row represents the destination of the pass.

Table 5: Adjancy matrix representing passes among the Huskies players in match 1

value	D1	D2	D3	D4	D5	F1	F2	F3	G1	M1	M2	M3	M4	M5
D1	0	14	8	1	3	3	3	1	2	11	1	4	1	0
D2	11	0	18	0	0	1	3	0	1	1	0	1	0	0
D3	6	6	0	9	0	1	2	1	6	4	5	6	1	1
D4	0	1	4	0	0	1	2	2	0	1	3	2	1	0
D5	1	0	0	0	2	1	8	0	0	4	0	1	0	0
F1	2	0	0	0	1	0	5	1	0	1	0	0	0	1
F2	3	0	4	2	4	3	0	1	0	14	0	4	2	1
F3	0	1	1	0	0	1	3	0	0	0	1	2	0	0
G1	2	0	2	0	2	2	0	0	1	3	0	0	0	0
M1	13	3	0	3	2	5	10	2	1	1	1	8	2	1
M2	0	2	2	4	0	1	0	2	1	1	0	7	0	0
M3	6	3	8	2	6	0	6	1	0	9	4	0	0	0
M4	0	0	0	2	0	0	2	2	0	1	0	2	0	0
M5	0	0	0	0	0	0	2	0	0	0	0	0	1	0

Table 6: Adjancy matrix representing passes among the Huskies players in match 2

D1	D2	D3	D4	D6	F1	F2	G1	M1	M2	M3	M4	M6	M7	
D1	0	4	3	0	5	0	0	7	3	0	1	0	0	0
D2	4	0	3	0	0	0	0	1	1	0	0	1	0	0
D3	3	2	0	1	0	2	2	8	1	0	2	0	0	0
D4	0	0	4	0	0	1	0	1	0	0	2	1	0	0
D6	0	0	0	0	1	8	2	1	2	0	0	2	0	0
F1	0	0	0	0	2	0	3	0	2	0	0	1	0	1
F2	1	1	2	3	3	3	0	0	3	1	1	1	0	0
G1	1	0	1	0	1	1	1	0	0	0	0	0	0	0
M1	3	1	2	0	8	3	5	1	0	0	4	1	1	0
M2	0	0	0	3	0	1	1	1	0	0	0	0	2	0
M3	3	1	1	6	0	1	2	1	1	0	0	0	1	0
M4	1	0	1	0	2	0	2	0	2	0	1	0	0	0
M6	0	0	0	2	0	0	2	0	0	0	1	0	0	0
M7	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Appendix B References programes

Here are simulation programmes we used in our model as follow.

Input R source:

```
A=read.csv("R1.csv")
B=as.matrix(A)
library('igraph')
q=graph_from_adjacency_matrix(B,weighted=TRUE)
plot(q)
motif=graph.motifs(q,size=3)
```

Input matlab source for Regression NN:

```
function [trainedModel, validationRMSE] = trainRegressionModel(trainingData)

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'ceil', 'successfulpassesceil', 'contribution', 'substitution', 'tempopasso
predictors = inputTable(:, predictorNames);
response = inputTable.Value;
isCategoricalPredictor = [false, false, false, false, false, false];

% Train a regression model
% This code specifies all the model options and trains the model.
regressionTree = fitrtree(...
    predictors, ...
    response, ...
    'MinLeafSize', 4, ...
    'Surrogate', 'off');

% Create the result struct with predict function
predictorExtractionFcn = @(t) t(:, predictorNames);
treePredictFcn = @(x) predict(regressionTree, x);
trainedModel.predictFcn = @(x) treePredictFcn(predictorExtractionFcn(x));

% Add additional fields to the result struct
trainedModel.RequiredVariables = {'ceil', 'contribution', 'eventflow', 'substitution', 'succe
trainedModel.RegressionTree = regressionTree;
trainedModel.About = 'This struct is a trained model exported from Regression Learner R2019b.
trainedModel.HowToPredict = sprintf('To make predictions on a new table, T, use: \n  yfit = c
c.RequiredVariables \nVariable formats (e.g. matrix/vector, datatype) must match the original

% Extract predictors and response
% This code processes the data into the right shape for training the
% model.
inputTable = trainingData;
predictorNames = {'ceil', 'successfulpassesceil', 'contribution', 'substitution', 'tempopasso
predictors = inputTable(:, predictorNames);
response = inputTable.Value;
isCategoricalPredictor = [false, false, false, false, false, false];

% Perform cross-validation
partitionedModel = crossval(trainedModel.RegressionTree, 'KFold', 5);

% Compute validation predictions
validationPredictions = kfoldPredict(partitionedModel);

% Compute validation RMSE
validationRMSE = sqrt(kfoldLoss(partitionedModel, 'LossFun', 'mse'));
```

Input matlab source for Neural Network Function:

```
function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)

% Input 1
x1_step1.keep = [1 2 3 5 6];
x1_step2.xoffset = [0;0.675363;0.242144;1.319558;2.106894];
x1_step2.gain = [0.327999998688;0.620646744940565;0.237616384505131;0.920644580096309;1.99808
x1_step2.ymin = -1;

% Layer 1
b1 = [0.44543603601616510446;-0.35427386853115577781;3.0046013024277815617;0.52931453048562
IW1_1 = [-2.8596417008235377466 -1.7633302850391752159 0.58870209151239749623 1.2995875482925

% Layer 2
b2 = -0.84281526054514566137;
LW2_1 = [3.2685228087150157705 3.0022005780863545432 -1.3578854929286001685 -2.916053857964

% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 0.112284799164601;
y1_step1.xoffset = 0.02297;

% ===== SIMULATION ========

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX
    X = {X};
end

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},1); % samples/series
else
    Q = 0;
end
```

```
% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    X{1,ts} = X{1,ts}';
    temp = removeconstantrows_apply(X{1,ts},x1_step1);
    Xp1 = mapminmax_apply(temp,x1_step2);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);
    Y{1,ts} = Y{1,ts}';
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end
end

% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Remove Constants Input Processing Function
function y = removeconstantrows_apply(x,settings)
y = x(settings.keep,:);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end
```

Input python source:

```
#This code is contributed by Neelam Yadav
d=40
import random
def getUserInput():
    action=input("Please input your action (shot, pass, duel, help)")
    return action
action=getUserInput()
t=0
i=False
state = (d, t, i)
dt=3.74
dd=5
def PlayRule(state):
    print(random.random())
    rand=random.random()
    if(i==True):
        print("Huskie hodls the ball")
        if(d==0):
            print("Huskie! d==0")
            state=(20,t+dt,True)
        elif(d<60):
            print("Huskie! d<60")

            if(rand<0.245645):
                state=(d-dd,t+dt,0)
```

```
                print("dueled")

            elif(rand<0.986847 and rand >0.245645):
                state=(d+dd,t+dt,0)
                print("successful pass")
            else:
                state=(d-dd,t+dt,0)
                print("foul")
        elif(d>60 or d==60):
            print("Huskie! d>=60")
            if(action=="shot"):
                print("Action is shot")
                if(rand<0.967511):
                    state=(d-dd,t+dt,0)
                    print("A")
                elif(rand>0.967511 and rand <0.986847):
                    state=(80,t+dt,0)
                    print("B")
                else:
                    state=(d-dd,t+dt,0)
                    print("C")
            elif(action=="pass"):
                print("Action is pass")
                if(rand<0.245645):
                    state=(d-dd,t+dt,0)
                    print("PassA")
                elif(rand<0.986847 and rand >0.245645):
                    state=(d+dd,t+dt,0)
                    print("PassB")
                else:
                    state=(d-dd,t+dt,0)
                    print("PassC")


    elif(i==False):
        print("others have ball")
        if(action=="duel"):
            print("It is duel")
            if (rand<0.259356):
                print("Successful Pass in duel")
                state=(d+dd,t+dt,1)
            elif(rand>0.259356 and rand<0.996544):
                print("Failed Duel")
                state=(d-dd,t+dt,0)
            else:
                print("Foul while duel")
                state=(d-dd,t+dt,0)

        elif(action=="help"):
            print("It is help")
            if(rand<0.741202):
                print("Successful Pass")
                state=(d-dt,t+dt,1)
            elif(rand>0.741202 and rand<0.996596):
                print("Failed to help")
                state=(d-dd,t+dt,0)
            else:
                print("Fouled here")
                state=(d-dd,t+dt,0)
```

Input python source for MDP:

```
import os

### Model (MDP problem)

class TransportationMDP(object):
    def __init__(self, d, t, i):
       # N = number of blocks
        self.d = d
        self.t = t
        self.i = i
    def startState(self):
        return (40,0,1)
    def isEnd(self, state):
        return self.t==5400
    def actions(self, state):
        # return list of valid actions
        result = []
        if state<60:
            result.append(’pass’)
        if state>90:
            result.append(’shot’)
        if state>70 and state<90:
```

```
            result.append('duel')
        if state>60 and state<70:
            result.append('help')
        return result
    def succProbReward(self, state, action):
        # return list of (newState, prob, reward) triples
        # state = s, action = a, newState = s'
        # prob = T(s, a, s'), reward = Reward(s, a, s')
        result = []
        if action=='pass':
            failProb = 1-0.74
            stateS=(self.d+5,self.t+3.47,1)
            stateF=(self.d-5,self.t+3.47,0)
            result.append((stateS, 1.-failProb, 0.00689830139499833))
            result.append((stateF, failProb, -0.0101926084490302))
        elif action=='shot':
            failProb = 1-0.019
            stateS=(80,self.t+3.47,0)
            stateF=(self.d-5,self.t+3.47,1)
            result.append((stateS, 1.-failProb, 2.71700810738816))
            result.append((stateF, failProb, 0.008))
        elif action=='duel':
            failProb = 1-0.259356
            stateS=(self.d+5,self.t+3.47,1)
            stateF=(self.d-5,self.t+3.47,0)
            result.append((stateS, 1.-failProb, 0.0148347712638812))
            result.append((stateF, failProb, 0))
        elif action=='help':
            failProb = 1-0.741202
            stateS=(self.d+5,self.t+3.47,1)
            stateF=(self.d-5,self.t+3.47,0)
            result.append((stateS, 1.-failProb, 0.00689830139499833))
            result.append((stateF, failProb, -0.05))
        return result
    def discount(self):
        return 1.
    def states(self):
        return range(1,self.t+1)

# Inference (Algorithms)

def valueIteration(mdp):
    # initialize
    print("VI")
    GGG=1
    V = {} # state -> Vopt[state]
    for state in mdp.states():
        V[state] = 0.

    def Q(state, action):
        print("Q")
        return sum(prob*(reward + mdp.discount()*V[newState]) \
                for newState, prob, reward in mdp.succProbReward(state, action))

    while GGG<1000:
        print("GGG")
        # compute the new values (newV) given the old values (V)
        newV = {}
        for state in mdp.states():
            print("test For")
            if mdp.isEnd(state):
                newV[state] = 0.
            else:
                newV[state] = max(Q(state, action) for action in mdp.actions(state))
        # check for convergence
#        if max(abs(V[state]-newV[state]) for state in mdp.states())<1e-10:
#            break
        V = newV
        print("new check")
        # read out policy
        pi = {}
        for state in mdp.states():
            print("read out policy")
            if mdp.isEnd(state):
                pi[state] = 'none'
            else:
                pi[state] = max((Q(state, action), action) for action in mdp.actions(state))

        # print stuff out
        os.system('clear')
        print('{:20} {:20} {:20}'.format('s', 'V(s)', 'pi(s)'))
        for state in mdp.states():
            print(mdp.states()[i])
            print(V[state])
            print(pi[state])
```

```
        input()
        GGG=GGG+1

mdp = TransportationMDP(40,0,1)

#for i in range(1000):
#    print(mdp.actions(i))
#print(mdp.succProbReward(3, 'pass'))
#print(mdp.succProbReward(3, 'tram'))
valueIteration(mdp)
```

Input Time series source:

```
%Seasonal time series
s=4;
n=342;
m1=length(x);
for i = s+1 : m1
    y(i-s)=x(i)-x(i-s);
end
w=diff(y);
m2=length(w);
k=0;
for i = 0:3
    for j = 0:3
        if i==0&j==0
            continue
        elseif i == 0
            ToEstMd=arima('MALags',1:j,'Constant',0);
        elseif j==0
            ToEstMd=arima('ARLags',1:i,'Constant',0);
        else
            ToEstMd=arima('ARLags',1:i,'MALags',1:j,'Constant',0);
        end
        k=k+1; R(k)=i; M(k)=j;
        [EstMd,EstParamCov,logL,info]=estimate(ToEstMd,w');
        numParams=sum(any(EstParamCov));
        [aic(k),bic(k)]=aicbic(logL, numParams, m2);
    end
end
fprintf('R,M,AIC,BIC\n%f');
check = [R',M',aic',bic']
r=input('input R=');  m=input('input M=');%use 3 in this paper
ToEstMd = arima('ARLags',1:r,'MALags',1:m,'Constant',0);
[EstMd,EstParamCov,LogL,info]=estimate(ToEstMd,w');
w_Forecast=forecast(EstMd,n,'Y0',w')
yhat=y(end) + cumsum(w_Forecast)
for j = 1:n
    x(m1+j) = yhat(j) + x(m1 + j - s);
end
xhat=x(m1 + 1: end)
```