Linköping University | Department of Computer and Information Science
Master thesis, 30 ECTS | Master thesis in Statistics and Data Mining
202019 | LIU-IDA/LITH-EX-A--2019/001--SE

Markov Decision Processes and ARIMA models to analyze and predict Ice Hockey player's performance

Carles Sans Fuentes

Supervisor : Patrick Lambrix
Examiner : Oleg Sysoev

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Abstract

In this thesis, player's performance on ice hockey is modelled to create new metrics by match and season for players. AD-trees have been used to summarize ice hockey matches using state variables, which combine context and action variables to estimate the impact of each action under that specific state using Markov Decision Processes. With that, an impact measure has been described and four player metrics have been derived by match for regular seasons 2007-2008 and 2008-2009. General analysis have been performed for these metrics and ARIMA models have been used to analyze and predict players performance. The best prediction achieved in the modelling is the mean of the previous matches. The combination of several metrics including the ones created in this thesis could be combined to evaluate player's performance using salary ranges to indicate whether a player is worth hiring/maintaining/firing.

Acknowledgments

I would like to express my gratitude to my Linköping University thesis supervisor Patrick Lambrix for the help provided in the development of the thesis, giving insightful comments and suggestions that helped me out developing it, as well as giving me short but intense master classes on hockey. Additionally, I would like to thank Niklas Carlsson for his insights and evaluation of my mid-results to check whether they were making sense or not. I would also like to thank my related thesis field colleague Dennis Ljung for the patience, help in understanding the data and productive discussions while deepening the common concepts we shared on our thesis.

Additionally, I would like to express the deepest appreciation to my opponent Merhawi Tewolde for reviewing my thesis in depth and giving valuable feedback.

Lastly, but not least, I would like to express the fact that this thesis would have been impossible without the community behind numerous open-source and open-access projects.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

Sport analytics has been traditionally used to understand the relevant features of a sport for a player to succeed as well as to summarize the most important information on matches and players using statistical Key Performance Indicators (KPI). Evaluating performance and characteristics of elite athletes [6] or assessing important metrics by sport categories (e.g. net and wall games, invasion games, and striking and fielding games) [9] have tried to give better understanding on sport dynamics.

The usage of video-recording games, the increase in computer power and the rise of machine learning techniques have enabled to evaluate sports in a way impossible before. The evaluation of traditional and new metrics can provide insights to further understanding the game dynamics of a particular sport (e.g. core strategies, effects of particular events on a sequence of events).

In a sport as ice hockey, it is crucial for teams to understand well game dynamics, players' performance and successful scoring strategies to develop and improve their playing performance. This may result, with good management, to more audience in matches, better sponsors, more money to fulfill and increase the objectives of the team.

Markov Decision Processes (MDP), in the view of Reinforcement Learning, provides a general framework to deal with decision making under particular events. With that, one can not only learn which are the best actions to take under a particular state, but also being able to see through historical data which are the most useful/successful strategies to win a match. Also, the creation of more complex valuation metrics for teams and players' valuation (e.g. understanding which players play best/worst together) can be of useful insight for hiring/firing purposes, even for gamblers to bet with meaningful knowledge.

1.2 Aim

The underlying purpose of this thesis is to create new performance metrics for ice hockey players by match, using MDPs on actions performed by players on a match (for the NHL data from Routley and Schulte's article [22]), to forecast players' performance based on their previous performances along the season.

1.3 Research questions

The research questions that this paper addresses are:

1. Can a MDP/RL be used to evaluate actions under certain time-series events ?
2. How can I store time series data for the usage of a MDP?
3. Can I predict player's performance based on their performances in the previous matches ?
4. Is there a way to use MDPs to create a metric that evaluates players for hiring/maintaining/firing purposes?

1.4 Literature Review

Markov Decision Processes (MDP) is a framework classically used for taking sequential decisions under uncertainty in a perfectly observable state. It was developed during the 1950s by Bellman [3], and it is currently used in several fields such as robotics, economics, sports, medicine or manufacturing.

In order to learn which actions are the best ones (i.e. they give higher rewards), the sequence of events with their associated actions need to be explored. Littman [14] explained the theory of a MDP in the view of the Q-learning algorithm from the reinforcement learning theory to find transition probabilities to states and best strategies on the framework of two agents with opposed goals while being in a common environment [8].

There is a lot of research made on player performance for different sports. Some related papers on other sports are fitting a MDP on rugby matches to evaluate which sequences of events are more prone to scores to find the most effective rugby strategies [2], or evaluating the EPV (Expected Point Value) per possession on basketball taking into account players configuration in the field and the ball position [4].

In ice hockey, one of the main metric used across years is the +/- metric: the difference on scoring between your team and the opponent relative to the current players on the ice. In other words, if you are a player on the ice rink and your team scores, you will be awarded with a +1. On the other hand, if you are on the ice ring and the opponent team scores, you as a player will get -1 point. This non-marginal metric has been tried to be improved in several papers [18],[17], [7] by using regressions (eg. logistic, ridge regression) that try to assess which player has had more impact marginalizing other players' valuation.

Schuckers [23] gave a value to all events performed as a measure to how nearer the possibility of a goal was in the following 20 seconds. Routley and Schulte [22] used the NHL dataset without a time constraint taking into account the following context: the Zone an event was happening, the Manpower Differential (MD) across teams, the Period in which the match was happening and the Goal Difference (GD) between teams. With this, they created a MDP process that gives players a valuation based on the improved probability of an event to scoring vs receiving a score, extending it lately to a better dataset taking into account more actions and better locations on the field [24], [25]. Also, Liu and Schulte [15] used LSTM (Deep Neural Networks) to calculate player's performance on the NHL database using more context variables such as the remaining time of the match or whether and action was a success.

MDPs have also been used to calculate the probability of winning a match given that an action is performed by a player under certain context variables (Goal and Manpower differential) [11]. Moreover, the impact of players goals under context has been calculated to know which players are more valuable [20]. Also, the best pairs of players have been calculated for each team based on goal performance, taking into account positions of players and the time these players has played together each match [16].

2 Theory

This thesis deals with discrete variables and states. Therefore, the theory described in this project is focused on discrete data, even if a large part of it is common with continuous data. The notation as well as the explanation used in the article follows the notation and the structure of the MIT book *Decision Making Under Uncertainty: Theory and Application (2015)* [12].

2.1 Markov Decision Processes (MDP)

Markov Decision Processes (MDPs) provide a framework for sequential decision making in situations where the outcome of a given action is non-deterministic. More specifically, an agent chooses action a_t at time t observing a state s_t based on an associated Reward $R(s_t, a_t)$. This state evolves according to a defined state transition function $T(s'|s_t, a_t)$ which depends on the action a_t taken at that particular state. The system modeled is assumed to follow the Markov property, meaning that its state at time t depends only on its state at time $t-1$.

Illustration 2.1 represents graphically a Markov Decision Process with States (S), Actions (A), Rewards (R) for 3 general sequential processes.

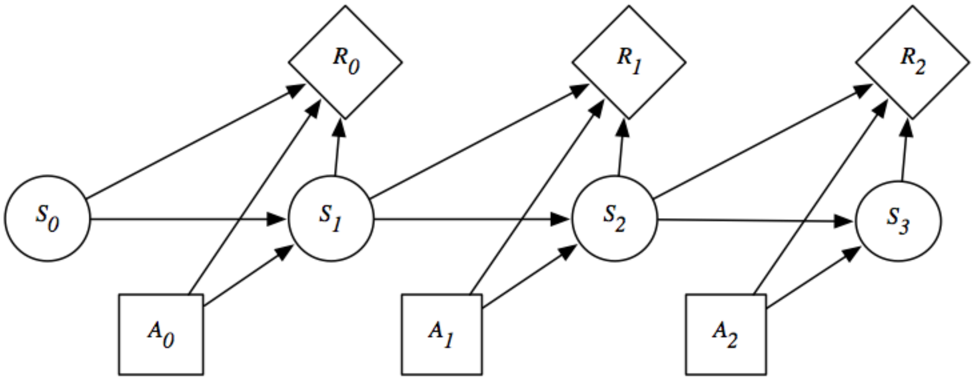


Figure 2.1: Graphical illustration of a Markov Decision Process with States (S), Actions (A), Rewards (R) for 3 general sequential processes(from *Artificial Intelligence: Foundations of Computational Agents* [21], figure 9.12).

Transition probabilities between states depend on the action taken. Depending on this action, one can decide which events are more likely to happen if one knows the desired outcome of the next event.

To learn what is the probability of state s' occurring after the state s under the choice a certain action, let's define first the following concepts following Littman's notation [14]:

- Occ(s) is the number of times state s occurs in the data
- Occ(s, s') is the number of times state s is followed by state s'.
- Then, the probability of state s' occurring after the state s is defined as

  $$Pr(s,s') = \frac{Occ(s,s')}{Occ(s)} = T(s'|s_t, a_t),$$

  also called the transition probability. The set of all possible transitions between states is then T.

In other words, a MPD can be explained by the tuple (S,A,T,R):

- S, the set of states
- A, the set of actions the agent can engage in
- T, the set of of all possible transitions. In a specific way, the transition probability $Pr(s,s')$ stands for the probability of state $s' = s_{t+1}$ occurring after state $s = s_t$ at time t given that an action $a = a_t$ has been taken, which is

  $$Pr(s,s') = Pr(s' = s_{t+1} \mid s = s_t, a = a_t) = \frac{Occ(s,s')}{Occ(s)} = T(s'|s_t, a_t)$$

- R, the set of rewards. For a specific state s_t, the reward is transitioning from s_t to s_{t+1} dependent on the action taken at each state, as $R(s,s')$. If the reward is not time dependent, then it is just R(s).

MDP example

A clear example of a MDP is the modelling of an ice hockey player's decision in control of the puck in the offensive zone of the field: The 4-tuple presented before <(S,A,R,T,)> would be used to describe what each variable would represent:

- States, representing the set of conditions a specific player might be in under which she/he performs an action (e.g. player shots the puck in the offensive zone (Zone), winning 1-0 (score of the match), same number of players per team (manpower differential) and first period (period the match is in)). The latter described is only one state, created by the combination of several variables, stated in between parenthesis. Therefore, it can be seen that there might be a lot of different states on a match, as many combinations as the multiplication of the values each variable can take.
- Actions the player can perform, which might be to pass, shot or dribble.
- Rewards the player receives for engaging into an action that improves statistically the probability of scoring or reducing the probability of receiving a goal. As an example, the reward might be set to 1 for the states that involve scoring a goal and -1 for receiving a goal (i.e. whatever the zone, the score of a match or the period values are). All the other states have a reward of 0.
- Transition function, representing the set of probabilities of the following state given the action performed in a particular current state (e.g. related to the example set in States: the probability of scoring given a shot in the light of the context the player is in).

The Utility and Reward function

The choice of an action at each point t in time has an associated reward r_t. In a finite horizon problem (finite number of states, i.e. n states), the total utility is defined by the sum of associated rewards to the action performed:

$$\sum_{t=0}^{n-1} r_t \tag{2.1}$$

The problem of this simple approach is that for an infinite time horizon (i.e. $n = \infty$) the reward function could reach the value ∞, so that no specific action would be preferred under this case. The simplest way to solve it is to add a discount factor γ^t (which can take values between 0 and 1) that is affected by time t. With that, a nearer reward would be preferable in case of equal rewards.

$$\sum_{t=0}^{n-1} \gamma^t r_t \tag{2.2}$$

That is one way to deal with the infinite horizon problem, but there are other ways such as calculating the average reward:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{t=0}^{n-1} r_t \tag{2.3}$$

Defining policies and utilities

A policy that determines the choice of an action at a point in time t given the state s_t we are in is expressed as $\pi_t(s_t)$, (i.e. recall the Markov assumption from the beginning of the section: the current state s defines all previous states). When transitions and rewards are stationary (i.e. they are constant across time), a policy π can be expressed as $\pi(s)$, since it is not time dependent. The expected utility of a policy $\pi(s)$ is denoted as $U^\pi(s)$. The optimal policy, expressed as π^* at state s is then defined as the policy that maximizes the expected utility:

$$\pi^*(s) = \arg\max_\pi U^\pi(s) \tag{2.4}$$

Calculating the expected utility from a specific policy is called *policy evaluation*. Let's assume now that we want to calculate the utility of a policy with t time steps. If t is 0 (the policy is not executed), then $U_0^\pi(s) = 0$. If the policy is execute for 1 step, then we get that $U_1^\pi(s) = R(s, \pi(s))$, being the reward associated to step 1. If we know the utility associated to π for t-1 steps, then the utility of π for t time steps is:

$$U_t^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s'|s, \pi(s)) U_{t-1}^\pi(s') \tag{2.5}$$

For an infinite time horizon with discounted rewards (i.e. with discount factor γ), equation 2.5 shows that the Utility of a sequence of policies is explained by the reward got at the current state given the policy π chosen at that state plus the sum of all future states defined by the Transition function (which defines with probability the exact future state given the current policy and state) and its associated Utility function at that future state multiplied by the discount factor γ.

$$U^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s'|s, \pi(s)) U^\pi(s') \tag{2.6}$$

Solving a MDP: the value iteration algorithm

There exists several ways to solve Markov Decision Processes, such as *value iteration* or *policy iteration*. However, in this thesis there will only be presented the *value iteration* one since it is the one used in the thesis for simplicity in implementation.

When trying to optimize a MDP policy, we are trying to get the highest expected utility possible. This is done through choosing the action that maximize the rewards. For an associated time horizon n, the optimal utility function (only for time horizon equal to n) is carried out following equation 2.7:

$$U_n(s) = \max_a (R(s,a) + \sum_{s'} T(s'|s,a) U_{t-1}(s')) \tag{2.7}$$

In the case of an infinite time horizon, the optimal utility is calculated using equation 2.8:

$$U^*(s) = \max_a (R(s,a) + \gamma \sum_{s'} T(s'|s,a) U^*(s')) \tag{2.8}$$

Once the optimal utility function is know, the optimal value function $\pi(s)$ can be estimated from the optimal policy $U_n^*(s)$ by just updating the estimate of U^* using 2.8. Then, optimal policy $\pi(s)$ is extracted using equation 2.9:

$$\pi(s) = \arg\max_a (R(s,a) + \gamma \sum_{s'} T(s'|s,a) U^*(s')) \tag{2.9}$$

I provide below the pseudocode for the Value Iteration algorithm [26] using the notation of this section.

Algorithm 1 Dynamic Programming for Value Iteration

```
Require: Markov Game Model, convergence criterion c, maximum number of iterations M
lastValue = 0
currentValue = 0
converged = False
for $i = 1; i \leqslant M, i \leftarrow i + 1$ do
    for all states s in the Markov Game Model do
        if $converged == False$ then
            $U_{i+1} = R(s,a) + \gamma \frac{1}{Occ(s)} \sum_{(s,s')\in E} Occ(s,s') U_i(s'))$
            $currentValue = currentValue + |U_{i+1}|$
        end if
    end for
    if $converged == False$ then
        if $\frac{(currentValue - lastValue)}{currentValue} < c$ then
            converged = True
        end if
    end if
    lastValue = currentValue
    currentValue = 0
end for
It outputs all the values of the optimal utilities for each state
```

In order to speed up convergence on the value iteration algorithm, convergence is set to be reached when $|(currentValue - lastValue)| < c$ being c a parameter of the programmer's choice known as the Bellman residual.

2.2 AD-tree

Motivation for AD-trees on MDP

Let's imagine we are on a grid of 4x4. Therefore, we have 16 possible states. Let's imagine we can take take four actions in each state: up, down, right or left. However, when one chooses an action, there is a possibility that you go in another direction (e.g. choosing to go up, but for some reason you go down), but one does not know this probability. However, we know 20 people have been walking through this grid, and we know their states and actions taken. From here, one could learn the probability of going from one state to another given an action taken, which is 16x4 combinations of states x actions.

Now let's imagine there is some reward to get to a specific point in the grid, but there are also negative rewards trough the path, such that the actions should be taken minimizing the probability of getting into negative reward places, and then choosing the best action at each state (see figure 2.2).

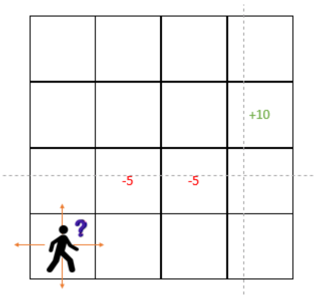


Figure 2.2: Graphical illustration of a person trying to choose the best set of actions and therefore the best path from one specific state in a grid of 4x4 states with 4 possible actions to choose in every state.

To do so, one could be counting every time the transitions from one state to another given an action from the data of the people explained above. Nevertheless, it would be handy to have this calculated for each possible state beforehand, since these transitions might be calculated more than once since several states would be used several times for different paths before getting to the best path. This is specially important when one has a lot of states and actions that can be taken. For instance, calculating the transition probabilities from a state to another several times with 10 million people with paths and a grid of 1000x1000 states with 80 possible actions could get totally unfeasible. However, if those probabilities and occurrences of states are stored beforehand, this would speed a lot the calculations.

Introduction to the AD-tree

An AD-tree is a machine learning method developed by B. Anderson and A. Moore [1] to summarize relations of events that happen sequentially in a database. Their main purpose was establishing associative rules to extract some general rules (e.g. what is the probability that after a shot there is a goal) based on support and confidence measures, explained below in this same section.

Following their article's notation, a literal is an attribute-value pair such as a variable *event* taking the value of "shot", such that "*event* = shot". However, we could think of a literal also as a unique combination of variables, englobed in a state. For instance, if a literal has two variables (called states in our MDP in previous sections), for instance the variable *event* with unique values of "shot", "goal", "pass" and and *zone* with values "offense", defense", there exist in this case 3x2 possible combinations of literals (or states), such as "shot" and "offense"

Then, L is all possible combinations of literals in the database (eg. in our example, the 6 possible combinations). If l1 being "shot" and "offense" happens before l2 being "goal" and "offense", then an association is created as $l1 \rightarrow l2$. This association rule implies that $l1 \rightarrow l2$, where $l1, l2 \subset L$, and $l1 \cap l2 = \emptyset$. l1 will refer to the first event, being the antecedent, and l2 to the conditional event, called the consequent. Then, we can understand a time series dataset as a compound of rules, each literal associated by time with the following literal.

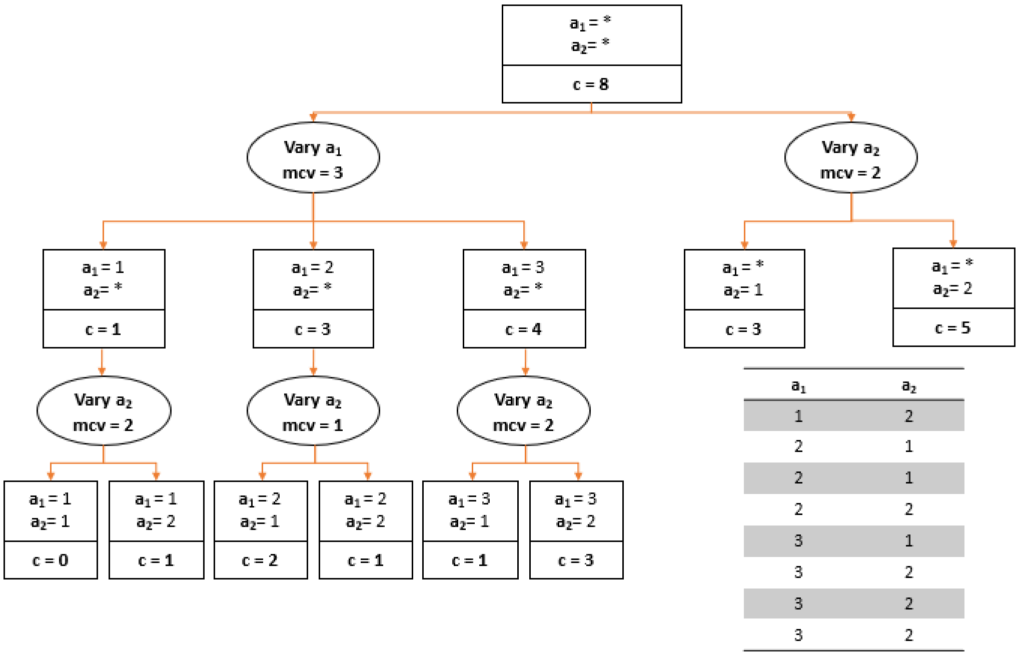
The support measure is a metric defined as the number of records in the database that match all the attribute-value pairs in S. In other words, it is counting how many times an association of events l1, l2 occur on the dataset, also expressed as $supp(l1 \cup l2)$. The more times an association occurs, the higher its support and the higher its importance.

The confidence measure is another metric defined as the percentage of records that matches $l1 \cup l2$ out of all records that match l1. That is, the fraction between support of the association rule $l1 \cup l2$ and the support of the antecedent l1. It is expressed as $conf(l1 \cup l2) = \frac{supp(l1 \cup l2)}{supp(l1)}$. The higher the confidence measure, the stronger and the more probable an association is, giving more importance to that specific association rule.

The AD-tree Data Structure

An AD-tree is a summary of a database with support measures and counts for each association and literal. This means that all possible queries have been calculated in advance to provide basic statistics.

To do that, a basic structure to record all the queries is needed (a tree structure with its support). There are two main components in the AD-tree structure. An ADnode represents a query and stores the number of times that query happens. Each ADnode, (shown as rectangles in figure 2.3) has child-nodes called "vary nodes" (shown as ovals). These "vary nodes" do not store counts but group ADnodes with one only feature. It can also contain the most common value (mcv). The "vary nodes" child of an ADnode has one child for each value v_j for feature a_i. These grandchildren ADnodes specialize the grandparent's query by storing the counts of the specific queries. The counts (c) of every specific association happening are also shown in the graph.



a_1	a_2
1	2
2	1
2	1
2	2
3	1
3	2
3	2
3	2

Figure 2.3: Graphical representation of an ADtree based on the dataset shown in the same figure. ADnodes are shown as rectanges. Vary nodes are shown as ovals.

2.3 Time series

The time series explained in this thesis only serve the purpose of understanding the thesis work. Therefore, theory and models will not be explained in depth. To better understand the models that are going to be presented in this paper, *Time Series Analysis With Applications in R*, [5] is a book that provides a lot of practical examples using R's software. Also, the notation used in this part of the thesis follows the book's notation.

Motivation for Time Series modelling

The main objective behind time series modelling is being able to predict certain behaviours of variables through time. Given that the objective of the thesis is trying to predict player's performance valuation based on previous performances, time series modelling can be quite handy due to the ability of modelling different types of trends from previous observations and then forecast the following performances. It is often alleged that forecasting players' performance for match 26 may not depend on matches 1 and 2, but on the most recent previous matches (e.g. the last 5 matches). This could be taken as an evaluation of how good that player is in that specific point in time, therefore used to predict their next match performance.

The concept of Stationarity

Let Y be a time series data. In order to evaluate whether this data has the structure of a stochastic process, it is assumed that the laws in which all stochastic processes occur do not change over time. This assumption is called stationarity. Let's assume that Y_t and Y_s are specific points in time in our Y data. The 3 main laws that define whether a process is stationary (referring in this thesis for weak stationarity) are:

- The mean function is constant over time: $\mathbf{E}(Y_t) = \mathbf{E}(Y_{t-k})$, where k is smaller than t as a value.
- The covariance between Y_t and Y_s being s<t depends only on time difference ∣t-s∣ but not on actual times of s and t. Therefore: $Cov(Y_t, Y_s) = Cov(Y_{0,|t-s|})$
- The variance is : $Var(Y_t) < \infty$

The most typical stationary process is the white noise process, which is a sequence of a independent and identically distributed (iid) random variables e_t.

ARMA Models

An autoregressive process (AR) is a regression process where the current observation is a combination of the p previous observations plus some difference explained as e_t. This model was proposed by Yule in 1926:

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + ... + \phi_p Y_{t-p} + e_t$$

A moving average process (MA) is a another regression process based on q previous white-noises with some specific weights. This model was firstly introduced by Slutsky in 1927:

$$Y_t = e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q}$$

ARIMA Models

In real life, though, time series data is not often stationary. Nevertheless, there exist techniques which might help us dealing with it.

Differencing is one of the most used methods for making data stationary, which is calculating the difference in Y for lags. As an example, a first order difference would be $\bigtriangledown Y_t = Y_t - Y_{t-1}$. Another method that is used together with differencing is taking logarithms of the data to avoid modelling exponential growth in the data set.

An Integrated autoregressive moving average models (ARIMA), introduced by Box and Jenkins in 1976, is a regression process that takes into account AR, differencing and MA processes to model time series. Considering W_t as $Y_t - Y_{t-1}$, an ARIMA process can be defined as

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + ... + \phi_p W_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} \tag{2.10}$$

which can be expressed as :

$$\begin{aligned} Y_t - Y_{t-1} = \phi_1(Y_{t-1} - Y_{t-2}) + \phi_2(Y_{t-2} - Y_{t-3}) + ... + \phi_p(Y_{t-p} - Y_{t-p-1}) \\ + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} \end{aligned} \tag{2.11}$$

and therefore

$$\begin{aligned} Y_t = (1 + \phi_1)Y_{t-1} + (\phi_2 - \phi_1)Y_{t-2} + ... + (\phi_p - \phi_{p-1})Y_{t-p} + \theta_p Y_{t-p-1} \\ + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q} \end{aligned} \tag{2.12}$$

which looks like an ARMA process. By using ARIMA processes, AR and MA processes are taken into account on one single process making it really interesting for its application.

Estimation and forecasting of coefficients

In order to evaluate which ARIMA model fits best the data, some function is required that maximizes the probability of a model θ given the data. This is done trough the log likelihood (LL) function 2.13.

$$LL(\theta) = -\frac{T}{2} log(2\pi) - \frac{T}{2} log(\sigma^2) - \sum_{t=1}^{T} \frac{Y_t^2}{2\sigma^2} \tag{2.13}$$

, where T is the length of the T is the length of the time series. In order not to overfit the data to a model with a lot of parameters, there exist several functions that try to eliminate this problem by penalizing models with many parameters k, called the Akaike Criterion information (AIC). The model chosen would be the one minimum AIC value:

$$AIC = 2k - 2log(L)$$

, where L is the likelihood.

With that, an algorithm for generating efficient guesses of parameter values is also required, as well as initial values to these parameters to calculate the likelihoods of the model parameters. For this thesis, the *auto.arima()* function from the forecast package in R is used. This function already has an algorithm which is the Hyndman-Khandakar algorithm to automate the choice of the parameter values and ARIMA modelling [10].

Once the number and the value of the parameters is estimated, the forecast is done by supplying the data into the ARIMA model and comparing it to the real value. Typical statistics measures to understand the difference between the value forecasted x' and the real value x (i.e. the residuals) are the Mean Error (ME), the mean across all the residuals, the Root Mean Squared Error (RMSE), and the Mean Absolute Error (MAE) with the formulas below:

$$ME = \frac{1}{n} \sum_{i=1}^{n} (x'_i - x_i) \tag{2.14}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \mid (x'_i - x_i) \mid \tag{2.15}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x'_i - x_i)^2} \tag{2.16}$$

where n is the length of forecasts.

The smaller these measures are, the better, since it means that a prediction is more accurate.

3 Domain: Hockey Rules and Hockey Data

3.1 Hockey Rules

Ice hockey, documented for the first time in 1773 in the book *Sports and Pastimes, to Which Are Prefixed, Memoirs of the Author: Including a New Mode of Infant Education*, by Richard Johnson, is a two-team team sport played on an ice rink with 5 players plus a goalkeeper for each team. The objective is to score more goals than the opponent in 3 periods of 20 minutes. On regular season, if the 3rd period ends up with a draw, there is an extra period of 5 minutes where the first to score immediately wins. In case there is no score, on a regular season, there will be shootouts for both teams. In playoffs, new periods of 20 minutes will continue until one team scores.

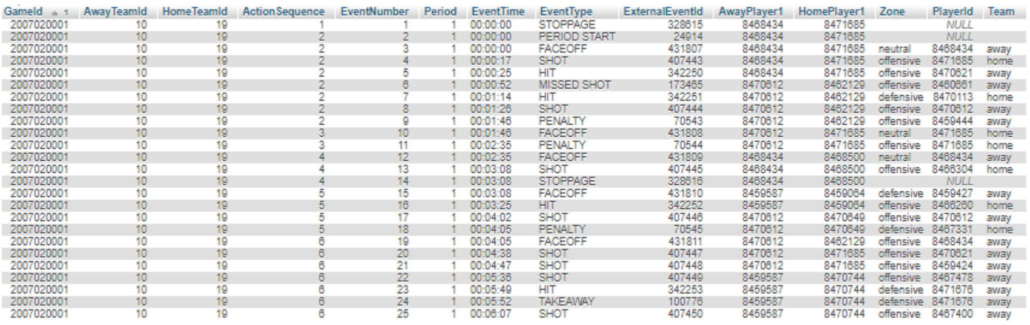
After a penalty occurs, the player who occasioned it will sit in the penalty box for a certain amount of minutes depending on the severity of the fault. The penalized team will be shorthanded, creating a Manpower Differential (MD) between the teams. A team has *powerplay* when their team has more players on the field than their opponent, resulting into manpower advantage. A *powerplay* goal means that a team scored with positive *Manpower differential*. The opposite is called a *shorthanded goal*.

3.2 Hockey Data

The datasets this project will use are the NHL dataset from Routley and Schulte's paper in 2015 [22], scrapped from *www.nhl.com* and some statistics related to NHL hockey players for seasons 2007-2008 and 2008-2009 extracted from *http://www.dropyourgloves.com*

NHL Hockey Dataset

The NHL dataset is a relational database with 2,827,467 play-by-play events recorded by the NHL, containing complete 2007-2014 seasons (i.e. regular season and playoff games) and the first 512 games of the 2014-2015 regular season. General information about the dataset is displayed in Table 3.2a. Also, a general view of the most important table is shown in figure 3.1.



GameId	AwayTeamId	HomeTeamId	ActionSequence	EventNumber	Period	EventTime	EventType	ExternalEventId	AwayPlayer1	HomePlayer1	Zone	PlayerId	Team
2007020001	10	19	1	1	1	00:00:00	STOPPAGE	328615	8468434	8471685		NULL	
2007020001	10	19	2	2	1	00:00:00	PERIOD START	24914	8468434	8471685		NULL	
2007020001	10	19	2	3	1	00:00:00	FACEOFF	431807	8468434	8471685	neutral	8468434	away
2007020001	10	19	2	4	1	00:00:17	SHOT	407443	8468434	8471685	offensive	8471685	home
2007020001	10	19	2	5	1	00:00:25	HIT	342250	8468434	8471685	offensive	8470621	away
2007020001	10	19	2	6	1	00:00:52	MISSED SHOT	173465	8470612	8462129	offensive	8460661	away
2007020001	10	19	2	7	1	00:01:14	HIT	342251	8470612	8462129	defensive	8470113	home
2007020001	10	19	2	8	1	00:01:26	SHOT	407444	8470612	8462129	offensive	8470612	away
2007020001	10	19	2	9	1	00:01:46	PENALTY	70543	8470612	8462129	offensive	8459444	away
2007020001	10	19	3	10	1	00:01:46	FACEOFF	431808	8470612	8471685	neutral	8471685	home
2007020001	10	19	3	11	1	00:02:35	PENALTY	70544	8470612	8471685	offensive	8471685	home
2007020001	10	19	4	12	1	00:02:35	FACEOFF	431809	8468434	8468500	neutral	8468434	away
2007020001	10	19	4	13	1	00:03:08	SHOT	407445	8468434	8468500	offensive	8466304	home
2007020001	10	19	4	14	1	00:03:08	STOPPAGE	328616	8468434	8468500		NULL	
2007020001	10	19	5	15	1	00:03:08	FACEOFF	431810	8459587	8459064	defensive	8459427	away
2007020001	10	19	5	16	1	00:03:25	HIT	342252	8459587	8459064	offensive	8466260	home
2007020001	10	19	5	17	1	00:04:02	SHOT	407446	8470612	8470649	offensive	8470612	away
2007020001	10	19	5	18	1	00:04:05	PENALTY	70545	8470612	8470649	defensive	8467331	home
2007020001	10	19	6	19	1	00:04:05	FACEOFF	431811	8470612	8462129	offensive	8468434	away
2007020001	10	19	6	20	1	00:04:38	SHOT	407447	8470612	8471685	offensive	8470621	away
2007020001	10	19	6	21	1	00:04:47	SHOT	407448	8470612	8471685	offensive	8459424	away
2007020001	10	19	6	22	1	00:05:36	SHOT	407449	8459587	8470744	offensive	8467478	away
2007020001	10	19	6	23	1	00:05:49	HIT	342253	8459587	8470744	defensive	8471676	away
2007020001	10	19	6	24	1	00:05:52	TAKEAWAY	100776	8459587	8470744	defensive	8471676	away
2007020001	10	19	6	25	1	00:06:07	SHOT	407450	8459587	8470744	offensive	8467400	away

Figure 3.1: Illustration of the main raw variables in the main table of the SQL database.

The raw variables, extracted from different relational tables using MySQL are:

- *GameId*: Id number identifying the season year and the number of a match in that season.
- *AwayTeamId*: Id related to the team playing away from their home.
- *HomeTeamId*: Id related to the team playing at home.
- *Action Sequence*: Number related to the sequence of plays in a match.
- *Event Number*: Unique number related to each event happening in a match.
- *Period*: Number identifying the period a match is in.
- *EventTime*: Time of an event happening in a period of a match.
- *EventType*: Event or action happening in the match.
- *ExternalEventId*: Id used to relate one table to other tables of events. It is important to know that *ExternalEventId* is not unique, so it must be used together with *GameId* and *EventNumber*.
- *AwayPlayer1*: Example of a variable where the id of an away player is shown. There are 9 *AwayPlayer* variables for each row in the table, accounting for the away players that could be playing at the same time. It can take null values, meaning no player is taken into account in that specific cell.
- *HomePlayer1*: Example of a variable where the id of a home player is shown. There are 9 *HomePlayer* variables for each row in the table, accounting for the home players that could be playing at the same time. It can take null values, meaning no player is taken into account in that specific cell.
- *Team*: It specifies, in case *EventType* is an action, which team is performing the action. It can take 'Home' and 'Away' values.
- *Zone*: It describes, from the point of view of the team performing the action, the zone in which the action described in *EventType* is happening.
- *PlayerId*: It describes, in case *EventType* is an action, the id associated to the player performing an action.

Table 3.2b shows the values the variable *EventType* can take. Each value can either be an action event or start/end event. An action event represents those events performed by players, whereas start/end events are events not performed by players. Each time event is continuous of another time event. That means we have a time series dataset. Additionally,

Nº of events	2,827,467
Nº sequences of events	590,924
Nº of games	9,220
Nº of players	1,951
Nº of teams	32

(a) General Information.

Action Events	**Start/End Events**
Faceoff	Period Starts
Missed Shot	Period Ends
Shot	Early Intermission Starts
Hit	Early Intermission Ends
Blocked Shot	Stoppage
Giveaway	Penalty
Goal	Game End
Takeaway	Shootout completed
	Game off

(b) Types of events.

Figure 3.2: The NHL Dataset.

each event performed by a player (an action) has an associated zone in which the event occurs (Home, Neutral, Away) and which team performs that event (Home or Away team).

Two extra variables are created to give context in which the ice hockey match is happening:

- Goal Differential (*GD*): It defines by how many goals the home team is winning by, as

  $$GD = HomeGoals - AwayGoals$$

  , until that specific time of the match. A GD of 1 means that the home team wins by 1 goal whereas a GD of -1 stands for the away team winning by 1. When a match starts, the variable GD takes a value of 0.

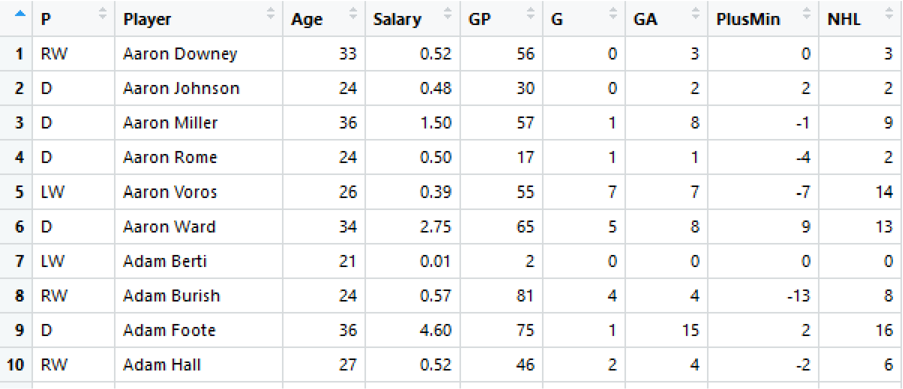
- Manpower Differential (*MD*): It defines the difference of players in the rink between both teams following the exact same logic as GD, as

  $$MD = HomePlayersinRink - AwayPlayersinRink$$

  , calculated from evaluating how many playerids there are in the *AwayPlayer* and *HomePlayer* at each event.

Player statistics dataset

The webpage *www.dropyourgloves.com* is a database for statistics on ice hockey. The data scrapped describes the general performance metrics for all ice hockey players that have played during a season. There have been scrapped 2 tables, each one accounting for the players and their resume statistics for seasons 2007-2008 and 2008-2009.



	P	Player	Age	Salary	GP	G	GA	PlusMin	NHL
1	RW	Aaron Downey	33	0.52	56	0	3	0	3
2	D	Aaron Johnson	24	0.48	30	0	2	2	2
3	D	Aaron Miller	36	1.50	57	1	8	-1	9
4	D	Aaron Rome	24	0.50	17	1	1	-4	2
5	LW	Aaron Voros	26	0.39	55	7	7	-7	14
6	D	Aaron Ward	34	2.75	65	5	8	9	13
7	LW	Adam Berti	21	0.01	2	0	0	0	0
8	RW	Adam Burish	24	0.57	81	4	4	-13	8
9	D	Adam Foote	36	4.60	75	1	15	2	16
10	RW	Adam Hall	27	0.52	46	2	4	-2	6

Figure 3.3: Illustration of variables extracted from *www.dropyourgloves.com*.

The variables shown in 3.3 are the following ones:

- Position (*P*): The position in which players are playing. It can take the following values: Right Wing (RW), Left Wing (LW), Forward (F), Defense (D), Goalie (G) and Central (C). C, LW and RW have been preprocessed to be Forward (F), since all of them make reference to players playing in the offense.
- *Player*: The name of the player.
- *Salary*: The salary of the player, in millions of american dollars.
- Games Played (*GP*): The number of games played by a player.
- Goals (*G*): The number of goals performed.
- Goals Assisted (*GA*): The number of goals assisted.
- *PlusMin*: The *PlusMin* measure stands for the plus minus (+/-) measure, which accounts for the number of goals a team has scored when that specific player was on the rink minus the number of scores a team has received when that player was on the rink.
- Points (*NHL*): The sum of number of goals assisted and goals, such that $Points = GA + G$.

4 Method

The methodology of this thesis follows the diagram described in figure 4.1.

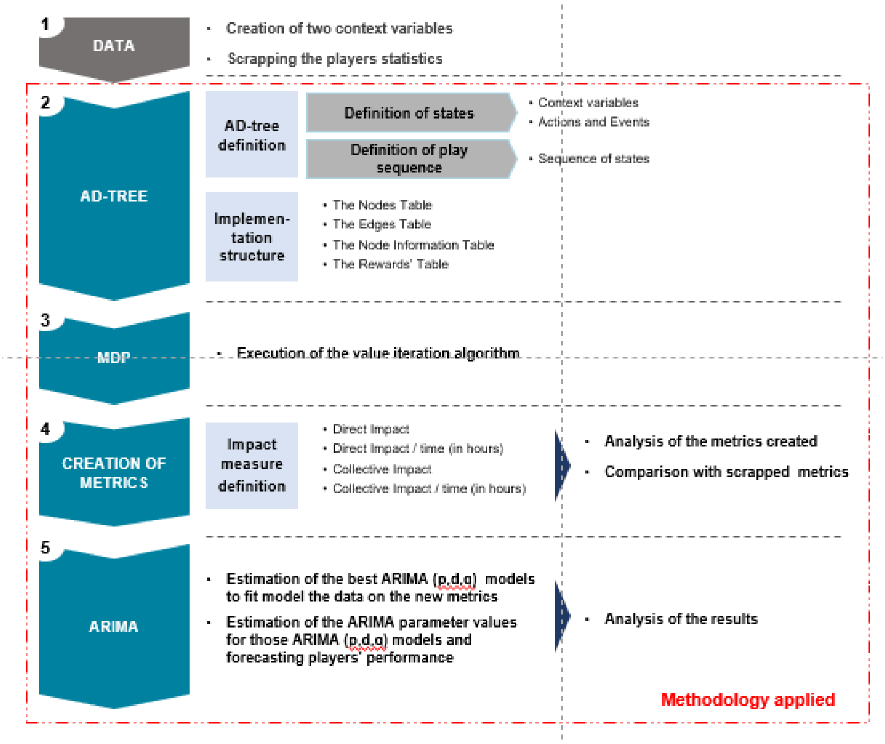

1
DATA
Creation of two context variables
Scrapping the players statistics
2
AD-TREE
AD-tree definition
Definition of states
Context variables
Actions and Events
Definition of play sequence
Sequence of states
Implemen-tation structure
The Nodes Table
The Edges Table
The Node Information Table
The Rewards' Table
3
MDP
Execution of the value iteration algorithm
4
CREATION OF METRICS
Impact measure definition
Direct Impact
Direct Impact / time (in hours)
Collective Impact
Collective Impact / time (in hours)
Analysis of the metrics created
Comparison with scrapped metrics
5
ARIMA
Estimation of the best ARIMA (p,d,q) models to fit model the data on the new metrics
Estimation of the ARIMA parameter values for those ARIMA (p,d,q) models and forecasting players' performance
Analysis of the results
Methodology applied

Figure 4.1: Diagram summary of the methodology.

After creating two context variables *goal differential* (GD) and *manpower differential* (MD) and scrapping players statistics for seasons 2007-2008 and 2008-2009, an AD-tree is used to summarize the states of a match. Combining context variables (*GD, MD* and *Period*) and actions variables (*EventType, Zone* and *Team*), one gets all the possible combinations of contexts and actions that can happen in a match, called states. By defining also what a playing sequence is, several tables are implemented to summarize the occurrences of states happening after others. Also, rewards are set to specific states where the *EventType* = *goal* is happening. With this, the value iteration algorithm is applied to calculate the closeness of each state to score, one for the states of the home team, and one for the away team.

Following Routley and Schulte's article [22], an *impact measure* is created. This measure calculates the contribution to score of each action in a match. Also, the time each player plays in a match is calculated. By using this *impact measure*, four other statistical metrics are created by match and player, having time series statistics of players' performance for each of the four metrics, i.e. 82 match performances (related to the n^{o} of matches for the regular season in the ice hockey league) for each player in the league, for seasons 2007-2008 and 2008-2009.

Besides general analysis of these metrics and a comparison with the statistical metrics scrapped from *www.dropyourgloves.com*, for each derived metric, the new metrics are fitted, being player performance through season, to estimate the most typical number of parameters in ARIMA (p,d,q) models. The 98% most frequent ARIMA (p,d,q) models are chosen to estimate, for each case, the parameter values related to players performance through 5 matches, extracted from the metrics created, and then forecast 1 step ahead match performance. Finally, a comparison of the different ARIMA (p,d,q) models is done to evaluate which ARIMA (p,d,q) model works better.

4.1 The AD-tree definition

Definition of context variables

In sports, not only is it important to understand the actions players take, but also under which circumstances that player is deciding and performing that specific action. In ice hockey, the probability between scoring a goal given that one team has more players in the field might be higher than when the same team has less. Probably, a reckless pass with fewer players in the rink might imply a higher risk of receiving a takeaway and therefore receiving a score. Also, it might not be equal the probability of scoring an extra goal in period 3 when you are losing by 1 goal than when you are winning by 3.

Thus, taking into consideration the context in which events are performed might be crucial to understand a game itself and evaluate which are the players that play better. With that, the choice of context variables that are important for the game are *Goal Differential*, *Manpower Differential* and *Period* [22].

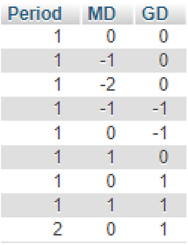
Those context variables are the following ones displayed in the table 4.1.

Notation	Variable Name	Range
GD	Goal Differential	[-8,8]
MD	Manpower Differential	[-3,3]
P	Period	[1,7]

Table 4.1: Context variables associated to each state.

Table 4.1 is showing the range in which those context variable might happen. As a summary of them, that means our events might happen under 17x7x7 combinations of context variables. That results in a total of 833 context states where each action could happen.

An example of the values the context variables can take is displayed in figure 4.2.



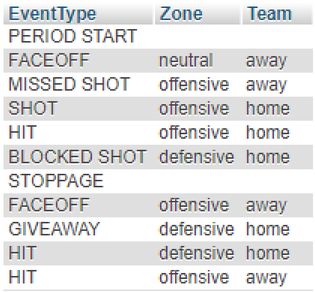
Period	MD	GD
1	0	0
1	-1	0
1	-2	0
1	-1	-1
1	0	-1
1	1	0
1	0	1
1	1	1
2	0	1

Figure 4.2: Illustrative example of possible context variables.

Definition of action events and a play sequence

As explained in the dataset information section 3.2, there are 8 main actions a player can perform (see table 3.2b). Those actions have associated a zone Z in which the action is performed (Offensive, Neutral or Defensive) subjected to the teams T performing the action (Home, Away).

An example of the values an action event can take is displayed in figure 4.3.

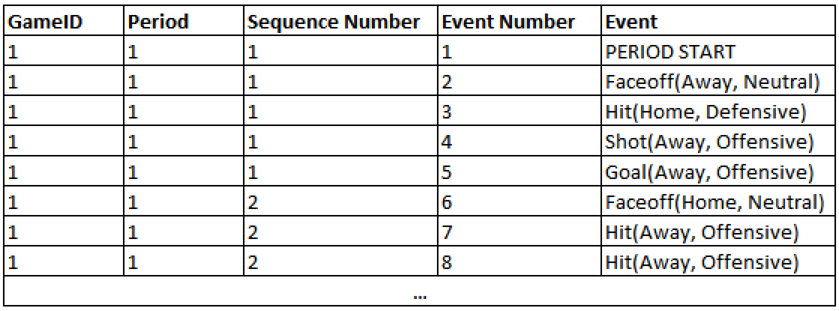


EventType	Zone	Team
PERIOD START		
FACEOFF	neutral	away
MISSED SHOT	offensive	away
SHOT	offensive	home
HIT	offensive	home
BLOCKED SHOT	defensive	home
STOPPAGE		
FACEOFF	offensive	away
GIVEAWAY	defensive	home
HIT	defensive	home
HIT	offensive	away

Figure 4.3: Illustrative example of possible action events.

A playing sequence is a series of action events. Each play sequence begins with a start marker. If the play sequence is complete, it will end with an end marker. However, it may be that there are sequences that are not complete. Start/End markers are those events in the right column of table 3.2b, adding faceoffs and shots as starting markers and goals as ending markers. To display it clearer here, action events will be explained in the notation form of a(T,Z) [13], where T is Team and Z is Zone. Goal(Home, Offensive) is an example of the a(T,Z) structure.

An illustrative example is displayed below in figure 4.4:



GameID	Period	Sequence Number	Event Number	Event
1	1	1	1	PERIOD START
1	1	1	2	Faceoff(Away, Neutral)
1	1	1	3	Hit(Home, Defensive)
1	1	1	4	Shot(Away, Offensive)
1	1	1	5	Goal(Away, Offensive)
1	1	2	6	Faceoff(Home, Neutral)
1	1	2	7	Hit(Away, Offensive)
1	1	2	8	Hit(Away, Offensive)
...				

Figure 4.4: Illustrative example of the variables got in the time series dataset for a hockey match with format a(T,Z) of the events.

Definition a state

Following Routley and Schulte's notation, a state s is denoted as $s < x, h >$ where x englobes the combination of context variables (being GD, MD and P) and h the action history (the variables in table 3.2b with zone Z, and the team T performing that action). If the sequence h is empty, the state is purely a context node. A context node is a state which does not have any action associated (i.e. the *NodeName* variable is set to 'state', with no values in team and zone variables). An example of this can be found in figure 4.5. Recall that a state is what it has been called previously a literal, and that is the terminology that will be used on the trees pseudocode. States and literals are the same.

4.2 The AD-tree implementation and structure

The pseudocode for the AD-tree is presented algorithm 2. For each literal in the dataset, it is evaluated whether that specific literal has already occurred. If it has, the support of that specific unique node (i.e. literal or state) is incremented by one, and then the edges (i.e. the evaluation of a literal happening after another one) are also evaluated. If that specific literal has not occurred, a new node for that literal is added with support one. In the same way, if a specific relation of edges has not occurred, then a new node for edges is added. The AD-tree is done when no more literals (i.e. rows in the dataset) are left to summarize.

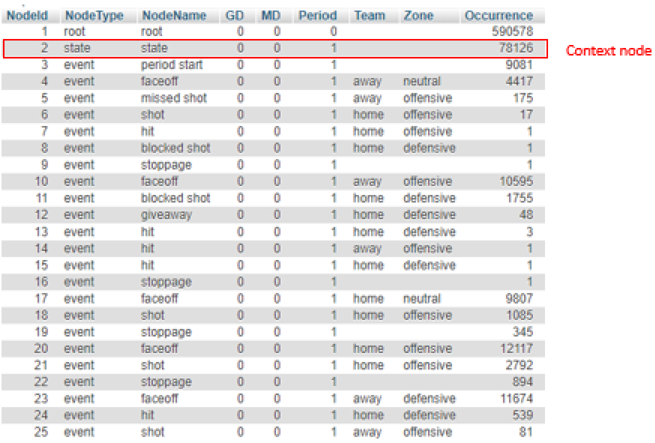
Algorithm 2 Build AD-Tree

```
function AddLiteralsToTree($Literals$)
    $currentNode \leftarrow Root$
    $previousNode \leftarrow none$
    $addedLeafLast \leftarrow false$
    for $literal$ in $Literals$ do
        $newNode \leftarrow$ CreateNode($literal$)
        if $currentNode == Root$ then
            $currentNode.Occurrence \leftarrow currentNode.Occurrence + 1$
            $stateNode \leftarrow CreateStateNode(literal)$
            $nextNode \leftarrow currentNode.FindChild(stateNode)$
            if $nextNode == none$ then
                $currentNode.AddChild(stateNode)$
                $stateNode.parent \leftarrow currentNode$
                $nextNode \leftarrow stateNode$
                $WriteEdge(currentNode, nextNode)$
            end if
            $IncrementEdgeOccurrence(currentNode, nextNode)$
            $currentNode \leftarrow nextNode$
            if $addedLeafLast$ then
                $nextNode \leftarrow previousNode.FindChild(currentNode)$
                if $nextNode == none$ then
                    $previousNode.AddChild(currentNode)$
                    $WriteEdge(previousNode, currentNode)$
                end if
                $IncrementEdgeOccurrence(previousNode, currentNode)$
                $addedLeafLast \leftarrow false$
            end if
            $currentNode.Occurrence \leftarrow currentNode.Occurrence + 1$
        end if
        $nextNode \leftarrow currentNode.FindChild(newNode)$
        if $nextNode == none$ then
            $currentNode.AddChild(newNode)$
            $newNode.parent \leftarrow currentNode$
            $nextNode \leftarrow newNode$
            $WriteEdge(currentNode, nextNode)$
        end if
        $IncrementEdgeOccurrence(currentNode, NextNode)$
        $previousNode \leftarrow currentNode$
        $currentNode \leftarrow nextNode$
        $currentNode.Occurrence \leftarrow currentNode.Occurrence + 1$
        if $literal == Endliteral$ then
            $currentNode \leftarrow Root$
            $addedLeafLast \leftarrow true$
        end if
    end for
end function
```

In order to deal with a large number of states or literals, an AD-tree has been created using a SQL database with four tables. Each table tries to store different basic useful information for the posterior calculus of the closeness of each state or literal to score a goal using a MDP.

The Nodes table

The Nodes table is the AD-tree table with all the summarization of all literals related to s$< x, h >$, being the dataset. Each row/node/state/literal is a unique combination of literals. Additionally, a variable *NodeId* is added to each state/literal working as a unique key. Finally, the *Occurrence* variable stands for how many times each state/literal event happens in the whole dataset, or in other words, storing the support of each literal (see figure 4.5).



NodeId	NodeType	NodeName	GD	MD	Period	Team	Zone	Occurrence
1	root	root	0	0	0			590578
2	state	state	0	0	1			78126
3	event	period start	0	0	1			9081
4	event	faceoff	0	0	1	away	neutral	4417
5	event	missed shot	0	0	1	away	offensive	175
6	event	shot	0	0	1	home	offensive	17
7	event	hit	0	0	1	home	offensive	1
8	event	blocked shot	0	0	1	home	defensive	1
9	event	stoppage	0	0	1			1
10	event	faceoff	0	0	1	away	offensive	10595
11	event	blocked shot	0	0	1	home	defensive	1755
12	event	giveaway	0	0	1	home	defensive	48
13	event	hit	0	0	1	home	defensive	3
14	event	hit	0	0	1	away	offensive	1
15	event	hit	0	0	1	home	defensive	1
16	event	stoppage	0	0	1			1
17	event	faceoff	0	0	1	home	neutral	9807
18	event	shot	0	0	1	home	offensive	1085
19	event	stoppage	0	0	1			345
20	event	faceoff	0	0	1	home	offensive	12117
21	event	shot	0	0	1	home	offensive	2792
22	event	stoppage	0	0	1			894
23	event	faceoff	0	0	1	away	defensive	11674
24	event	hit	0	0	1	home	defensive	539
25	event	shot	0	0	1	away	offensive	81

Context node

Figure 4.5: Example of the Nodes table.

The Edges table

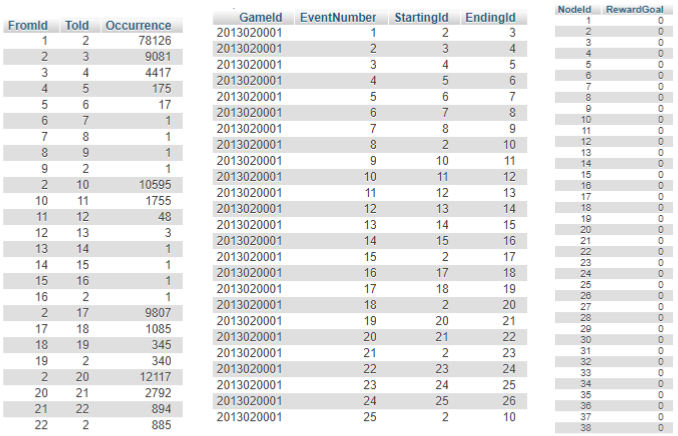
The Edges table stores the links between nodes in the dataset and its occurrence. The variables *FromId* and *ToId* refers to the link created between two nodes (see figure 4.6a). These are the only paths considered later on as possible transitions from one node to another node in the MDP. Non-existent pair of sequences are taken as impossible to occur and therefore with transition probability equal to 0.

The Node information table

The Node information table stores the original pairs of unique keys from the original table *(GameId, EventNumber)* with the matching state *NodeId* called *EndingId* occurring in that action and the previous state *StartingId* related to that specific *EndingId* (see figure 4.6b). Each *NodeId* is unique, but different *(GameId, EventNumber)* tuples can have the same *NodeId* value.

The Reward table

The Reward table stores the reward values associated to each *NodeId*. The values the *RewardGoal* variable takes are 0, 1 or -1 (see figure 4.6c). Since the main objective of ice hockey is scoring goals to win a match, the reward associated for scoring a goal is determined to be +1. The reward for receiving a goal then is -1, and 0 otherwise. In other papers such as in Routley and Shulte [22], the rewards given are 0 and 1 for Goals. By giving -1 to receiving a goal, certain actions under which there are a lot of possible events which can finish in other teams scoring will have a lower utility value U, which could be seen as well as a measure of risk under certain situations.



FromId	ToId	Occurrence
1	2	78126
2	3	9081
3	4	4417
4	5	175
5	6	17
6	7	1
7	8	1
8	9	1
9	2	1
2	10	10595
10	11	1755
11	12	48
12	13	3
13	14	1
14	15	1
15	16	1
16	2	1
2	17	9807
17	18	1085
18	19	345
19	2	340
2	20	12117
20	21	2792
21	22	894
22	2	885

GameId	EventNumber	StartingId	EndingId
2013020001	1	2	3
2013020001	2	3	4
2013020001	3	4	5
2013020001	4	5	6
2013020001	5	6	7
2013020001	6	7	8
2013020001	7	8	9
2013020001	8	2	10
2013020001	9	10	11
2013020001	10	11	12
2013020001	11	12	13
2013020001	12	13	14
2013020001	13	14	15
2013020001	14	15	16
2013020001	15	2	17
2013020001	16	17	18
2013020001	17	18	19
2013020001	18	2	20
2013020001	19	20	21
2013020001	20	21	22
2013020001	21	2	23
2013020001	22	23	24
2013020001	23	24	25
2013020001	24	25	26
2013020001	25	2	10

NodeId	RewardGoal
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0
31	0
32	0
33	0
34	0
35	0
36	0
37	0
38	0

(a) The Edges table

(b) The Node information table

(c) The Reward table

Figure 4.6: Examples of the AD-tree tables

4.3 The Markov Decision Process

The reward's value per state

The reward given here differs from Routleys paper and it is defined as follows: for any state with a play sequence starting with home/away and finishing with that same team scoring goal home/away, they will get a 1. However, the reward for receiving a goal by the other team is -1. Otherwise, the reward is set to be 0.

Once the AD-tree has been created, a Markov Decision Process has been trained to compute the utility values related to closeness of the next home/away team scoring, being two MDPs with $M = 100{,}000$ iterations and a convergence c of 0.0001 (using Bellman's convergence). The pseudocode for the value iteration algorithm is provided in algorithm 1.

Once it converges, the two variables, accounting for the utility of each unique event in the dataset, are stored in a table with the *NodeId* reference.

4.4 Creation of metrics

Once each pair of *(GameId, EventNumber)* with an event a(T,Z) (as in figure 4.4) has been associated with its utility derived from the MDP execution, several metrics has been created for posterior evaluation.

In order to understand these metrics, a couple of measures are introduced, from which other metrics have been derived:

- The impact of a player's action stands for how much that specific player has contributed to score with the action performed compared to the previous state (e.g. If we are in *GameId, eventNumber* with *NodeId* = n, utility = 0.3 and an action performed by a player, then the previous state is related to the node's information table where relations between that ids are also stored according to (*GameId, EventNumber*). If the previous n-1 *NodeId* value is 0.4, then the impact or contribution of the player with his action is $0.3 - 0.4 = -0.1$).
$$impact(s, a) = U_T(s, a) - U_T(s)$$
where T is the team performing the current action, s refers to the *NodeId* state, and a to action.

- The time each player has played in each match has also been calculated from the data set.

In order to understand well the metrics, table 4.2 presents the basic action sets as done in Lambrix and Ljung 's article [16]:

A is the set of all state-action-pairs $< s, a >$ where action a is performed in state s during a match.
$A_i(p_k)$ is the set of state-action-pairs when player p_k is on the ice during a match. $A_i(p_{tp\|k})$ is the set of state-action-pairs that all team players tp related to p_k have performed while p_k is on the rink, including p_k state-action-pairs.
$A_p(p_k)$ is the set of state-action-pairs where the action is performed by player p_k during a match: $A_p(p_k) \subseteq A_i(p_k)$.

Table 4.2: Basic action sets, extracted from Lambrix and Ljung 's article [16]

Then, the metrics derived are displayed and described in table 4.3:

The direct impact (*Direct*) of a player per match is the sum of the impact values of the actions performed by the player: $Direct(p_k) = \sum_{<s,a> \in A_p(p_k)} impact(s, a)$
The direct impact divided by time (*Directh* or *Direct/h*) of a player per match is the sum of the impact values of the actions performed by the player divided by the time h the player has played in that match (expressed in hours): $Directh(p_k) = \frac{1}{h} \sum_{<s,a> \in A_p(p_k)} impact(s, a)$
The Collective impact (*Collective*) of a player per match is the sum of the impact values of the actions performed by their team while that specific player is on the rink: $Collective(p_k) = \sum_{<s,a> \in A_p(p_{tp\|k})} impact(s, a)$ The Collective impact divided by time (*Collectiveh* or *Collective/h*) of a player per match is the sum of the impact values of the actions performed by their team while that specific player is on the rink, including their own actions: $Collectiveh(p_k) = \frac{1}{h} \sum_{<s,a> \in A_p(p_{tp\|k})} impact(s, a)$

Table 4.3: The four derived metrics

All metrics are calculated for the regular season, which are 82 matches in the NHL, getting 1 vector of 82 values for each player in each season. The result per metric will be a matrix of 82xm where m is the number of players in that season. This is done for the four derived metrics and for all regular seasons.

4.5 General analysis of the metrics

The analysis, nonetheless, it is done only on 2 regular seasons: 2007-2008 and 2008-2009. Also, the overall values of the derived metrics for each player at the end of the league are also calculated and analyzed. Moreover, the overall values will be divided, according to each player, by the amount of games they have played in that specific season. In this case, the metric will have the same name with games played *GP* in front of the metric (e.g.*GPDirect*). These overall metrics will be compared to the general statistical metrics scrapped from *www.dropyourgloves.com* taking into consideration the position of the player in the field (whether it is a Central (C), Defender (D), Right Wing (RW), Left Wing (LW), Goalie (G) or Forward (F). Note that C, RW, LW and F are all Forward positions, and the analysis is performed replacing these positions for the Forward one, since they indeed are) to evaluate their behavior in the *Results* section. The statistical metrics scrapped are those described in the 3.2 section. For those cases where there are players that have played in 2 or 3 different teams during the same season, performance has been joined in a sole vector using the *GameId* variable as an order variable. In the case where a player plays for 2 teams in a specific number match (eg. match 32), the match with higher performance is the one taken into account. Moreover, scatterplots are displayed to evaluate whether it exists some relation between salary and player performance for the regular season and by match, taking into consideration the position of the player. Finally, the top 10 players for the four metrics are extracted.

4.6 Time series forecasting of metrics

In this section, it will be briefly explained how are the ARIMA (p,d,q) models chosen for prediction purposes and how the player's performance forecast has been done. Table 4.4 illustrates how the time series data is structured for a specific metric.

CountGame	8451774	8459462	8459574	8460620	8462035
1	4.94	3.38	2.995	0.41	1.38
2	3.23	3.51	1.64	0.37	1.53
3	3.06	2.02	2.69	0.50	2.80
4	2.02	2.03	1.10	0.28	1.42
5	1.53	3.88	2.22	0.00	2.79
6	3.54	4.39	2.11	0.00	0.63
7	2.16	2.24	1.64	0.00	2.37
8	2.89	3.81	3.18	-0.01	3.93
9	1.31	0.81	2.01	0.45	2.44
10	2.58	3.62	0.00	0.19	1.49

Table 4.4: Illustrative example of a metric structure for time series analysis. The *CountGame* variable relates to the number of matches a regular season have. The other column ids are associated to players performance for each match.

It can be then understood that for each player id in the column names, there are 82 player performances. This is done for each metric and for the time each player has played in each match, for seasons 2007-2008 and 2008-2009. That is, a total of 10 tables.

The choice of the best ARIMA models

For each player taken into consideration in each metric, an ARIMA (p,d,q) model has been fitted to evaluate which are the most typical ARIMA (p,d,q) models taking into account the number of games played by each player (later classified in ranges of 10 for the seasons 2007-2008 and 2008-2009 (e.g. the 10 range accounts for players that has played between 5-14 matches, and so on)) and the position the player plays in the rink, as described in section 4.5. That means that if a player has not played during a match, accounted in the time table , the associated value performance in that match (which would be 0 on the metrics) is eliminated and therefore shortened.

The selection of ARIMA (p,d,q) models performed in the previous subsection establishes the best general models to forecast over data. Therefore, one best ARIMA model is established for each player by comparing different models according to AIC, AICc or BIC value criterias, standing for a penalization to models that may overfit the data. In other words, a search is conducted over all possible ARIMA models using Maximum Likelihood Estimation (MLE). The MLE is a typical method to estimate parameters based on the data one has. In this case, it is used to know how many parameters are to be used, evaluating how the data is better fitted given different values to parameters and therefore deciding on their number.

The pseudocode to understand analytically how the analysis has been done is provided in algorithm 3:

Algorithm 3 The choice of the ARIMA (p,d,q) models

```
Require: list of metrics, seasonyears, playerposition, list of timeMatrix
count = 0
BestModel = matrix()
for $i = 1; i \leqslant length(seasonvect), i \leftarrow i + 1$ do
    for $j = 1; j \leqslant length(playerposition), j \leftarrow j + 1$ do
        for metric in list of metrics do
            for player in metric do
                count +=1
                eliminate matches where the player has not played using timeMatrix[player]
                $BestModel_{count} = [fitbestarima(p, d, q), metric, playerposition[j]]$
            end for
        end for
    end for
end for
It outputs bestModel with the best ARIMA(p,q,d) model for each player taking into account position and and metric
```

All ARIMA (p,d,q) models which occurred to be the best one over a threshold of 2% of the times have been selected for this part.

Estimate and forecast of player's performance using ARIMA(p,d,q) models

The ARIMA (p,d,q) models to estimate players' performance

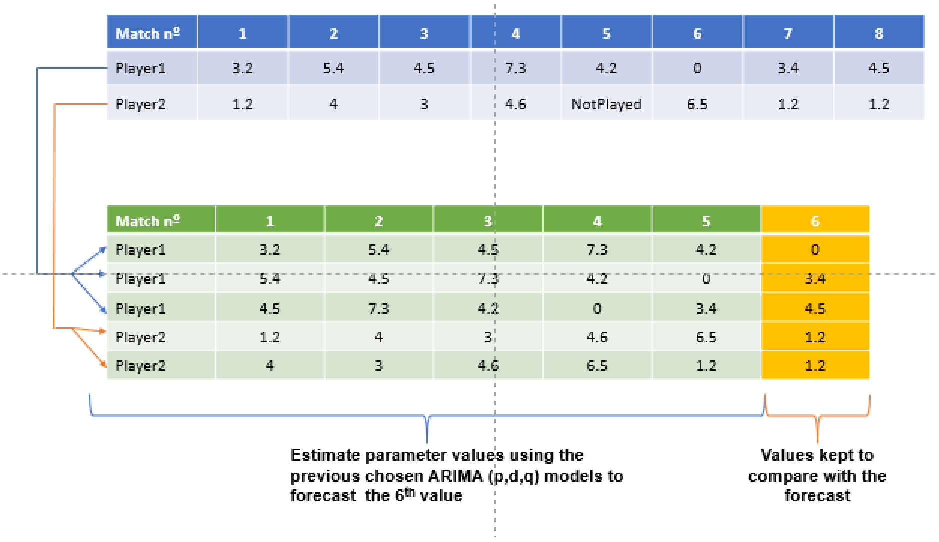
Additionally to the ARIMA(p,d,q) models selected in the previous section, one extra ARIMA model, called *best*, is added. It accounts for the ARIMA model that better fits the previous m matches at each moment.

The choice of length for fitting and forecasting ARIMA models

The choice of how many observations to take to estimate the parameter values has been settled to 5 and the prediction is set to be 1-step ahead. The motivation behind this choice is that we are interested in trying to predict player's performance through season based on their last previous matches, intuitively understanding that as their current playing status. Also, a short number of observations has also been chosen given the general fit of the ARIMA models over the threshold, which is max(p,d,q) equal to 2, displayed later in the results section.

The estimation of the parameters for the ARIMA (p,d,q) models and the forecast

All possible 6 consecutive combinations of player performances through the whole season have been taken for each metric and each player playing in the forward and defender positions (goalies have been excluded given the bad dependence with salary). For each combination, the ARIMA(p,d,q) models have been trained (making previously data stationary) for the first 5 observations and trained and predicted on the six one, as shown in figure 4.7.



Match nº	1	2	3	4	5	6	7	8
Player1	3.2	5.4	4.5	7.3	4.2	0	3.4	4.5
Player2	1.2	4	3	4.6	NotPlayed	6.5	1.2	1.2

Match nº	1	2	3	4	5	6
Player1	3.2	5.4	4.5	7.3	4.2	0
Player1	5.4	4.5	7.3	4.2	0	3.4
Player1	4.5	7.3	4.2	0	3.4	4.5
Player2	1.2	4	3	4.6	6.5	1.2
Player2	4	3	4.6	6.5	1.2	1.2

Estimate parameter values using the previous chosen ARIMA (p,d,q) models to forecast the 6th value
Values kept to compare with the forecast

Figure 4.7: Illustrative example of the selection of players' performance for the estimation and forecasting of their performances. For each player, the maximum number of combinations of players' performance is extracted for series length of 6 observations, training on the first 5 and forecasting 1 time ahead that is compared to the 6th value performance.

With this, the statistical measures ME, RMSE and MAE are displayed to evaluate the best model for each metric.

5 Results

In this section, it is firstly presented a general analysis of the four different player performance metrics for seasons 2007-2008 and 2008-2009 (section 5.1), showing as well the top 10 players for each metric. Additionally, players valuation are plotted to show the dependence with salary. This is done for each metric to put into context players' performance by position in the rink.

After that, ARIMA analysis is presented (section 5.2) for each metric taking into consideration position and number of games played per player, discretized the latter in ranges of 10. Forecasting has also been done for seasons 2007-2008 and 2008-2009 with the 5 most frequent ARIMA models and the best ARIMA model selected each time (called *best*, as explained in the method section) fitting the previous five games and then predicting the next game player's performance. It is presented then some accuracy measures for the predicted data such as ME, RMSE or MAE to identify the best models.

5.1 General evaluation of the metrics

In figure 5.1, the valuation (i.e. players' performance) of five players chosen at random through the regular season are exemplified. It seems that player's valuation are quite random across the league, and in some cases, it even looks like some noise across the mean.

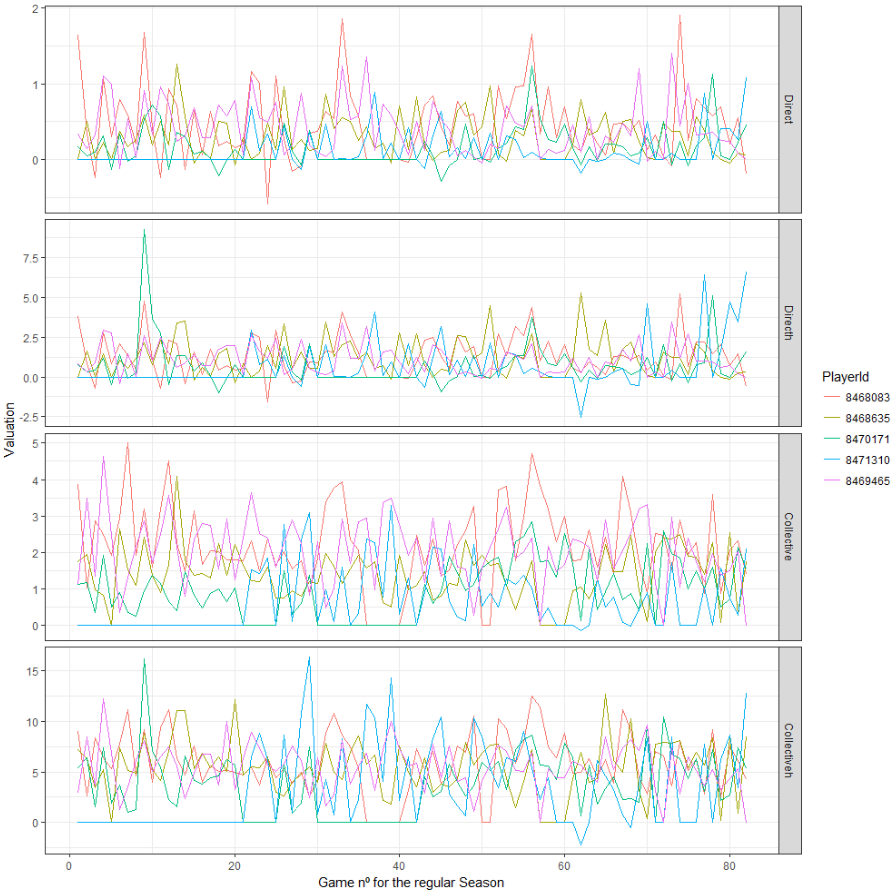
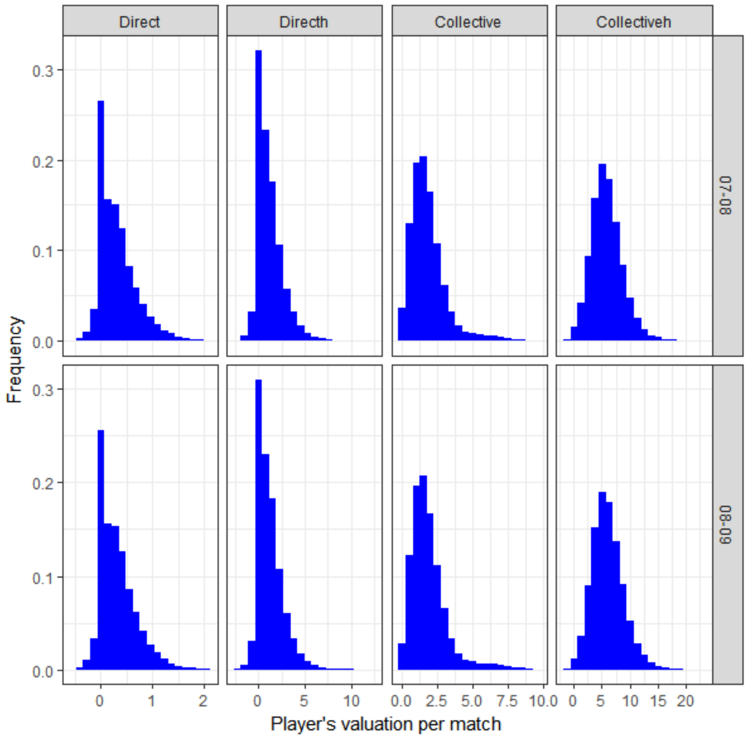
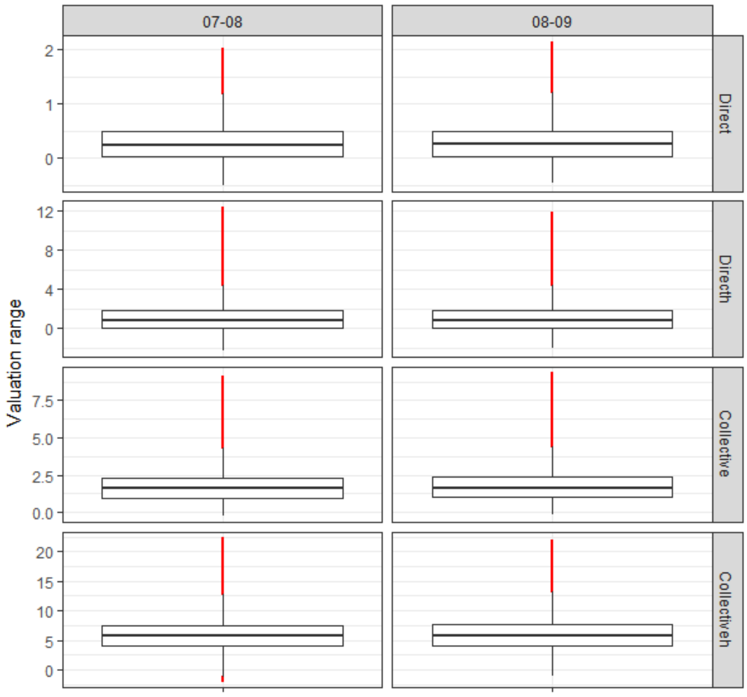


Figure 5.1: Example of five players valuation through the regular season.

In figure 5.2, the distribution of the 99.8% players' performance values in matches during regular seasons 2007-2008 and 2008-2009 is displayed (excluding 0.01% tails) for each metric.



(a) Histogram distribution of player's performance values in matches.

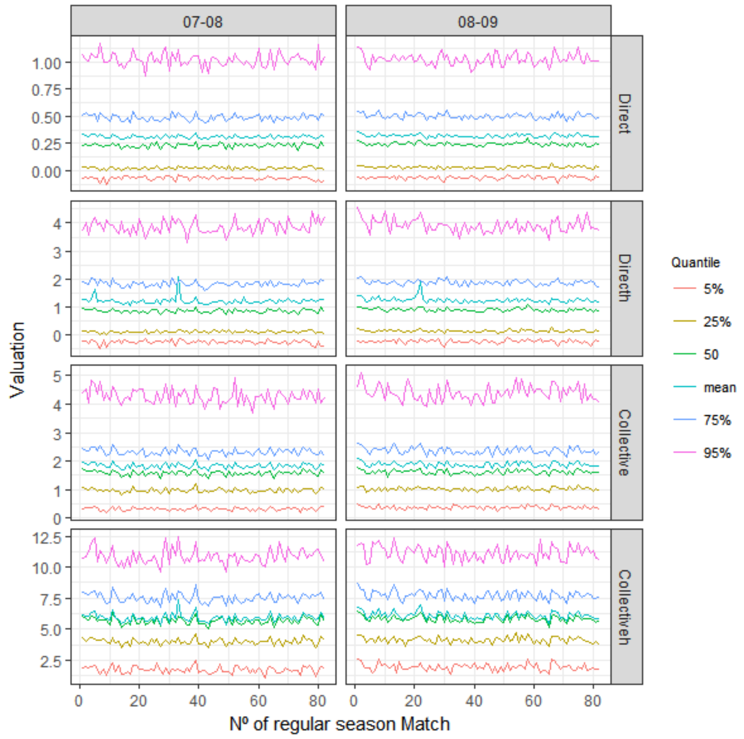


(b) Box plot distribution of player's valuation in matches. The box encloses values from $Q1 - 1.5(Q3 - Q1)$ to $Q3 + 1.5(Q3 - Q1)$, where Q1 and Q3 are the 25% and 75% quantiles respectively.

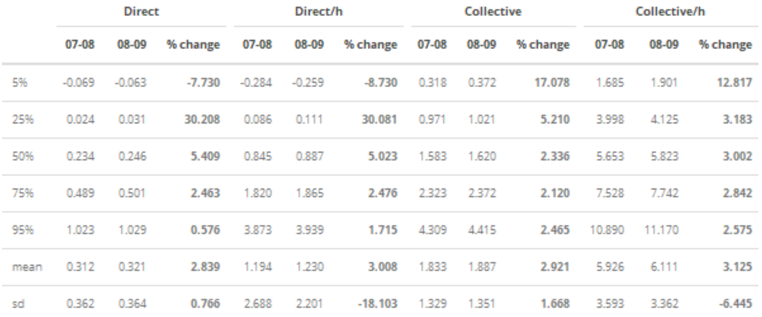
Figure 5.2: Distribution of the 99.8% player's performance values in matches during regular seasons 2007-2008 and 2008-2009 per metric (excluding 0.01% tails).

The players valuation distribution on matches for both the histogram and the boxplot are really similar across both years. It can also be seen in all metrics that the valuation distribution is skewed to the right.

In figure 5.3, the quantile performances are shown through the 82 matches for seasons 2007-2008 and 2008-2009, as well as a table showing the general quantiles year performance with its mean, standard deviation and change compared to the previous year.



(a) Quantile plot of players' valuation per match, season and metric.



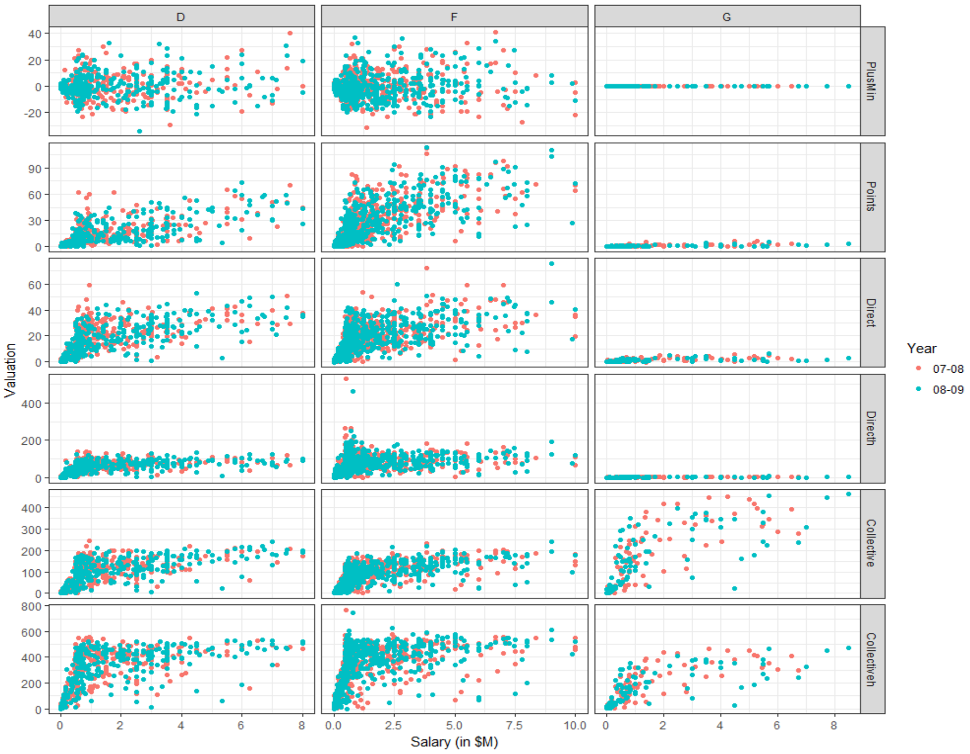
	Direct			Direct/h			Collective			Collective/h		
	07-08	08-09	% change	07-08	08-09	% change	07-08	08-09	% change	07-08	08-09	% change
5%	-0.069	-0.063	**-7.730**	-0.284	-0.259	**-8.730**	0.318	0.372	**17.078**	1.685	1.901	**12.817**
25%	0.024	0.031	**30.208**	0.086	0.111	**30.081**	0.971	1.021	**5.210**	3.998	4.125	**3.183**
50%	0.234	0.246	**5.409**	0.845	0.887	**5.023**	1.583	1.620	**2.336**	5.653	5.823	**3.002**
75%	0.489	0.501	**2.463**	1.820	1.865	**2.476**	2.323	2.372	**2.120**	7.528	7.742	**2.842**
95%	1.023	1.029	**0.576**	3.873	3.939	**1.715**	4.309	4.415	**2.465**	10.890	11.170	**2.575**
mean	0.312	0.321	**2.839**	1.194	1.230	**3.008**	1.833	1.887	**2.921**	5.926	6.111	**3.125**
sd	0.362	0.364	**0.766**	2.688	2.201	**-18.103**	1.329	1.351	**1.668**	3.593	3.362	**-6.445**

(b) Quantile table of players' valuation season and metric taking into account the standard deviation of the metrics.

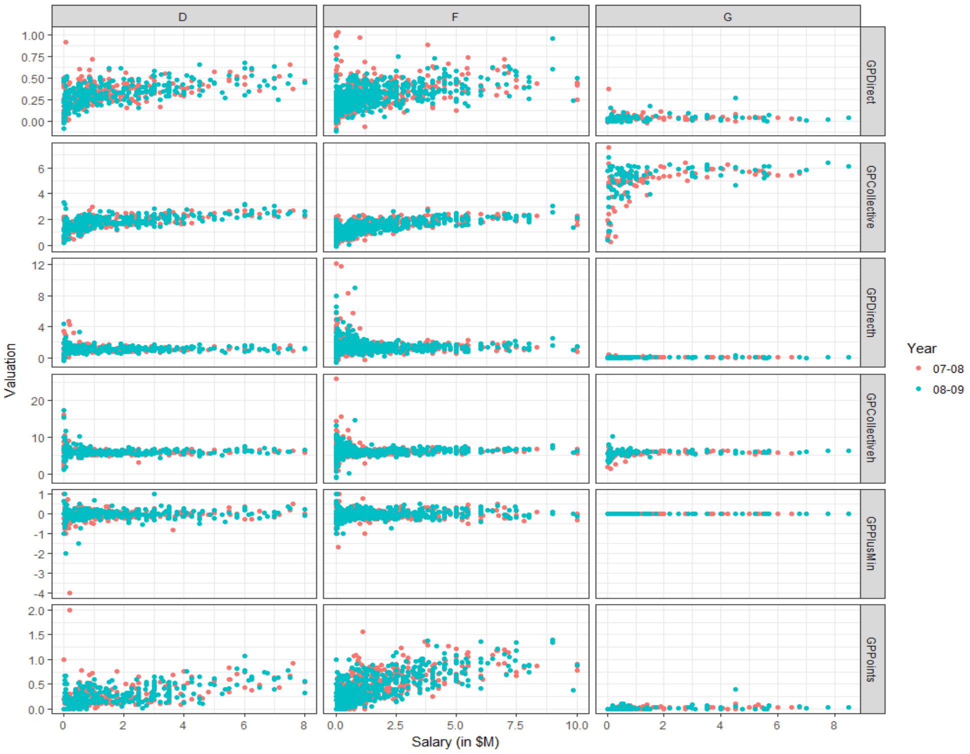
Figure 5.3: Quantile analysis of players' valuation in matches during regular seasons 2007-2008 and 2008-2009.

The quantile plot shows that players range of performance through season is quite stable. All metrics except the Collectiveh (Collective/time (hours)) have a clear separation between the mean and the median (the 50% quantile).

In figure 5.4, the dependence between players' valuation metrics on salary (Valuation∼ Salary) based on players' general position is displayed for metrics accounted at the end of the season and divided by the number of games played (the ones with GP) in 5.3(b).



(a) Dependence between players' valuation metrics on salary at the end of the season.



(b) Dependence between players' valuation metrics on salary at the end of the season divided by games played (GP) for each player.

Figure 5.4: Dependence between players' valuation metrics on salary (Valuation∼ Salary) based on players general position.

Below, the top 10 players for each metric and season are displayed, also showing the top 10 players on the Collective metric without the goalies.

PlayerName	Position	Age	Salary	GP	G	GA	PlusMin	Points	**Direct**	Directh	Collective	Collectiveh
2007												
Alex Ovechkin	F	22	3.83	82	65	47	28	112	**71.96**	182.65	232.56	588.85
Dion Phaneuf	D	22	0.94	82	17	43	12	60	**59.22**	134.05	246.12	559.67
Rick Nash	F	23	5.50	80	38	31	3	69	**59.01**	181.80	158.82	485.99
Jarome Iginla	F	30	7.00	82	50	48	27	98	**58.94**	161.92	204.12	560.88
Dustin Brown	F	23	1.18	78	33	27	-13	60	**53.78**	156.41	171.40	501.48
Brenden Morrow	F	28	4.10	82	32	42	23	74	**51.15**	146.62	171.59	504.57
Zdeno Chara	D	30	7.50	77	17	34	14	51	**50.74**	117.69	203.78	468.89
Trent Hunter	F	27	1.55	82	12	29	-17	41	**50.31**	167.65	153.36	508.27
Mike Green	D	22	0.85	82	18	38	6	56	**48.26**	122.63	219.72	545.08
Pavel Datsyuk	F	29	6.70	82	31	66	41	97	**48.22**	134.68	198.44	559.41
2008												
Alex Ovechkin	F	23	9.00	79	56	54	8	110	**75.93**	194.34	239.89	612.23
Dustin Brown	F	24	2.60	80	24	29	-15	53	**59.76**	177.60	178.34	540.84
Shea Weber	D	23	4.50	81	23	30	1	53	**53.14**	136.10	201.19	511.36
Evgeni Malkin	F	22	3.83	82	35	78	17	113	**50.76**	134.92	220.41	591.75
Dion Phaneuf	D	23	7.00	79	11	36	-11	47	**50.34**	122.64	240.57	532.49
Vincent Lecavalier	F	28	7.17	77	29	38	-9	67	**49.46**	143.99	188.17	549.37
Sheldon Souray	D	32	6.25	81	23	30	1	53	**49.38**	125.86	203.08	514.73
Jeff Carter	F	24	4.50	82	46	38	23	84	**48.88**	141.78	189.35	548.30
Rick Nash	F	24	6.50	78	40	39	11	79	**48.88**	145.11	171.59	498.26
Martin St. Louis	F	33	5.00	82	30	50	4	80	**47.82**	135.55	204.19	569.06

Table 5.1: Top 10 Players performance for 2007-2008 and 2008-2009 for the Direct metric.

PlayerName	Position	Age	Salary	GP	G	GA	PlusMin	Points	Direct	**Directh**	Collective	Collectiveh
2007												
Mike Rupp	F	27	0.5	64	3	6	-8	9	15.74	**529.76**	49.09	769.03
Riley Cote	F	25	0.48	70	1	3	2	4	14.55	**263.64**	38.90	570.09
Jeff Cowan	F	31	0.72	46	0	1	-5	1	10.62	**262.03**	39.03	316.68
DJ King	F	23	0.5	61	3	3	-4	6	12.06	**227.25**	36.18	485.79
Eric Godard	F	27	0.47	74	1	1	-8	2	10.80	**214.19**	35.18	505.55
Jared Boll	F	21	0.74	75	5	5	-4	10	27.41	**201.09**	66.26	472.20
Alex Ovechkin	F	22	3.83	82	65	47	28	112	71.96	**182.65**	232.56	588.85
Rick Nash	F	23	5.5	80	38	31	3	69	59.01	**181.80**	158.82	485.99
Raitis Ivanans	F	28	0.48	73	6	2	-10	8	20.88	**176.36**	57.73	445.11
David Clarkson	F	23	0.8	81	9	13	1	22	33.91	**168.22**	97.97	476.47
2008												
Derek Boogaard	F	26	0.8	51	0	3	3	3	10.83	**460.55**	33.48	747.36
Eric Godard	F	28	0.72	71	2	2	-3	4	14.15	**249.95**	35.40	539.28
Dan Carcillo	F	23	0.83	73	3	11	-15	14	33.14	**219.86**	94.20	501.42
Jody Shelley	F	32	0.72	70	2	2	-6	4	20.11	**196.58**	52.38	505.46
Alex Ovechkin	F	23	9	79	56	54	8	110	75.93	**194.34**	239.89	612.23
Evgeni Artyukhin	F	25	0.89	73	6	10	1	16	35.72	**189.73**	94.61	504.91
Mike Rupp	F	28	0.5	72	3	6	-2	9	24.52	**188.26**	73.94	517.70
Cam Janssen	F	24	0.55	56	1	3	-5	4	13.11	**187.93**	36.51	478.01
Matt Cooke	F	30	1.17	76	13	18	0	31	36.32	**185.57**	117.48	500.05
Riley Cote	F	26	0.52	63	0	3	-7	3	12.10	**184.08**	36.36	543.95

Table 5.2: Top 10 players performance for 2007-2008 and 2008-2009 for the Directh metric (Direct/time(hours))

PlayerName	Position	Age	Salary	GP	G	GA	PlusMin	Points	Direct	Directh	**Collective**	Collectiveh
2007												
Henrik Lundqvist	G	25	4.25	72	0	0	0	0	1.31	1.29	**449.62**	453.19
Miikka Kiprusoff	G	31	3.60	76	0	2	0	2	4.13	4.07	**447.26**	464.24
Evgeni Nabokov	G	32	5.00	77	0	2	0	2	3.00	2.98	**440.79**	453.59
Martin Brodeur	G	35	5.20	77	0	4	0	4	4.58	4.55	**419.29**	418.00
Ryan Miller	G	27	2.50	76	0	1	0	1	4.11	4.55	**418.40**	430.52
Cam Ward	G	23	2.00	69	0	1	0	1	1.67	1.70	**417.03**	440.68
Tomas Vokoun	G	31	5.30	69	0	6	0	6	2.87	2.87	**398.65**	413.01
Roberto Luongo	G	28	6.50	73	0	3	0	3	2.09	1.78	**393.72**	407.58
Vesa Toskala	G	30	1.38	66	0	5	0	5	1.73	1.74	**380.84**	391.68
Rick DiPietro	G	26	4.50	63	0	6	0	6	5.75	5.67	**372.26**	382.34
2008												
Miikka Kiprusoff	G	32	8.50	76	0	3	0	3	3.03	3.31	**462.40**	476.85
Marty Turco	G	33	5.70	74	0	5	0	5	6.24	6.48	**454.18**	467.39
Henrik Lundqvist	G	26	7.75	70	0	2	0	2	1.66	1.62	**447.99**	450.00
Cam Ward	G	24	2.50	67	0	1	0	1	2.24	3.09	**396.95**	410.44
Evgeni Nabokov	G	33	5.50	62	0	1	0	1	2.59	2.59	**380.95**	381.38
Marc-Andre Fleury	G	24	3.50	62	0	1	0	1	2.87	2.87	**375.41**	385.18
Niklas Backstrom	G	30	3.10	71	0	0	0	0	4.61	5.22	**373.02**	384.31
Dwayne Roloson	G	39	3.00	63	0	1	0	1	2.62	3.25	**353.05**	374.33
Steve Mason	G	20	0.85	61	0	0	0	0	1.63	1.65	**352.33**	357.26
Ilya Bryzgalov	G	28	4.00	64	0	2	0	2	0.69	0.70	**346.39**	364.53

Table 5.3: Top 10 players performance for 2007-2008 and 2008-2009 for the Collective metric

The Collective metric for a player stands for the collective impact of their mate players and himself while on the rink. On a general purpose, players tend to go and lose the puck after a shot or nearer the offensive zone while retaking it nearby their zone, usually trying to make movements to improve the chances on scoring. Then, it is normal that the collective metric is positive since actions performed by the other team are not taken into account on the opponents. Therefore, those who play during a longer time on the rink are those with highest valuation, which is accounting for the valuation of their team. That is why goalkeepers are in the 10 first positions of this metric, despite of the fact they do not score goals. Also, the more matches they play, the more time they play as well, making their collective performance bigger.

PlayerName	Position	Age	Salary	GP	G	GA	PlusMin	Points	Direct	Directh	**Collective**	Collectiveh
2007												
Dion Phaneuf	D	22	0.94	82	17	43	12	60	59.22	134.05	**246.12**	559.67
Alex Ovechkin	F	22	3.83	82	65	47	28	112	71.96	182.65	**232.56**	588.85
Tomas Kaberle	D	29	4.25	82	8	45	-8	53	38.32	93.36	**221.93**	551.72
Mike Green	D	22	0.85	82	18	38	6	56	48.26	122.63	**219.72**	545.08
Andrei Markov	D	29	5.75	82	16	42	1	58	42.37	105.18	**213.81**	530.37
Nicklas Lidstrom	D	37	7.60	76	10	60	40	70	29.04	66.41	**205.68**	480.18
Jarome Iginla	F	30	7.00	82	50	48	27	98	58.94	161.92	**204.12**	560.88
Zdeno Chara	D	30	7.50	77	17	34	14	51	50.74	117.69	**203.78**	468.89
Lubomir Visnovsky	D	31	2.05	82	8	33	-18	41	32.64	83.52	**201.34**	523.00
Roman Hamrlik	D	33	5.50	77	5	21	7	26	37.79	93.89	**201.29**	509.39
2008												
Dion Phaneuf	D	23	7.00	79	11	36	-11	47	50.34	122.64	**240.57**	532.49
Alex Ovechkin	F	23	9.00	79	56	54	8	110	75.93	194.34	**239.89**	612.23
Evgeni Malkin	F	22	3.83	82	35	78	17	113	50.76	134.92	**220.41**	591.75
Dan Boyle	D	32	6.67	77	16	41	6	57	36.11	88.65	**219.94**	539.81
Chris Pronger	D	34	6.25	82	11	37	0	48	43.40	99.89	**217.92**	503.72
Mike Green	D	23	6.00	68	31	42	24	73	46.41	106.62	**214.33**	493.09
Nicklas Backstrom	F	21	2.40	82	22	66	16	88	37.12	111.83	**214.19**	630.43
Braydon Coburn	D	23	1.20	80	7	21	7	28	40.78	100.10	**211.64**	516.12
Andrei Markov	D	30	5.75	78	12	52	-2	64	38.03	96.17	**209.18**	527.62
Mark Streit	D	31	4.10	74	16	40	6	56	39.38	97.60	**206.59**	504.31

Table 5.4: Top 10 players performance for 2007-2008 and 2008-2009 for the Collective metric without goalkeeper positions

PlayerName	Position	Age	Salary	GP	G	GA	PlusMin	Points	Direct	Directh	Collective	**Collectiveh**
2007												
Mike Rupp	F	27	0.50	64	3	6	-8	9	15.74	529.76	49.09	**769.03**
Alex Ovechkin	F	22	3.83	82	65	47	28	112	71.96	182.65	232.56	**588.85**
Nicklas Backstrom	F	20	2.40	82	14	55	13	69	28.80	86.12	187.96	**577.10**
Riley Cote	F	25	0.48	70	1	3	2	4	14.55	263.64	38.90	**570.09**
Jarome Iginla	F	30	7.00	82	50	48	27	98	58.94	161.92	204.12	**560.88**
Viktor Kozlov	F	32	2.50	81	16	38	28	54	32.25	108.51	168.31	**560.82**
Dion Phaneuf	D	22	0.94	82	17	43	12	60	59.22	134.05	246.12	**559.67**
Pavel Datsyuk	F	29	6.70	82	31	66	41	97	48.22	134.68	198.44	**559.41**
Eric Staal	F	23	4.50	82	38	44	-2	82	34.85	96.67	199.04	**552.43**
Tomas Kaberle	D	29	4.25	82	8	45	-8	53	38.32	93.36	221.93	**551.72**
2008												
Derek Boogaard	F	26	0.80	51	0	3	3	3	10.83	460.55	33.48	**747.36**
Nicklas Backstrom	F	21	2.40	82	22	66	16	88	37.12	111.83	214.19	**630.43**
Alex Ovechkin	F	23	9.00	79	56	54	8	110	75.93	194.34	239.89	**612.23**
Arron Asham	F	30	0.64	78	8	12	0	20	18.24	143.65	83.85	**605.67**
Evgeni Malkin	F	22	3.83	82	35	78	17	113	50.76	134.92	220.41	**591.75**
Eric Staal	F	24	5.00	81	40	35	15	75	39.45	110.57	203.62	**581.78**
Brandon Dubinsky	F	22	0.64	82	13	28	-6	41	38.83	137.89	165.14	**580.76**
Nikolai Zherdev	F	24	3.25	82	23	35	6	58	37.57	133.72	165.89	**579.38**
Chris Thorburn	F	25	0.54	81	7	8	-10	15	23.15	143.05	93.92	**570.34**
Martin St. Louis	F	33	5.00	82	30	50	4	80	47.82	135.55	204.19	**569.06**

Table 5.5: Top 10 players performance for 2007-2008 and 2008-2009 for the Collectiveh metric (Collective/time(hours))

The Top 10 players on the Direct metric seems to perform relatively well in terms of Points (Goals and Assists). This makes sense since both metrics offer quite a similar dependence line with Salary. As previously stated, the top 10 Players in the Collective metric are Goalkeepers, since they are the ones gathering all the impact valuation of their teams 5.3. Therefore, those metrics should be taken into account under position circumstances, and not only under total general valuation.

Metrics accounting for time (Directh and Collectiveh) outputs players with low goals and assists on a general basis who might have played little in a lot of matches but who have had a great impact while they were in the rink.

5.2 Time Series Analysis

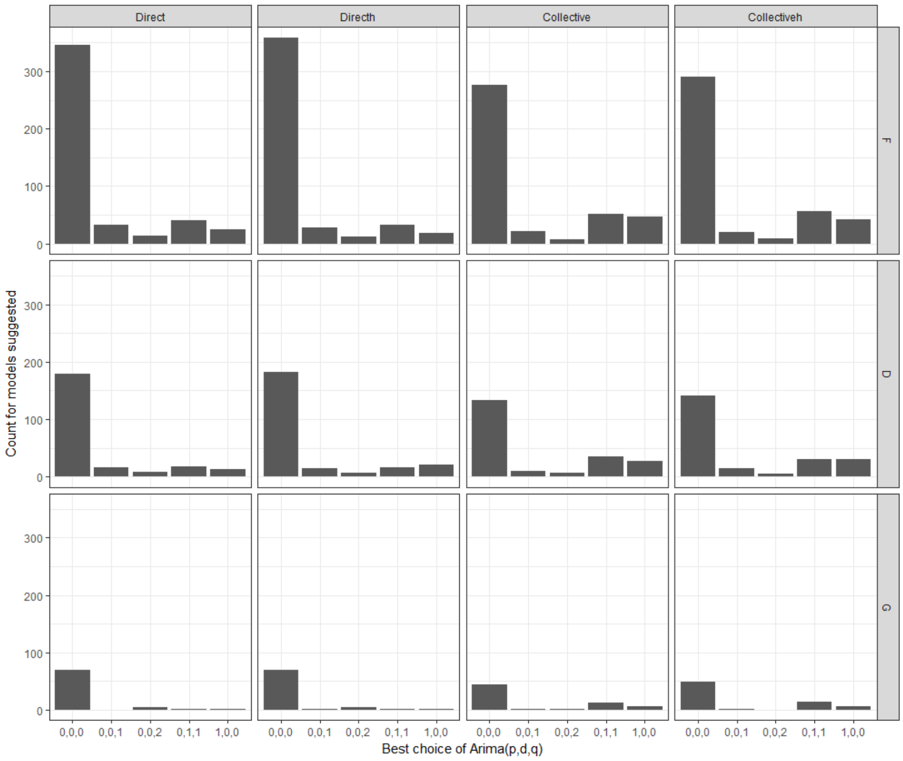
The choice of an Arima Model

ArimaModel	Position	Dataset	Frequency	Prob
0,0,0	F	Direct	346	0.7554585
0,0,0	F	Directh	359	0.7960089
0,0,0	F	Collective	277	0.6873449
0,0,0	F	Collectiveh	290	0.6937799
0,0,0	D	Direct	180	0.7627119
0,0,0	D	Directh	182	0.7583333
0,0,0	D	Collective	134	0.6261682
0,0,0	D	Collectiveh	141	0.6351351
0,0,0	G	Direct	69	0.8961039
0,0,0	G	Directh	69	0.8846154
0,0,0	G	Collective	45	0.6617647
0,0,0	G	Collectiveh	49	0.6901408

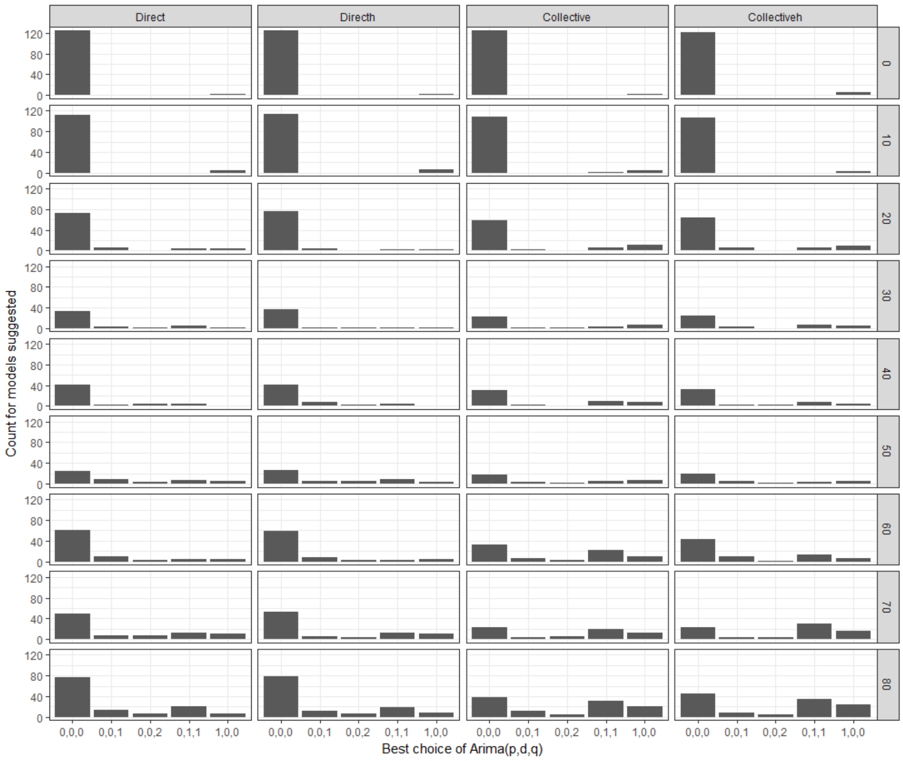
Table 5.6: Top arima models per metric (*Dataset*) and position for seasons 2007-2008 and 2008-2009 together. Only models with the highest frequency per position and metric are shown in the table).

In table 5.6, the most and only typical model is displayed for each metric and position, with the number of times chosen (*Frequency*) and the percentage out of all cases by position (*Position*) and metric (*Dataset*), called *Prob*. It can be seen that in all cases the most preferred model is the ARIMA(0,0,0), which stands for the mean of the previous observations.

In plots 5.5, it is shown the distribution of the 98% most frequent ARIMA models by Position(a), Range(b) and Metric. Both plots show quite an outstanding dominance of the ARIMA(0,0,0) model. The ranges in the second histogram have been discretized rounding them up. For instance, range 0 will englobe those players who have played between 0 and 4 matches. Then, range 10 will englobe those players who have played between 5 and 14 matches, and so on.



(a) Histogram distribution of the 98% most frequent arima models by position and metric.



(b) Histogram distribution of the 98% most frequent arima models by range and metric.

Figure 5.5: Histogram distribution of the 98% most frequent arima models by range, position and metric.

Forecast of Player's performance using ARIMA models

In this section, a summary table is displayed to understand the forecast prediction of each metric.

Model	ME	RMSE	MAE	Dataframe
0,0,2	0.011	0.438	0.325	Direct
0,0,1	0.015	0.414	0.310	Direct
1,0,0	0.011	0.408	0.303	Direct
0,1,1	0.034	0.439	0.313	Direct
0,0,0	0.007	0.366	0.273	Direct
best	-0.204	0.438	0.295	Direct
0,0,2	0.072	3.371	1.354	Directh
0,0,1	0.069	3.314	1.300	Directh
1,0,0	0.059	3.002	1.247	Directh
0,1,1	0.180	3.772	1.311	Directh
0,0,0	0.040	2.828	1.127	Directh
best	-0.817	2.774	1.144	Directh
0,0,2	0.031	1.089	0.835	Collective
0,0,1	0.027	1.030	0.795	Collective
1,0,0	0.033	1.009	0.775	Collective
0,1,1	0.063	1.026	0.767	Collective
0,0,0	0.027	0.908	0.708	Collective
best	-0.241	1.073	0.786	Collective
0,0,2	0.167	5.089	3.298	Collectiveh
0,0,1	0.148	4.906	3.144	Collectiveh
1,0,0	0.173	4.635	3.040	Collectiveh
0,1,1	0.328	5.147	3.031	Collectiveh
0,0,0	0.131	4.243	2.765	Collectiveh
best	-1.054	4.685	3.004	Collectiveh

Table 5.7: Analysis of the residuals for the predicted player performance on the most frequent ARIMA models

In table 5.7, the Mean Error, the Root Mean Squared Error and the Mean Absolute Error are the measures used to evaluate the forecast of the player's performance for one step-ahead. It can be seen that the best model is again the ARIMA (0,0,0) for all cases since it is the one being smaller for most metrics and measures.

6 Discussion

In this section, the general analysis and the time-series forecast are discussed for each metric. Then, methods used in this thesis are also discussed.

6.1 Results

General evaluation of the metrics

Player's valuation varies a lot from match to match. Beforehand, it seems very challenging to find a model that would fit well player's performance. As already mentioned in the results section, players valuation distribution on matches for the histogram, boxplot and the quantile plots in 5.2 and 5.3 is really similar across both years and stable (i.e. the range of values are very similar). That means that the metrics are therefore usable for other years prediction, meaning that there are no peaks or big changes on valuation for some specific weeks or years. It has also been commented the fact that the valuation distribution is skewed to the right, possibly meaning that players often perform between some ranges, and that there are few players who consistently may have some outstanding performances that are larger than the average of the players performance.

In figure 5.3, all metrics except the Collectiveh (Collective/time (hours)) have a clear separation between the mean and the median (the 50% quantile). This also makes sense since the distribution of matches seen in 5.2 is skewed to the right in all cases but in the Collectiveh metric where the distribution, although it is still a little bit skewed, looks much more normal than the other ones.

In figure 5.4, the PlusMinus measure does not seem to be a good metric to predict players Salary at all, since there is no dependence between salary and the metric values. This makes sense since the PlusMinus measure accounts for the number of goals that occur while a player is on the field. Even though a specific player scores a lot goals, if that specific team receives a lot of goals while he is on the rink, the value of their performance for this metric will be low or even negative, therefore not making it good to predict good players. The Direct and the Points measures seem to be fairly good to predict players Salary for Foward and Defender players. Also, the Collective measure performs better for the prediction in general players performance. This might be misleading since the Collective measure takes into account the sum of all impact actions of players in the rink while that specific player is in the rink. Of course, if a player has been playing good through the season, that player will continue playing even more time since he is proven useful to the team. Also, this measure can be misleading and must be interpreted only between players playing in similar position and similar number of matches. Since goalkeepers are the players who play most of the time, they are getting all impact actions of their teams and that is why their performances are the highest ones. Both Directh and Collectiveh seem to have some kind of logarithmic dependence, which could be interesting to predict players salary based on performances.

The second figure in 5.4, which is related to the same metrics but divided by Games Played in each case, shows that GPPoints, GPDirect and GPCollective are be able to capture the linear dependence between players' performance and Salary for Forward and Defender players. Nevertheless, metrics divided by time do not seem to have any impact on players performance. That makes sense, since the Games Played (GP) measure takes into account to some extent the time a player plays (e.g.the more matches a player plays, the more is playing).

Time series forecasting of metrics

The best time series ARIMA model for forecasting players performance has proven to be an ARIMA (0,0,0) which is indeed the mean of all previous matches. In other words: based on my original dataset and on my impact measures, I cannot predict better than the mean. This means that players performance may not depend on the n previous matches as sometimes believed when people say that a player is *on fire* for some consecutive matches but it just happens that player has normal ups and downs, non-related to n previous matches performances, but just on their average level performance through the whole season. This goes in line with the hot hands fallacy [19], which tries to proof that a person who experiences a successful outcome in a random event (e.g. a match) has not a greater chance of success in successive trials.

6.2 Method

The definition of a state

In this thesis, I have assumed to have as state nodes the combination of 3 context variables (Manpower Differential, Goal Differential, Period) and actions associated with the Team (T) having the puck or winning the action and a categorical Zone (Z) in which the action is happening based on the team's performing the action (i.e a(T,Z)). It would be interesting to include new state variables such as a time remaining in the quarter, meaning that maybe when almost no time is left, the actions taken may have a lower impact on the scoring due to time pressure or duration of the possession until the moment of the action, meaning that the probability of scoring given that you have had the puck might differ when the length of the possession is significantly different.

The choice of the reward function

The choice of the reward value for actions in the markov decision process has not been discussed in this article nor on the ones I have read. Nevertheless, it would be interesting to do a study of the impact of different sets of rewards for more than one action at a time, and seeing how that affects the impact of a specific actions and their change.

Time series forecasting

In terms of time series modelling and prediction, only ARIMA models have been evaluated. Other models such as Neural Networks or Gaussian Processes could be tried to evaluate whether there could be a general model that could fit players evaluation better than the mean.

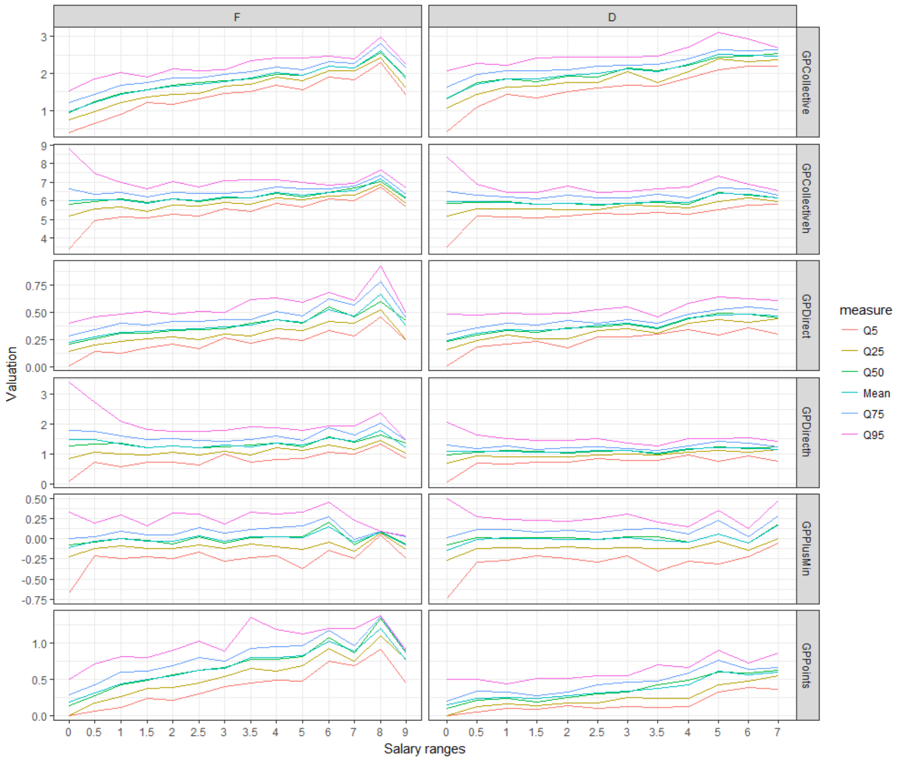


Figure 6.1: Quantile distribution of the regression on players performance by game to Salary Range (Valuation Salary). The range of Salary for a specific number is meant to be from that specific number to the next, (eg. 0 is from 0-0.5M). This is done for each Combination of Positions and Metric for Season 2007-2008 and 2008-2009, excluding Goalkeepers and Forward

Suggestion for further research

In the light of the results got and the shape of players performance, I sincerely do not think that any model would fit significantly well the data. Instead, I would try to get a richer data set in terms of performing actions. For instance, having passes or pucks in the action variable would enable to have a better precision of the action's impact and therefore of the player performance as in [24], [25] or [15]. Also, the new dataset could have non-discretized or less discretized the Zone variable to quantify better the impact actions as in the latter papers above.

As a matter of fact, I have not given any clear rule to decide whether to hire/fire/maintain a player. Some new research could involve quantile analysis to decide whether a player would be fired/hired/maintained. In figure 6.1, I have discretized the salary range of players such that a specific number goes from that number to the next one on the x-axis (e.g. if 0, from 0-0.5M). All players performance have been allocated in Salary ranges and then, for each discrete salary range, quantile performances are shown by position and range for several metrics. With that, a combination of metrics could be settled such that if a player appears to be over the 75% quantile for their specific range, then that player could be a potential hiring, and viceversa.

Ethical issues

The data used in this thesis is public and therefore there are no private potential issues. Nevertheless, the top 10 players shown for the different derived metrics could affect the sensitivity of the players not appearing on the list. Also, the automatization of the hiring/firing process in sports and other jobs in the future could be a potential issue to discuss, as well as the implications this might arise.

7 Conclusion

In this thesis, I have used AD-trees to summarize Ice Hockey matches using state variables, which englobes context and action variables, and then using Markov Decision Processes to estimate the impact of each action under that specific state. With that, an impact measure has been described and four player metrics have been derived by match for regular Seasons 2007-2008 and 2008-2009. From that, general data analysis have been performed to understand the metrics, model player's performance using ARIMA and forecasting the following match through season.

1. **How can a MDP/RL be used to evaluate actions under certain time-series events ?** By using AD-trees to summarize actions events under context, one can possibly reward the actions one is interested in happening (e.g. as in my thesis GOAL) to evaluate how other actions have an impact in reaching that action.

2. **How can I store time series data for the usage of a MDP?** An AD-tree is a good solution to summarize actions under context. It is important to understand well data and also the context under which specific variables are happening to provide a meaningful insight. In this case, context variables such as manpower differential (MD), goal differential (GD) and Period (P) has been chosen as context variables and the player actions taking into account the Team performing the action and the Zone in which the action is occurring.

3. **Can I predict player's performance based on their performances in the previous matches ?** One can try to do it using ARIMA models. Nevertheless, the result got is that no better result is achieved rather than the mean of the previous *m* matches.

4. **Is there a way to use MDPs to create a metric that evaluates players for hiring/maintaining/firing purposes?**

   Several metrics has been created to achieve that purpose (Direct, Direct/h, Collective and Collective/h). Neverthess, none of them seems good enough by itself to decide using one specific metric whether a player should be hired/fired/maintained. Therefore, not an specific answer has been given to that particular question. Nevertheless, in the further research section 6.2, a combination of relevant metrics per position of a player could be done using quantile analysis and salary range of a player to decide whether to hire/fire/maintain a specific player.

Bibliography

[1] Brigham S. Anderson and Andrew W. Moore. "ADtrees for Fast Counting and for Fast Learning of Association Rules". In: (1998), pp. 134–138. URL: http://www.aaai.org/Library/KDD/1998/kdd98-020.php.

[2] James F. Barkell, Alun Pope, Donna O'Connor, and Wayne G. Cotton. "Predictive game patterns in world rugby sevens series games using Markov chain analysis". In: International Journal of Performance Analysis in Sport 17.4 (2017), pp. 630–641. ISSN: 14748185. DOI: 10.1080/24748668.2017.1381459.

[3] Richard Bellman. "A Markovian Decision Process". In: Indiana Univ. Math. J. 6 (4 1957), pp. 679–684. ISSN: 0022-2518.

[4] Dan Cervone, Alexander D'Amour, Luke Bornn, and Kirk Goldsberry. "POINTWISE: Predicting Points and Valuing Decisions in Real Time with NBA Optical Tracking Data". In: MIT Sloan Sports Analytics Conference (2014), pp. 1–9. URL: http://dl.frz.ir/FREE/papers-we-love/sports%7B%5C_%7Danalytics/2014-ssac-pointwise-predicting-points-and-valuing-decisions-in-real-time.pdf.

[5] Jonathan D Cryer and Kung-Sik Chan. Time Series Analysis. With Applications to R. January. 2008, p. 487. ISBN: 9780387759586. DOI: 10.1007/978-0-387-75959-3.

[6] Will G. Hopkins, John Hawley, and Louise Burke. "Design and analysis of research on sport performance enhancement". In: Medicine and science in sports and exercise 31 (Apr. 1999), pp. 472–85. DOI: 10.1097/00005768-199905000-00022.

[7] Robert B. Gramacy, Shane T. Jensen, and Matt Taddy. "Estimating player contribution in hockey with regularized logistic regression". In: Journal of Quantitative Analysis in Sports 9.1 (2013), pp. 97–111. ISSN: 15590410. DOI: 10.1515/jqas-2012-0001.

[8] R. A. Howard. Dynamic Programming and Markov Processes. Cambridge, MA: MIT Press, 1960.

[9] Mike D. Hughes and Roger M. Bartlett. "The use of performance indicators in performance analysis". In: Journal of Sports Sciences 20.10 (2002). PMID: 12363292, pp. 739–754. DOI: 10.1080/026404102320675602.

[10] Rob Hyndman and Yeasmin Khandakar. "Automatic Time Series Forecasting: The forecast Package for R". In: Journal of Statistical Software, Articles 27.3 (2008), pp. 1–22. ISSN: 1548-7660. DOI: 10.18637/jss.v027.i03. URL: https://www.jstatsoft.org/v027/i03.

[11] Edward H. Kaplan, Kevin Mongeon, and John T. Ryan. "A Markov Model for Hockey: Manpower Differential and Win Probability Added". In: INFOR: Information Systems and Operational Research 52.2 (2014), pp. 39–50. DOI: 10.3138/infor.52.2.39. URL: https://doi.org/10.3138/infor.52.2.39.

[12] Mykel J. Kochenderfer, Christopher Amato, Girish Chowdhary, Jonathan P. How, Hayley J. Davison Reynolds, Jason R. Thornton, Pedro A. Torres-Carrasquillo, N. Kemal Üre, and John Vian. Decision Making Under Uncertainty: Theory and Application. 1st. The MIT Press, 2015. ISBN: 0262029251, 9780262029254.

[13] Hector Levesque, Fiora Pirri, and Ray Reiter. "Foundations for the Situation Calculus". In: Linköping Electronic Articles in Computer and Information Science 3.18 (1998), p. 18. URL: http://www.ep.liu.se/ea/cis/1998/018/.

[14] Michael L. Littman. "Markov games as a framework for multi-agent reinforcement learning". In: Machine Learning Proceedings 1994 (1994), pp. 157–163. ISSN: 00493848. DOI: 10.1016/B978-1-55860-335-6.50027-1.

[15] Guiliang Liu and Oliver Schulte. "Deep Reinforcement Learning in Ice Hockey for Context-Aware Player Evaluation". In: MIT Sloan Sports Analytics Conference (2015), pp. 3442–3448.

[16] Dennis Ljung, Niklas Carlsson, and Patrick Lambrix. "Player pairs valuation in ice hockey". In: MIT Sloan Sports Analytics Conference (2018), pp. 14–24. URL: http://ceur-ws.org/Vol-2284/#paper-02.

[17] Brian Macdonald. "Adjusted Plus-Minus for NHL Players using Ridge Regression with Goals, Shots, Fenwick, and Corsi". In: Journal of Quantitative Analysis in Sports 8.3 (2012), pp. 1–24. ISSN: 15590410. DOI: 10.1515/1559-0410.1447. arXiv: 1201.0317.

[18] Brian Macdonald. "An Expected Goals Model for Evaluating NHL Teams and Players". In: MIT Sloan Sports Analytics Conference (2012), pp. 1–8. ISSN: 10769757.

[19] Joshua Miller and Adam Sanjurjo. Surprised by the Gambler's and Hot Hand Fallacies? A Truth in the Law o
Working Papers 552. IGIER (Innocenzo Gasparini Institute for Economic Research), Bocconi University, 2015. URL: https://EconPapers.repec.org/RePEc:igi:igierp:552.

[20] Stephen Pettigrew. "Assessing the offensive productivity of NHL players using in-game win probabilities 2 A win probability metric for NHL games". In: MIT Sloan Sports Analytics Conference (2015), pp. 1–9.

[21] David Poole and Alan Mackworth. Artificial Intelligence: Foundations of Computational Agents. 2nd ed. Cambridge, UK: Cambridge University Press, 2017. ISBN: 978-0-521-51900-7. URL: http://artint.info/2e/html/ArtInt2e.html.

[22] Kurt Routley and Oliver Schulte. "A Markov Game Model for Valuing Player Actions in Ice Hockey". In: Uncertainty in Artificial Intelligence (UAI). 2015, pp. 782–791.

[23] Michael Schuckers and James Curro. "Total Hockey Rating (THoR): A comprehensive statistical rating of National Hockey League forwards and defensemen based upon all on-ice events". In: MIT Sloan Sports Analytics Conference 7 (2013), pp. 1–10. URL: http://www.sloansportsconference.com/wp-content/uploads/2013/Total%20Hockey%20Rating%20(THoR)%20A%20comprehensive%20statistical%20rating%20of%20National%20Hockey%20League%

20forwards%20and%20defensemen%20based%20upon%20all%20on-ice%20events.pdf.

[24] Oliver Schulte, Mahmoud Khademi, Sajjad Gholami, Zeyu Zhao, Mehrsan Javan, and Philippe Desaulniers. "A Markov Game model for valuing actions , locations , and team performance in ice hockey". In: Data Mining and Knowledge Discovery 31.6 (2017), pp. 1735–1757. ISSN: 1573-756X. DOI: 10.1007/s10618-017-0496-z.

[25] Oliver Schulte, Zeyu Zhao, Computing Science, Mehrsan Javan, and Philippe Desaulniers. "Apples-to-Apples : Clustering and Ranking NHL Players Using Location Information and Scoring Impact". In: MIT Sloan Sports Analytics Conference (2017), pp. 1–14.

[26] Richard S. Sutton and Andrew G. Barto. Introduction to Reinforcement Learning. 1st. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262193981.