StarCraft Winner Prediction

Norouzzadeh Ravari, Yaser; Bakkes, Sander; Spronck, Pieter

StarCraft Winner Prediction

Yaser Norouzzadeh Ravari and **Sander Bakkes** and **Pieter Spronck**

Tilburg center for Cognition and
Communication
Tilburg University
Tilburg, the Netherlands
{y.NorouzzadehRavari, s.c.j.bakkes, p.spronck}@uvt.nl

Abstract

In game-playing, a challenging topic is to investigate an evaluation function that accurately predicts which player will be the winner of a two-player match. Our work investigates to what extent it is possible to predict the winner of a StarCraft match, regardless of the races that are involved. We developed models for individual match types, and also general models for predicting the winner of non-symmetric matches, symmetric matches, and general matches. The contribution of this paper is (1) a generic and relatively accurate model for winner prediction in StarCraft, and (2) a detailed analysis of which features are the principal component in accurately predicting the winner in this complex game. Specially, our results show that we can predict the winner of a match with an accuracy of more than 63% in average over all time slices, regardless of the time slice and the combination of the match types. A study of which features are most important for the prediction of the match results, shows that the economic aspects of StarCraft matches are the strongest predictors for winning, followed by the use micro commands.

Introduction

Among AI researchers, Real-Time Strategy (RTS) games have been a popular research domain in the past decade. In particular, the complex, partially observable, and dynamic environments of RTS games motivate AI researchers to study different approaches and techniques to create strong AI, analyzing the games, and modeling players. In Particular, winner prediction is a highly relevant topic of AI research. In StarCraft, winner prediction is challenging because players have many action choices, in a discrete environment where players manage their units concurrently. Moreover, the strategy of players depends on the match type. This increases the complexity of winner prediction.

StarCraft has been a popular RTS game since 1998. In StarCraft, players gather resources to strengthen their economy. To provide military power, they must spend resources to construct buildings, research new technologies, and training units. The goal of the game is to destroy all of the opponent's bases and armies. StarCraft has various maps that differ in dimension, arrangement of resources, and the areas that are build-able and walk-able.

Figure 1: Terran vs. Zerg

StarCraft includes three different playable races: Terran, Zerg, and Protoss. The player chooses one of the races to play at the start of a match. While the races are well-balanced, they each need a different playing style. Very generally speaking, Terran are defensively strong and employ slow units, Zerg are offensively strong and employ fast units, and Protoss form a middle ground. Figure 1 shows a battle between Terran and Zerg.

Since each race needs a different playing style, using a playing style that is not fitting the chosen race may very well lead to defeat. It is possible to recognize a playing style already early in a game, meaning that sometimes it is possible to accurately predict the ultimate winner of a match already in the first few minutes of the match.

By analyzing replays from StarCraft competitions, we can use our model as an internal evaluation function for bots to improve their game play. Winner prediction in matches where a player faces an AI bot tends to be relatively easy: AI bots are fairly weak players and tend to lose against players who have some experience with the game. Consequently, considerable research has been invested in this area to make AI bots stronger players. In human vs. human matches, winner prediction is not as straightforward.

To our knowledge, the only study into winner prediction for StarCraft in human vs. human matches was limited to both players using Protoss (Erickson and Buro 2014). In our study, we investigate winner prediction both for symmetric

matches (Protoss vs. Protoss, Zerg vs. Zerg, Terran vs. Terran) but also for non-symmetric matches (Protoss vs. Terran, Protoss vs. Zerg, Terran vs. Zerg). We compare the relative importance of match and player skill and style features for the purpose of winner prediction.

We pose the following questions:

- To what extent is it possible to predict the winner for non-symmetric matches?

- To what extent is it possible to design a general model for winner prediction in all matches?

- What is the comparative importance of individual features for the winner prediction?

- Is there a difference in relative importance of features for non-symmetric and symmetric matches?

In the following sections, we present related work; then, an overview of our method, the dataset that we used, and the features that we extracted are described in the method section. We continue with the experimental setup and results. Afterwards, we discuss about the results. Finally, we mention the conclusions that we draw.

Related work

In our research, we build a model of StarCraft players. This is a challenging task, as RTS games have a very large state space (Robertson and Watson 2015) and are only partially observable (Ontanón et al. 2013).

Player modeling encompasses a player's in-game behavior (Robertson and Watson 2015; Ortega et al. 2013; Yannakakis et al. 2013; Holmgård et al. 2014) including actions, skills, and strategies. Player modeling in RTS games has been studied from different perspectives. Gagné et al (Gagné, El-Nasr, and Shaw 2011) used telemetry and visualization to understand how players learn and play a basic RTS game. They reported that their approach does not suffice to understand players.

Since RTS games are partially observable, not all behaviors of an opponent can be known at all times. To model the opponent, different techniques have been used. Schadd et al (Schadd, Bakkes, and Spronck 2007) classified an opponent's playing style and strategy in the RTS game SPRING. They found it difficult to determine opponent strategy in the early game. Dereszynski et al (Dereszynski et al. 2011) successfully used a statistical model for predicting opponent behavior and strategy in StarCraft.

Multiple researchers have investigated detection of player skills in RTS games. Avontuur et al (Avontuur, Spronck, and Van Zaanen 2013) built a model to determine a player's StarCraft league based on observations of player features during the early game stages. Thompson et al (Thompson et al. 2013) examined the differences between player skills across the leagues. They reported that experts have automated many behaviors, i.e., the higher a player's skill, the less control they need to to spend on basic game tasks, and thus have room to develop other skills.

Park et al (Park et al. 2012) and Hsieh and Sun (Hsieh and Sun 2008) predict opponent strategy by analyzing build orders. In (Synnaeve and Bessiere 2011) Synnaeve and Pierre

presented a Bayesian model to predict the first strategy of the opponent in RTS games. Hsieh and Sun used case-based reasoning for this purpose. They managed to model different strategies that could then be recognized. They did this for all three races. On a limited winner-prediction scale, Stanescu et al (Stanescu et al. 2013) showed that the winner of a small battle in StarCraft can be predicted with high accuracy. Bakkes et al (Bakkes, Spronck, and van den Herik 2007) predicted the outcome of the RTS game SPRING using the phase of the game. Hsu et al in (Hsu, Hung, and Tsay 2013) utilized an evolutionary method to predict the winning rate between EISBot and human player for ZvZ, ZvT, and ZvP match types. They formulated the winner prediction as an optimization task. Their approach achieved 61% accuracy on average for ZvZ and less than 2% for ZvT and ZvP.

Predicting match up outcome is more challenging than combat outcome. During the match up, players lose their units or buildings during combats that affect the math up outcome. Meanwhile, the number of units and their locations changes and thus, the player has to adjust his strategy. If a suitable prediction model can be built, an interesting application would be the possibility of game personalization. Moreover, it can be used as an evaluation function to design AI bots that behave like human players.

Closest to what we intend to do with our research, is the work by Erickson et al (Erickson and Buro 2014), who used state evaluation to predict the winner of a StarCraft match in human vs. human play. They limited themselves to symmetric matches between Protoss players in games of a particular length. In contrast, in our work we investigate all races, in all possible match ups, with less limitation on game length.

Method

Overview of the method

StarCraft is a zero-sum game, but in some matches there is no winner in our replays. Therefore, we filtered the matches that do not have a winner, and we represent the winner prediction as a binary classification problem: win(1), and lose(0).

We follow two approaches: individual models for each match type, and mixed models. The individual models include six binary classifiers for PvT, PvZ, TvZ, PvP, ZvZ, and TvT matches. We used P, T, and Z for Protoss, Terran, and Zerg races respectively. The mixed models include the following tree binary classifiers: a model for non-symmetric matches (PvT, PvZ, and TvZ), a model for symmetric matches (PvP, ZvZ, and TvT), and a general model for all matches.

Data

We used the dataset that was provided by (Robertson and Watson 2014). This dataset has been created based on human vs. human replays from professional players that were collected by Synnaeve et al (Synnaeve and Bessiere 2012). The database includes replay data and state information provided by the Brood War API (BWAPI).

Table 1: Number of replays in the used database (Robertson and Watson 2014).

Race	PvT	PvZ	TvZ	PvP	ZvZ	TvT
Number of replays	2017	840	812	392	199	395
Number of replays(After filtering)	1490	579	612	263	115	298

Table 1 shows the number of replays for each match type. We filtered the replays to exclude replays with a length of less than 10 to have reasonable data for feature extraction. Also, we removed replays with a length more than 50 minutes, in order to limit the diversity of the replays' length.

We computed the fractions of victories in non-symmetric matches in our dataset. The results show Protoss won a fraction of 0.55 of the matches vs. Terran and 0.51 vs. Zerg. The winning rate of Terran vs. Zerg was 0.56. This implies that the winner/loser classes are balanced in our dataset with respect to the percentage of winning in different match types.

In the dataset, each match is played on a unique map. In StarCraft, the size of a map is measured as the number of tiles. In our dataset, over 60% of maps have a size of 128×128 tiles. All other maps are smaller, with the smallest measuring 96×128 tiles.

Features

In this section, we explain how features are extracted from the dataset. The features are time-dependent or time-independent. The time-dependent features are extracted for each player in 10-second intervals. We extracted *unspent resources* and *income* as follows (Erickson and Buro 2014): R_t is the total of resources (minerals and vespene gas) at time t (increments in intervals of 1 seconds), and T is the passed time in seconds (T always being a multiple of 180 seconds). The *unspent resources* U (i.e., how many resources are available on average at any given time) are calculated as:

$$U = (\sum_{t=1,2,...,T} R_t)/T$$

The *income* I is computed as the total resources R_{tot} collected over time T, averaged per second:

$$I = R_{tot}/T$$

For each feature, over the last 3 minutes we calculated the mean, the variance, and the difference between the two players. For instance, let b_t denote the number of build commands during t, t being a multiple of 10 seconds. Then, B_T is an array of b_t during last 180 seconds: $B_T = [b_{t_1}, b_{t_2}, ..., b_{t_{18}}]$. We computed $mean(B_T)$ and $var(B_T)$. In addition, if b_{A_t}, and b_{B_t} are number of build commands for player A and B during during 10-second interval t, the difference between players A and B in the number of build commands for the past 180 seconds is calculated as:

$$d_T = \sum_{t=T-180}^{T} (b_{A_t} - b_{B_t})$$

Table 2: Proposed features

Time-dependent	Time-independent
move	number of regions
build	buildable ratio tiles
tech	walkable ratio tiles
hold	average of choke distances
siege	height levels ratio
burrow	map dimension
micro	
macro	
control	
strategy	
tactic	
unique regions	
region value	
commands diversity	

The list of proposed features are summarized in table 2. The data set also included a race indicator. After the filteration, we collected 24k, 9k, and 9k samples for PvT, PvZ, and TvZ respectively. For symmetric matches, we have 3K, 1K, and 4K samples for PvP, ZvZ, and TvT respectively.

Time-dependent features

Expert players use time more efficiently when they play StarCraft (Thompson et al. 2013). To capture skills of players in this regard, we used the following features.

First, we counted the frequency of commands for each match type, and we found that the most frequent commands include: *move, build, tech, hold, siege, and burrow*. The order of command frequencies differs across the match types.

We categorized the commands into *micro* and *macro* commands. A command is considered *micro* if it does not cost minerals or gas; otherwise, it is considered *macro*. Then, we computed the number of *micro* and *macro* commands during each 10 seconds for each player.

Inspired by (Ontanón et al. 2013), we put the commands in one of three categories: *control, strategy*, and *tactic* commands. We computed the number of commands in each category for 10 seconds intervals per player.

Regions are extracted by the method that authors in (Perkins 2010) proposed. A region include adjacent walkable tiles that do not include choke points. We counted the number of *unique regions* that have a building for a player during each 10 seconds. The game assigns buildings different values. For a player we also stored the sum of the building values minus the sum of the opponent's building values as *region value*.

Time-independent features

To study the effect of maps on the winner prediction, we recorded some features that reflect the static characteristics of the map. The size of the map is indicated by the total number of regions.

Maps contain different areas, including; *buildable areas, walkable areas* and *average of choke distances*. The height

Table 3: Winner prediction performance across non-symmetric matches in terms of accuracy. A=APM and economy features, B=time-dependent features, C=time-independent features

Model	Features	PvT	PvZ	TvZ	PvP	ZvZ	TvT
baseline		0.55	0.51	0.56	0.50	0.50	0.50
RF	A,B,C	0.591	0.611	0.502	0.502	0.515	0.491
GBRT	A,B,C	0.595	0.623	0.502	0.502	0.507	0.483
RF	A,B	0.644	0.634	0.624	0.643	0.582	0.639
GBRT	A,B	0.637	0.634	0.624	0.639	0.581	0.635

of an area is one of six different *height* levels. For each map, we counted the number of *buildable* tiles, and we computed the ratio of the total number of buildable areas to the total number of tiles.

We did the same for the other types of areas. Since maps have different dimensions, we included the dimension of the map in terms of length and width as number of tiles.

Experimental setup

In this section, we explain our winner prediction models across the StarCraft races that are mentioned in section .

We formulated the winner prediction as a binary classification task to predict if a player wins (1) or loses (0). As the first step, we designed an individual model for each match type. Then, we mixed the models to combine the winner predictions for different match types.

The individual models are six binary classifiers for winner prediction for PvT, PvZ, TvZ, PvP, ZvZ, and PvP matches. The three mixed models are: a model only for non-symmetric matches, a model only for symmetric matches, and a model for all matches (general model).

We employed two state-of-the-art classification methods: Gradient Boosting Regression Trees (GBRT) (Friedman 2002) and Random Forest (RF) (Breiman 2001). These are implemented in the Scikit-learn Python package.

GBRT uses ensemble of trees to learn the target variable. It is robust to different features, does not need to normalize the inputs, and it can handle non-linear dependencies between the feature values and the output. Moreover, it computes feature importance that is a value in [0, 1]. The higher values shows the most important feature. RF has shown high performance in many classification tasks. It is an ensemble of decision tree classifiers, but it can handle the overfitting issue in decision trees.

We did 10-fold cross validation on the samples. To avoid bias, for any match the samples are either in the training set or in the test set, but not in both.

Results

In this section, we present the results of our approaches for winner prediction in StarCraft. The first approach uses individual models for each match type, and the second approach uses mixed models.

Table 4: Winner prediction performance across mixed match types in terms of accuracy. A=APM and economy features, B=time-dependent features, C=time-independent features

Model	Features	NonSym	Sym	General
RF	A,B,C	0.575	0.497	0.591
GBRT	A,B,C	0.577	0.499	0.593
RF	A,B	0.639	0.637	0.639
GBRT	A,B	0.635	0.634	0.635

Prediction per match type

The winner prediction results across the match types are summarized in table 3. The table also includes the baseline victory fractions. The baseline represents the majority winning rate in all match types according to our dataset. The performance of the models is presented in terms of accuracy. The features are grouped into three categories: Category A contains actions per minute (APM), income, and unspent resources, category B contains time-depended features, and category C contains time-independent features. We compared the performances in two cases: modeling using all mentioned features (A,B,C), and modeling excluding time-independent features (A,B). The reason to exclude the time-independent features from the second modeling approach is that player strength, and therefore chances at victory, tend not to be influenced by static map features, which are the core of category C.

We attempted to improve the results for both approaches by employing random forest for feature selection, but we did not observe a significant improvement in the predictions. Therefore these results are left out of the paper.

From the table it can be observed that with the (A,B,C) modeling approach, a small improvement to winner prediction over the baseline can be achieved for PvT and PvZ matches (for the PvT matches, a *very* small improvement). No improvement is achieved for the other matches.

However, for the (A,B) modeling approach, a considerable improvement of winner prediction over the baseline is achieved for all match types.

From these results, we see that time-independent features seem to have a negative effect on most predictions. Thus, we may assume that the inclusion of map properties in the feature set leads to detrimental results of the classification. Since our data set contains mainly replays of expert players, it seems that they are capable of incorporating map properties in their playing style, regardless of match type.

Prediction for mixed match types

As we mentioned earlier, the winner prediction is possible across the match types by individual models. In the next step, we are interested to see how accurately we can predict the match results when we mix the races. Therefore, we employed three mixed models: one for non-symmetric match types, one for symmetric match type, and one for all match type (general model).

The prediction performance of the mixed models are

Table 5: Top time-depended features per match type

PvT	PvZ	TvZ	PvP	ZvZ	TvT
Income (0.203)	Income (0.189)	Income (0.198)	Income (0.219)	Micro (0.233)	Income (0.206)
Unspent (0.141)	Unspent (0.157)	Unspent (0.140)	Unspent (0.201)	Unspent (0.229)	Unspent(0.192)
Micro (0.094)	Micro (0.129)	Micro (0.140)	Micro (0.174)	Income (0.217)	Micro (0.161)
Control (0.091)	Control (0.096)	Control (0.095)	Control (0.140)	Control (0.092)	Control (0.134)
Region value (0.076)	Region value (0.080)	Region value (0.067)	Region value (0.033)	Region value (0.031)	Region value(0.032)
Unique regions (0.052)	Unique regions (0.035)	Unique regions(0.044)	Unique regions (0.030)	Slice (0.022)	Unique regions(0.027)
Builds (0.020)	Slice (0.027)	Race (0.027)	Slice (0.017)	Unique commands (0.012)	Slice (0.025)
Slice (0.020)	Race (0.024)	Slice (0.027)	Builds (0.013)	Tech (0.008)	player distance (0.019)
APM (0.017)	Unique commands (0.017)	Burrow (0.018)	Unique commands (0.009)	Unique regions (0.008)	Unique commands (0.015)
Unique commands (0.016)	Burrow (0.016)	Unique commands(0.012)	APM (0.009)	Tactics (0.007)	APM (0.011)

shown in table 4. The first two rows represent the performance of the models that use all features, while the last two rows show the performance for the models without time-independent features.

The table shows a similar result as found for the models for the individual match types: when all features are included, the models do not perform well, while when time-dependent features are removed from the data, all models perform reasonably well with an accuracy of more than 63%, even for the generalized model that predicts the results for all match types.

Top features per match type

Table 5 presents the relative importance of top 10 time-depended features for individual models, of which the results are given in table 3 as the models for feature sets (A,B). The importance rates are given between parentheses.

Our feature set includes three variations of features (mean, variance, and difference between players). For the top feature list we ignored variations of the features. For instance, if mean and variance of income are amongst the top features, we only included 'income' on the list once; however, we summed the importance rates for the different variations of a feature, and ranked them by these sums.

Generally, most features have some predictive value for each of the match types, and when examining the rankings, we see that they tend to be ordered similarly across the match types, with some notable exceptions. Income and unspent resources are always amongst the top three features for all match types. This shows that having a strong economy is a an important element to win the a match for any of the match types.

The biggest exceptions are found for the ZvZ matches. In ZvZ, *micro* commands have a stronger predictive value compared to the other match types. According to the table 5, while the importance rate of *micro* commands (0.233) in ZvZ is close to the importance rates of *income* (0.229) and *unspent* (0.229), in the other match types *micro* commands are placed in the third rank of the top features, and have a considerably lower importance rate. This shows that ZvZ matches have to be approached by the players in a different way than they approach the other match types.

Control and *region value* are strong predictive features across all match types. *Control* commands are issued on a unit, and include move, gather, build, and repair; i.e.,

they are a combination of micro and macro commands. Thy reflect the general process of enriching the economy and spending resources on buildings. *Region value* is the difference between the values of the players' buildings during the specified time interval. I.e., it reflects how the resources are spent to construct buildings.

Top features for mixed match types

The top 10 features, with their importance rates, for each of the mixed models that do not include time-independent features, are given in table 6. The importance rates are presented in parenthesis.

Income is the most predictive feature for all of the mixed models. For the non-symmetric and symmetric match types, again *income* and *unspent* are the most predictive features. For the mixed models, *unspent* is moved to the third place in the ranking, while *region value* is in second place – however, the importance of *unspent* is still very close to the importance of *region value*. This means that for all match types, economic features play a decisive role in determining the match outcome.

From the table we can see that the top six features are the same for each of the combined match types, though they sometimes appear in a slightly different order. We also see that of these six features, for symmetric matches, there is a considerable gap between the importance of the top-4 features, and the features on the fifth and sixth place. For the other two combined match types, that gap is found between the sixth and seventh ranked features. From this we conclude that *income*, *unspent*, *micro*, and *control* are the most important features overall, while in non-symmetric matches *region values* and *unique regions* also play a role in determining the match outcomes.

Discussion

From the results we found, we conclude that including time-independent features in the data set actually has a detrimental effect on the classification algorithms, creating classifiers that perform worse than those created using a data set without these time-independent features. We offer the following explanation for this observation:

Each match is divided into multiple time-slices (180 seconds); each slice from a match has the same winner, and also exactly the same time-independent features and thus, there are correlations among several samples in training set.

Table 6: Top time-depended features for mixed models

non-Sym	Sym	General
Income (0.181)	Income(0.184)	Income (0.177)
Unspent (0.118)	Unspent(0.150)	Region value (0.112)
Region value (0.107)	Micro(0.138)	Unspent (0.104)
Control (0.074)	Control(0.118)	Control (0.079)
Micro (0.074)	Region value (0.044)	Micro (0.071)
Unique regions (0.062)	Unique regions (0.043)	Unique regions (0.066)
Race (0.028)	APM (0.019)	Slice (0.023)
Slice (0.025)	Slice (0.019)	Race (0.023)
APM (0.017)	Unique commands (0.016)	APM (0.020)
Unique commands (0.017)	Builds (0.012)	Unique commands (0.018)

Therefore, a classification algorithm may uncover a strong relationship between these time-independent features and the ultimate winner. However, since the time-slices of each match are stored only in one specific fold for the evaluation, in the fold that is used as test the relationships found in the folds used for training are non-existent. Therefore, the inclusion of time-independent features creates classifiers that work well on a training set but not as well on a test set.

We surmise that there still might be an interesting relationship between time-independent features and the ultimate winner of a match, but such a relationship cannot be found using our approach with match slices. A separate classification run using a data set that only stores features of complete matches may uncover such relationships.

As for the individual features, we see that the general class of *micro* features ranks fairly high in victory prediction, but that the two most important features (*income* and *unspent*) for winner prediction are both *macro* features. Therefore we conclude that while micro commands are important for winning StarCraft matches, the strategic and tactical aspects of StarCraft, which are exemplified by macro actions, have more importance overall.

In this work, we studied winner prediction in non-symmetric match types by individual models and mixed models. Our results show that both approaches manage to predict the winner with a considerably higher accuracy than the baseline models. The general model for all match types achieved a performance above 63%. The comparative importance of features shows economic features are the strongest predictors across match types. The list of the top-10 features in symmetric models and non-symmetric models are more or less the same, but the rank and the importance rate of the features differs.

Conclusion

In this work, we studied the winner prediction of a matches across StarCraft races using individual and mixed models for match types. Our work is the first work in comparing the performance of winner prediction across the races, and analyzing the relative importance of the features in this task.

The individual models for match types show that winner prediction is possible for all of the match types, with an accuracy of 63% or higher for all match types except ZvZ, as

long as only time-dependent features are included in the data set.

Moreover, we designed more general models that contain non-symmetric match types, symmetric match types, and all match types. The results show that these mixed models manage to predict the match winner, also with an accuracy of 63% or higher.

For all classifiers, the top-10 features used for prediction are more or less the same, with economic features having the highest predictive value in all cases, followed by micro commands.

Our results improve considerably on previous work done in this area, where only symmetric matches were used, and where accuracies achieved were much lower than we managed to find. Further improvements might still be possible, if more detailed features of matches are incorporated in the data set.

References

Avontuur, T.; Spronck, P.; and Van Zaanen, M. 2013. Player skill modeling in starcraft ii. In *AIIDE*.

Bakkes, S.; Spronck, P.; and van den Herik, J. 2007. Phase-dependent evaluation in rts games. In *Proceedings of the 19th Belgian-Dutch Conference on Artificial Intelligence*, 3–10.

Breiman, L. 2001. Random forests. *Machine learning* 45(1):5–32.

Dereszynski, E. W.; Hostetler, J.; Fern, A.; Dietterich, T. G.; Hoang, T.-T.; and Udarbe, M. 2011. Learning probabilistic behavior models in real-time strategy games. In *AIIDE*.

Erickson, G. K. S., and Buro, M. 2014. Global state evaluation in starcraft. In *AIIDE*.

Friedman, J. H. 2002. Stochastic gradient boosting. *Comput. Stat. Data Anal.* 38(4):367–378.

Gagné, A. R.; El-Nasr, M. S.; and Shaw, C. D. 2011. A deeper look at the use of telemetry for analysis of player behavior in rts games. In *Entertainment Computing–ICEC 2011*. Springer. 247–257.

Holmgård, C.; Liapis, A.; Togelius, J.; and Yannakakis, G. N. 2014. Personas versus clones for player decision mod-

eling. In *Entertainment Computing–ICEC 2014*. Springer. 159–166.

Hsieh, J.-L., and Sun, C.-T. 2008. Building a player strategy model by analyzing replays of real-time strategy games. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, 3106–3111. IEEE.

Hsu, C.-J.; Hung, S.-S.; and Tsay, J.-J. 2013. An efficient framework for winning prediction in real-time strategy game competitions. *Innovation, Communication and Engineering* 239.

Ontanón, S.; Synnaeve, G.; Uriarte, A.; Richoux, F.; Churchill, D.; and Preuss, M. 2013. A survey of real-time strategy game ai research and competition in starcraft. *Computational Intelligence and AI in Games, IEEE Transactions on* 5(4):293–311.

Ortega, J.; Shaker, N.; Togelius, J.; and Yannakakis, G. N. 2013. Imitating human playing styles in super mario bros. *Entertainment Computing* 4(2):93–104.

Park, H.; Cho, H.-C.; Lee, K.; and Kim, K.-J. 2012. Prediction of early stage opponents strategy for starcraft ai using scouting and machine learning. In *Proceedings of the Workshop at SIGGRAPH Asia*, 7–12. ACM.

Perkins, L. 2010. Terrain analysis in real-time strategy games: An integrated approach to choke point detection and region decomposition. *AIIDE* 10:168–173.

Robertson, G., and Watson, I. D. 2014. An improved dataset and extraction process for starcraft ai. In *FLAIRS Conference*.

Robertson, G., and Watson, I. 2015. Building behavior trees from observations in real-time strategy games. In *Innovations in Intelligent SysTems and Applications (INISTA), 2015 International Symposium on*, 1–7. IEEE.

Schadd, F.; Bakkes, S.; and Spronck, P. 2007. Opponent modeling in real-time strategy games. In *GAMEON*, 61–70.

Stanescu, M.; Hernandez, S. P.; Erickson, G.; Greiner, R.; and Buro, M. 2013. Predicting army combat outcomes in starcraft. In *AIIDE*. Citeseer.

Synnaeve, G., and Bessiere, P. 2011. A bayesian model for opening prediction in rts games with application to starcraft. In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, 281–288. IEEE.

Synnaeve, G., and Bessiere, P. 2012. A dataset for starcraft ai\ & an example of armies clustering. *arXiv preprint arXiv:1211.4552*.

Thompson, J. J.; Blair, M. R.; Chen, L.; and Henrey, A. J. 2013. Video game telemetry as a critical tool in the study of complex skill learning. *PloS one* 8(9):e75129.

Yannakakis, G. N.; Spronck, P.; Loiacono, D.; and André, E. 2013. Player modeling. *Dagstuhl Follow-Ups* 6.