A meta analysis of tournaments and an evaluation of performance in the Iterated Prisoner's Dilemma.

Nikoleta E. Glynatsi, Vincent A. Knight

Abstract

The Iterated Prisoner's Dilemma has been used for decades as a model of behavioural interactions. From the celebrated performance of Tit for Tat, to the introduction of the zero-determinant strategies, to the use of sophisticated structures such as neural networks, the literature has been exploring the performance of strategies in the game for years. The results of the literature, however, have been relying on the performance of specific strategies in a finite number of tournaments. This manuscript evaluates 195 strategies' effectiveness in more than 40000 tournaments. The top ranked strategies are presented, and moreover, the impact of features on their success are analysed using machine learning techniques. The analysis determines that the cooperation ratio of a strategy in a given tournament compared to the mean and median cooperator is the most important feature. The conclusions are distinct for different types of tournaments. For instance a strategy with a theory of mind would aim to be the mean/median cooperator in standard tournaments, whereas in tournaments with probabilistic ending it would aim to cooperate 10% of the times the median cooperator did.

1 Background

The Iterated Prisoner's Dilemma (IPD) is a repeated two player game that models behavioural interactions, and more specifically, interactions where self-interest clashes with collective interest. At each turn of the game both players, simultaneously and independently, decide between cooperation (C) and defection (D) whilst having memory of their prior interactions. The payoffs for each player, at each turn, is influenced by their own choice and the choice of the other player. The payoffs of the game are generally defined by:

$$\begin{pmatrix} R & S \\ T & P \end{pmatrix}$$

where $T > R > P > S$ and $2R > T + S$. The most common values used in the literature [17] are $R = 3, P = 1, T = 5, S = 0$. These values are also used in this work.

Conceptualising strategies and understanding the best way of playing the game has been of interest to the scientific community since the formulation of the game in 1950 [29]. Following the computer tournaments of Axelrod in the 1980's [15, 16], a strategy's performance in a round robin computer tournament became a common evaluation technique for newly designed strategies. Today more than 200 strategies exist in the literature and several tournaments, following on from Axelrod's, have been undertaken [20, 35, 39, 59, 60].

In the 80's, Axelrod performed two computer tournaments [15, 16]. The contestants were strategies submitted in the form of computer code. They competed against all other entries, a copy of themselves and a random strategy. The winner was decided on the average score a strategy achieved. The winner of both tournaments was the simple strategy Tit For Tat which cooperated on the first turn and then simply copied the previous action of it's opponent. Due to the strategy's strong performance in both tournaments, and moreover in a series of evolutionary experiments [17], Tit For Tat was thought to be the most robust basic strategy in the IPD.

However, further research proved that the strategy had weaknesses, and more specifically, it was shown that the strategy suffered in environments with noise [20, 28, 49, 58]. This was mainly due to the strategy's lack of generosity and contrition. The strategy was quick to punish a defection, and in a noisy environment it could lead to a repeated cycle of defections and cooperations. Some new strategies, more robust in tournaments with noise, were soon introduced and became the new protagonists of the game. These include Nice and Forgiving [20], Pavlov [51] and Generous Tit For Tat [52].

In 2004, a 20$^{\text{th}}$ Anniversary Iterated Prisoner Dilemma Tournament took place with 233 entries. This time the winning strategy was not designed on a reciprocity based approach but on a mechanism of teams [26, 27, 56]. A team from Southampton University took advantage of the fact that a participant was allowed to submit multiple strategies. They submitted a total of 60 strategies that could recognise each other and colluded to increase one members score. This resulted with three of the strategies to be ranked in the top spots. The performance of the Southampton University team received mixed attention, though they had won the tournament as stated in [25] "technically this strategy violates the spirit of the Prisoner's Dilemma, which assumes that the two prisoners cannot communicate with one another".

Another set of IPD strategies that have received a lot of attention are the zero-determinant strategies (ZDs) [54]. By forcing a linear relationship between the payoffs ZDs can ensure that they will never receive less than their opponents. The American Mathematical Society's news section stated that "the world of game theory is currently on fire". ZDs are indeed a set of mathematically unique strategies and robust in pairwise interactions, however, their simplicity and extortionate behaviour have been tested. In [35] a tournament containing over 200 strategies, including ZDs, was ran and none of them ranked in top spots. Instead, the top ranked strategies were a set of trained strategies based on lookup tables [14], hidden markov models [35] and finite state automata [47].

Though only a select pieces of work have been discussed, the IPD literature is rich, and new strategies and competitions are being published every year [34]. The question, however, still remains the same: what is the best way to play the game? Compared to other works, whereas a few selected strategies are evaluated on a small number of tournaments, this manuscript evaluates the performance of 195 strategies in 45686 tournaments. These tournaments do not consist of just standard round robin tournaments, but also tournaments with noise and tournaments with a probabilistic ending. The later part of the paper, evaluates the impact of features on the performance of the strategies using modern machine learning techniques. These features include measures regarding a strategy's behaviour and measures regarding the tournaments. The data set used in this work has been made publicly available [33] and can be used for further analysis and insights.

The different tournament types as well as the data collection, which is made possible due to an open source package called Axelrod-Python, are covered in Section 2. Section 3, focuses on the best performing strategies for each type of tournament and overall. Section 4, explores the traits which contribute to good performance, and finally the results are summarised in Section 5. This manuscripts uses several parameters. These are introduced in the following sections, however, the full set of parameters and their definitions are given in Appendix A. A list of all strategies including citations of their origins is given in Appendix B.

2 Data collection

For the purposes of this manuscript a data set containing results of IPD tournaments has been generated and is available at [33]. This was done using the open source package Axelrod-Python library [4] (APL), and more specifically, version 3.0.0. APL allows for different types of IPD computer tournaments to be simulated whilst containing a list of over 180 strategies. Most of these are strategies described in the literature with a few exceptions being strategies that have been contributed specifically to the package. This paper make use of 195 strategies implemented in version 3.0.0. A list of the strategies is given in the Appendix B. Although APL features several tournament types, this work considers standard, noisy, probabilistic ending and noisy probabilistic ending tournaments.

Standard tournaments, are tournaments similar to that of Axelrod's in [15]. There are N strategies which all play an iterated game of n number of turns against each other. Note that self interactions are not included. Similarly, **noisy tournaments** have N strategies and n number of turns, but at each turn there is a probability p_n that a player's action will be flipped. **Probabilistic ending tournaments**, are of size N and after each turn a match between strategies ends with a given probability p_e. Finally, **noisy probabilistic ending** tournaments have both a noise probability p_n and an ending probability p_e. For smoothing the simulated results a tournament is repeated for k number of times. This was allowed to vary in order to evaluate the effect of smoothing. The winner of each tournament is based on the average score a strategy achieved and not by the number of wins.

The process of collecting tournament results implemented in this manuscript is described by Algorithm 1. For each trial a random size N is selected, and from the 195 strategies a random list of N strategies is chosen. For the given list of strategies a standard, a noisy, a probabilistic ending and a noisy probabilistic ending tournament are performed and repeated k times. The parameters for the tournaments, as well as the number of repetitions, are selected once for each trial. The parameters and their respective minimum and maximum values are given by Table 1.

parameter	parameter explanation	min value	max value
N	number of strategies	3	195
k	number of repetitions	10	100
n	number of turns	1	200
p_n	probability of flipping action at each turn	0	1
p_e	probability of match ending in the next turn	0	1

Table 1: Data collection; parameters' values

The source code for the data collection, as well as the source code for the analysis, which will be discussed in the following sections, have been written following best practices [5, 21]. It has been packaged and is available here.

A total of 11420 trials of Algorithm 1 have been run. For each trial the results for 4 different tournaments were collected, thus a total of 45686 (11420×4) tournament results have been retrieved. Each tournament outputs a result summary in the form of Table 2. Each strategy have participated on average in 5154 tournaments of each type. The strategy with the maximum participation in each tournament type is Inverse Punisher with 5639 entries. The strategy with the minimum entries is EvolvedLookerUp 1 1 1 which was selected in 4693 trials.

The result summary, Table 2, has N rows because each row contains information for each strategy that participated in the tournament. The information includes the strategy's rank, median score, the rate with which the strategy cooperated (C_r), its match win count and the probability that the strategy cooperated in the opening move. Moreover, the probabilities of a strategy being in any of the four states (CC, CD, DC, DD),

Algorithm 1: Data collection Algorithm

foreach *seed* $\in [0, 11420]$ **do**

  $N \leftarrow$ randomly select integer $\in [N_{min}, N_{max}]$;

  players $\leftarrow$ randomly select N players;

  $k \leftarrow$ randomly select integer $\in [k_{min}, k_{max}]$;

  $n \leftarrow$ randomly select integer $\in [n_{min}, n_{max}]$;

  $p_n \leftarrow$ randomly select float $\in [p_{n\,min}, p_{n\,max}]$;

  $p_e \leftarrow$ randomly select float $\in [p_{e\,min}, p_{e\,max}]$;

  result standard $\leftarrow$ Axelrod.tournament(players, n, k);

  result noisy $\leftarrow$ Axelrod.tournament(players, n, p_n, k);

  result probabilistic ending $\leftarrow$ Axelrod.tournament(players, p_e, k);

  result noisy probabilistic ending $\leftarrow$ Axelrod.tournament(players, p_n, p_e, k);

return *result standard, result noisy, result probabilistic ending, result noisy probabilistic ending*;

and the rate of which the strategy cooperated after each state. A feature that has been manually included is the **normalised rank**. The rank of a given strategy, denoted as R, can vary between 0 and $N-1$. Thus, the normalised rank, denoted as r, is calculated as a strategy's rank divided by the tournament's size $N-1$. In the next section the performance of these strategies is evaluated based on their normalised rank.

						Rates							
Rank	Name	Median score	Cooperation rating (C_r)	Win	Initial C	CC	CD	DC	DD	CC to C	CD to C	DC to C	DD to C
0	EvolvedLookerUp2 2 2	2.97	0.705	28.0	1.0	0.639	0.066	0.189	0.106	0.836	0.481	0.568	0.8
1	Evolved FSM 16 Noise 05	2.875	0.697	21.0	1.0	0.676	0.020	0.135	0.168	0.985	0.571	0.392	0.07
2	PSO Gambler 1 1 1	2.874	0.684	23.0	1.0	0.651	0.034	0.152	0.164	1.000	0.283	0.000	0.136
3	PSO Gambler Mem1	2.861	0.706	23.0	1.0	0.663	0.042	0.145	0.150	1.000	0.510	0.000	0.122
4	Winner12	2.835	0.682	20.0	1.0	0.651	0.031	0.141	0.177	1.000	0.441	0.000	0.462
...	...	...	...	...	...	...	...	...	...	...	...	...	...

Table 2: Output result of a single tournament.

3 Top ranked strategies

This section evaluates the performance of 195 IPD strategies. The performance of each strategy is evaluated in four tournament types, which were presented in Section 2, followed by an evaluation of their performance over all the 45686 simulated tournaments of this work.

Each strategy participated in multiple tournaments of the same type (on average 5154). For example Tit For Tat participated in a total of 5114 tournaments of each type. The strategy's normalised rank distribution in these is given in Figure 1. A value of $r = 0$ corresponds to a strategy winning the tournament where a value of $r = 1$ corresponds to the strategy coming last. Because of the strategies' multiple entries their performance is evaluated based on the **median normalised rank** denoted as $\bar{r}$.

The top 15 strategies for each tournament type based on $\bar{r}$ are given in Table 3.

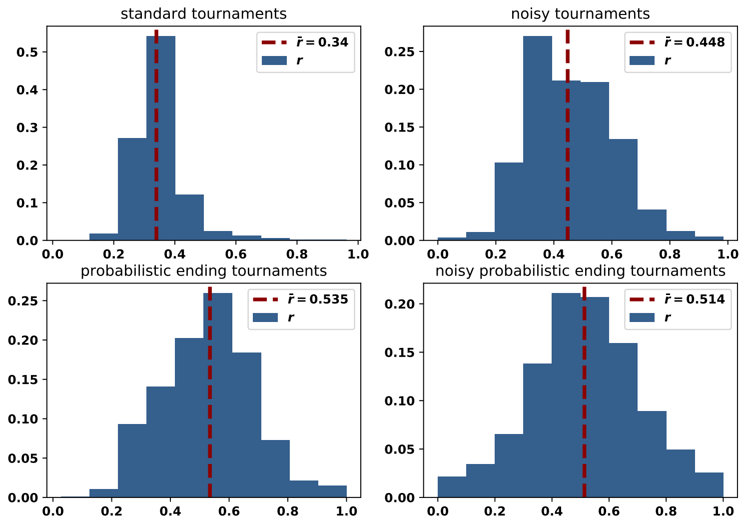


Figure 1: Tit For Tat's r distribution in tournaments. The best performance of the strategy has been in standard tournaments where it achieved a $\bar{r}$ of 0.34.

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending	
	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$
0	Evolved HMM 5	0.00667	Grumpy	0.14020	Fortress4	0.01266	Alternator	0.30370
1	Evolved FSM 16	0.00995	e	0.19388	Defector	0.01429	ϕ	0.30978
2	EvolvedLookerUp2 2 2	0.01064	Tit For 2 Tats	0.20617	Better and Better	0.01587	e	0.31250
3	Evolved FSM 16 Noise 05	0.01667	Slow Tit For Two Tats	0.20962	Tricky Defector	0.01875	π	0.31686
4	PSO Gambler 2 2 2	0.02143	Cycle Hunter	0.21538	Fortress3	0.02174	Limited Retaliate	0.35263
5	Evolved ANN	0.02878	Risky QLearner	0.22222	Gradual Killer	0.02532	Anti Tit For Tat	0.35431
6	Evolved ANN 5	0.03390	Retaliate 3	0.22887	Aggravater	0.02778	Retaliate 3	0.35563
7	PSO Gambler 1 1 1	0.03704	Cycler CCCCCD	0.23507	Raider	0.03077	Limited Retaliate 3	0.35563
8	Evolved FSM 4	0.04891	Retaliate 2	0.23913	Cycler DDC	0.04545	Retaliate	0.35714
9	PSO Gambler Mem1	0.05036	Defector Hunter	0.24038	Hard Prober	0.05128	Retaliate 2	0.35767
10	Winner12	0.06011	Retaliate	0.24177	SolutionB1	0.06024	Limited Retaliate 2	0.36134
11	Fool Me Once	0.06140	Hard Tit For 2 Tats	0.25000	Meta Minority	0.06077	Hopeless	0.36842
12	DBS	0.07143	ShortMem	0.25286	Bully	0.06081	Arrogant QLearner	0.40651
13	DoubleCrosser	0.07200	Limited Retaliate 3	0.25316	Fool Me Forever	0.07080	Cautious QLearner	0.40909
14	BackStabber	0.07519	Limited Retaliate	0.25706	EasyGo	0.07101	Fool Me Forever	0.41764

Table 3: Top performances for each tournament type based on $\bar{r}$.

In standard tournaments 10 out of the 15 top strategies are introduced in [35]. These are strategies based on finite state automata (FSM), hidden markov models (HMM), artificial neural networks (ANN), lookup tables (LookerUp) and stochastic lookup tables (Gambler) that have been trained using reinforcement learning algorithms (evolutionary and particle swarm algorithms). They have been trained to perform well against the strategies in APL in a standard tournament, thus their performance in the specific setting was anticipated. DoubleCrosser, and Fool Me Once, are strategies not from the literature but from the APL. DoubleCrosser is a strategy that makes use of the number of turns because it is set to defect on the last two rounds. The strategy was expected to not perform as well in tournaments where the number of turns is not specified, but the strategy did not perform well in tournaments with noise either. Finally, Winner 12 [46] and DBS [13] are both from the the literature. DBS is strategy specifically designed for noisy environments, however, it ranks highly only in standard ones.

Figure 2 gives the distributions of r for the top ranked strategies. The distributions are skewed towards zero and the highest median, of the top 15 strategies, is at 0.075. This indicates that the top ranked strategies perform well in any given standard tournament, regardless of the opponents and the number of turns.

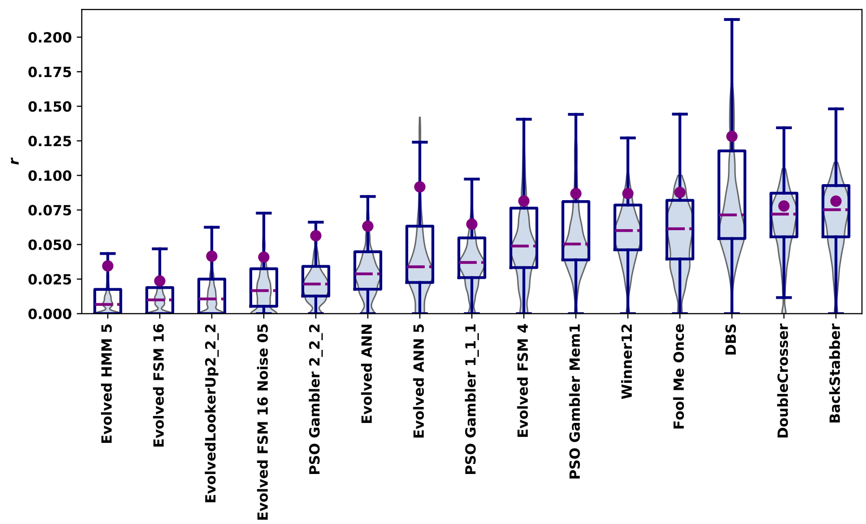


Figure 2: r distributions of top 15 strategies in standard tournaments.

The top strategies in noisy tournaments are shown in Figure 3. These include deterministic strategies, such as Tit For 2 Tats [16], Slow Tit For Two Tats [4], Hard Tit For 2 Tats [60] and Cycler CCCCCD, and strategies which decide their actions based on the cooperations to defections ratio, such as ShortMem [23], Grumpy and e [4]. Slow Tit For Two Tats is the same strategy as Tit For 2 Tats, and at the time of writing this manuscript the contributors of [4] made a new release where the strategy has been removed. However, for the purpose of this work the strategy is kept. The Retaliate and Limited Retaliate strategies are implemented in APL by the same contributor. They are strategies designed to defect if the opponent has tricked them more often than $x\%$ of the times that they have done the same. Finally, in 4^{th} and 9^{th} place are Hunter strategies which try to extort strategies that play cyclically and defectors.

From Figure 3, it is evident that the normalised rank distributions in noisy environments are more variant with higher medians compared to standard tournaments. The distributions are bimodal. This indicates that although the top ranked strategies mainly performed well, there are several tournaments that they ranked in the bottom half. The bimodality of the r distributions can be explained by Figure 3 which gives the r distributions for the top 6 strategies over the noise probability p_n. Note that for $p_n = 0.5$ a strategy corresponds to a random player and for $p_n = 1$ a strategy is behaving in the exact opposite way than its design.

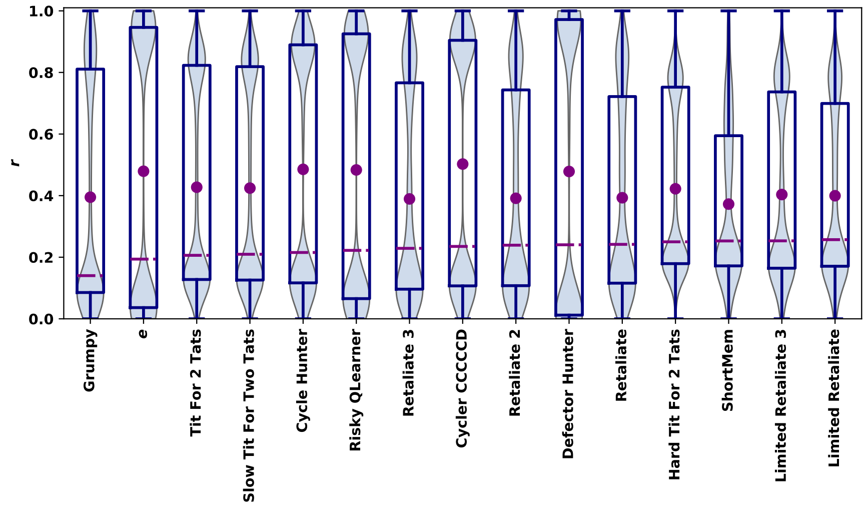


Figure 3: r distributions for best performed strategies in noisy tournaments.

From Figure 4 it is evident that the strategies highly ranked in noisy environments did so because of their performance in tournaments with $p_n > 0.5$. In tournaments with a noise probability lower than 0.5 the strategies performed poorly which is why their r distributions are bimodal. If during the data collection a p_n strictly less 0.5 was considered then the top ranked strategies would be different. There are a total of 5661 trials where $p_n < 0.5$ and the top ranked strategies are given in Table 4. The median ranks are lower than before and the top spots are mainly overtaken by Meta strategies which include NMWE deterministic and NMWE Long Memory. The Meta strategies [4] create a team of strategies for themselves and choose to play as a member of their team based on their scores against a given opponent.

Name	$\bar{r}$
MEM2	0.06135
Spiteful Tit For Tat	0.06344
Nice Meta Winner	0.06620
Grudger	0.06667
Meta Winner Long Memory	0.07339
Forgiver	0.07362
Fool Me Once	0.07362
Meta Winner	0.07487
Meta Winner Memory One	0.07621
Meta Winner Finite Memory	0.07692
Meta Winner Deterministic	0.07792
NMWE Deterministic	0.08696
NMWE Long Memory	0.08696
CollectiveStrategy	0.08696
Defector	0.08889

Table 4: Top performances in 5661 noisy tournaments where $p_n < 0.5$.

The 15 top ranked strategies in probabilistic ending tournaments include Fortress 3, Fortress 4 (both introduced in [11]), Raider [12] and Solution B1 [12], which are strategies based on finite state automata introduced by Daniel and Wendy Ashlock. These strategies have been evolved using reinforcement learning, however, there were trained to maximise their payoffs in tournaments with fixed turns (150 specifically) and not in probabilistic ending ones. In probabilistic ending tournaments it appears that the top ranks are mostly occupied by defecting strategies. These include Better and Better, Gradual Killer, Hard Prober (all from [1]), Bully (Reverse Tit For Tat) [50] and Defector. Thus, it's surprisingly that EasyGo and Fool Me

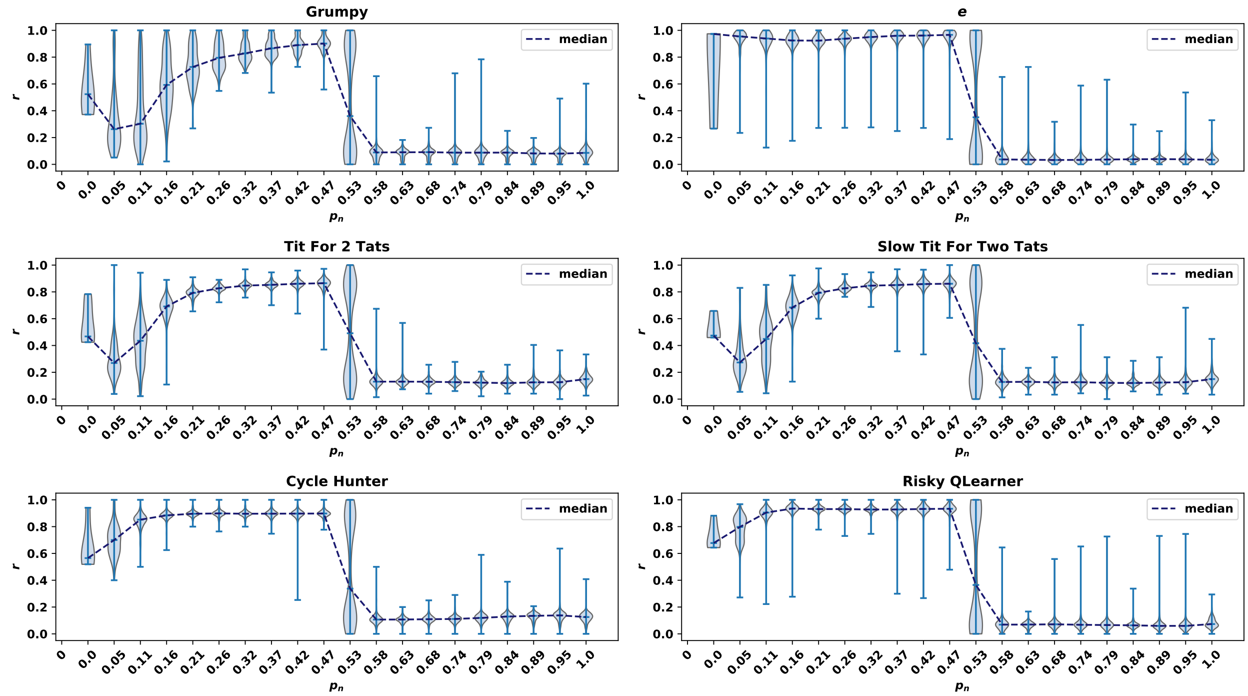


Figure 4: r distributions for top 6 strategies in noisy tournaments over the probability of noisy (p_n).

Forever which are strategies that will defect until their opponent defect, then they will cooperate until the end, ranked 14$^{\text{th}}$ and 15$^{\text{th}}$. Upon inspection, it was found that they are actually the same strategy. This was not known to the authors at the time of data collection. Figure 5 verifies that their performance is the same. Both strategies have repeatedly ranked highly and there are cases for which they were the winners of the tournament.

The distributions of the normalised rank in probabilistic ending tournaments, shown in Figure 5, are less variant than those of noisy tournaments. The medians of the top 15 strategies are lower than 0.1 and the distributions are skewed towards 0. Though the large difference between the means and the medians indicates some outliers, the strategies have overall performed well in the probabilistic ending tournaments that they participated.

The distributions of r for the top 6 strategies in probabilistic ending tournaments over p_e are given in Figure 6. Figure 6 shows that the 6 strategies start of with a high median rank, however, their ranked decreased as the the probability of the game ending increased and at the point of $p_e = 0.1$ they became the dominant strategies in their respective tournaments. In essence, what is demonstrated is that defecting strategies did better when the likelihood of the game ending in the next turn increased, which is inline with the Folk Theorem [31]. If tournaments where the probability of the game ending was less than 0.1 were considered then the the top ranked spots are not dominated by just defecting strategies anymore, as shown in Table 5. Instead the effective strategies are now the Meta strategies, trained strategies, Grudger [4] and Spiteful Tit for Tat [1].

In tournaments with both noise and an unspecified number of turns several of the top ranked strategies are strategies that were highly ranked in noisy tournaments. However, strategies from the top ranks in probabilistic ending tournaments did not rank highly here. Other strategies include π, ϕ which are based on the same approach as e. The distributions of r shown in Figure 7 have the largest median values compared

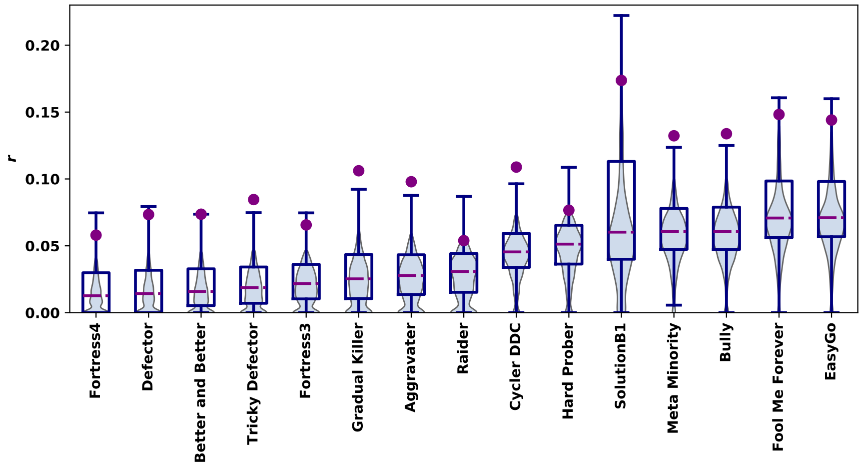


Figure 5: r distributions for best performed strategies in probabilistic ending tournaments.

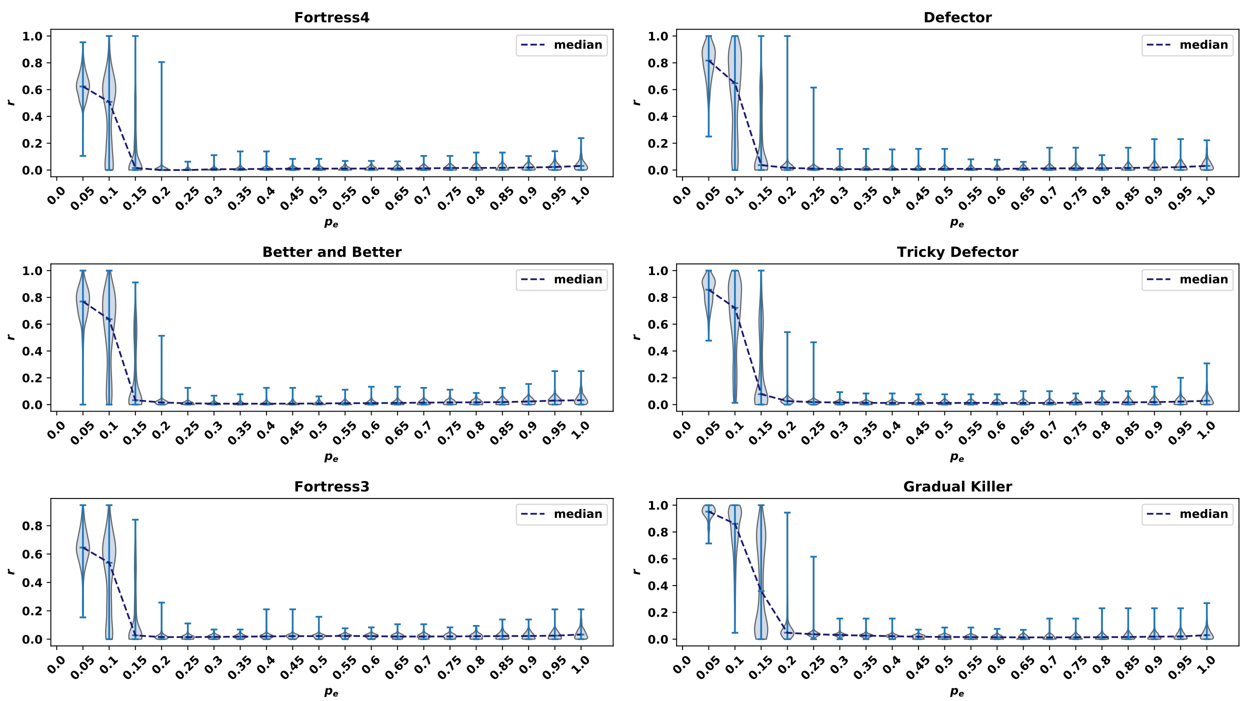


Figure 6: r distributions for top 6 strategies in probabilistic ending tournaments over p_e.

to the top rank strategies of the other tournament types. A subset of noisy probabilistic ending tournaments has been considered such that $p_e < 0.1$ and $p_n < 0.5$. The top ranked strategies are given in Table 6 and it is shown that the Meta strategies which performed well in noisy tournaments with $p_n < 0.5$, perform well once again even the number of turns is not specified. Moreover, several strategies that did well in probabilistic ending tournaments such as Fortress 3, Fortress 4, Defector and Better and Better are effective here as well.

Up till now, the performances of the 195 strategies have been evaluated for individual tournament types. The distributions of r for the tournament types indicate that for probabilistic ending and standard tournaments

Name	$\bar{r}$
Evolved FSM 16	0.00000
Evolved FSM 16 Noise 05	0.01266
MEM2	0.02715
Evolved HMM 5	0.04423
EvolvedLookerUp2 2 2	0.04870
Spiteful Tit For Tat	0.05958
Nice Meta Winner	0.06842
NMWE Finite Memory	0.06923
Grudger	0.06985
NMWE Deterministic	0.07018
NMWE Long Memory	0.07407
Nice Meta Winner Ensemble	0.07595
EvolvedLookerUp1 1 1	0.07692
NMWE Memory One	0.08000
NMWE Stochastic	0.08475

Table 5: Top performances in 1139 probabilistic ending tournaments with $p_e < 0.1$

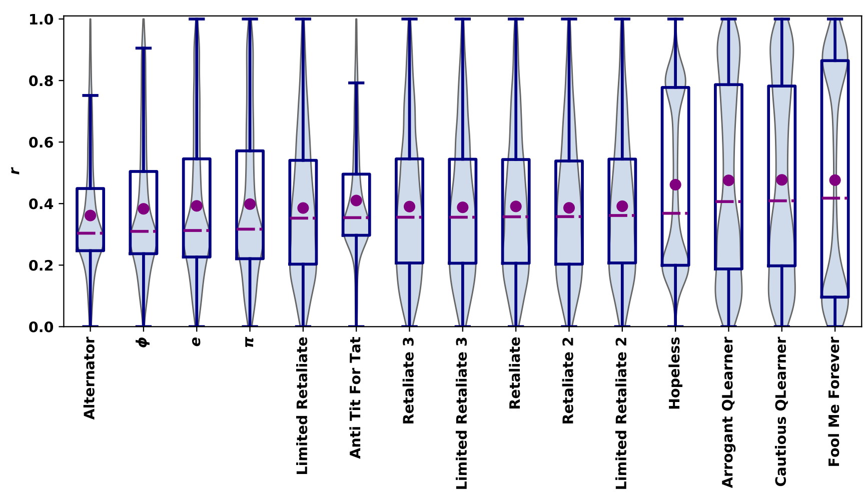


Figure 7: r distributions for best performed strategies in noisy probabilistic ending tournaments.

Name	$\bar{r}$
Defector	0.00552
Better and Better	0.01055
Aggravater	0.01399
Fortress4	0.02100
Tricky Defector	0.03857
Meta Winner Long Memory	0.04878
Meta Winner Memory One	0.04955
Meta Winner Finite Memory	0.04972
Meta Winner Stochastic	0.05128
Meta Winner Deterministic	0.05195
Meta Winner	0.05333
Meta Winner Ensemble	0.05882
Fortress3	0.06956
CollectiveStrategy	0.07692
Prober 3	0.08018

Table 6: Top performances in 568 probabilistic ending tournaments with $p_e < 0.1$ and $p_n < 0.5$.

successful strategies do exist. For these settings, the top 15 strategies have frequently ranked in the top spots with only a few exceptions. Contrarily, it appears that noise cause variation in the normalised ranks, and the strategies can always guarantee a spot in the top ranks.

The data set considered in this work, described in Section 2, contains a total of 45686 tournament results. For this part of the manuscript the strategies are ranked based on the median normalised rank they achieved over the entire data set. The top 15 strategies are given in Table 7 and their normalised rank distributions are given in Figure 8.

Name	$\bar{r}$
Limited Retaliate 3	0.28609
Retaliate 3	0.29630
Retaliate 2	0.30250
Limited Retaliate 2	0.30328
Limited Retaliate	0.31000
Retaliate	0.31707
BackStabber	0.32381
DoubleCrosser	0.33136
Nice Meta Winner	0.34921
PSO Gambler 2 2 2 Noise 05	0.35146
Grudger	0.35156
Evolved HMM 5	0.35714
NMWE Memory One	0.35714
Nice Meta Winner Ensemble	0.35870
Forgetful Fool Me Once	0.35884

Table 7: Top performances over all the tournaments

The top ranks include strategies that have been previously mentioned. The set of Retaliate strategies occupy the top spots followed by BackStabber and DoubleCrosser. The distributions of the Retaliate strategies have no statistical difference. Thus, in an IPD tournament where the type is not specified, playing as any of the Retaliate strategies seems like a good approach. DoubleCrosser performed well in standard tournaments and the strategy is just an extension of BackStabber. It should be noted that these strategies can be characterised as "cheaters". The source code of the strategies allows them to known the number of turns in a match (if they are specified). PSO Gambler and Evolved HMM 5 are trained strategies introduced in [35] and Nice Meta Winner and NMWE Memory One are strategies based on teams. Grudger is a strategy from Axelrod's original tournament and Forgetful Fool Me Once is based on the same approach as Grudger. Overall the top 15 strategies are fundamentally different. Some are cheaters, some are complex, others are simple deterministic strategies and strategies based on teams. The results of 45686 tournaments used in this work imply the following: there is not a single type of strategy which can performance well in any IPD interaction.

This section presented the winning strategies in a series of IPD tournaments. In standard tournaments the top spots were dominated by complex strategies that had been trained using reinforcement learning techniques. In noisy environments, whether the number of turns was fixed or not, the winning strategies were deterministic strategies designed to defect if the opponent tricked them more than a current amount of the times that they had tricked their opponent. However, if a value of noise strictly less than 0.5 was considered, then the successful strategies were strategies based on the behaviour of many strategies. In probabilistic ending tournaments most of the highly ranked strategies were defecting strategies and trained finite state automata, all by the authors of [11, 12]. These strategies ranked high due to their performance in tournaments where the probability of the game ending after each turn was bigger than 0.1. Finally the performance of all 195 strategies over the 45686 tournaments in this manuscript was assessed on $\bar{r}$. The top ranked strategies were a mixture of behaviours that did well in standard tournaments and tournaments with noise, as well as a few strategies based on teams.

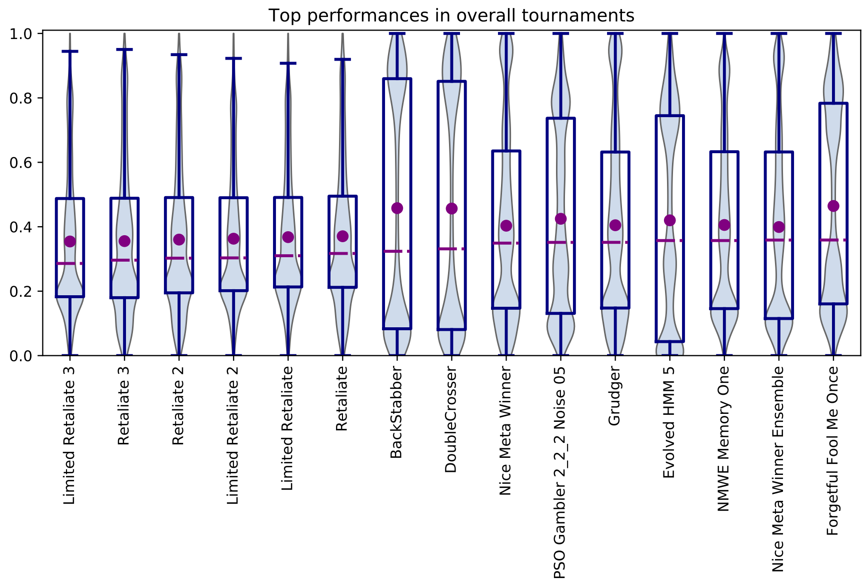


Figure 8: r distributions for best performed strategies in the data set [33].

The results of this section imply that successful strategies for specific settings exist for an IPD tournament. The top ranked strategies in both standard tournaments and tournaments with probabilistic ending, managed to rank in the top 10% of the tournament most of the times. Strategies in noisy environments demonstrated that no strategy can be consistently successful, except if the value of noise is constrained to less than a half. Overall, there has been not a single strategy that has shown to perform well in more than one setting. The aim of the next section is to understand the features that made these strategies successful, in each setting separately but also overall.

4 Evaluation of performance

The aim of this section is to explore the features that contribute to a strategy's successful performance. The features explored are measures regarding a strategy's behaviour, along with measures regarding the tournaments the strategies competed in. These are given in Table 8.

APL makes use of classifiers to classify strategies according to various dimensions. These determine whether a strategy is stochastic or deterministic, whether it makes use of the number of turns or the game's payoffs. The memory usage feature is calculated as the memory size of strategy (which is specified in the strategies implementation in the APL) divide by the number of turns. For example, Evolved FSM 16 Noise 05 has a memory size of 16 and participated in a tournament where n was 134. In the given tournament Evolved FSM 16 Noise 05 has a memory usage of 0.119. For tournaments with a probabilistic ending the number of turns was not collected, so the memory usage feature is not used for probabilistic ending tournaments. The SSE is a feature introduced in [41] which shows how close a strategy is to behaving as a ZDs, and subsequently, in an extortionate way. The method identifies the ZDs closest to a given strategy and calculates the algebraic distance between them, defined as SSE. A SSE value of 1 indicates no extortionate behaviour at all whereas a value of 0 indicates that a strategy is behaving a ZDs. The rest of the features considered are the CC to C, CD to C, DC to C, and DD to C rates as well as cooperating ratio of a strategy. The minimum, maximum, medium and median cooperating ratios of each tournament are also included, and finally the number of turns,

feature	feature explanation	source	value type	min value	max value
stochastic	If a strategy is stochastic	strategy classifier from APL	boolean	Na	Na
makes use of game	If a strategy makes used of the game information	strategy classifier from APL	boolean	Na	Na
makes use of length	If a strategy makes used of the number of turns	strategy classifier from APL	boolean	Na	Na
memory usage	The memory size of a strategy divided by the number of turns	memory size from APL	float	0	1
SSE	A measure of how far a strategy is from ZD behaviour	method described in [41]	float	0	1
max cooperating rate (C_{max})	The biggest cooperating rate in a given tournament	result summary	float	0	1
min cooperating rate (C_{min})	The smallest cooperating rate in a given tournament	result summary	float	0	1
median cooperating rate (C_{median})	The median cooperating rate in a given tournament	result summary	float	0	1
mean cooperating rate (C_{mean})	The mean cooperating rate in a given tournament	result summary	float	0	1
C_r / C_{max}	A strategy's cooperating rate divided by the maximum	result summary	float	0	1
C_r / C_{min}	A strategy's cooperating rate divided by the minimum	result summary	float	0	1
C_r / C_{median}	A strategy's cooperating rate divided by the median	result summary	float	0	1
C_r / C_{mean}	A strategy's cooperating rate divided by the mean	result summary	float	0	1
C_r	The cooperating ratio of a strategy	result summary	float	0	1
CC to C rate	The probability a strategy will cooperate after a mutual cooperation	result summary	float	0	1
CD to C rate	The probability a strategy will cooperate after being betrayed by the opponent	result summary	float	0	1
DC to C rate	The probability a strategy will cooperate after betraying the opponent	result summary	float	0	1
DD to C rate	The probability a strategy will cooperate after a mutual defection	result summary	float	0	1
p_n	The probability of a player's action being flip at each interaction	trial summary	float	0	1
n	The number of turns	trial summary	integer	1	200
p_e	The probability of a match ending in the next turn	trial summary	float	0	1
N	The number of strategies in the tournament	trial summary	integer	3	195
k	The number of repetitions of a given tournament	trial summary	integer	10	100

Table 8: The features which are included in the performance evaluation analysis.

the number of strategies, the number of repetitions and the probabilities of noise and the game ending are also included.

Table 9 shows the correlation coefficients between the features of Table 8 the median score and the median normalised rank. Note that the correlation for the classifiers is not included because they are binary variables and they will be evaluated using a different method. The correlation coefficients for all the features in Table 8 against themselves have also been calculated and a graphical representation can be found in the Appendix C.

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending		Overall	
	r	median score	r	median score	r	median score	r	median score	r	median score
CC to C rate	-0.501	0.501	0.414	-0.504	0.408	-0.323	0.260	0.022	-0.501	0.501
CD to C rate	0.226	-0.199	0.456	-0.330	0.320	-0.017	0.205	-0.220	0.226	-0.199
C_r	-0.323	0.384	0.711	-0.678	0.714	-0.832	0.579	-0.135	-0.323	0.384
C_r / C_{max}	-0.323	0.381	0.616	-0.551	0.714	-0.833	0.536	-0.116	-0.323	0.381
C_r / C_{mean}	-0.331	0.358	0.731	-0.740	0.721	-0.861	0.649	-0.621	-0.331	0.358
C_r / C_{median}	-0.331	0.353	0.652	-0.669	0.712	-0.852	0.330	-0.466	-0.331	0.353
C_r / C_{min}	0.109	-0.080	-0.358	0.250	-0.134	0.150	-0.368	0.113	0.109	-0.080
C_{max}	-0.000	0.049	0.000	0.023	-0.000	0.046	0.000	-0.004	-0.000	0.049
C_{mean}	-0.000	0.229	-0.000	0.271	0.000	0.200	0.000	0.690	-0.000	0.229
C_{median}	0.000	0.209	-0.000	0.240	-0.000	0.187	-0.000	0.673	0.000	0.209
C_{min}	0.000	0.084	0.000	-0.017	-0.000	0.007	-0.000	0.041	0.000	0.084
DC to C rate	0.127	-0.100	0.509	-0.504	-0.018	0.033	0.341	-0.016	0.127	-0.100
DD to C rate	0.412	-0.396	0.533	-0.436	-0.103	0.176	0.378	-0.263	0.412	-0.396
N	0.000	-0.009	-0.000	0.002	-0.000	0.003	-0.000	0.001	0.000	-0.009
k	0.000	-0.002	-0.000	0.003	-0.000	0.001	-0.000	-0.008	0.000	-0.002
n	0.000	-0.125	-0.000	-0.024	-	-	-	-	0.000	-0.125
p_e	-	-	-	-	0.000	0.165	0.000	-0.058	-0.001	0.001
p_n	-	-	-0.000	0.207	-	-	-0.000	-0.650	0.002	-0.000
Make use of game	-0.003	-0.022	0.025	-0.082	-0.053	-0.108	0.013	-0.016	-0.003	-0.022
Make use of length	-0.158	0.124	0.005	-0.123	-0.025	-0.090	0.014	-0.016	-0.154	0.117
SSE	0.473	-0.452	0.463	-0.337	-0.156	0.223	0.305	-0.259	0.473	-0.452
memory usage	-0.082	0.095	-0.007	-0.017	-	-	-	-	-0.084	0.095
stochastic	0.006	-0.024	0.022	-0.026	0.002	-0.130	0.021	-0.013	0.006	-0.024

Table 9: Correlations table between the features of Table 8 the normalised rank and the median score.

In standard tournaments the features CC to C, C_r, C_r/C_{max} and the cooperating ratio compared to C_{median} and C_{mean} have a moderate negative effect on the normalised rank, and a moderate positive on the median score. The SSE error and the DD to C have the opposite effects. Thus, in standard tournaments behaving cooperatively corresponds to a more successful performance. Even though being nice pays off, that's not true against defective strategies. Cooperating after a mutual defection lowers a strategy's success. Figure 9 confirms that the winners of standard tournaments always cooperate after a mutual cooperation and almost always defects after a mutual defection.

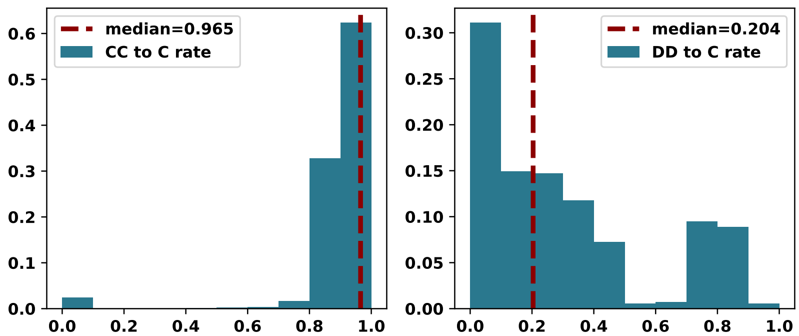


Figure 9: Distributions of CC to C and DD to C for the winners in standard tournaments.

Compared to standard tournaments, in both noisy and in probabilistic ending tournaments the higher the rates of cooperation the lower a strategy's success and median score. A strategy would want to cooperate less than both the mean and median cooperator in such settings. In probabilistic ending tournaments the correlations coefficients have a larger values, indicating a stronger effect. Thus a strategy will be punished more by it's cooperative behaviour in probabilistic ending environments, this was seen in Section 4 as well. The distributions of the C_r of the winners in both tournaments is given by Figure 10. It confirms that the winners in noisy tournaments cooperated less than 35% of the times and in probabilistic ending tournaments less than 10%. In noisy probabilistic ending tournaments and in over all the tournaments' results, the only features that had a moderate affect are $C_r/C_{\text{mean}}, C_r/C_{\text{max}}$ and C_r. In such environments cooperative behaviour appears to be punished by not as much as in noisy and probabilistic ending tournaments.

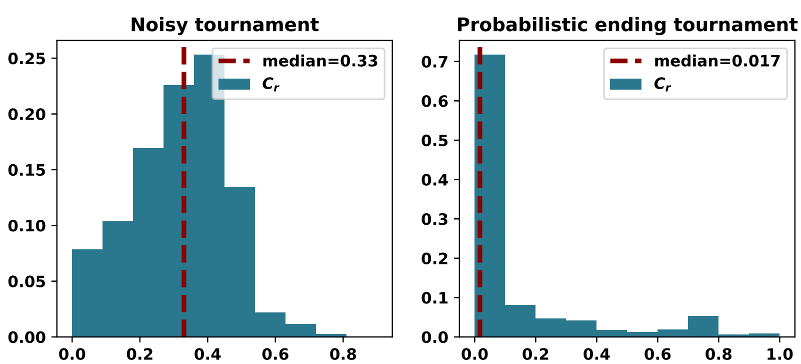


Figure 10: C_r distributions of the winners in noisy and in probabilistic ending tournaments.

To further explore the features that contribute to a strategy's success the performances are divided into clusters based on whether they were successful or not. A random forest approach [22] is then applied to each performance to predict the cluster to which it has been assigned to. This allows for the importance of each feature in the classification to be calculated. In essence, to calculate which are the important features when one is trying to predict whether a performance was successful or not.

The performances are clustered into successful and unsuccessful clusters based on 4 different approaches.

More specifically:

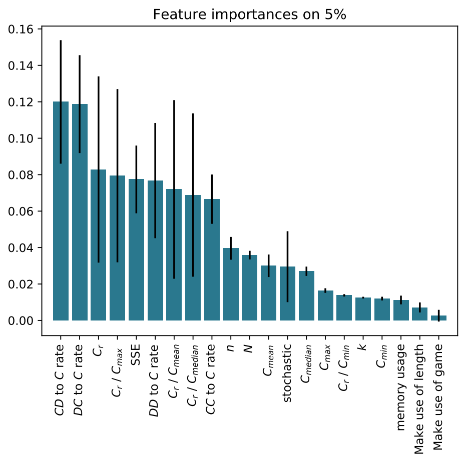
- **Approach 1:** The performances are divided into two clusters based on whether their performance was in the top 5% of their respective tournaments. Thus, whether r was smaller or larger than 0.05.
- **Approach 2:** The performances are divided into two clusters based on whether their performance was in the top 25% of their respective tournaments. Thus, whether r was smaller or larger than 0.25.
- **Approach 3:** The performances are divided into two clusters based on whether their performance was in the top 50% of their respective tournaments. Thus, whether r was smaller or larger than 0.50.
- **Approach 4:** The performances are clustered based on their normalised rank and their median score by a $k-$means algorithm [8]. The number of clusters is not deterministically chosen but it is based on the silhouette coefficients [57].

The random forest method constructs many individual decision trees and the predictions from all trees are pooled to make the final prediction. The random forest models are trained on a training set of 70% of the tournaments results. The accuracy of each model based on R^2 and the number of clusters for each tournament type (because in the case of Approach 4 it is not deterministically chosen) are given by Table 10. The out of the bag error (OOB) [36] has also been calculated. The models fit well, and a high value of both the accuracy measures on the test data and the OOB error indicate that the model is not over fitting.

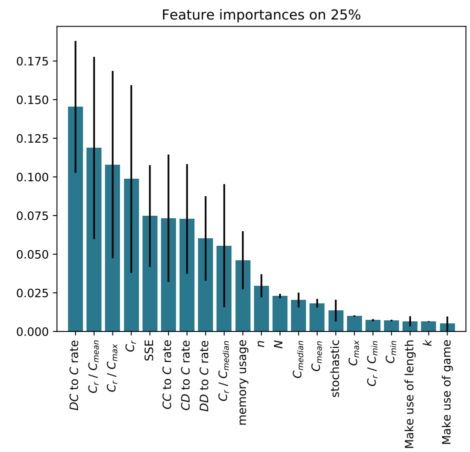
Tournament type	Clustering Approach	Number of clusters	R^2 training data	R^2 test data	R^2 OOB score
standard	Approach 1	2	0.998831	0.987041	0.983708
	Approach 2	2	0.998643	0.978626	0.969202
	Approach 3	2	0.998417	0.985217	0.976538
	Approach 4	2	0.998794	0.990677	0.982959
noisy	Approach 1	2	0.996677	0.950572	0.935383
	Approach 2	2	0.996677	0.950572	0.935383
	Approach 3	2	0.996677	0.950572	0.935383
	Approach 4	3	0.996677	0.950572	0.935383
probabilistic ending	Approach 1	2	0.999592	0.995128	0.992819
	Approach 2	2	0.999592	0.995128	0.992819
	Approach 3	2	0.999592	0.995128	0.992819
	Approach 4	2	0.999592	0.995128	0.992819
noisy probabilistic ending	Approach 1	2	0.990490	0.813905	0.791418
	Approach 2	2	0.990490	0.813905	0.791418
	Approach 3	2	0.990490	0.813905	0.791418
	Approach 4	4	0.990490	0.813905	0.791418
over 45686 tournaments	Approach 1	2	0.993396	0.913409	0.898059
	Approach 2	2	0.993396	0.913409	0.898059
	Approach 3	2	0.993396	0.913409	0.898059
	Approach 4	3	0.993396	0.913409	0.898059

Table 10: Accuracy metrics for random forest models.

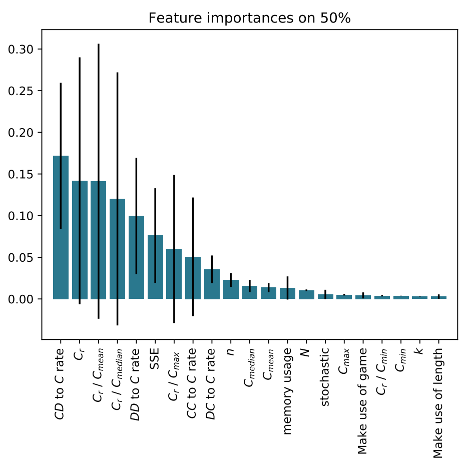
The importance that the features of Table 8 had on each random forest model while the perfomances were clustered based on the different approaches have been calculated and are given by Figures 11, 12, 13, 14 and 15. These show that the classifiers stochastic, make use of game and make use of length have no significant effect, and several of the features that are highlighted by the importance are inline with the correlation results. Moreover, the smoothing parameter k appears to no have a significant effect either. The most important features based on the random forest analysis were C_r/C_{median} and C_r/C_{mean}.



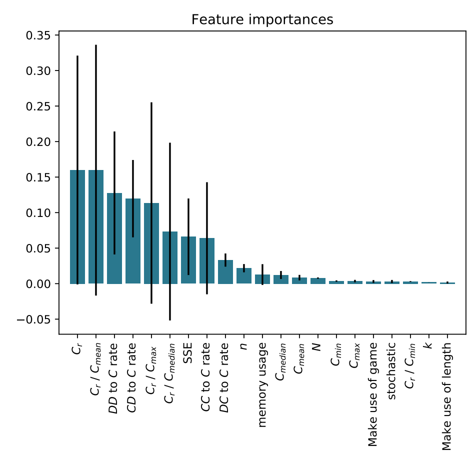
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.

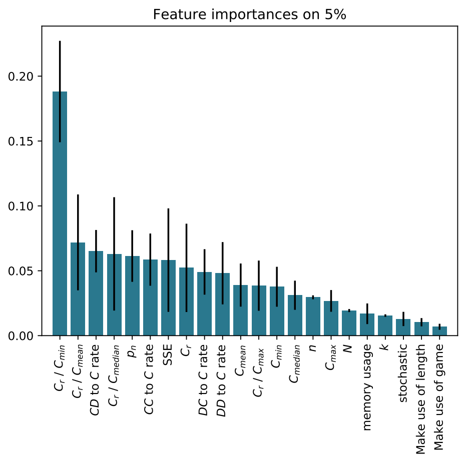


(c) Importance of features for clusters on 50% performance.

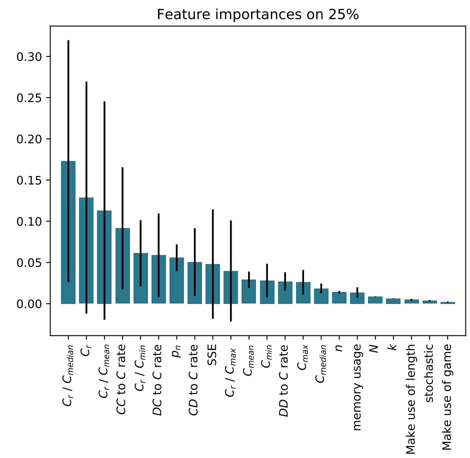


(d) Importance of features for clusters based on $k$means algorithm.

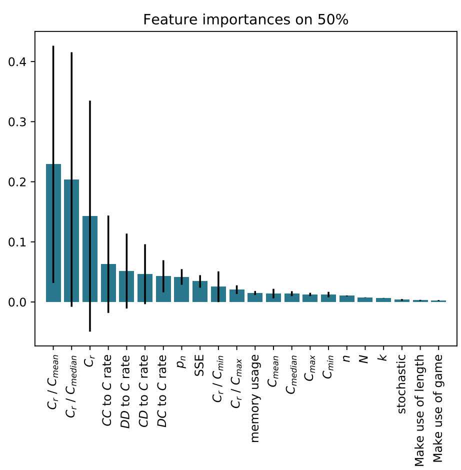
Figure 11: Importance of features in standard tournaments for different clustering methods.



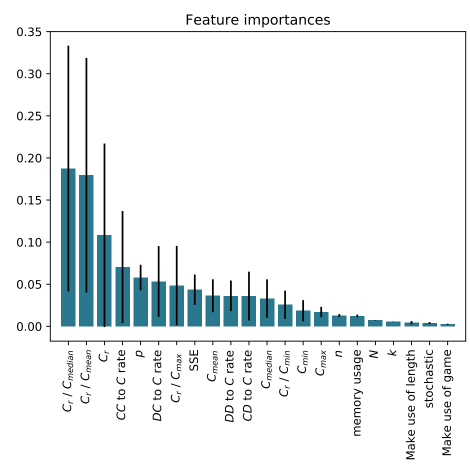
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.

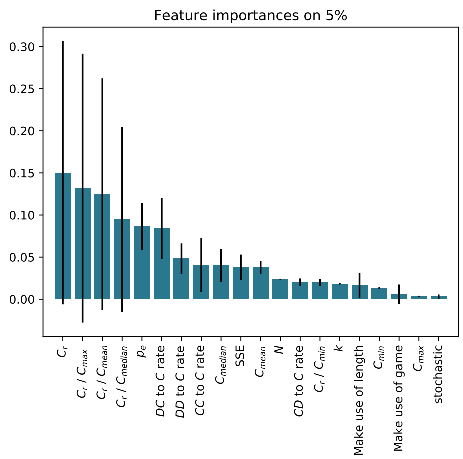


(c) Importance of features for clusters on 50% performance.

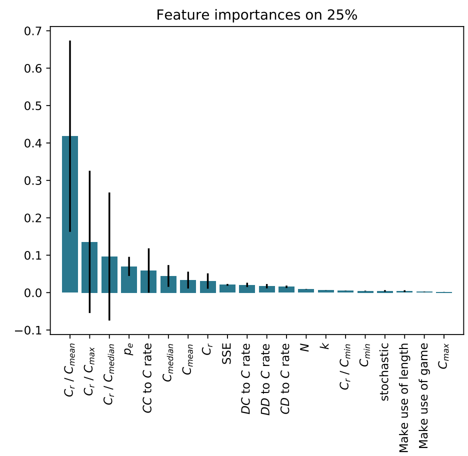


(d) Importance of features for clusters based on $k$means algorithm.

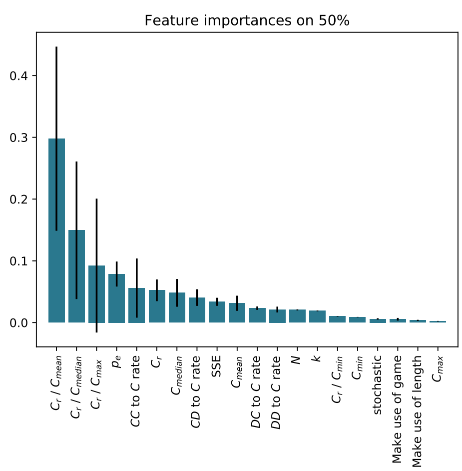
Figure 12: Importance of features in noisy tournaments for different clustering methods.



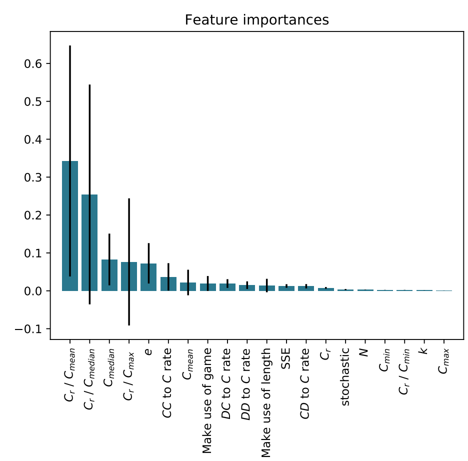
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.

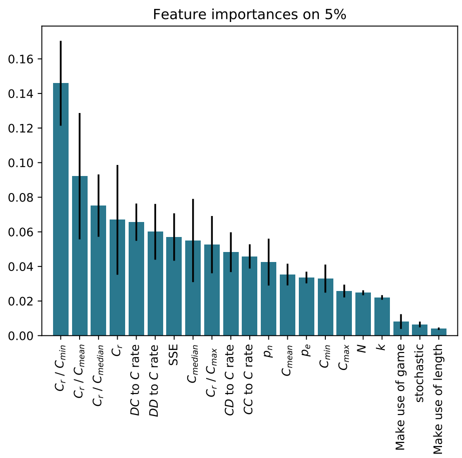


(c) Importance of features for clusters on 50% performance.

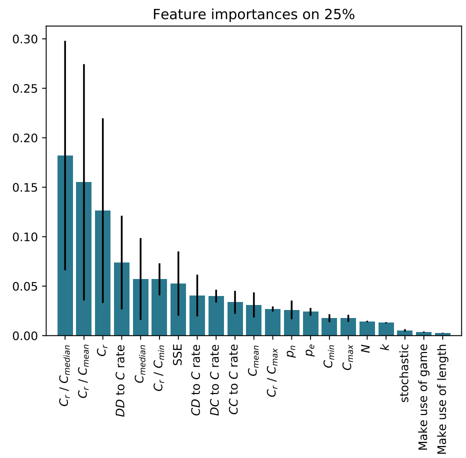


(d) Importance of features for clusters based on $k$means algorithm.

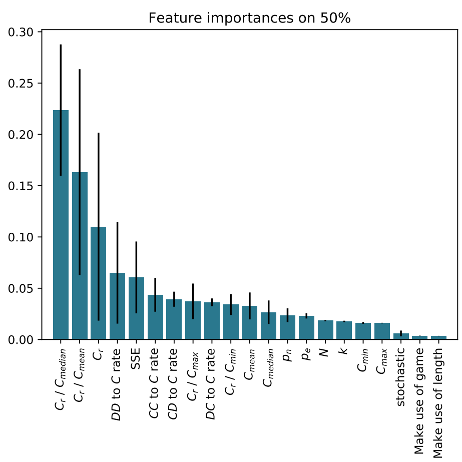
Figure 13: Importance of features in probabilistic ending tournaments for different clustering methods.



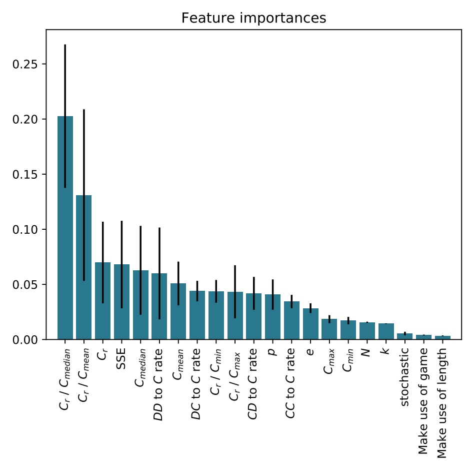
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.

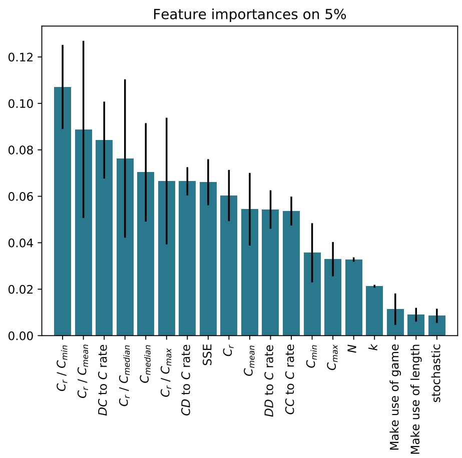


(c) Importance of features for clusters on 50% performance.

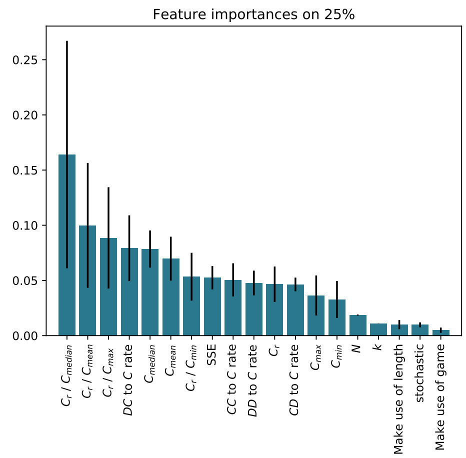


(d) Importance of features for clusters based on $k$means algorithm.

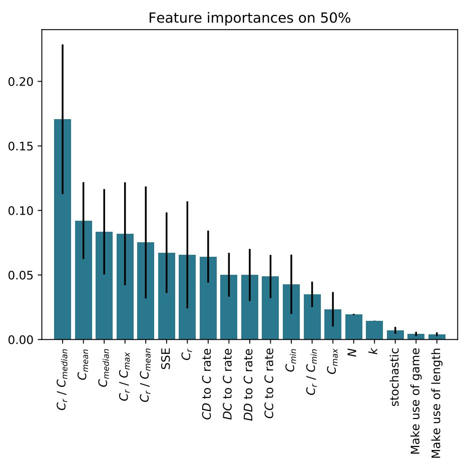
Figure 14: Importance of features in noisy probabilistic ending tournaments for different clustering methods.



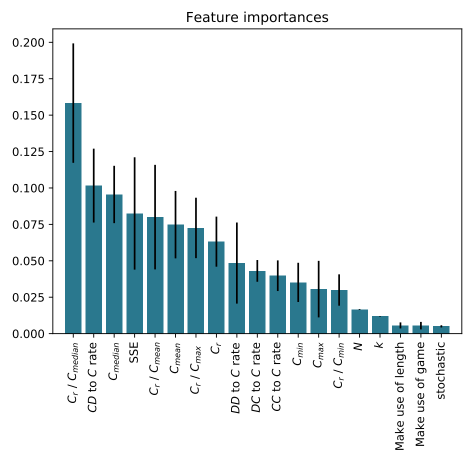
(a) Importance of features for clusters on 5% performance.



(b) Importance of features for clusters on 25% performance.



(c) Importance of features for clusters on 50% performance.



(d) Importance of features for clusters based on $k$means algorithm.

Figure 15: Importance of features over all the tournaments for different clustering methods.

The effect of both these features can be further explored. In Figure 16 the distributions of C_r/C_{mean} and C_r/C_{median} are given for the winners in standard tournaments. A value of $C_r/C_{\text{mean}} = 1$ imply that the cooperating ratio of the winner was the same as the mean/median cooperating ratio of the tournament. In standard tournaments, the mean for both ratios is 1. Therefore, an effective strategy in standard tournaments was the mean/median cooperator of its respective tournament. In comparison, Figure 17 shows the distributions of the features for the winners in noisy tournaments where the mean is at 0.67. Thereupon the winners cooperated 67% of the times the mean/median cooperator did. This analysis is applied to the rest of the tournaments and the distributions are given by Figures 18, 19 and 20. In a tournament with noisy and a probabilistic ending the winners cooperated 60%, whereas in settings that the type of the tournament can vary between the types considered in this work the winners cooperated 67% of the times the mean or median cooperator did. Finally, in probabilistic ending tournament it has already been mentioned that defecting strategies prevail and this result is once again confirmed in this section.

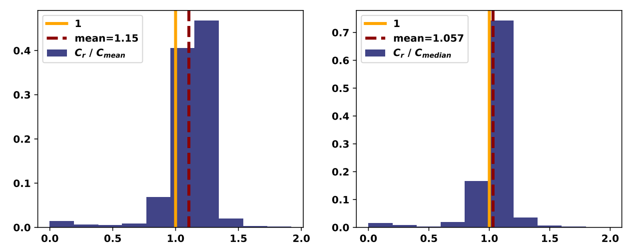


Figure 16: Distributions of C_r/C_{median} and C_r/C_{median} for winners of standard tournaments.

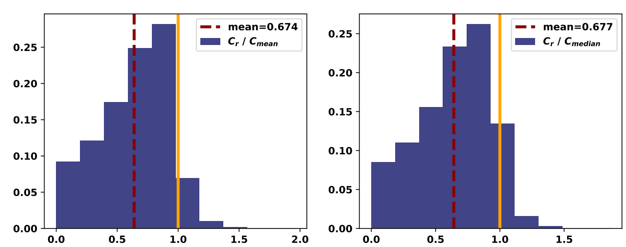


Figure 17: Distributions of C_r/C_{median} and C_r/C_{median} for winners of noisy tournaments.

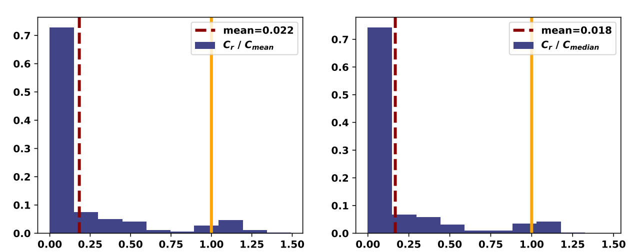


Figure 18: Distributions of C_r/C_{median} and C_r/C_{median} for winners of probabilistic ending tournaments.

In this section the effect of several features, regarding a strategy's behaviour and the tournament in which it participated on its performance were presented. This was done using two approaches. Correlation coefficients and a random forest analysis. The results of these are summarised in the following section.

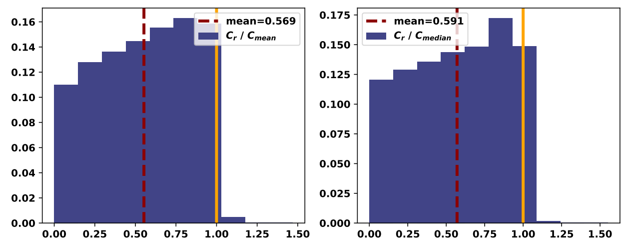


Figure 19: Distributions of C_r/C_{median} and C_r/C_{median} for winners of noisy probabilistic ending tournaments.

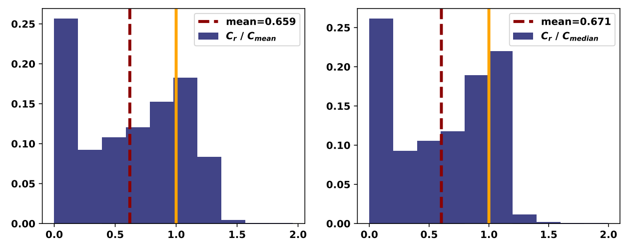


Figure 20: Distributions of C_r/C_{median} and C_r/C_{median} for winners of over all the tournaments.

5 Conclusion

This manuscript has explored the performance of 195 strategies of the Iterated Prisoner's Dilemma in a large number of computer tournaments. The results of the analysis demonstrated that, although for specific tournament types such as standard and probabilistic ending tournaments, dominant strategies exist there is not a single dominant type of strategies if the environments vary. Moreover, a strategy with a theory of mind should aim to adapt its behaviour based on the mean and median cooperators and should in general not be too cooperative.

The 195 strategies used in this manuscript have been mainly from the literature, and they have been accessible due to an open source software called the Axelrod-Python library. The software was used to generate a total of 45686 computer tournaments results with different number of strategies and different participants each time. The data collection was described in Section 2. In Section 3, the tournaments results were used to present the top performances. The data set contained results from four different settings, and these were also studied individually. In standard tournaments complex strategies trained using reinforcement learning ranked in the top spots. Some of these strategies ranked again in the top spots in probabilistic ending tournaments when a p_e of less 0.1 was considered. In probabilistic ending tournaments p_e was designed to vary between 0 and 1. It was demonstrated that for values larger than 0.1, as stated in the Folk Theorem, defecting strategies were winning the tournaments because there was a high likelihood of the game ending in the next turn. In tournaments with noise the median ranks of the top 15 strategies had the highest values and the r distributions were bimodal. The top rank strategies were performing both well and bad, and this indicates that in noisy tournaments there are not strategies that can guarantee winning. Overall, the top ranked strategies differed from one tournament type to another and the mechanism behind the winning strategies were all different. Even strategies designed to perform well in one setting were demonstrated to be better in others.

Section 4, covered an analysis of performance based on several features associated with a strategy and with the environments it was competing. The results of this analysis showed that a strategy's characteristics such as whether or not it's stochastic, and the information it used regarding the game had no effect on the strategy's success. The most important features have been those that compared the strategy's behaviour to it's environment. The cooperating ratio of the strategy compared to the mean and median cooperator was highlighted as the most important feature in the analysis. More specifically, if a strategy were to enter a tournament with a theory of mind of its environment it would choose to be the median cooperator in standard tournaments, to cooperate 10% of the time the median cooperator did in probabilistic ending tournaments and 60% in noisy and noisy probabilistic tournaments. Lastly, if a strategy was aware of the opponents but not of the setting of the tournament, a strategy would be more likely to be successful if it were to identify the median cooperator and cooperated 67% of the times that they did.

The data set described in this work contains the largest number of IPD tournaments, to the authors knowledge, and available at [33]. Further data mining could be applied and provide new insights in the field.

References

[1] Lifl (1998) prison. `http://www.lifl.fr/IPD/ipd.frame.html`. Accessed: 2017-10-23.

[2] A strategy with novel evolutionary features for the iterated prisoner's dilemma. *Evolutionary Computation*, 17(2):257–274, 2009.

[3] Prisoner's dilemma tournament results. `https://www.lesswrong.com/posts/hamma4XgeNrsvAJv5/prisoner-s-dilemma-tournament-results`, 2011.

[4] The Axelrod project developers . Axelrod: 3.0.0, April 2016.

[5] Mark Aberdour. Achieving quality in open-source software. *IEEE software*, 24(1):58–64, 2007.

[6] Christoph Adami and Arend Hintze. Evolutionary instability of zero-determinant strategies demonstrates that winning is not everything. *Nature communications*, 4(1):2193, 2013.

[7] Eckhart Arnold. Coopsim v0.9.9 beta 6. https://github.com/jecki/CoopSim/, 2015.

[8] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[9] Daniel Ashlock, Joseph Alexander Brown, and Philip Hingston. Multiple opponent optimization of prisoner's dilemma playing agents. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(1):53–65, 2015.

[10] Daniel Ashlock and Eun-Youn Kim. Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies. *IEEE Transactions on Evolutionary Computation*, 12(5):647–659, 2008.

[11] Wendy Ashlock and Daniel Ashlock. Changes in prisoner's dilemma strategies over evolutionary time with different population sizes. In *2006 IEEE International Conference on Evolutionary Computation*, pages 297–304. IEEE, 2006.

[12] Wendy Ashlock, Jeffrey Tsang, and Daniel Ashlock. The evolution of exploitation. In *2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 135–142. IEEE, 2014.

[13] Tsz-Chiu Au and Dana Nau. Accident or intention: that is the question (in the noisy iterated prisoner's dilemma). In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 561–568. ACM, 2006.

[14] R. Axelrod. The evolution of strategies in the iterated prisoner's dilemma. *Genetic Algorithms and Simulated Annealing*, pages 32–41, 1987.

[15] Robert Axelrod. Effective choice in the prisoner's dilemma. *Journal of Conflict Resolution*, 24(1):3–25, 1980.

[16] Robert Axelrod. More effective choice in the prisoner's dilemma. *Journal of Conflict Resolution*, 24(3):379–403, 1980.

[17] Robert Axelrod and William D Hamilton. The evolution of cooperation. *science*, 211(4489):1390–1396, 1981.

[18] Jeffrey S Banks and Rangarajan K Sundaram. Repeated games, finite automata, and complexity. *Games and Economic Behavior*, 2(2):97–117, 1990.

[19] Bruno Beaufils, Jean-Paul Delahaye, and Philippe Mathieu. Our meeting with gradual, a good strategy for the iterated prisoner's dilemma. In *Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pages 202–209, 1997.

[20] J. Bendor, R. M. Kramer, and S. Stout. When in doubt... cooperation in a noisy prisoner's dilemma. *The Journal of Conflict Resolution*, 35(4):691–719, 1991.

[21] Fabien CY Benureau and Nicolas P Rougier. Re-run, repeat, reproduce, reuse, replicate: transforming code into scientific contributions. *Frontiers in neuroinformatics*, 11:69, 2018.

[22] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[23] A Carvalho, H Rocha, F Amaral, and F Guimaraes. Iterated prisoner's dilemma-an extended analysis. *Iterated Prisoner's Dilemma-An extended analysis*, 2013.

[24] Andre LC Carvalho, Honovan P Rocha, Felipe T Amaral, and Frederico G Guimaraes. Iterated prisoner's dilemma-an extended analysis. 2013.

[25] C. Crick. A new way out of the prisoner's dilemma: Cheat. `https://spectrum.ieee.org/computing/software/a-new-way-out-of-the-prisoners-dilemma-cheat`.

[26] J.P. Delahaye. L'altruisme perfectionné. *Pour la Science (French Edition of Scientific American)*, 187:102–107, 1993.

[27] J.P. Delahaye. Logique, informatique et paradoxes, 1995.

[28] C. Donninger. *Is it Always Efficient to be Nice? A Computer Simulation of Axelrod's Computer Tournament*. Physica-Verlag HD, Heidelberg, 1986.

[29] M. M. Flood. Some experimental games. *Management Science*, 5(1):5–26, 1958.

[30] Marcus R Frean. The prisoner's dilemma without synchrony. *Proceedings of the Royal Society of London B: Biological Sciences*, 257(1348):75–79, 1994.

[31] Drew Fudenberg and Eric Maskin. The folk theorem in repeated games with discounting or with incomplete information. In *A Long-Run Collaboration On Long-Run Games*, pages 209–230. World Scientific, 2009.

[32] Marco Gaudesi, Elio Piccolo, Giovanni Squillero, and Alberto Tonda. Exploiting evolutionary modeling to prevail in iterated prisoner's dilemma tournaments. *IEEE Transactions on Computational Intelligence and AI in Games*, 8(3):288–300, 2016.

[33] Nikoleta E. Glynatsi. A data set of 45686 Iterated Prisoner's Dilemma tournaments' results, October 2019.

[34] Nikoleta E. Glynatsi and Vincent A. Knight. A bibliometric study of research topics, collaboration and influence in the field of the iterated prisoner's dilemma, 2019.

[35] Marc Harper, Vincent Knight, Martin Jones, Georgios Koutsovoulos, Nikoleta E Glynatsi, and Owen Campbell. Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma. *PloS one*, 12(12):e0188046, 2017.

[36] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[37] Christian Hilbe, Martin A Nowak, and Arne Traulsen. Adaptive dynamics of extortion and compliance. *PloS one*, 8(11):e77886, 2013.

[38] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[39] Graham Kendall, Xin Yao, and Siang Yew Chong. *The iterated prisoners' dilemma: 20 years on*, volume 4. World Scientific, 2007.

[40] Graham Kendall, Xin Yao, and Siang Yew Chong. *The iterated prisoners' dilemma: 20 years on*, volume 4. World Scientific, 2007.

[41] Vincent A. Knight, Marc Harper, Nikoleta E. Glynatsi, and Jonathan Gillard. Recognising and evaluating the effectiveness of extortion in the iterated prisoner's dilemma. *CoRR*, abs/1904.00973, 2019.

[42] David Kraines and Vivian Kraines. Pavlov and the prisoner's dilemma. *Theory and decision*, 26(1):47–79, 1989.

[43] Steven Kuhn. Prisoner's dilemma. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 2017 edition, 2017.

[44] Jiawei Li, Philip Hingston, Senior Member, and Graham Kendall. Engineering Design of Strategies for Winning Iterated Prisoner ' s Dilemma Competitions. 3(4):348–360, 2011.

[45] Jiawei Li, Graham Kendall, and Senior Member. The effect of memory size on the evolutionary stability of strategies in iterated prisoner ' s dilemma. X(X):1–8, 2014.

[46] Philippe Mathieu and Jean-Paul Delahaye. New winning strategies for the iterated prisoner's dilemma. *Journal of Artificial Societies and Social Simulation*, 20(4):12, 2017.

[47] J. H. Miller. The coevolution of automata in the repeated prisoner's dilemma. *Journal of Economic Behavior and Organization*, 29(1):87 – 112, 1996.

[48] Shashi Mittal and Kalyanmoy Deb. Optimal strategies of the iterated prisoner's dilemma problem for multiple conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 13(3):554–565, 2009.

[49] P. Molander. The optimal level of generosity in a selfish, uncertain environment. *The Journal of Conflict Resolution*, 29(4):611–618, 1985.

[50] John H Nachbar. Evolution in the finitely repeated prisoner's dilemma. *Journal of Economic Behavior & Organization*, 19(3):307–326, 1992.

[51] Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56, 1993.

[52] Martin A Nowak and Karl Sigmund. Tit for tat in heterogeneous populations. *Nature*, 355(6357):250, 1992.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[54] William H Press and Freeman J Dyson. Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences*, 109(26):10409–10413, 2012.

[55] Arthur J Robson. Efficiency in evolutionary games: Darwin, nash and the secret handshake. *Journal of theoretical Biology*, 144(3):379–396, 1990.

[56] A. Rogers, A Rogers, RK Dash, SD Ramchurn, P Vytelingum, and NR Jennings. Coordinating team players within a noisy iterated prisoner's dilemma tournament. *Theoretical computer science.*, 377(1-3):243–259, 2007.

[57] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[58] R. Selten and P. Hammerstein. Gaps in harley's argument on evolutionarily stable learning rules and in the logic of "tit for tat". *Behavioral and Brain Sciences*, 7(1):115–116, 1984.

[59] David W Stephens, Colleen M McLinn, and Jeffery R Stevens. Discounting and reciprocity in an iterated prisoner's dilemma. *Science*, 298(5601):2216–2218, 2002.

[60] Alexander J Stewart and Joshua B Plotkin. Extortion and cooperation in the prisoner's dilemma. *Proceedings of the National Academy of Sciences*, 109(26):10134–10135, 2012.

[61] E Tzafestas. Toward adaptive cooperative behavior. *From Animals to animals: Proceedings of the 6th International Conference on the Simulation of Adaptive Behavior (SAB-2000)*, 2:334–340, 2000.

[62] Unknown. www.prisoners-dilemma.com. http://www.prisoners-dilemma.com/, 2017.

[63] Pieter Van den Berg and Franz J Weissing. The importance of mechanisms for the evolution of cooperation. *Proceedings of the Royal Society B: Biological Sciences*, 282(1813):20151382, 2015.

[64] S. Walt, S. C. Colbert, and G. Varoquaux. The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[65] Jianzhong Wu and Robert Axelrod. How to cope with noise in the iterated prisoner's dilemma. *Journal of Conflict resolution*, 39(1):183–189, 1995.

6 Acknowledgements

A variety of software have been used in this work:

- The Axelrod-Python library for IPD simulations [4].
- The Matplotlib library for visualisation [38].
- The Numpy library for data manipulation [64].
- The scikit-learn library for data analysis [53].

A A summary of parameters

A summary of the parameters and features used in this work are given by Table 11.

feature	feature explanation
stochastic	If a strategy is stochastic
makes use of game	If a strategy makes used of the game information
makes use of length	If a strategy makes used of the number of turns
memory usage	The memory size of a strategy divided by the number of turns
SSE	A measure of how far a strategy is from extortionate behaviour
$C_{\max}$	The biggest cooperating rate in the tournament
$C_{\min}$	The smallest cooperating rate in the tournament
C_{median}	The median cooperating rate in the tournament
C_{mean}	The mean cooperating rate in the tournament
C_r / $C_{\max}$	A strategy's cooperating rate divided by the maximum
C_r / $C_{\min}$	A strategy's cooperating rate divided by the minimum
C_r / C_{median}	A strategy's cooperating rate divided by the median
C_r / C_{mean}	A strategy's cooperating rate divided by the mean
C_r	The cooperating ratio of a strategy
CC to C rate	The probability a strategy will cooperate after a mutual cooperation
CD to C rate	The probability a strategy will cooperate after being betrayed by the opponent
DC to C rate	The probability a strategy will cooperate after betraying the opponent
DD to C rate	The probability a strategy will cooperate after a mutual defection
p_n	The probability of a player's action being flip at each interaction
n	The number of turns
p_e	The probability of a match ending in the next turn
N	The number of strategies in the tournament
k	The number that a given tournament is repeated

Table 11: The features which are included in the performance evaluation analysis.

B List of strategies

The strategies used in this study which are from Axelrod-Python library version 3.0.0.

1. ϕ [4]
2. π [4]
3. e [4]
4. ALLCorALLD [4]
5. Adaptive [44]
6. Adaptive Pavlov 2006 [40]
7. Adaptive Pavlov 2011 [44]
8. Adaptive Tit For Tat: 0.5 [61]
9. Aggravater [4]
10. Alexei [3]
11. Alternator [17, 48]
12. Alternator Hunter [4]
13. Anti Tit For Tat [37]
14. AntiCycler [4]
15. Appeaser [4]
16. Arrogant QLearner [4]
17. Average Copier [4]
18. Backstabber [4]
19. Better and Better [1]
20. Bully [50]
21. Calculator [1]
22. Cautious QLearner [4]
23. Champion [16]
24. CollectiveStrategy [2]
25. Contrite Tit For Tat [65]
26. Cooperator [17, 48, 54]
27. Cooperator Hunter [4]
28. Cycle Hunter [4]
29. Cycler CCCCCD [4]
30. Cycler CCCD [4]
31. Cycler CCCDCD [4]
32. Cycler CCD [48]
33. Cycler DC [4]
34. Cycler DDC [48]
35. DBS [13]
36. Davis [15]
37. Defector [17, 48, 54]
38. Defector Hunter [4]
39. Double Crosser [4]
40. Desperate [63]
41. DoubleResurrection [7]
42. Doubler [1]
43. Dynamic Two Tits For Tat [4]

44. EasyGo [44, 1]
45. Eatherley [16]
46. Eventual Cycle Hunter [4]
47. Evolved ANN [4]
48. Evolved ANN 5 [4]
49. Evolved ANN 5 Noise 05 [4]
50. Evolved FSM 16 [4]
51. Evolved FSM 16 Noise 05 [4]
52. Evolved FSM 4 [4]
53. Evolved HMM 5 [4]
54. EvolvedLookerUp1 1 1 [4]
55. EvolvedLookerUp2 2 2 [4]
56. Eugine Nier [3]
57. Feld [15]
58. Firm But Fair [30]
59. Fool Me Forever [4]
60. Fool Me Once [4]
61. Forgetful Fool Me Once [4]
62. Forgetful Grudger [4]
63. Forgiver [4]
64. Forgiving Tit For Tat [4]
65. Fortress3 [11]
66. Fortress4 [11]
67. GTFT [32, 51]
68. General Soft Grudger [4]
69. Gradual [19]
70. Gradual Killer [1]
71. Grofman[15]
72. Grudger [15, 18, 19, 63, 44]
73. GrudgerAlternator [1]
74. Grumpy [4]
75. Handshake [55]
76. Hard Go By Majority [48]
77. Hard Go By Majority: 10 [4]
78. Hard Go By Majority: 20 [4]
79. Hard Go By Majority: 40 [4]
80. Hard Go By Majority: 5 [4]
81. Hard Prober [1]
82. Hard Tit For 2 Tats [60]
83. Hard Tit For Tat [62]
84. Hesitant QLearner[4]
85. Hopeless [63]
86. Inverse [4]
87. Inverse Punisher [4]
88. Joss [15, 60]
89. Knowledgeable Worse and Worse [4]
90. Level Punisher [7]
91. Limited Retaliate 2 [4]
92. Limited Retaliate 3 [4]
93. Limited Retaliate [4]
94. MEM2 [45]
95. Math Constant Hunter [4]
96. Meta Hunter Aggressive [4]
97. Meta Hunter [4]
98. Meta Majority [4]
99. Meta Majority Finite Memory [4]
100. Meta Majority Long Memory [4]
101. Meta Majority Memory One [4]
102. Meta Minority [4]
103. Meta Mixer [4]
104. Meta Winner [4]
105. Meta Winner Deterministic [4]
106. Meta Winner Ensemble [4]
107. Meta Winner Finite Memory [4]
108. Meta Winner Long Memory [4]
109. Meta Winner Memory One [4]
110. Meta Winner Stochastic [4]
111. NMWE Deterministic [4]
112. NMWE Finite Memory [4]
113. NMWE Long Memory [4]
114. NMWE Memory One [4]
115. NMWE Stochastic [4]
116. Naive Prober [44]
117. Negation [62]
118. Nice Average Copier [4]
119. Nice Meta Winner [4]
120. Nice Meta Winner Ensemble [4]
121. Nydegger [15]
122. Omega TFT [40]
123. Once Bitten [4]
124. Opposite Grudger [4]
125. PSO Gambler 1 1 1 [4]
126. PSO Gambler 2 2 2 [4]
127. PSO Gambler 2 2 2 Noise 05 [4]
128. PSO Gambler Mem1 [4]
129. Predator [11]
130. Prober [44]
131. Prober 2 [1]
132. Prober 3 [1]
133. Prober 4 [1]
134. Pun1 [11]

135. Punisher [4]
136. Raider [12]
137. Random Hunter [4]
138. Random: 0.5 [15, 61]
139. Remorseful Prober [44]
140. Resurrection [7]
141. Retaliate 2 [4]
142. Retaliate 3 [4]
143. Retaliate [4]
144. Revised Downing [15]
145. Ripoff [10]
146. Risky QLearner [4]
147. SelfSteem [24]
148. ShortMem [24]
149. Shubik [15]
150. Slow Tit For Two Tats [4]
151. Slow Tit For Two Tats 2 [1]
152. Sneaky Tit For Tat [4]
153. Soft Go By Majority [17, 48]
154. Soft Go By Majority 10 [4]
155. Soft Go By Majority 20 [4]
156. Soft Go By Majority 40 [4]
157. Soft Go By Majority 5 [4]
158. Soft Grudger [44]
159. Soft Joss [1]
160. SolutionB1 [9]
161. SolutionB5 [9]
162. Spiteful Tit For Tat [1]
163. Stalker [23]
164. Stein and Rapoport [15]
165. Stochastic Cooperator [6]
166. Stochastic WSLS [4]
167. Suspicious Tit For Tat [19, 37]
168. TF1 [4]
169. TF2 [4]
170. TF3 [4]
171. Tester [16]
172. ThueMorse [4]
173. ThueMorseInverse [4]
174. Thumper [10]
175. Tit For 2 Tats (**Tf2T**) [17]
176. Tit For Tat (**TfT**) [15]
177. Tricky Cooperator [4]
178. Tricky Defector [4]
179. Tullock [15]
180. Two Tits For Tat (**2TfT**) [17]
181. VeryBad [24]
182. Willing [63]
183. Win-Shift Lose-Stay (**WShLSt**) [44]
184. Win-Stay Lose-Shift (**WSLS**) [42, 51, 60]
185. Winner12 [46]
186. Winner21 [46]
187. Worse and Worse[1]
188. Worse and Worse 2[1]
189. Worse and Worse 3[1]
190. ZD-Extort-2 v2 [43]
191. ZD-Extort-2 [60]
192. ZD-Extort-4 [4]
193. ZD-GEN-2 [43]
194. ZD-GTFT-2 [60]
195. ZD-SET-2 [43]

C Correlation coefficients

A graphical representation of the correlation coefficients for the features in Table 8.

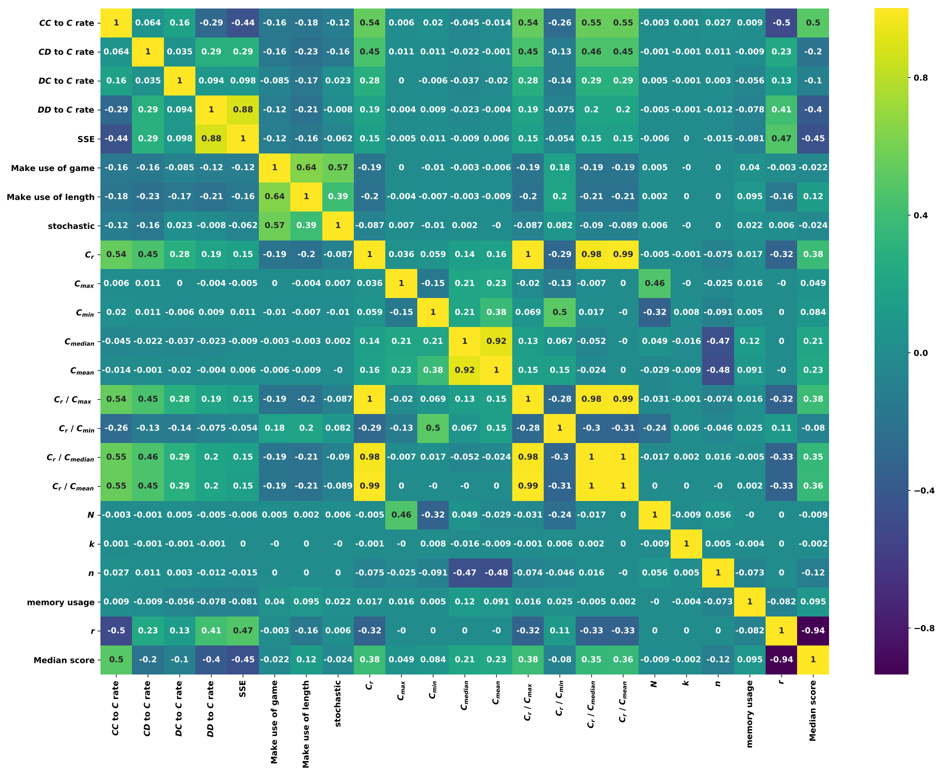


Figure 21: Correlation coefficients of features in Table 8 for standard tournaments

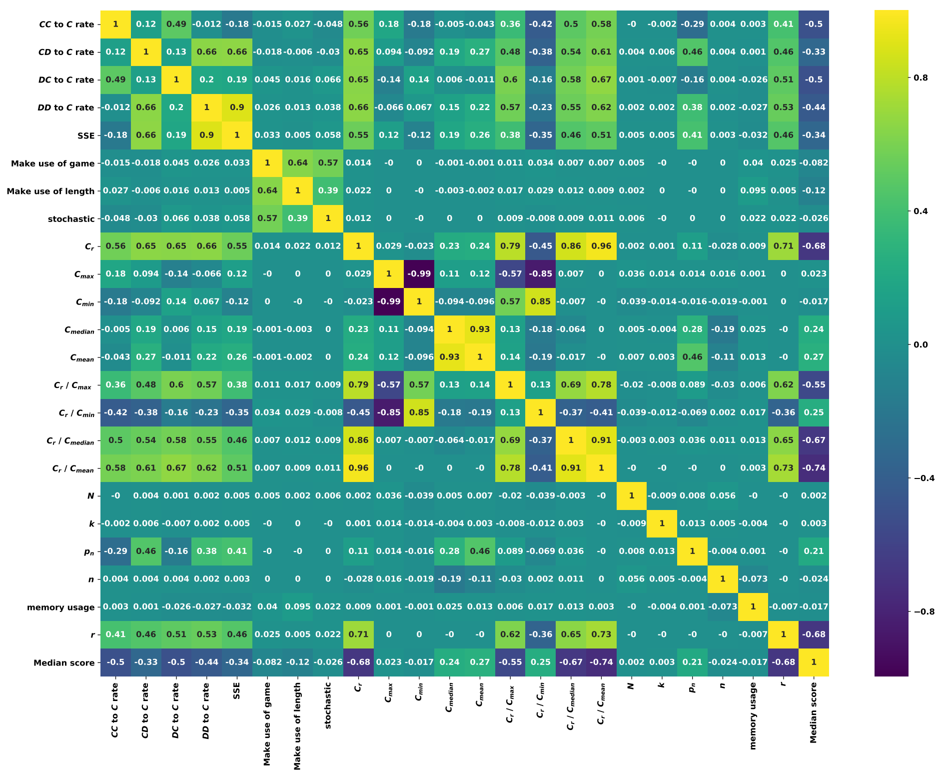


Figure 22: Correlation coefficients of features in Table 8 for noisy tournaments

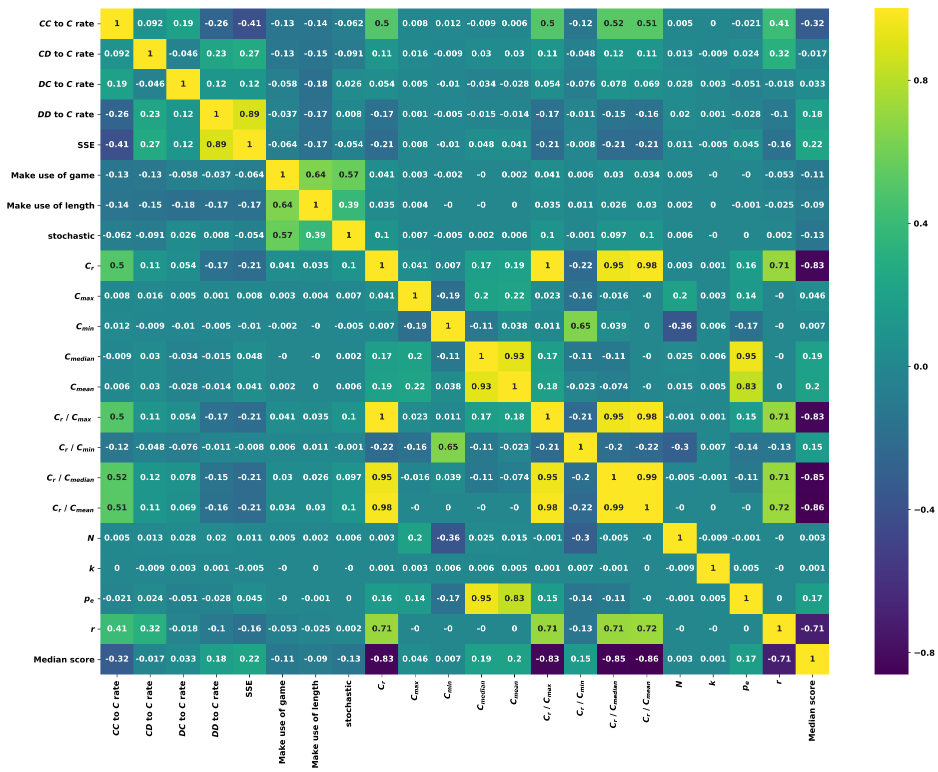


Figure 23: Correlation coefficients of features in Table 8 for probabilistic ending tournaments

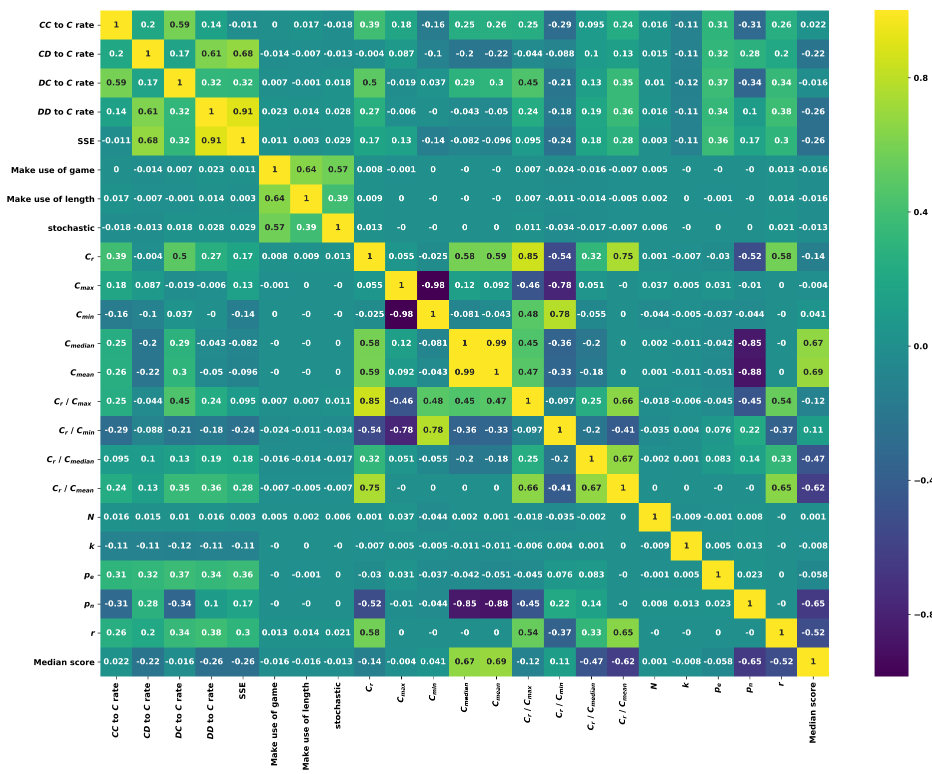


Figure 24: Correlation coefficients of features in Table 8 for noisy probabilistic ending tournaments

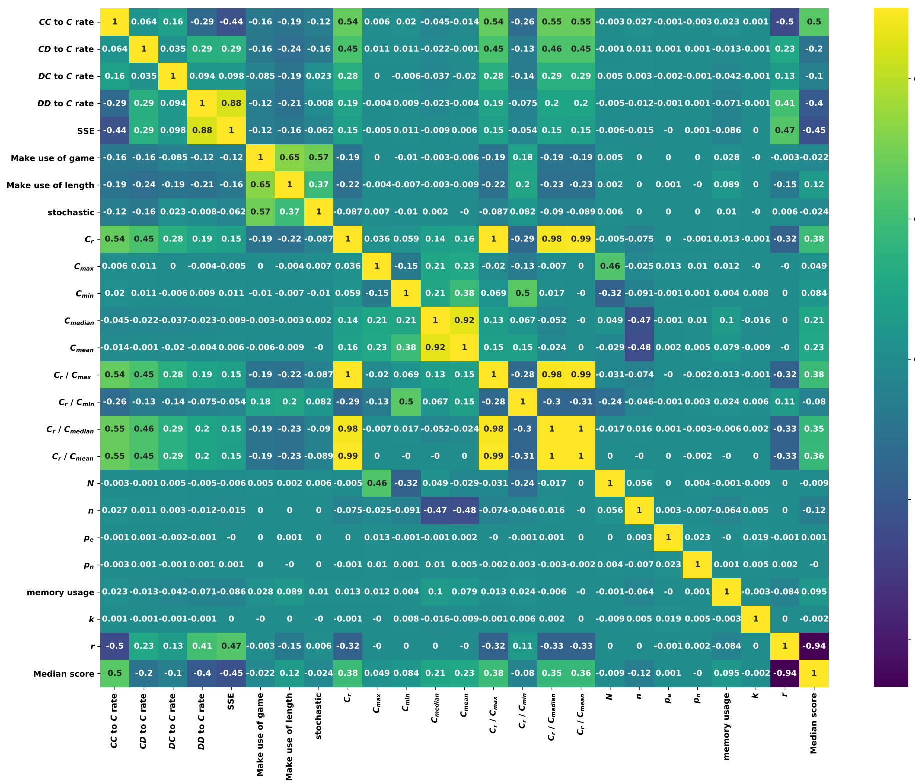


Figure 25: Correlation coefficients of features in Table 8 for data set