Properties of Winning Iterated Prisoner's Dilemma Strategies.

Nikoleta E. Glynatsi, Vincent A. Knight, Marc Harper

Abstract

Researchers have explored the performance of Iterated Prisoner's Dilemma strategies for decades: from the celebrated performance of Tit for Tat, to the introduction of the zero-determinant strategies, to the use of sophisticated learning structures such as neural networks, many new strategies have been introduced and tested in a variety of tournaments and population dynamics. Typical results in the literature, however, rely on performance against a small number of somewhat arbitrarily selected strategies in a very small number of tournaments, casting doubt on the generalizability of conclusions. We analyze a large collection of 195 typically known strategies in 45686 tournaments, present the top performing strategies across multiple tournament types, and distill their salient features. The results show that there is not yet a single strategy that performs well in diverse Iterated Prisoner's Dilemma scenarios. Nevertheless there are several properties that heavily influence the best performing strategies, refining the properties described by R. Axelrod in light of recent and more diverse opponent populations. These are: be nice, be provocable and forgiving, be a little envious, be clever, and adapt to the environment, which includes the parameters of the tournament (e.g. noise) and the population of opponents. More precisely, we find that strategies perform best when their probability of cooperation matches the total tournament population's aggregate cooperation probabilities, or a proportion thereof in the case of noisy and probabilistically ending tournaments, and that the manner in which a strategy achieves the ideal cooperation rate is crucial. The features of high performing strategies reveal why strategies such as Tit For Tat performed historically well in tournaments and why zero-determinant strategies typically do not fare well in tournament settings.

1 Background

The Iterated Prisoner's Dilemma (IPD) is a repeated two player game that models behavioural interactions, 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), given 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 defined by:

	Cooperate (C)	Defect (D)
Cooperate (C)	R	S
Defect (D)	T	P

where typically $T > R > P > S$ and $2R > T + S$. The most common values used in the literature [15] 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 [24]. Following the computer tournaments of R. Axelrod in the 1980's [12, 13], a strategy's performance in a round robin computer tournament became a common evaluation technique for newly designed strategies. Many tournaments have followed Axelrod's [18, 17, 56, 34, 57, 30] and today more hundreds of strategies exist in the literature.

The winner of both of R. Axelrod's tournaments [12, 13] was the simple strategy Tit For Tat (TFT) which cooperates on the first turn and thereafter copies the previous action of its opponent, retailiating against defections with a defection, and forgiving a defection if followed by a cooperation. R. Axelrod concluded that the strategy's robustness was due to four properties, which he adapted into four suggestions on doing well in an IPD:

- Do not be envious by striving for a payoff larger than the opponent's payoff
- Be "nice"; Do not be the first to defect
- Reciprocate both cooperation and defection; Be provocable to retaliation and forgiveness
- Do not be too clever by scheming to exploit the opponent

As a result of the strategy's strong performance in both tournaments, and moreover in a series of evolutionary experiments [15], TFT was often claimed to be a highly robust (and sometimes the most robust) strategy for the IPD.

There are strategies which have built upon TFT and the reciprocity based approach. In [17] a strategy called Gradual was introduced, constructed to have the same qualities as those of TFT with one addition. Gradual has a memory of the previous rounds of play of the game, recording the number of defections by the opponent and punishing them with a growing number of defections. It then enters a calming state in which it cooperates for two rounds. A strategy with the same intuition as Gradual is Adaptive Tit for Tat [58]. Adaptive Tit for Tat maintains a continually updated estimate of the opponent's behaviour and uses this estimate to condition its future actions.

Other work has built upon the limitations of TFT, and in some cases have shown that suggestions made by R. Axelrod did not necessarily apply in alternative environmental settings. In [18, 23, 45, 55] it was shown that TFT suffered in environments with noise. This was mainly due to the strategy's lack of generosity and contrition. Since TFT immediately punishes a defection, in a noisy environment it can get stuck in a repeated cycle of defections and cooperations. Some new strategies, more robust in tournaments with noise, were soon introduced, including Nice and Forgiving [18], Generous Tit For Tat [48], and Pavlov (aka Win Stay Lose Shift) [47], as well as later variants such as OmegaTFT [35]. The introduction of a new strategy is often accompanied by a claim that the new strategy is the best known, despite only being tested against a small number of opponents or against specific classes of opponents not necessarily representative of all possible or all published strategies. The lack of testing against formally defined strategies and tournament winners is understandable given the effort required to implement the hundreds of published IPD strategies [1], yet calls into question any claims of superiority or robustness of newly introduced strategies.

A set of envious IPD strategies were introduced called zero-determinant strategies (ZDs) in [51]. These strategies attempt to force a linear relationship between stationary payoffs against other memory-one opponents, potentally ensuring that they receive a higher average payout. While ZDs were introduced with a small tournament in which some were reportedly successful [57], this result has not generally held in future work. In [30] a series of strategies trained using reinforcement learning were introduced, and a tournament

[1]Implementing prior strategies faithfully is often extremely difficult or impossible due to insufficient descriptions and lack of published implementations or code.

containing over 200 strategies featured no ZDs ranked in top spots. Instead, the top ranked strategies were a set of "clever" (in the sense of R. Axelrod's characteristics) trained strategies based on lookup tables [14], hidden Markov models [30], and finite state automata [43]. Similarly, in [42], a set of evolutionarily-trained strategies, and a pre-selected set of known strategies, outperformed a collection of 12 ZDs.

Though only select pieces of work have been discussed, there is a broad collection of strategies in the literature, and new strategies and competitions are being published frequently [29]. The question, however, still remains the same: what is the best way to play the game?

Compared to other works, where typically a few selected or introduced strategies are evaluated on a small number of tournaments and/or small number of opponents, this manuscript evaluates the performance of 195 strategies in 45686 tournaments. Furthermore a large portion of the strategies used in this manuscript are drawn from the known and named strategies in IPD literature, including many previous tournament winners, in contrast to other work that may have randomly generated many essentially arbitrary strategies (typically restrained to a class such as memory-one strategies, or those of a certain structural form such as finite state machines or deterministic memory two strategies). Additionally, our tournaments come in a number of variations including standard tournaments emulating R. Axelrod's original tournaments, tournaments with noise, probabilistic match length, and both noise and probabilistic match length. This diversity of strategies and tournament types yields new insights and tests earlier claims in alternative settings against known powerful strategies.

The later part of the paper evaluates the impact of various strategy features on the performance of the strategies using standard statistical and machine learning techniques. These features include measures regarding a strategy's behaviour and measures regarding the tournaments. The outcomes of our work reinforce the discussion started by R. Axelrod and concludes that the properties of a successful strategy in the IPD are:

- ~~Do not be envious~~ Be a little bit envious
- Be "nice" in non-noisy environments or when game lengths are longer
- Reciprocate both cooperation and defection; Be provocable and forgiving
- ~~Do not be too clever~~ It's ok to be clever
- Adapt to the environment; Adjust to the mean population cooperation

The different tournament types as well as the data collection, made possible due to an open source library called Axelrod-Python (APL), are covered in Section 2. The data set generated for this work has been made publicly available [28] and can be used for further analysis and insights. 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, introduced in the following sections. The full set of parameters and their definitions are given in Appendix A.

2 Data collection

The data set generated for this manuscript was created with APL version 3.0.0. APL allows for different types of IPD computer tournaments to be simulated and contains a large list of strategies. Most of these are strategies described in the literature with a few exceptions of strategies that have been contributed specifically to the package. This paper makes use of 195 strategies implemented in version 3.0.0. A list of the strategies is given in the Appendix E. 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 R. Axelrod's well-known tournaments [12]. 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 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 [4, 19] and is available here.

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*;
```

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.

A result summary (Table 2) has N number of 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), and the rate of which the strategy cooperated after each state. The **normalised rank** feature that is manually added. The rank R of a given. strategy can vary between 0 and $N-1$. Thus, the normalised rank, denoted as r, is calculated as a strategy's rank divided by $N-1$.

						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

The performance of each strategy is evaluated in four tournament types, as 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 TFT 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. The data collection process was designed such that the probabilities of noise and ending of the match varied between 0 and 1. However, commonly used values for these probabilities are values less than 0.1. Thus, Table 3 also includes the top 15 strategies in noisy tournaments with $p_n < 0.1$ and probabilistic ending tournaments with $p_e < 0.1$.

The r distributions for the top ranked strategies of Table 3 are given by Figure 2.

In standard tournaments 10 out of the 15 top strategies are introduced in [30]. 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 a subset of the strategies in APL in a standard tournament, thus their performance in the specific setting was anticipated. DoubleCrosser, BackStabber and Fool Me Once, are strategies not from the literature but from the APL. DoubleCrosser is an extension of BackStabber and both strategies make use of the number of turns because they are set to defect on the last two rounds. It should be noted that these strategies can be characterised as "cheaters" because the source code of the strategies allows them to know the number of turns in a match (unless the match has a probabilistic ending). These strategies were expected to not perform

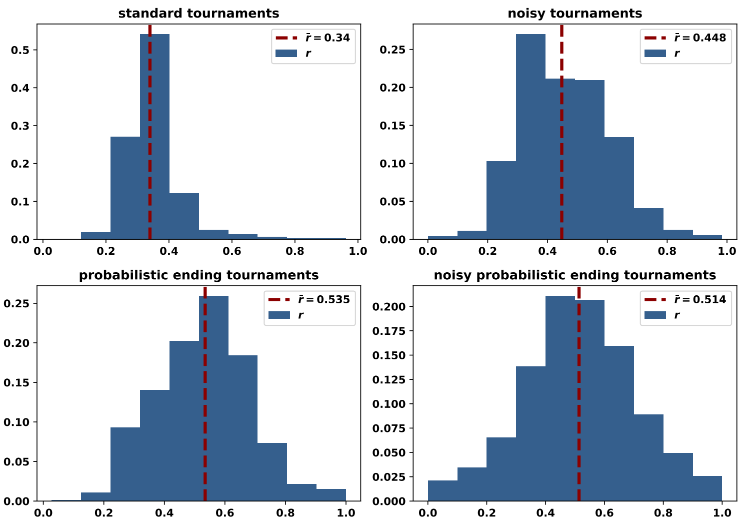
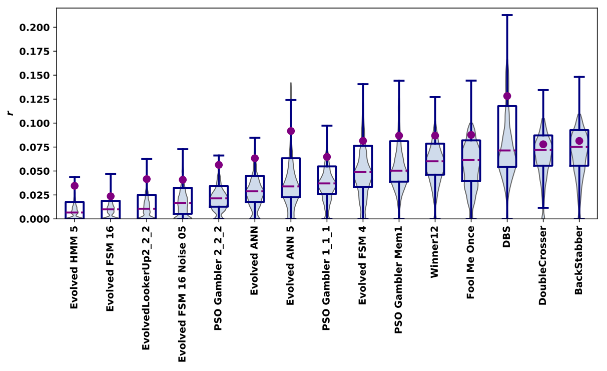


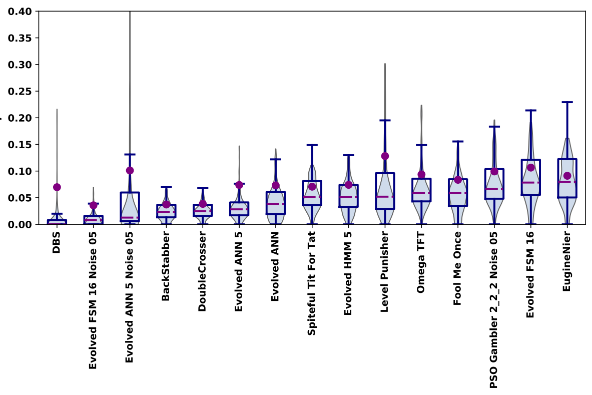
Figure 1: TFT's r distribution in tournaments. Lower values of r correspond to better performances. The best performance of the strategy has been in standard tournaments where it achieved a $\bar{r}$ of 0.34.

	Standard		Noisy		Noisy ($p_n < 0.1$)		Probabilistic ending		Probabilistic ending ($p_e < 0.1$)		Noisy probabilistic ending	
	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$	Name	$\bar{r}$
0	Evolved HMM 5	0.007	Grumpy	0.140	DBS	0.000	Fortress4	0.013	Evolved FSM 16	0.000	Alternator	0.304
1	Evolved FSM 16	0.010	e	0.194	Evolved FSM 16 Noise 05	0.008	Defector	0.014	Evolved FSM 16 Noise 05	0.013	ϕ	0.310
2	EvolvedLookerUp2 2 2	0.011	Tit For 2 Tats	0.206	Evolved ANN 5 Noise 05	0.013	Better and Better	0.016	MEM2	0.027	e	0.312
3	Evolved FSM 16 Noise 05	0.017	Slow Tit For Two Tats	0.210	BackStabber	0.024	Tricky Defector	0.019	Evolved HMM 5	0.044	π	0.317
4	PSO Gambler 2 2 2	0.021	Cycle Hunter	0.215	DoubleCrosser	0.025	Fortress3	0.022	EvolvedLookerUp2 2 2	0.049	Limited Retaliate	0.353
5	Evolved ANN	0.029	Risky QLearner	0.222	Evolved ANN 5	0.028	Gradual Killer	0.025	Spiteful Tit For Tat	0.060	Anti Tit For Tat	0.354
6	Evolved ANN 5	0.034	Retaliate 3	0.229	Evolved ANN	0.038	Aggravater	0.028	Nice Meta Winner	0.068	Limited Retaliate 3	0.356
7	PSO Gambler 1 1 1	0.037	Cycler CCCCCD	0.235	Spiteful Tit For Tat	0.051	Raider	0.031	NMWE Finite Memory	0.069	Retaliate 3	0.356
8	Evolved FSM 4	0.049	Retaliate 2	0.239	Evolved HMM 5	0.051	Cycler DDC	0.045	NMWE Deterministic	0.070	Retaliate	0.357
9	PSO Gambler Mem1	0.050	Defector Hunter	0.240	Level Punisher	0.052	Hard Prober	0.051	Grudger	0.070	Retaliate 2	0.358
10	Winner12	0.060	Retaliate	0.242	Omega TFT	0.059	SolutionB1	0.060	NMWE Long Memory	0.074	Limited Retaliate 2	0.361
11	Fool Me Once	0.061	Hard Tit For 2 Tats	0.250	Fool Me Once	0.059	Meta Minority	0.061	Nice Meta Winner Ensemble	0.076	Hopeless	0.368
12	DBS	0.071	Limited Retaliate 3	0.253	PSO Gambler 2 2 2 Noise 05	0.067	Bully	0.061	EvolvedLookerUp1 1 1	0.077	Arrogant QLearner	0.407
13	DoubleCrosser	0.072	ShortMem	0.253	Evolved FSM 16	0.078	EasyGo	0.071	NMWE Memory One	0.080	Cautious QLearner	0.409
14	BackStabber	0.075	Limited Retaliate	0.257	EugineNier	0.080	Fool Me Forever	0.071	Winner12	0.085	Fool Me Forever	0.418

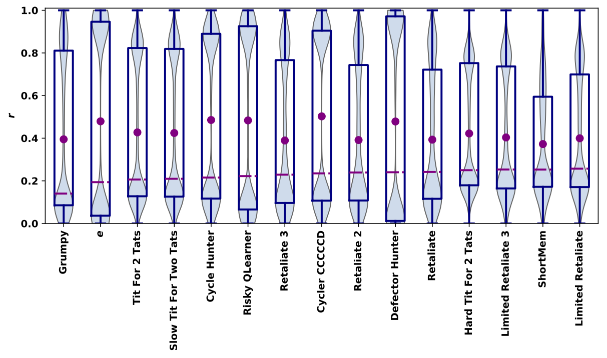
Table 3: Top performances for each tournament type based on $\bar{r}$. The results of each type are based on 11420 unique tournaments of each type. The results for noisy tournaments with $p_n < 0.1$ are based on 1151 tournaments, and for probabilistic ending tournaments with $p_e < 0.1$ on 1139. The top ranks indicate that trained strategies perform well in a variety of environments, but so do simple deterministic strategies. The normalised medians are close to 0 for most environments, except environments with noise not restricted to 0.1 regardless the number of turns. Noisy and noisy probabilistic ending tournaments have the highest medians. This implies that strategies from the collection of this work do not perform well in environments with high values of noise.



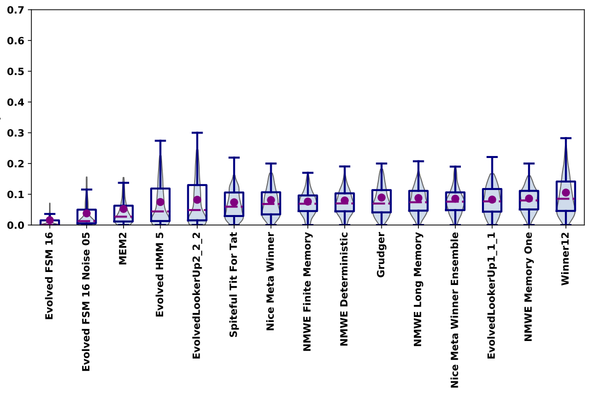
(a) r distributions of top 15 strategies in standard tournaments.



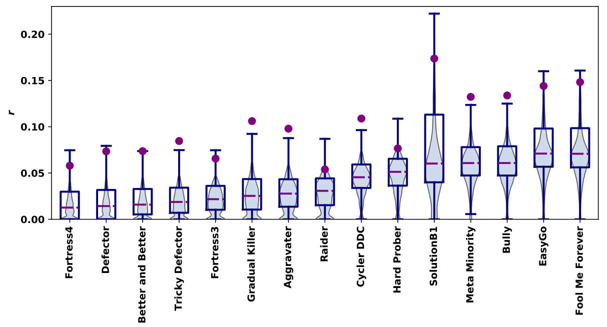
(b) r distributions of top 15 strategies in noisy tournaments with $p_n < 0.1$.



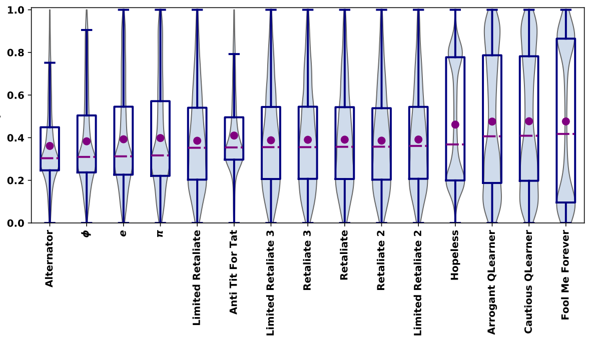
(c) r distributions of top 15 strategies in noisy tournaments.



(d) r distributions of top 15 strategies in 1139 probabilistic ending tournaments with $p_e < 0.1$.



(e) r distributions of top 15 strategies in probabilistic ending tournaments.



(f) r distributions of top 15 strategies in noisy probabilistic ending tournaments.

Figure 2: r distributions of the top 15 strategies in different environments. A lower value of $\bar{r}$ corresponds to a more successful performance. A strategy's r distribution skewed towards zero indicates that the strategy ranked highly in most tournaments it participated in. Most distributions are skewed towards zero except the distributions with unrestricted noise, supporting the conclusions from Table 3.

as well in tournaments where the number of turns is not specified. Finally, Winner 12 [42] and DBS [11] are both from the literature. DBS is a strategy specifically designed for noisy environments, however, it ranks highly in standard tournaments as well. Similarly the fourth ranked player, Evolved FSM 16 Noise 05, was trained for noisy tournaments yet performs well in standard tournaments. Figure 2a shows that these strategies typically perform well in any standard tournament in which they participate.

In the case of noisy tournaments with smaller noise $p_n < 0.1$ the top performed strategies include strategies specifically designed for noisy tournaments. These are DBS, Evolved FSM 16 Noise 05, Evolved ANN 5 Noise 05, PSO Gambler 2 2 2 Noise 05 and Omega Tit For Tat [35]. Omega TFT, a strategy designed to break the deadlocking cycles of CD and DC that TFT can fall into in noisy environments, places 10th. The rest of the top ranks are occupied by strategies which performed well in standard tournaments and deterministic strategies such as Spiteful Tit For Tat [1], Level Punisher [3], Eugine Nier [50]. Similarly to standard tournaments, the successful strategies in this given setting performed well overall the tournaments they participated in (see Figure 2b).

In contrast, the performance of the top ranked strategies in noisy environments when $p_n \in [0, 1]$ is bimodal. The top strategies include strategies which decide their actions based on the cooperation to defection ratio, such as ShortMem [22], Grumpy [52] and e [52], and the Retaliate strategies which are designed to defect if the opponent has tricked them more often than a given percentage of the times that they have done the same. The bimodality of the r distributions is explained by Figure 3 which demonstrates that the top 6 strategies were highly ranked due to the their performance in tournaments with $p_n > 0.5$, and that in tournaments with a noise probability lower than 0.5 they performed poorly. At a noisy level of 0.5 or greater, mostly cooperative strategies become mostly defectors and vice versa.

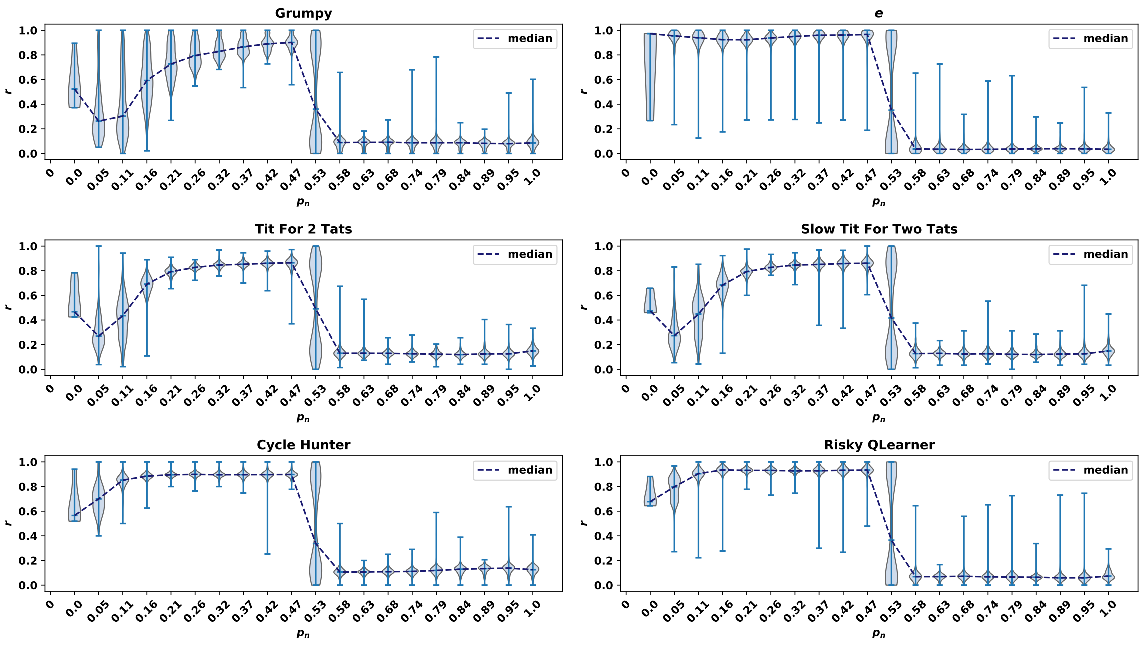


Figure 3: Normalised rank r distributions for top 6 strategies in noisy tournaments over the probability of noisy (p_n).

The new entrants to the most effective strategies list in probabilistic ending tournaments with $p_e < 0.1$ are a series of ensembled Meta strategies, trained strategies which performed well in standard tournaments, and Grudger [52] and Spiteful Tit for Tat [1]. The Meta strategies [52] utilize a team of strategies and aggregate the potential actions of the team members into a single action in various ways. Figure 2d indicates that these strategies performed well in any probabilistic ending tournament they competed in.

In probabilistic ending tournaments with $p_e \in [0, 1]$ the top ranks are mostly occupied by defecting strategies such as Better and Better, Gradual Killer, Hard Prober (all from [52]), Bully (Reverse Tit For Tat) [46] and Defector, and a series of strategies based on finite state automata introduced by Daniel Ashlock and Wendy Ashlock: Fortress 3, Fortress 4 (both introduced in [9]), Raider [10] and Solution B1 [10]. The success of defecting strategies in probabilistic ending tournaments is due to larger values of p_e which lead to shorter matches (the expected number of rounds is $1/p_e$), so the impact of the PD being iterated is subdued. This is captured by the Folk Theorem [26] as defecting strategies do better when the likelihood of the game ending in the next turn increases. This is demonstrated by Figure 4, which gives the distributions of r for the top 6 strategies in probabilistic ending tournaments over p_e.

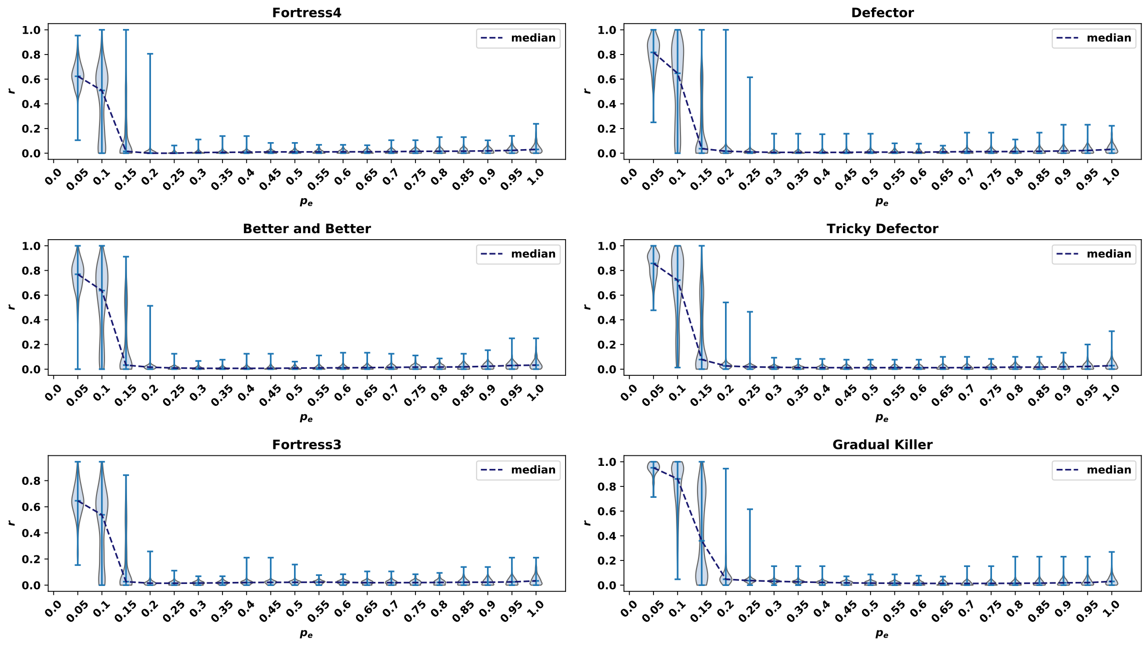


Figure 4: Normalised rank r distributions for top 6 strategies in probabilistic ending tournaments over p_e. The 6 strategies start of with a high median rank, however, their ranked decreased as the the probability of the game ending increased and at the point of $p_e = 0.1$.

The top performances in tournaments with both noise and a probabilistic ending and the top performances over the entire data set have the largest median values compared to the top rank strategies of the other tournament types, Figure 2f and Figure 5. The $\bar{r}$ for the top strategy is approximately at 0.3, indicating that the most successful strategy can on average just place at the top 30% of the competition.

On the whole, the analysis of this manuscript has shown that:

- In standard tournaments the dominating strategies were strategies that had been trained using reinforcement learning techniques.
- In noisy environments where the noise probability strictly less than 0.1 was considered, the successful strategies were strategies specifically designed or trained for noisy environments.
- In probabilistic ending tournaments most of the highly ranked strategies were defecting strategies and trained finite state automata, all by the authors of [9, 10]. These strategies ranked high due to their

Name	$\bar{r}$
Limited Retaliate 3	0.286
Retaliate 3	0.296
Retaliate 2	0.302
Limited Retaliate 2	0.303
Limited Retaliate	0.310
Retaliate	0.317
BackStabber	0.324
DoubleCrosser	0.331
Nice Meta Winner	0.349
PSO Gambler 2 2 2 Noise 05	0.351
Grudger	0.352
Evolved HMM 5	0.357
NMWE Memory One	0.357
Nice Meta Winner Ensemble	0.359
Forgetful Fool Me Once	0.359

Table 4: 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. PSO Gambler and Evolved HMM 5 are trained strategies introduced in [30] and Nice Meta Winner and NMWE Memory One are strategies based on teams. Grudger is a strategy from R. Axelrod's original tournament and Forgetful Fool Me Once is based on the same approach as Grudger.

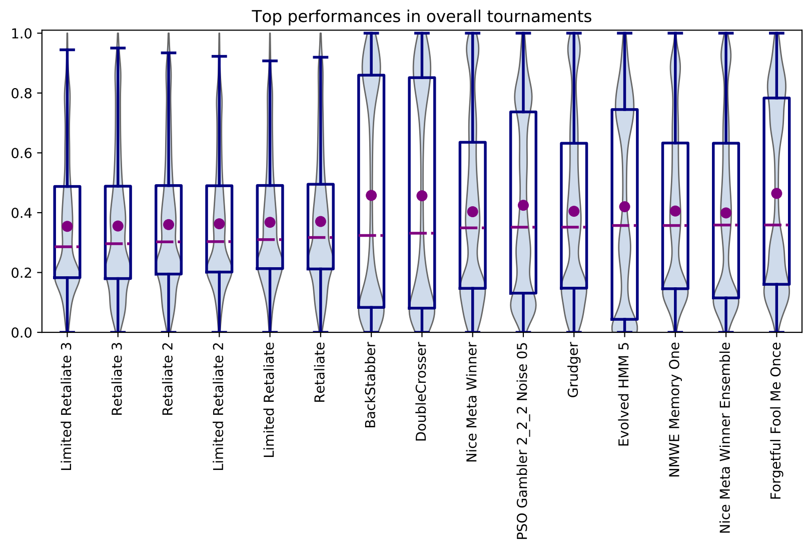


Figure 5: Normalised rank r distributions for best performed strategies in the data set [28]. A lower value of $\bar{r}$ corresponds to a more successful performance.

performance in tournaments where the probability of the game ending after each turn was bigger than 0.1.

- In probabilistic tournaments with p_e less than 0.1 the highly ranked strategies were strategies based on the behaviour of others.
- From the collection of strategies considered here, no strategy can be consistently successful in noisy environments, except if the value of noise is constrained to less than a 0.1.

Though there is not a single strategy that repeatably outranks all others in any of the distinct tournament types, or even across the tournament types, there are specific types of strategies have been repeatably ranked in the top ranks. These have been strategies that have been trained, strategies that retailiate, and strategies that would adapt their behaviour based on preassigned rules to achieve the highest outcome. These results contradict some of R. Axelrod's suggestions, and more specifically, the suggestions 'Do not be clever' and 'Do not be envious'. We dig deeper into the crucial strategy features for success in the following sections.

4 Evaluation of performance

For each strategy we have a variety of features, described in Table 5. These features are measures regarding a strategy's behaviour from the tournaments the strategies competed in as well as intrinsic properties such as whether a strategy is deterministic or stochastic.

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 [36]	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_{min} / C_r	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 5: The features which are included in the performance evaluation analysis. Stochastic, makes use of length and makes use of game are APL classifiers that 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 is calculated as the number of turns the strategy considers to make an action (which is specified in the APL) divided by the number of turns. The SSE (introduced in [36]) 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 as the sum of squared error (SSE). A SSE value of 1 indicates no extortionate behaviour at all whereas a value of 0 indicates that a strategy is behaving as 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 (C_{min}), maximum (C_{max}), mean (C_{mean}) and median (C_{median}) cooperating ratios of each tournament.

The memory usage of strategies is the number of rounds of play used by the strategy divided by the number of turns in each match. For example, Winner12 uses the previous two rounds of play, and if participating in a match with 100 turns its memory usage would be 2/100. For strategies with an infinite memory size, for example Evolved FSM 16 Noise 05, memory usage is equal to 1. Note that 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 correlation coefficients between the features of Table 5 the median score and the median normalised rank are given by Table 6. The correlation coefficients between all features of Table 5 have been calculated and a graphical representation can be found in the Appendix B.

	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	0.108	0.081
CD to C rate	0.226	-0.199	0.456	-0.330	0.320	-0.017	0.205	-0.220	0.281	-0.177
C_r	-0.323	0.384	0.711	-0.678	0.714	-0.832	0.579	-0.135	0.360	-0.124
C_r / C_{max}	-0.323	0.381	0.616	-0.551	0.714	-0.833	0.536	-0.116	0.395	-0.265
C_r / C_{mean}	-0.331	0.358	0.731	-0.740	0.721	-0.861	0.649	-0.621	0.428	-0.439
C_r / C_{median}	-0.331	0.353	0.652	-0.669	0.712	-0.852	0.330	-0.466	0.294	-0.405
C_r / C_{min}	0.109	-0.080	-0.358	0.250	-0.134	0.150	-0.368	0.113	0.000	0.280
C_{max}	-0.000	0.049	0.000	0.023	-0.000	0.046	0.000	-0.004	-0.000	0.553
C_{mean}	-0.000	0.229	-0.000	0.271	0.000	0.200	0.000	0.690	-0.000	0.544
C_{median}	0.000	0.209	-0.000	0.240	-0.000	0.187	-0.000	0.673	0.000	-0.250
C_{min}	0.000	0.084	0.000	-0.017	-0.000	0.007	-0.000	0.041	-0.161	-0.190
DC to C rate	0.127	-0.100	0.509	-0.504	-0.018	0.033	0.341	-0.016	0.173	-0.088
DD to C rate	0.412	-0.396	0.533	-0.436	-0.103	0.176	0.378	-0.263	0.237	-0.239
N	0.000	-0.009	-0.000	0.002	-0.000	0.003	-0.000	0.001	-0.000	-0.001
k	0.000	-0.002	-0.000	0.003	-0.000	0.001	-0.000	-0.008	0.000	-0.001
n	0.000	-0.125	-0.000	-0.024	-	-	-	-	0.000	-0.074
p_e	-	-	-	-	0.000	0.165	0.000	-0.058	0.000	0.055
p_n	-	-	-0.000	0.207	-	-	-0.000	-0.650	-0.000	-0.256
Make use of game	-0.003	-0.022	0.025	-0.082	-0.053	-0.108	0.013	-0.016	-0.004	-0.053
Make use of length	-0.158	0.124	0.005	-0.123	-0.025	-0.090	0.014	-0.016	-0.041	-0.026
SSE	0.473	-0.452	0.463	-0.337	-0.156	0.223	0.305	-0.259	0.233	-0.167
memory usage	-0.082	0.095	-0.007	-0.017	-	-	-	-	-0.053	0.046
stochastic	0.006	-0.024	0.022	-0.026	0.002	-0.130	0.021	-0.013	0.013	-0.048

Table 6: Correlations table between the features of Table 5 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 (smaller rank is better), 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 generally pays off that does not hold against defective strategies. Being more cooperative after a mutual defection, that is not retaliating, is associated to lesser overall success in terms of normalised rank. Figure 6 confirms that the winners of standard tournaments always cooperate after a mutual cooperation and almost always defect after a mutual defection.

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 correlation coefficients have larger values, indicating a stronger effect. Thus a strategy will be punished more by its cooperative behaviour in probabilistic ending environments, supporting the results of Section 4 as well. The distributions of the C_r of the winners in both tournaments are given by Figure 7. It confirms that the winners in noisy tournaments cooperated less than 35% of the time and in probabilistic ending tournaments less than 10%. In noisy probabilistic ending tournaments and over all the tournaments' results, the only features that had a moderate effect are $C_r/C_{\text{mean}}, C_r/C_{\text{max}}$ and C_r. In such environments cooperative behaviour appears to be punished less than in noisy and probabilistic ending tournaments.

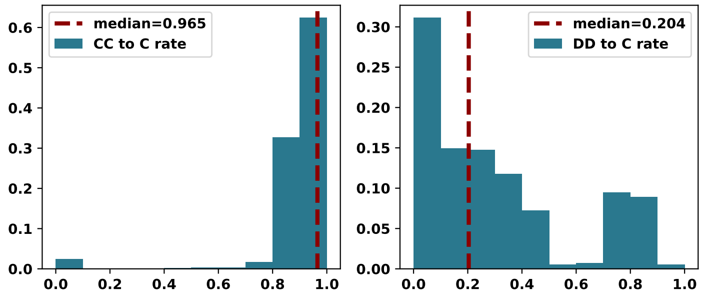


Figure 6: Distributions of CC to C and DD to C for the winners in standard tournaments.

Moreover, the manner in which a strategy achieves a given cooperation rate relative to the tournament population average is important. Playing a strategy that randomly cooperates with C_{mean} is unlikely to be as effective as it will randomly have cooperations matched with defections, and it may be punished for defections against cooperations by retailiating opponents. In contrast, TFT naturally achieves a cooperation rate near C_{mean} by virtue of copying its opponent's last move while also minimizing instances where it is exploited by an opponent (cooperating while the opponent defects), at least in non-noisy tournaments. [2] TFT forces the opponent to pay back the (C, D) round with a (D, C) round before returning to mutual cooperation. This explains why TFT performed well in Axelrod's original tournaments as most strategies submitted to those tournaments were typically cooperative and relatively few strategies played with an easily exploitable pattern. TFT does not appear in the top ranks in tournaments with more aggressive strategies because it is too nice. Strategies like Grudger will always defect after a fixed number of opponent defections, which allows them to effectively exploit strategies like Alternator or stochastic strategies that have a non-zero chance of cooperating after mutual defection, which TFT will not do. Moreover in a noisy environment these strategies will naturally tend toward always defecting, leading them to exploit strategies like Cooperator. In such a scenario the noisy environment effectively voids R. Axelrod's rule to be nice, allowing strategies to attempt exploitation, whereas in a noise-free environment, attempting exploitation can be risky if strategies that exhibit a Grudger-like behavior are present, reducing the overall value in attempting to exploit strategies like Cooperator.

Similarly, these results suggest an explanation regarding the intuitively unexpected effectiveness of memory-one strategies historically. Given that among the important features associated with success are the relative cooperation rate to the population average and the four memory-one probabilities of cooperating conditional on the previous round of play, these features can be optimized by a memory-one strategy such as TFT. Usage of more history becomes valuable when there are exploitable opponent patterns. This is indicated by the importance of SSE as a feature, showing that the first-approximation provided by a memory-one strategy is no longer sufficient.

A multivariate linear regression has been fitted to model the relationship between the features and the normalised rank. Based on the graphical representation of the correlation matrices given in Appendix B several of the features are highly correlated and have been removed before fitting the linear regression model. The features included are given by Table 7 alongside their corresponding p values in the distinct tournaments and their regression coefficients.

A multivariate linear regression has also be fitted on the median score. The coefficients and p values of the features can be found in Appendix D. This approach leads to similar conclusions.

The feature C_r/C_{mean} has a statistically significant effect across all models and a high regression coefficient. It

[2]This also explains why Tit For N Tats does not fare well for $N > 1$ – it fails to achieve the proper cooperation ratio by tolerating too many defections.

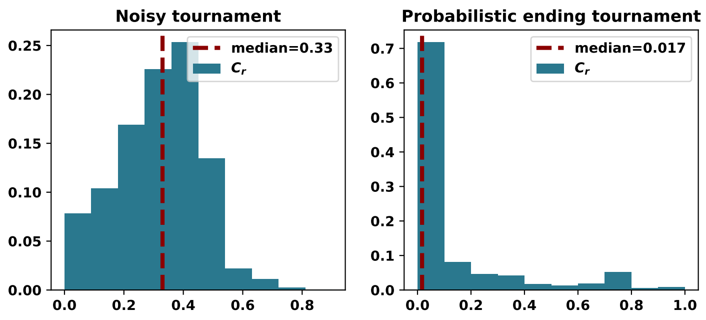


Figure 7: C_r distributions of the winners in noisy and in probabilistic ending tournaments.

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending		Overall	
	R adjusted: 0.541		R adjusted: 0.639		R adjusted: 0.587		R adjusted: 0.577		R adjusted: 0.242	
	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
CC to C rate	-0.042	0.000	-0.007	0.000	0.017	0.000	0.111	0.0	-0.099	0.0
CD to C rate	0.297	0.000	-0.068	0.000	0.182	0.000	0.023	0.0	0.129	0.0
C_r / C_{max}	-	-	1.856	0.000	-	-	1.256	0.0	-	-
C_r / C_{mean}	-0.468	0.000	-0.577	0.000	0.525	0.000	-0.120	0.0	0.300	0.0
C_{max}	-0.071	0.000	-	-	-0.022	0.391	1.130	0.0	-	-
C_{mean}	0.118	0.000	-2.558	0.000	-0.023	0.001	-1.489	0.0	-	-
C_{min}	-0.161	0.000	-1.179	0.000	-0.170	0.000	-	-	-	-
C_{min} / C_r	0.057	0.000	-0.320	0.000	0.125	0.000	-	-	-0.103	0.0
DC to C rate	0.198	0.000	0.040	0.000	-0.030	0.000	0.022	0.0	0.064	0.0
k	0.000	0.319	0.000	0.020	0.000	0.002	0.000	0.0	-	-
n	0.000	0.000	-	-	-	-	-	-	-	-
p_e	-	-	-	-	0.000	0.847	-0.083	0.0	-	-
p_n	-	-	-0.048	0.000	-	-	-	-	-	-
SSE	0.258	0.000	0.153	0.000	-0.041	0.000	0.100	0.0	0.056	0.0
constant	0.697	0.000	1.522	0.000	-0.057	0.019	-0.472	0.0	0.178	0.0
memory usage	-0.010	0.000	-0.000	0.035	-	-	-	-	-	-

Table 7: Results of multivariate linear regressions with r as the dependent variable. R squared is reported for each model.

has both a positive and negative impact on the normalised rank depending on the environment. For standard tournaments, Figure 8 gives the distributions of several features for the winners of standard tournaments. The C_r/C_{mean} distribution of the winner is also given in Figure 8. A value of $C_r/C_{\text{mean}} = 1$ implies that the cooperating ratio of the winner was the same as the mean cooperating ratio of the tournament, and in standard tournaments, the median is 1. Therefore, an effective strategy in standard tournaments was the mean cooperator of its respective tournament.

The distributions of SSE and CD to D rate for the winners of standard tournaments are also given in Figure 8. The SSE distributions for the winners indicate that the strategy behaved in a ZD way in several tournaments, however, not constantly. The winners participated in matches where they did not try to extortionate their opponents. Furthermore, the CD to D distribution indicates that if a strategy were to defect against the winners they would reciprocate on average with a probability of 0.5.

Similarly for the rest of the different tournaments types, and the entire data set the distributions of C_r/C_{mean}, SSE and CD to C ratio are given by Figures 9, 11, 12 and 13.

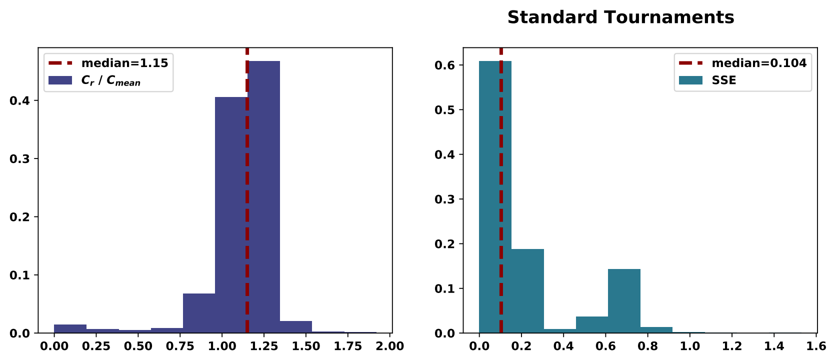


Figure 8: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of standard tournaments. A value of $C_r/C_{\text{mean}} = 1$ imply that the cooperating ratio of the winner was the same as the mean cooperating ratio of the tournament. An SSE distribution skewed towards 0 indicates a extortionate behaviour by the strategy.

Based on the C_r/C_{mean} distributions the successful strategies have adapted differently to the mean cooperator depending on the tournament type. In noisy tournaments where the median of the distribution is at 0.67, and thereupon the winners cooperated 67% of the time the mean cooperator did. In tournaments with noise and a probabilistic ending the winners cooperated 60%, whereas in settings that the type of the tournament can vary between all the types the winners cooperated 67% of the time the mean cooperator did. Lastly, in probabilistic ending tournaments above more defecting strategies prevail (Section 3), and this result is reflected here.

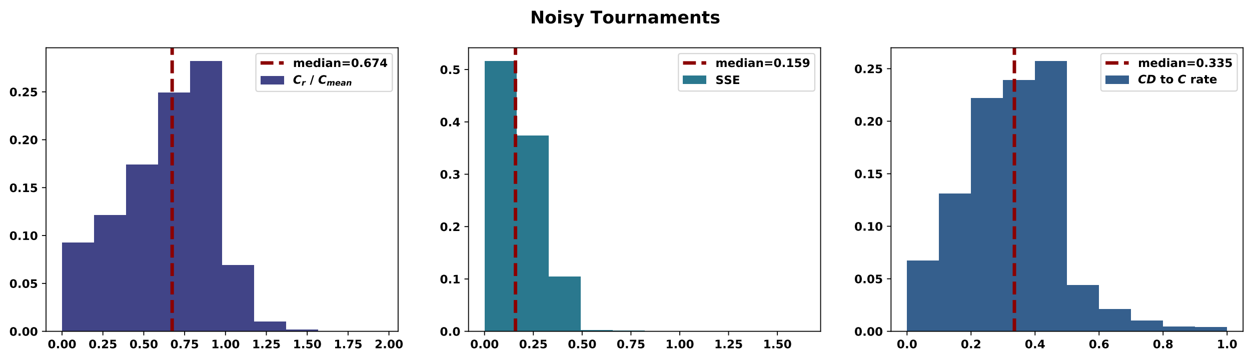
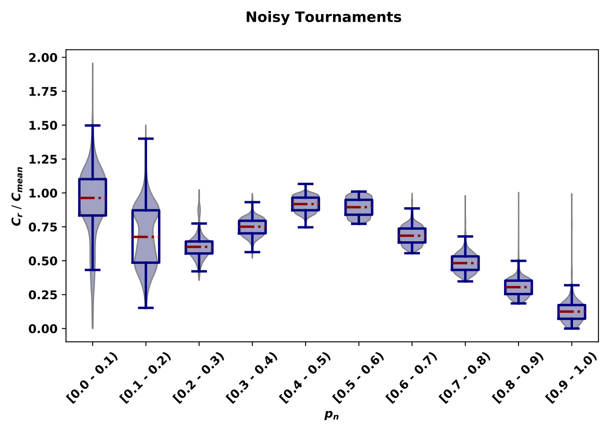


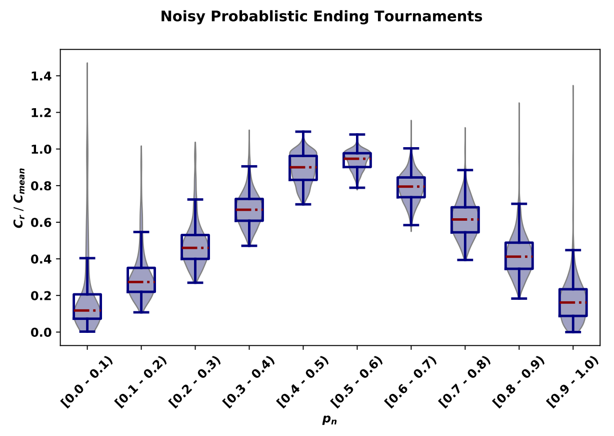
Figure 9: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of noisy tournaments.

The probability of noise has been observed to substantially affect optimal behaviour. Figure 10 gives the ratio C_r/C_{mean} for the winners in tournaments with noise, over the probability of noise. From Figure 10a it is clear that the cooperating only 67% of the time the mean cooperator did is optimal only when $p_n \in [0.2, 0.4)$ and $p_n \in [0.6, 0.7]$. In environments with $p_n < 0.1$ the winners want to be close to the mean cooperator, similarly to standard tournaments, and as the probability of noise is exceeding 0.5 (where the game is effectively inverted) strategies should aim to be less and less cooperative.

Figure 10 gives C_r/C_{mean} for the winners over p_n in tournaments with noise and a probabilistic ending. The optimal proportions of cooperations are different now that the number of turns is not fixed, successful strategies want to be more defecting that the mean cooperator, that only changes when p_n approaches 0.5. Figure 10 demonstrates how the adjustments to C_r/C_{mean} change over the noise in the to the environment, and thus supports how important adapting to the environment is for a strategy to be successful.



(a) C_r/C_{mean} distribution for winners in noisy tournaments over p_n.



(b) C_r/C_{mean} distribution for winners in noisy probabilistic ending tournaments over p_n.

Figure 10: C_r/C_{mean} distributions over intervals of p_n. These distributions model the optimal proportion of cooperation compared to C_{mean} as a function of (p_n).

The distributions of the SSE across the tournament types suggest that successful strategies exhibit some extortionate behaviour, but not constantly. ZDs are a set of strategies that are often envious as they try to exploit their opponents. The winners of the tournaments considered in this work are envious, but not as much as many ZDs. This highlights why TFT's early tournament success fails to generalize – it never attempts to defect against a cooperating or exploitable opponent (e.g. Alternator). Moreover, many of the strategies in the library will not tolerate exploitation attempts. A clever strategy can achieve mutual cooperation with stronger strategies while also being able to exploit weaker strategies. This is why ZDs fail to appear in the tops ranks – they try to exploit all opponents and cannot actively adapt back to mutual cooperation against stronger strategies, which requires more depth of memory. [3]

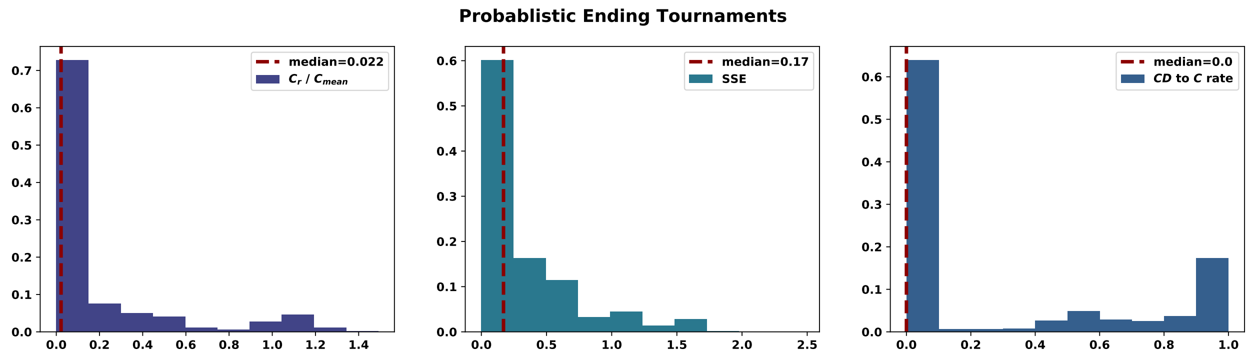


Figure 11: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of probabilistic ending tournaments.

The distributions of the CD to C rate evaluate the behaviour of a successful strategy after its opponent has defected against it. In standard tournaments it was observed that a successful strategy reciprocates with a probability of 0.5. This is distinct between the tournament types. In tournaments with noise a strategy is less likely to cooperate following a defection compared to standard tournaments, and in probabilistic ending

[3]Note that ZDs also tend to perform poorly in population games for a similar reason: they attempt to exploit other players using ZDs, failing to form a cooperative subpopulation. This makes them good invaders but poor resisters of invasion.

tournaments a strategy will reciprocate a defection. In a setting that the type of the tournament can vary between all the examined types a winning strategy would reciprocate on average with a probability of 0.58.

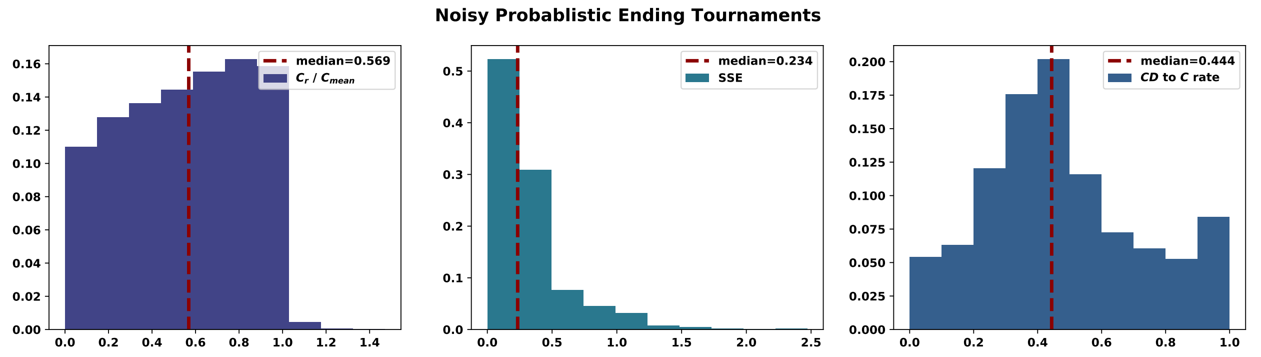


Figure 12: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners of noisy probabilistic ending tournaments.

Further statistically significant features with strong effects include $C_r/C_{\min}$, $C_r/C_{\max}$, $C_{\min}$ and $C_{\max}$. These add more emphasis on how important it is for a a strategy to adapt to its environment. Finally, the features number of turns, repetitions and the probabilities of noise and the game ending had no significant effects based on the multivariate regression models.

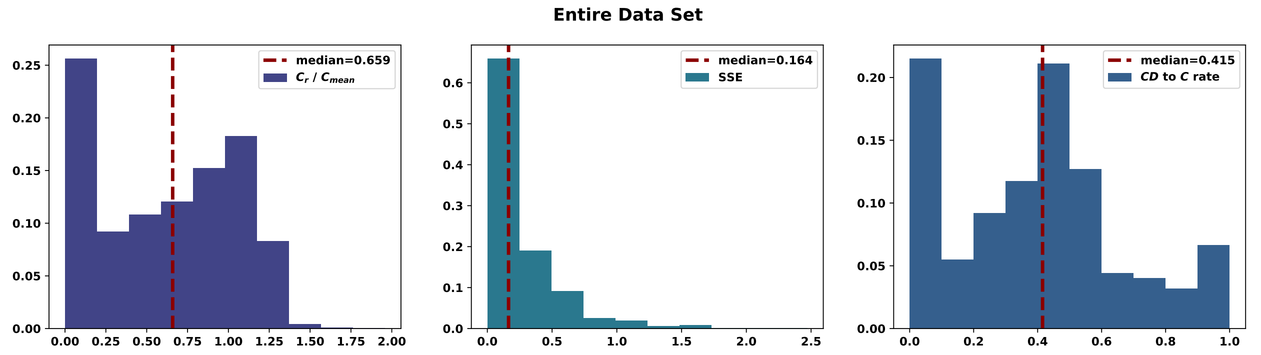


Figure 13: Distributions of C_r/C_{mean}, SSE and CD to C ratio for the winners over the tournaments of the entire data set.

A third method that evaluates the importance of the features in Table 5 using clustering and random forests can be found in the Appendix C. The results uphold the outcomes of the correlation and multivariate regression. It also evaluates the effects of the whether or not a strategy is stochastic, makes use of the knowledge of the utility values, or makes use of match length. These were not evaluated by the methods above because there are binary variables. The results showed that they have no significant effect on a strategy's performance.

5 Discussion

This manuscript has explored the performance of 195 strategies of the Iterated Prisoner's Dilemma in a large number of computer tournaments. We analyzed and extracted the salient features of the best performing strategies across various tournament types, casting the results in terms of Axelrod's original suggested features of good IPD strategies. Moreover, our results shed light on the historic performance of TFT, zero-determinant strategies, and memory one strategies generally. Strategies need to match their play to the cooperativeness of the tournament population and do so in a way that prevents or minimizes exploitation. Overall we see that complex or clever strategies can be effective, whether trained against a corpus of possible opponents or purposely designed to mitigate the impact of noise such as in the strategy DBS. Further, we showed that while the type of exploitation attempted by ZDs is not typically effective in tournaments, more sophisticated strategies capable of selectively exploiting weaker opponents while mutually cooperating with stronger opponents can be highly successful. This fact was also indicated numerically by the importance of the strategy feature SSE in the analysis of strategy features. These results highlight a central idea in evolutionary game theory in this context: the fitness landscape is a function of the population (where fitness in this case is tournament performance). While that may seem obvious now, it shows why historical tournament results on small or arbitrary populations of strategies have so often failed to produce generalizable results.

Highly noisy or tournaments with short matches favor less cooperative strategies. These environments mitigate the value of being nice. Uncertainty enables exploitation, reducing the ability of maintaining or enforcing mutual cooperation, while triggering grudging strategies to switch from typically cooperating to typically defecting. Accordingly, we find that in noisy tournaments the best performing players cooperate a lower rate than the tournament population on average. Nevertheless we found some strategies designed or trained for noisy environments were also highly ranked in noise-free tournaments. This indicates that strategy complexity is not necessarily a liability, rather it can confer adaptability to a more diverse set of environments.

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 and in noisy tournaments when p_n was less than 0.1. 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 where the noise can vary substantially, there were no strategies that can guarantee winning across a range of noise. However, if the probability of noise was constrained at 0.1 then strategies designed for noisy tournaments indeed performed well.

So what is the best way of playing the IPD? And is there a single dominant strategy for the IPD?

There was not a single strategy within the collection of the 195 strategies that managed to perform well in all the tournaments variations it competed in. Even if on average a strategy ranked highly in a specific environment this did not guarantee its success over the different tournament types. However, the results of sections 3 and 4 demonstrated that there are properties associated with the success of strategies. A few of the properties that have been identified by this manuscript's analysis contradict the properties of Axelrod [15]. Namely, in Section 3 it was shown that trained strategies dominated several tournaments across tournament types, hinting that successful IPD strategies are often clever or more complex than simple strategies like TFT. Most of the successful strategies highlighted in Section 3 were strategies that begin with cooperation.

Furthermore, in Section 3 and 4 it was shown that envious strategies performed well. Though these were not the most envious strategies in the tournaments (ZDs were included), these strategies benefited by being a bit envious. From Section 4 it was concluded that there is a significant importance in adapting to the environment, and more specifically in this work, to the mean cooperator. This section also demonstrated that a strategy should reciprocate, as suggested by Axelrod, but in some environments, such as standard and noisy, it should relax its readiness to do so.

Thus, the five properties successful strategies need to have in a IPD competition are:

- be nice, if the environment isn't noisy and the games are long
- be provocable and contrite,
- be a little envious,
- it's ok to be clever,
- adapt to the environment (including the population of strategies).

The data set described in this work contains the largest number of IPD tournaments, to the authors knowledge, and it available at [28]. 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] The prisoner's dilemma. http://www.prisoners-dilemma.com/, 2017.

[3] Eckhart A. Coopsim v0.9.9 beta 6. `https://github.com/jecki/CoopSim/`, 2015.

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

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

[6] D. Arthur and S. 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.

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

[8] D. Ashlock and E. Y. Kim. Fingerprinting: Visualization and automatic analysis of prisoner's dilemma strategies. *IEEE Transactions on Evolutionary Computation*, 12(5):647–659, 2008.

[9] W. Ashlock and D. 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.

[10] W. Ashlock, J. Tsang, and D. Ashlock. The evolution of exploitation. In *2014 IEEE Symposium on Foundations of Computational Intelligence (FOCI)*, pages 135–142. IEEE, 2014.

[11] T. C. Au and D. 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.

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

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

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

[15] R. Axelrod and W. D. Hamilton. The evolution of cooperation. *science*, 211(4489):1390–1396, 1981.

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

[17] B. Beaufils, J. P. Delahaye, and P. 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.

[18] 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.

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

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

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

[22] A. Carvalho, H. P. Rocha, F. T. Amaral, and F. G. Guimaraes. Iterated prisoner's dilemma-an extended analysis. 2013.

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

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

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

[26] D. Fudenberg and E. 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.

[27] M. Gaudesi, E. Piccolo, G. Squillero, and A. 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.

[28] N. E. Glynatsi. A data set of 45686 Iterated Prisoner's Dilemma tournaments' results. `https://doi.org/10.5281/zenodo.3516652`, October 2019.

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

[30] M. Harper, V. Knight, M. Jones, G. Koutsovoulos, N. E. Glynatsi, and O. Campbell. Reinforcement learning produces dominant strategies for the iterated prisoner's dilemma. *PloS one*, 12(12):e0188046, 2017.

[31] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[32] C. Hilbe, M. A. Nowak, and A. Traulsen. Adaptive dynamics of extortion and compliance. *PloS one*, 8(11):e77886, 2013.

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

[34] G. Kendall, X. Yao, and S. Y. Chong. *The iterated prisoners' dilemma: 20 years on*, volume 4. World Scientific, 2007.

[35] G. Kendall, X. Yao, and S. Y. Chong. *The iterated prisoners' dilemma: 20 years on*, volume 4. World Scientific, 2007.

[36] V. A. Knight, M. Harper, N. E. Glynatsi, and J. Gillard. Recognising and evaluating the effectiveness of extortion in the iterated prisoner's dilemma. *CoRR*, abs/1904.00973, 2019.

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

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

[39] J. Li, P. Hingston, S. Member, and G. Kendall. Engineering Design of Strategies for Winning Iterated Prisoner ' s Dilemma Competitions. 3(4):348–360, 2011.

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

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

[42] P. Mathieu and J. P. Delahaye. New winning strategies for the iterated prisoner's dilemma. *Journal of Artificial Societies and Social Simulation*, 20(4):12, 2017.

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

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

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

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

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

[48] M. A. Nowak and K. Sigmund. Tit for tat in heterogeneous populations. *Nature*, 355(6357):250, 1992.

[49] 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.

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

[51] W. H. Press and F. 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.

[52] The Axelrod project developers. Axelrod: 3.0.0. `http://dx.doi.org/10.5281/zenodo.807699`, April 2016.

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

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

[55] 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.

[56] D. W. Stephens, C. M. McLinn, and J. R. Stevens. Discounting and reciprocity in an iterated prisoner's dilemma. *Science*, 298(5601):2216–2218, 2002.

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

[58] E. Tzafestas. Toward adaptive cooperative behavior. 2:334–340, Sep 2000.

[59] 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.

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

[61] 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.

[62] J. Wu and R. 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 [52].
- The Matplotlib library for visualisation [33].
- The Numpy library for data manipulation [61].
- The scikit-learn library for data analysis [49].

A Parameters Summary

All the parameters used in this manuscript alongside their explanation are given by Table 8.

B Correlation coefficients

A graphical representation of the correlation coefficients for the features in Table 5.

Feature	Explanation
SSE	A measure of how far a strategy is from extortionate behaviour defined in [36].
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 cooperating rate in the tournament.
C_{min} / C_r	The minimum in the tournament divided by a strategy's cooperating rate.
C_r / C_{median}	A strategy's cooperating rate divided by the median cooperating rate in the tournament.
C_r / C_{mean}	A strategy's cooperating rate divided by the mean cooperating rate in the tournament.
C_r	The cooperating rate 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 flipped at each interaction.
n	The number of turns in a match.
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 8: The features which are included in the performance evaluation analysis.

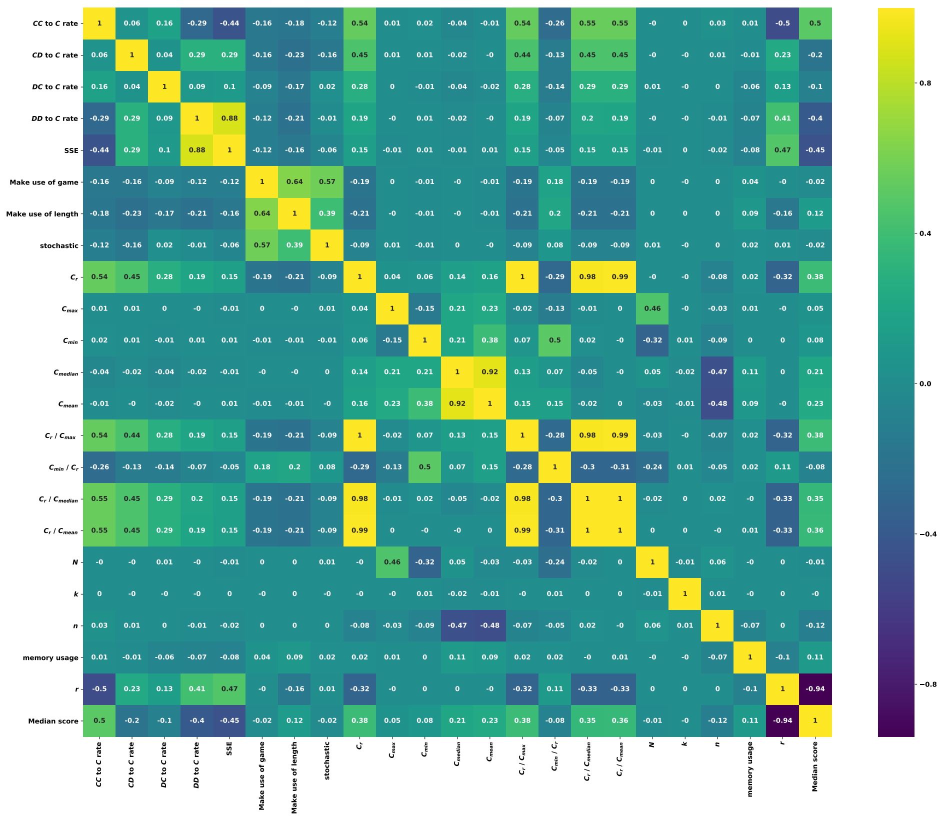


Figure 14: Correlation coefficients of features in Table 5 for standard tournaments

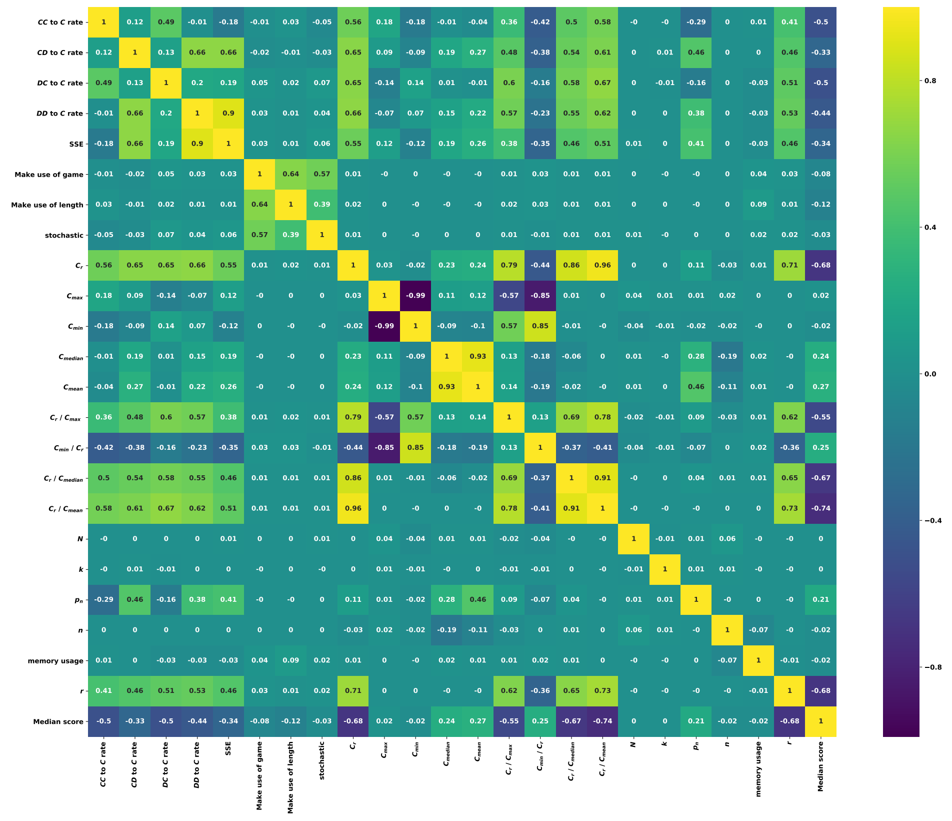


Figure 15: Correlation coefficients of features in Table 5 for noisy tournaments

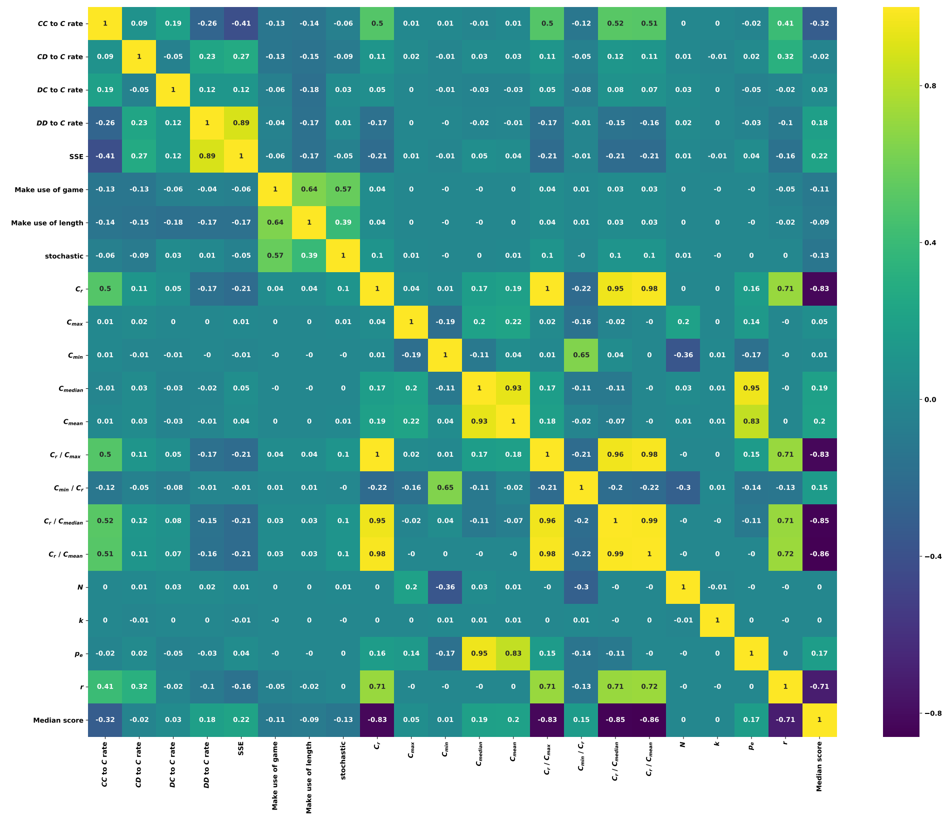


Figure 16: Correlation coefficients of features in Table 5 for probabilistic ending tournaments

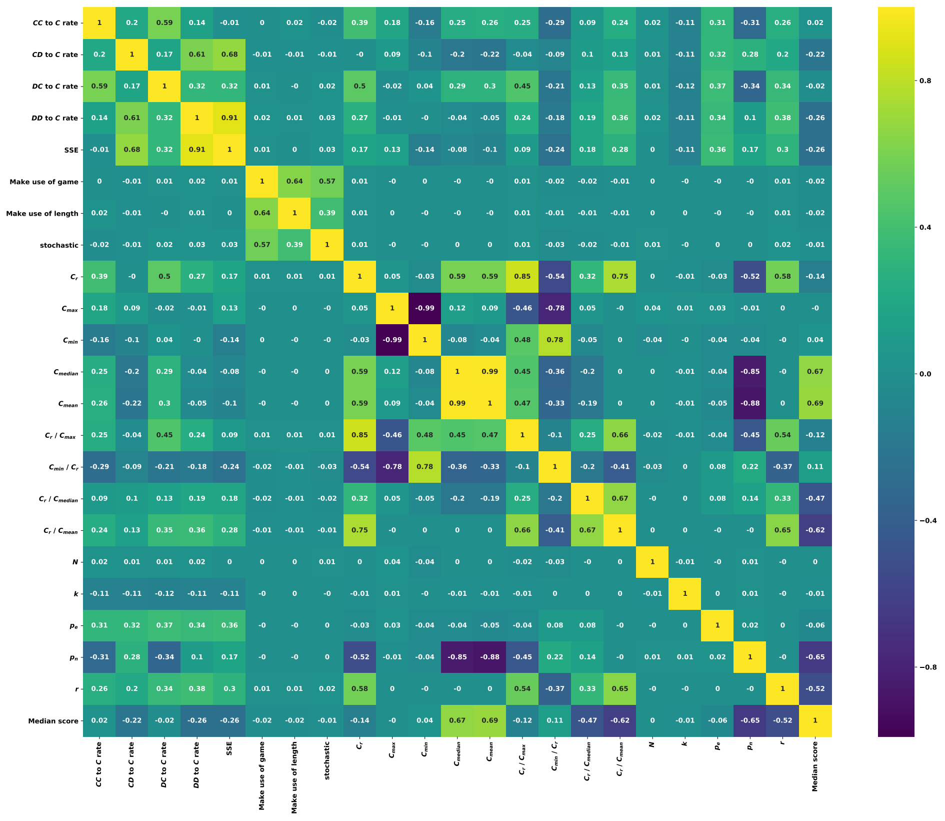


Figure 17: Correlation coefficients of features in Table 5 for noisy probabilistic ending tournaments

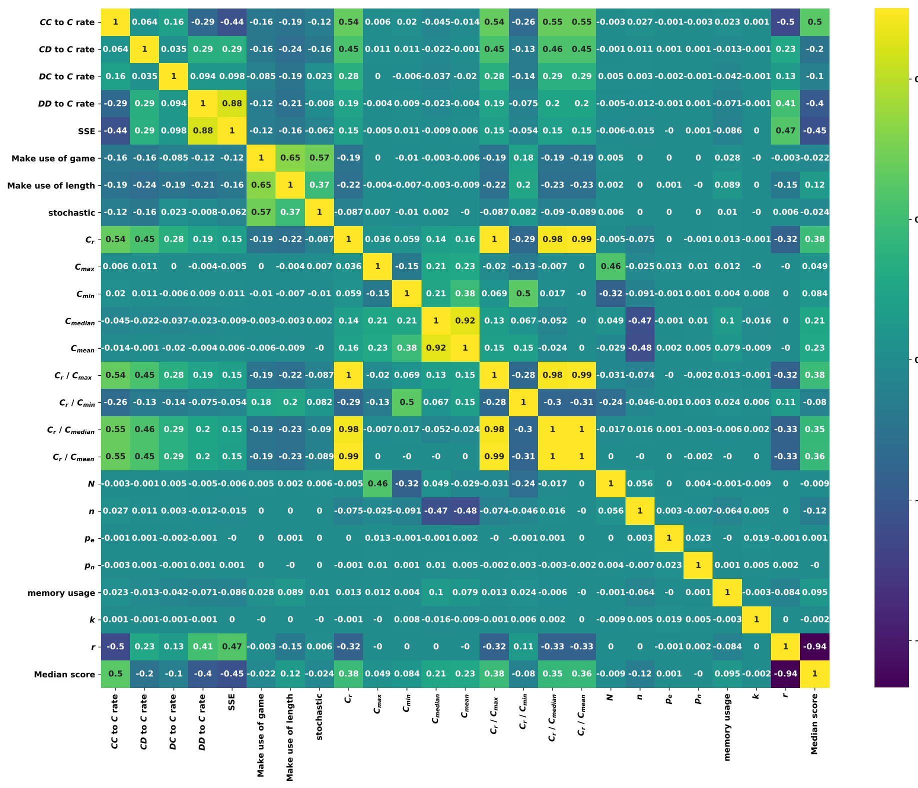


Figure 18: Correlation coefficients of features in Table 5 for data set

C Evaluation based on clustering and random forest.

The final method to evaluate the features importance in a strategy's success is a combination of a clustering task and a random forest algorithm. Initially the performances are clustered into different clusters based on them being 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 [6]. The number of clusters is not deterministically chosen but it is based on the silhouette coefficients [54].

Once the performances have been assigned to a cluster for each approach a random forest algorithm [20] is applied. The problem is a supervised problem where the random forest algorithm predicts the cluster to which a performance has been assigned to using the features of Table 5. 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 9. The out of the bag error (OOB) [31] 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.

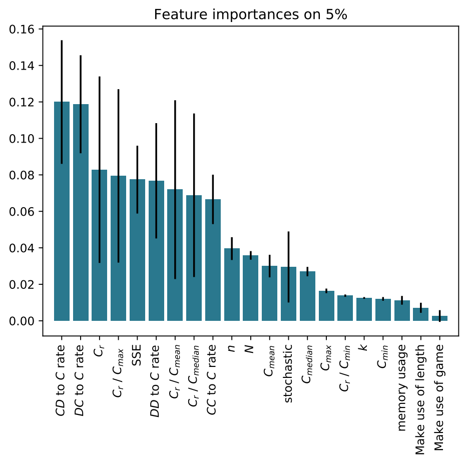
The importance that the features of Table 5 had on each random forest model are given by Figures 19, 20, 21, 22 and 23. These show that the classifiers stochastic, make use of game and make use of length have no significant effect, and several of the features that are highlighted by the importance are 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}.

D Multivariate linear regression on median score

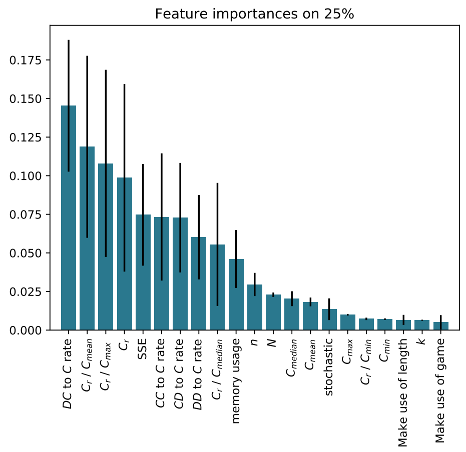
A multivariate linear regression has also been fitted to model the relationship between the features and the median score. The features included are given by Table 10 alongside their corresponding p values in the distinct tournaments and their regression coefficients.

E List of strategies

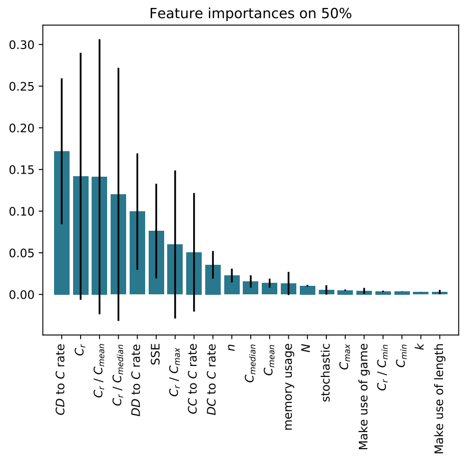
The strategies used in this study which are from Axelrod-Python library version 3.0.0.



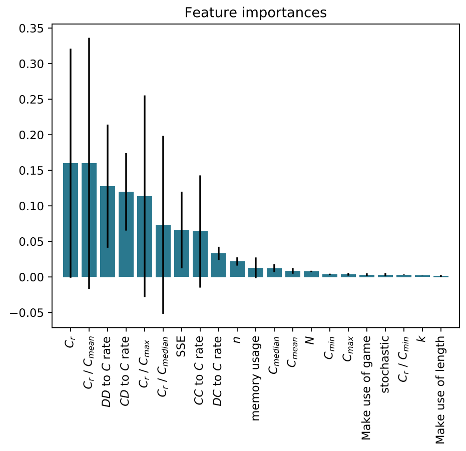
(a) Importance of features for clusters on 5% performance.



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

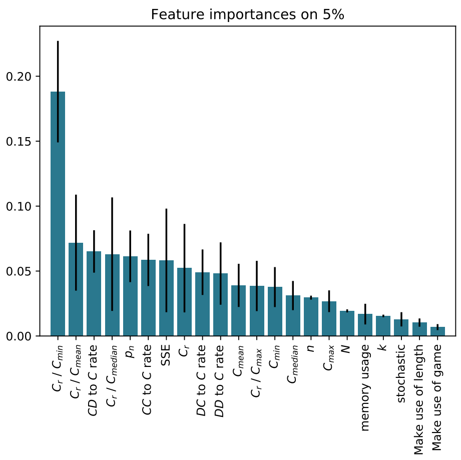


(c) Importance of features for clusters on 50% performance.

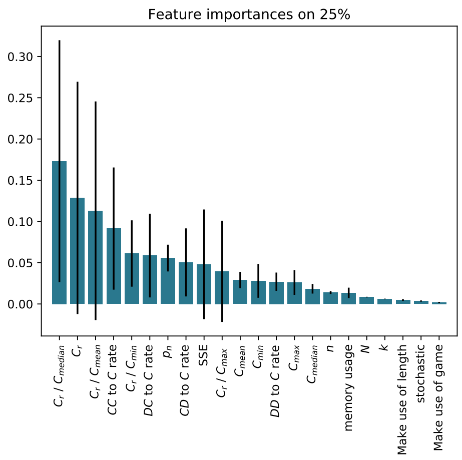


(d) Importance of features for clusters based on $k$means algorithm.

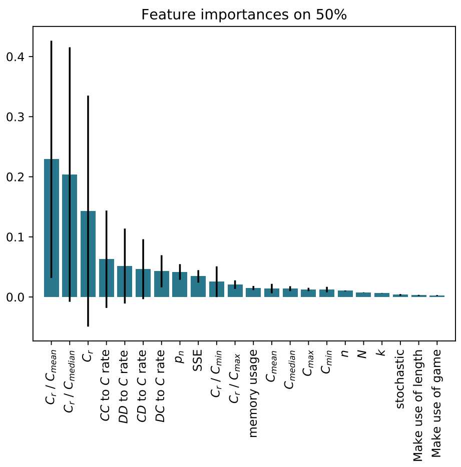
Figure 19: Importance of features in standard tournaments for different clustering methods.



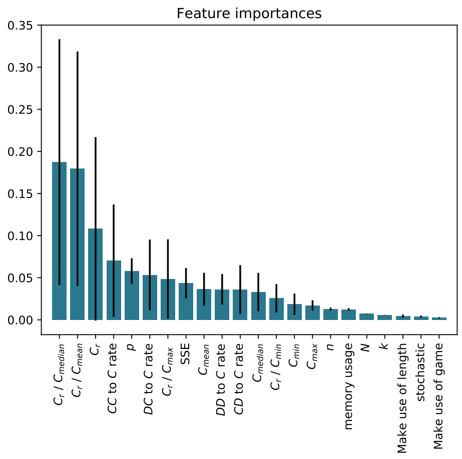
(a) Importance of features for clusters on 5% performance.



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

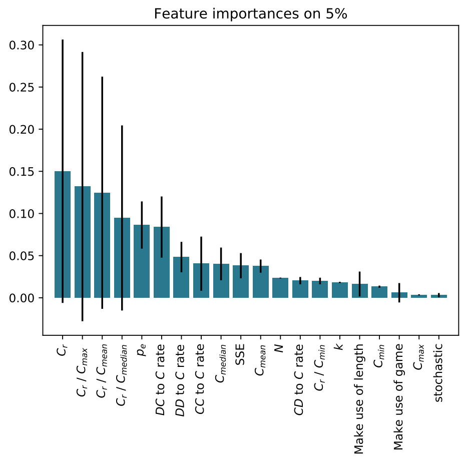


(c) Importance of features for clusters on 50% performance.

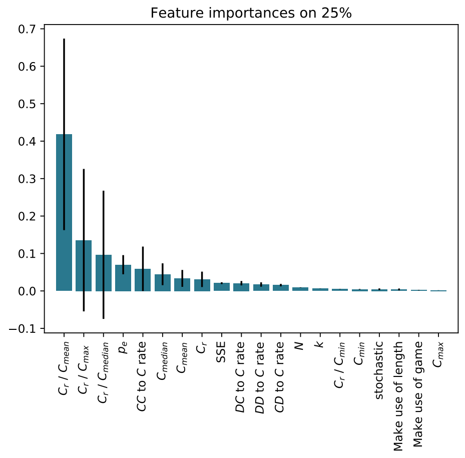


(d) Importance of features for clusters based on $k$means algorithm.

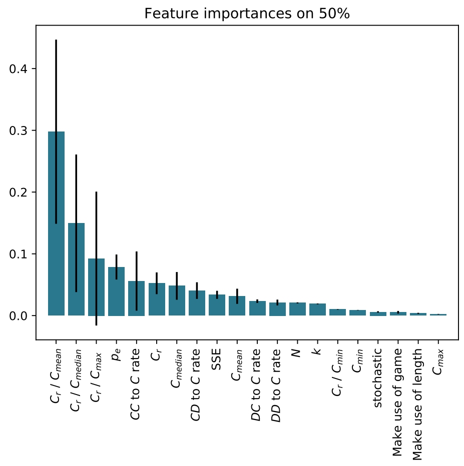
Figure 20: Importance of features in noisy tournaments for different clustering methods.



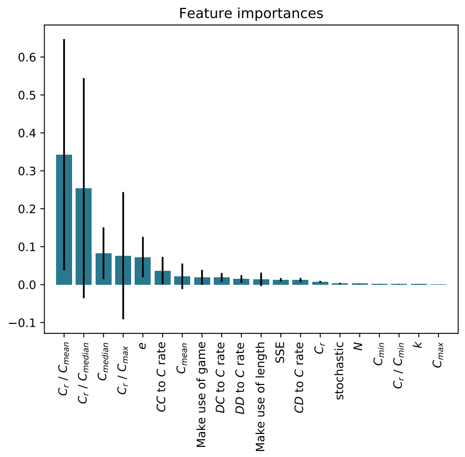
(a) Importance of features for clusters on 5% performance.



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

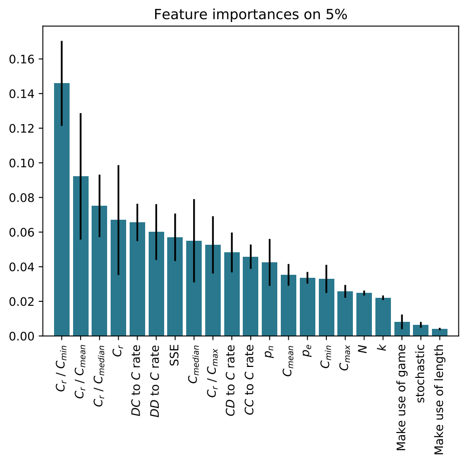


(c) Importance of features for clusters on 50% performance.

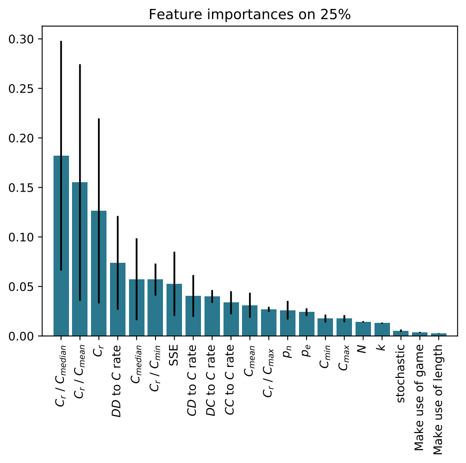


(d) Importance of features for clusters based on $k$means algorithm.

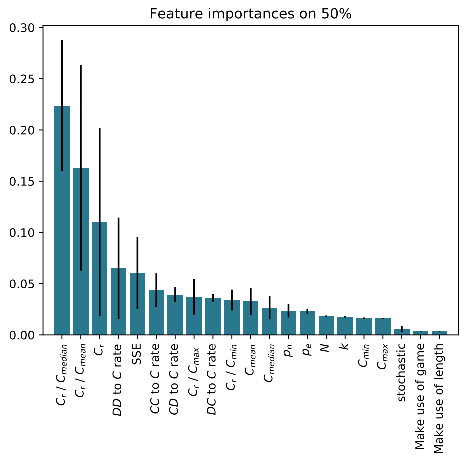
Figure 21: Importance of features in probabilistic ending tournaments for different clustering methods.



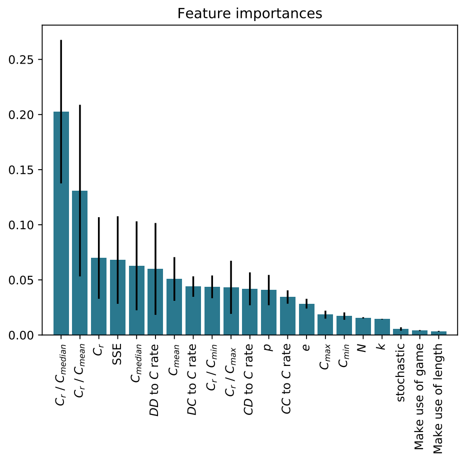
(a) Importance of features for clusters on 5% performance.



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

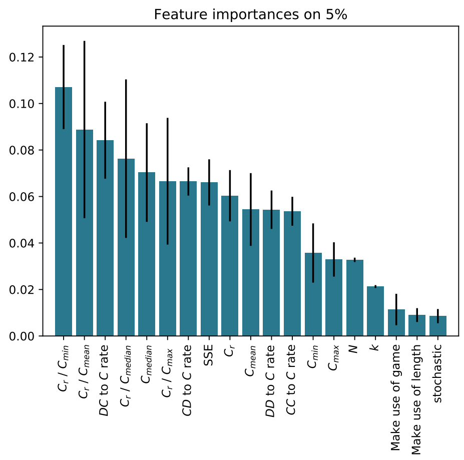


(c) Importance of features for clusters on 50% performance.

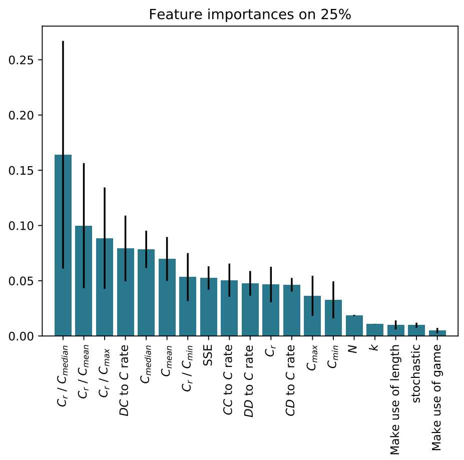


(d) Importance of features for clusters based on *k*means algorithm.

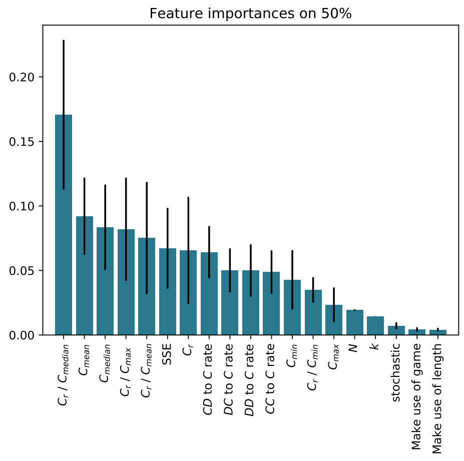
Figure 22: Importance of features in noisy probabilistic ending tournaments for different clustering methods.



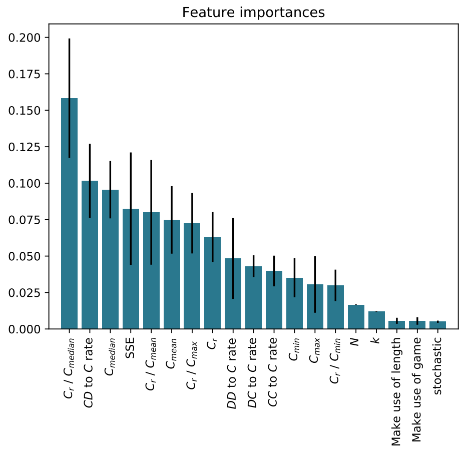
(a) Importance of features for clusters on 5% performance.



(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 23: Importance of features over all the tournaments for different clustering methods.

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	0.997786	0.972229	0.968332
	Approach 2	2	0.997442	0.963254	0.955219
	Approach 3	2	0.997152	0.953164	0.940528
	Approach 4	3	0.996923	0.950728	0.935444
probabilistic ending	Approach 1	2	0.997909	0.981490	0.978120
	Approach 2	2	0.997883	0.973492	0.967150
	Approach 3	2	0.990448	0.890068	0.875822
	Approach 4	2	0.999636	0.995183	0.992809
noisy probabilistic ending	Approach 1	2	0.995347	0.957846	0.952353
	Approach 2	2	0.992813	0.909346	0.898613
	Approach 3	2	0.990579	0.824794	0.806540
	Approach 4	4	0.989465	0.841652	0.824052
over 45686 tournaments	Approach 1	2	0.997271	0.972914	0.969198
	Approach 2	2	0.996323	0.951194	0.940563
	Approach 3	2	0.993707	0.906941	0.891532
	Approach 4	3	0.993556	0.913335	0.898453

Table 9: Accuracy metrics for random forest models.

1. ϕ [52]
2. π [52]
3. e [52]
4. ALLCorALLD [52]
5. Adaptive [39]
6. Adaptive Pavlov 2006 [35]
7. Adaptive Pavlov 2011 [39]
8. Adaptive Tit For Tat: 0.5 [59]
9. Aggravater [52]
10. Alexei [50]
11. Alternator [15, 44]
12. Alternator Hunter [52]
13. Anti Tit For Tat [32]
14. AntiCycler [52]
15. Appeaser [52]
16. Arrogant QLearner [52]
17. Average Copier [52]
18. Backstabber [52]
19. Better and Better [1]
20. Bully [46]
21. Calculator [1]
22. Cautious QLearner [52]
23. Champion [13]
24. CollectiveStrategy [40]
25. Contrite Tit For Tat [62]
26. Cooperator [15, 44, 51]
27. Cooperator Hunter [52]
28. Cycle Hunter [52]
29. Cycler CCCCCD [52]
30. Cycler CCCD [52]
31. Cycler CCCDCD [52]
32. Cycler CCD [44]
33. Cycler DC [52]
34. Cycler DDC [44]
35. DBS [11]
36. Davis [12]
37. Defector [15, 44, 51]
38. Defector Hunter [52]
39. Double Crosser [52]
40. Desperate [60]
41. DoubleResurrection [3]
42. Doubler [1]
43. Dynamic Two Tits For Tat [52]
44. EasyGo [39, 1]
45. Eatherley [13]
46. Eventual Cycle Hunter [52]
47. Evolved ANN [52]
48. Evolved ANN 5 [52]
49. Evolved ANN 5 Noise 05 [52]
50. Evolved FSM 16 [52]

	Standard		Noisy		Probabilistic ending		Noisy probabilistic ending		Overall	
	R adjusted: 0.576		R adjusted: 0.679		R adjusted: 0.816		R adjusted: 0.930		R adjusted: 0.318	
	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value	Coefficient	p-value
CC to C rate	0.043	0.000	-0.380	0.000	0.224	0.000	0.078	0.0	0.308	0.0
CD to C rate	-0.325	0.000	0.124	0.000	0.060	0.000	0.073	0.0	-0.014	0.0
C_r / C_{max}	-	-	-1.044	0.000	-	-	-1.251	0.0	-	-
C_r / C_{mean}	0.553	0.000	-0.101	0.000	-1.136	0.000	-0.089	0.0	-0.665	0.0
C_{max}	0.059	0.000	-	-	-0.044	0.086	-1.396	0.0	-	-
C_{mean}	1.837	0.000	3.078	0.000	1.506	0.000	3.645	0.0	-	-
C_{min}	0.156	0.000	1.528	0.000	0.311	0.000	-	-	-	-
C_{min} / C_r	-0.049	0.000	-0.378	0.000	-0.204	0.000	-	-	-0.257	0.0
DC to C rate	-0.204	0.000	0.074	0.000	0.066	0.000	0.066	0.0	0.007	0.0
k	-0.000	0.853	-0.000	0.987	0.000	0.008	0.000	0.0	-	-
n	-0.000	0.000	-	-	-	-	-	-	-	-
p_e	-	-	-	-	0.025	0.000	-0.095	0.0	-	-
p_n	-	-	0.124	0.000	-	-	-	-	-	-
SSE	-0.294	0.000	-0.319	0.000	0.055	0.000	0.010	0.0	-0.015	0.0
constant	0.925	0.000	1.536	0.000	2.466	0.000	2.299	0.0	2.924	0.0
memory usage	0.010	0.000	-0.004	0.000	-	-	-	-	-	-

Table 10: Results of multivariate linear regressions with the median score as the dependent variable. R squared is reported for each model.

51. Evolved FSM 16 Noise 05 [52]
52. Evolved FSM 4 [52]
53. Evolved HMM 5 [52]
54. EvolvedLookerUp1 1 1 [52]
55. EvolvedLookerUp2 2 2 [52]
56. Eugine Nier [50]
57. Feld [12]
58. Firm But Fair [25]
59. Fool Me Forever [52]
60. Fool Me Once [52]
61. Forgetful Fool Me Once [52]
62. Forgetful Grudger [52]
63. Forgiver [52]
64. Forgiving Tit For Tat [52]
65. Fortress3 [9]
66. Fortress4 [9]
67. GTFT [27, 47]
68. General Soft Grudger [52]
69. Gradual [17]
70. Gradual Killer [1]
71. Grofman[12]
72. Grudger [12, 16, 17, 60, 39]
73. GrudgerAlternator [1]
74. Grumpy [52]
75. Handshake [53]
76. Hard Go By Majority [44]
77. Hard Go By Majority: 10 [52]
78. Hard Go By Majority: 20 [52]
79. Hard Go By Majority: 40 [52]
80. Hard Go By Majority: 5 [52]
81. Hard Prober [1]
82. Hard Tit For 2 Tats [57]
83. Hard Tit For Tat [2]
84. Hesitant QLearner[52]
85. Hopeless [60]
86. Inverse [52]
87. Inverse Punisher [52]
88. Joss [12, 57]
89. Knowledgeable Worse and Worse [52]
90. Level Punisher [3]
91. Limited Retaliate 2 [52]
92. Limited Retaliate 3 [52]
93. Limited Retaliate [52]
94. MEM2 [41]
95. Math Constant Hunter [52]
96. Meta Hunter Aggressive [52]
97. Meta Hunter [52]
98. Meta Majority [52]
99. Meta Majority Finite Memory [52]
100. Meta Majority Long Memory [52]

101. Meta Majority Memory One [52]
102. Meta Minority [52]
103. Meta Mixer [52]
104. Meta Winner [52]
105. Meta Winner Deterministic [52]
106. Meta Winner Ensemble [52]
107. Meta Winner Finite Memory [52]
108. Meta Winner Long Memory [52]
109. Meta Winner Memory One [52]
110. Meta Winner Stochastic [52]
111. NMWE Deterministic [52]
112. NMWE Finite Memory [52]
113. NMWE Long Memory [52]
114. NMWE Memory One [52]
115. NMWE Stochastic [52]
116. Naive Prober [39]
117. Negation [2]
118. Nice Average Copier [52]
119. Nice Meta Winner [52]
120. Nice Meta Winner Ensemble [52]
121. Nydegger [12]
122. Omega TFT [35]
123. Once Bitten [52]
124. Opposite Grudger [52]
125. PSO Gambler 1 1 1 [52]
126. PSO Gambler 2 2 2 [52]
127. PSO Gambler 2 2 2 Noise 05 [52]
128. PSO Gambler Mem1 [52]
129. Predator [9]
130. Prober [39]
131. Prober 2 [1]
132. Prober 3 [1]
133. Prober 4 [1]
134. Pun1 [9]
135. Punisher [52]
136. Raider [10]
137. Random Hunter [52]
138. Random: 0.5 [12, 59]
139. Remorseful Prober [39]
140. Resurrection [3]
141. Retaliate 2 [52]
142. Retaliate 3 [52]
143. Retaliate [52]
144. Revised Downing [12]
145. Ripoff [8]
146. Risky QLearner [52]
147. SelfSteem [22]
148. ShortMem [22]
149. Shubik [12]
150. Slow Tit For Two Tats [52]
151. Slow Tit For Two Tats 2 [1]
152. Sneaky Tit For Tat [52]
153. Soft Go By Majority [15, 44]
154. Soft Go By Majority 10 [52]
155. Soft Go By Majority 20 [52]
156. Soft Go By Majority 40 [52]
157. Soft Go By Majority 5 [52]
158. Soft Grudger [39]
159. Soft Joss [1]
160. SolutionB1 [7]
161. SolutionB5 [7]
162. Spiteful Tit For Tat [1]
163. Stalker [21]
164. Stein and Rapoport [12]
165. Stochastic Cooperator [5]
166. Stochastic WSLS [52]
167. Suspicious Tit For Tat [17, 32]
168. TF1 [52]
169. TF2 [52]
170. TF3 [52]
171. Tester [13]
172. ThueMorse [52]
173. ThueMorseInverse [52]
174. Thumper [8]
175. Tit For 2 Tats (**Tf2T**) [15]
176. Tit For Tat (**TfT**) [12]
177. Tricky Cooperator [52]
178. Tricky Defector [52]
179. Tullock [12]
180. Two Tits For Tat (**2TfT**) [15]
181. VeryBad [22]
182. Willing [60]
183. Win-Shift Lose-Stay (**WShLSt**) [39]
184. Win-Stay Lose-Shift (**WSLS**) [37, 47, 57]
185. Winner12 [42]
186. Winner21 [42]
187. Worse and Worse[1]
188. Worse and Worse 2[1]
189. Worse and Worse 3[1]
190. ZD-Extort-2 v2 [38]
191. ZD-Extort-2 [57]
192. ZD-Extort-4 [52]
193. ZD-GEN-2 [38]
194. ZD-GTFT-2 [57]
195. ZD-SET-2 [38]