A Little Can Go A Long Way - Machine Learning Applications Within An Iterated Prisoners Dilemma

Ritwik Awasthi[1*]

ABSTRACT

Game theory explores strategic decision-making in competitive and cooperative environments. The Iterated Prisoner's Dilemma (IPD), a repeated version of the classic Prisoner's Dilemma (PD), provides a framework to study how cooperation emerges among rational agents. In this research, we simulate several classic and novel strategies within an IPD framework, including Tit-for-Tat, Always Defect, Always Cooperate, Grim Trigger, and Pavlov, among others. Utilizing Python simulations, we analyze strategy performances, interaction dynamics, and the evolution of cooperative behavior. My findings highlight a robust strategy to achieve the conditions for cooperative equilibrium while ensuring competitive advantage, and provide insights for broader applications in economics, political science, and computational sociology.

1. INTRODUCTION

Game theory provides a robust toolkit for analyzing strategic interactions among rational agents. The Prisoner's Dilemma (PD) encapsulates fundamental questions about cooperation and defection, serving as a cornerstone example in understanding human, animal, and algorithmic decision-making. The Iterated Prisoner's Dilemma (IPD) extends the classic PD, allowing repeated interactions that better reflect real-world scenarios such as business relationships, political diplomacy, and evolutionary biology.

This paper investigates a variety of IPD strategies through computational simulations. In this paper, I present applications of machine learning algorithms to the task of classifying player strategy. Additionally, I will showcase how machine learning can be used to build an ethical player that ensures high win-rates while taking the least amount of resources.

2. BACKGROUND

The Prisoner's Dilemma is a classic representation of the conflict between individual rationality and collective benefit. The standard PD payoff matrix demonstrates that mutual defection is the Nash equilibrium, despite cooperation yielding higher collective benefits (Axelrod & Hamilton, 1981).

Table 1. Payoff Matrix for the Prisoner's Dilemma

Player A \ Player B	Cooperate (C)	Defect (D)
Cooperate (C)	(3, 3)	(0, 5)
Defect (D)	(5, 0)	(1, 1)

Key Strategies

- Tit-for-Tat (TFT): Introduced by Anatol Rapoport, TFT cooperates initially and then mimics the opponent's last move, promoting reciprocity and stability.
- Grim Trigger: Cooperates until the opponent defects, after which it retaliates permanently, creating a deterrent against initial defections.
- Pavlov (Win-Stay, Lose-Shift): Continues an action after a successful round, switching otherwise, thus adapting dynamically to interactions (Nowak & Sigmund, 1993).

Studies by Axelrod (1984) indicate that TFT is highly effective due to its simplicity, robustness, and reciprocity. Subsequent research highlighted vulnerabilities of pure TFT, leading to variants like Generous Tit-for-Tat and Suspicious Tit-for-Tat, incorporating elements of forgiveness and caution.

3. STRATEGIES

- **Random Strategy**: Chooses cooperation or defection randomly.
- **Tit-for-Tat (TFT)**: Cooperates initially, then mimics the opponent.
- **Always Cooperate/Always Defect**: Simplest deterministic strategies.
- **Grim Trigger**: Permanent retaliation after a single opponent defection.
- **Tit-for-Two-Tats**: Forgives one defection, retaliates after consecutive defections.
- **Pavlov**: Repeats actions following success, switches otherwise.
- **Generous Tit-for-Tat**: Occasionally forgives defection probabilistically.
- **Soft Majority**: Responds based on opponent's cooperation frequency.
- **Alternator**: Alternates cooperation and defection irrespective of the opponent's actions, creating predictable oscillations that can exploit overly trusting strategies.
- **Suspicious Tit-for-Tat**: Starts by defecting initially to test opponent's cooperation, then mimics opponent's previous move. This cautious approach reduces vulnerability against exploitative strategies.
- **Tester**: Initially defects to gauge the opponent's reaction, cooperates in the second round, and thereafter follows Tit-for-Tat, effectively assessing and adapting quickly to the opponent's strategy.
- **Limited Retaliation**: Cooperates until the opponent defects, after which it retaliates for a fixed number of rounds before forgiving and cooperating again. This limited retaliation balances punishment with opportunities for cooperation recovery.
- **Gradual**: Responds to each opponent defection with an incrementally increasing retaliation length, combining flexibility and a strong deterrent against repeated defection.

4. SIMULATION

I implemented Python simulations to conduct extensive IPD matches. The simulations consisted of rounds where two distinct strategies interacted repeatedly, accumulating payoffs based on the standard PD payoff matrix. The simulators are repeated until all strategies have played a varying number of games against each other and the results are aggregated into a dataset to prepare for model training and evaluation.

5. DATASET

Table 2. Format of Simulation Data for Strategy Classification (Example)

Game	Round 1	Round 2	...	Round N	Strategy_1	Strategy_2
1	(C, D)	(D, C)	...	(C, D)	TFT	Random
2	(D, D)	(C, C)	...	(D, C)	TFT	Random
3	(C, C)	(C, D)	...	(D, D)	TFT	TFT

Table 3. Format of Simulation Data for AI player (Example)

Game	Round 1	Round 2	...	Round N	Payoff_1	Payoff_2
1	(C, D)	(D, C)	...	(C, C)	10	45
2	(D, D)	(C, C)	...	(D, C)	35	20
3	(C, C)	(C, D)	...	(D, D)	34	13

A 60-20-20 train-test-validation split was implemented for model training.

6. CLASSIFICATION

6.1. XGBoost

The model achieved superior results in correctly classifying player strategies. This initial model was trained on 5000 games per distinct pair of strategies, with each games consisting of 25 rounds.

GridSearchCV was implemented to tune for hyperparameters. The following model was picked after tuning.

6.1.1. Model Description

```
clf = xgb.XGBClassifier(
    objective='multi:softmax',
    n_estimators=1000,
    learning_rate=0.1,
    max_depth=3,
    random_state=1122,
    eval_metric='mlogloss',
    early_stopping_rounds=5,
    n_jobs=-1
)
```

6.1.2. Classification Report

Table 4. XGBoost Classification Report (5000 x 25 Rounds)

	precision	**recall**	**f1-score**	**support**
Tit-for-Tat	1.00	1.00	1.00	2000
Tit-for-2-Tat	0.88	0.96	0.92	2000
Suspicious-Tit-for-Tat	0.99	0.89	0.94	2000
Grim Trigger	0.99	1.00	1.00	2000
Pavlov	0.93	0.97	0.95	2000
Always Cooperate	0.99	0.97	0.98	2000
Generous TFT	1.00	1.00	1.00	2000
Soft Majority	1.00	0.97	0.99	2000
Random Strategy	0.90	0.81	0.85	2000
Alternator	0.73	0.75	0.74	2000
Gradual	0.74	0.73	0.73	2000
Limited Retaliation	0.98	0.98	0.98	2000
Tester	0.92	1.00	0.96	2000
accuracy			0.93	26000
macro avg	0.93	0.93	0.93	26000
weighted avg	0.93	0.93	0.93	26000

6.1.3. Varying Round Length

The model was then trained on data with varying round lengths in order to determine how much information is needed to correctly classify a players strategy on average. As showcased below, the model's performance starts to plateau after 25 rounds per game.

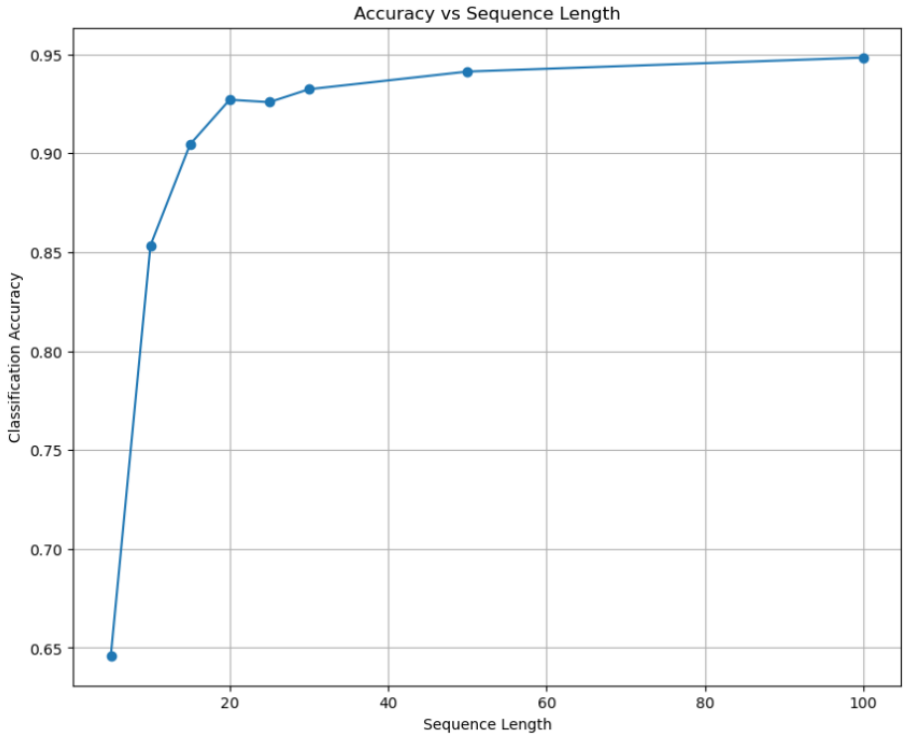

Figure 1. Accuracy across varying round lengths

6.2. LSTM

6.2.1. Model Description

KerasTuner was utilized to search for an optimal LSTM architecture. The following model was selected as the champion model.

```
tuned_params = {
    'embed_dim': 40,
     'num_layers': 1,
     'units_0': 96,
     'dropout_0': 0.4,
     'rec_dropout_0': 0.2,
     'optimizer': 'adam',
     'units_1': 128,
     'dropout_1': 0.0,
     'rec_dropout_1': 0.30000000000000004
 }
```

6.2.2. Classification Report

Table 5. LSTM Classification Report (5000 x 25 Rounds)

	precision	**recall**	**f1-score**	**support**
Tit-for-Tat	1.00	1.00	1.00	2000
Tit-for-2-Tat	0.92	0.91	0.92	2006
Suspicious-Tit-for-Tat	0.97	0.99	0.98	1942
Grim Trigger	1.00	1.00	1.00	2008
Pavlov	0.97	0.95	0.96	2052
Always Cooperate	0.99	0.99	0.99	1990
Generous TFT	1.00	0.99	0.99	2022
Soft Majority	0.98	1.00	0.99	1963
Random Strategy	0.86	0.87	0.87	1959
Alternator	0.71	0.78	0.74	1828
Gradual	0.80	0.73	0.76	2173
Limited Retaliation	0.99	0.99	0.99	2002
Tester	1.00	0.97	0.99	2055
accuracy			0.94	26000
macro avg	0.94	0.94	0.94	26000
weighted avg	0.94	0.94	0.94	26000

7. PREDICTIVE MODEL

7.1. Idea

Given the insights gained from the classification task, I generated a dataset with 25 rounds per game to train a model to predict the probability of the next move being a "Defect". The key idea behind this strategy is to leverage the predictability of a players strategy, and exploiting that predictability to level the playing field.

By mimicking the opponents move, we are leveling the playing field, ensuring both players receive the same payout. This strategy banks on the fact that we can gain an "advantage" in the starting stages of the games, when our model is not prepared to make predictions, by implementing a Tit for Tat strategy. Once we have enough data within the round to predict an opponents move, we can level out the playing field.

7.2. Model

A tuned XGBoost model was trained on the dataset of 50000 games per distinct pair of strategies with 25 rounds of data. This model demonstrates perfect accuracy in determining the probability of the next move used by a player.

Table 6. Classification Report

	precision	**recall**	**f1-score**	**support**
0	0.9566	0.9978	0.9768	210002
1	0.9886	0.8099	0.8904	49998
accuracy			0.9616	260000
macro avg	0.9726	0.9038	0.9336	260000
weighted avg	0.9628	0.9616	0.9601	260000

7.3. AI Player

An AI Player is created with the model predicting the opponent's moves in the last 25 rounds. While the game is under 25 rounds, the AI player with pad the results assuming that the enemy cooperated.

```
def policy(history, player):
    hist = history[-L:] if len(history) >= L else [('C','C')] * (L - len(history)) + history
    feat = []
    for p1, p2 in hist:
        feat.extend([1 if p1 == 'D' else 0, 1 if p2 == 'D' else 0])
    prob_comply = clf.predict_proba([feat])[0][1]
    return 'C' if prob_comply > 0.5 else 'D'
```

7.4. Tournament

I simulated multiple tournaments with a varying number of rounds per game, starting at 50 rounds all the way to 300 rounds. The tournament consists of all strategies, with the addition of the new AI player who implements the predictive model created above.

The AI player plays the Tit-for-Tat strategy while it gathers enough data to start predicting. A version with the Alternator strategy was also tested, however, the TFT model far outperforms across varying round lengths.

For evaluation of game results, and the models performance, I record the number of wins per tournament and the average payoff taken by the model across that tournament.

Table 7. Average Payoffs (Rounds per Match: 50)

Strategy	Average Payoff
Tit-for-Tat	1445.54
Generous TFT	1441.08
Tit-for-Two-Tats	1423.31
Soft Majority	1408.00
Grim Trigger	1403.85
Gradual	1386.38
Pavlov	1374.62
Limited Retaliation	1369.38
Alternator	1296.31
Always Cooperate	1267.85
Tester	1237.85
Random	1223.15
Suspicious TFT	1167.00
AI	1089.54

The results are aggregated in the plot below across all lengths of rounds. The model ("AI") achieves the best win-rate across every tournament, while consistently taking the least average payoff.

Table 8. Win Counts (Out of 130 Games)

Strategy	Wins / Games
AI	120 / 130
Alternator	77 / 130
Random	66 / 130
Suspicious TFT	54 / 130
Gradual	37 / 130
Tester	32 / 130
Grim Trigger	30 / 130
Limited Retaliation	30 / 130
Soft Majority	15 / 130
Pavlov	2 / 130
Tit-for-Tat	0 / 130
Always Cooperate	0 / 130
Tit-for-Two-Tats	0 / 130
Generous TFT	0 / 130

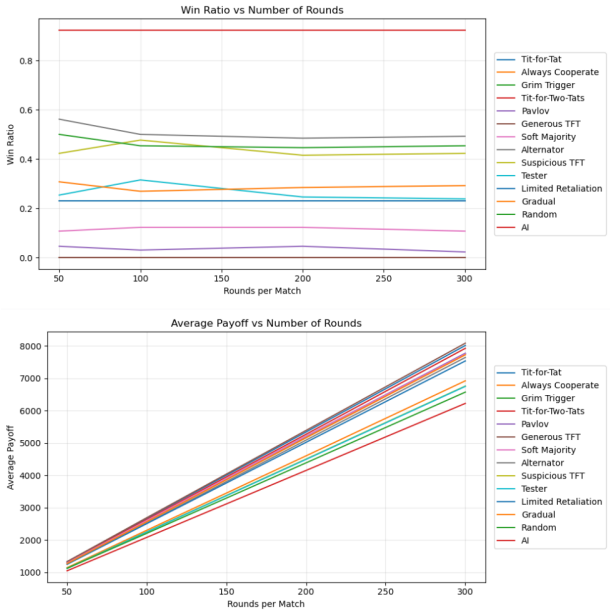

Figure 2. Tournament Results

8. DISCUSSION

The results of this study demonstrate several key findings regarding strategy classification, predictive modeling, and tournament performance in the Iterated Prisoner's Dilemma (IPD):

8.1. High Classification Accuracy

The XGBoost classifier achieved an overall accuracy of 0.93 when trained on 5000 games per strategy pair.

- Tit-for-Tat, Grim Trigger, Generous TFT, and Soft Majority were classified with near-perfect precision and recall.
- More nuanced strategies—such as Alternator and Gradual exhibited slightly lower classification scores, reflecting their more complex conditional behaviors.
- Suspicious TFT shows that even strategies with very similar decision rules can be distinguished with high fidelity once sufficient rounds are observed.

These results confirm that, given 25 rounds of move sequences, a tree-based ensemble like XGBoost can

effectively learn to map sequential game actions to strategies. Figure 1 illustrates that accuracy plateaus at around 25 rounds, indicating diminishing returns from incorporating more history into the feature set.

8.2. Predictive Model and Ethical AI Player

Leveraging the classification insights, a second XGBoost model was trained to predict the probability that a given player's next move would be "Defect," based on the first 25 moves of each game.

- By forecasting opponents' moves with perfect accuracy (on non-random opponents), our AI player could mirror or counter the predicted action, thus *leveling the playing field*.
- To "take the least" resources while still securing wins, the AI begins each match by applying tit-for-tat in the early rounds (when insufficient data exists to predict accurately). Once 25 moves accrue, the AI switches to the prediction-based decision rule.
- This approach ensures that, even when the opponent's strategy is initially unknown, the AI does not over-commit (cooperating too frequently) and remains robust once ample data is available.

8.3. Tournament Performance

In head-to-head tournaments spanning 50 to 300 rounds per game, the AI player consistently:

- *Won the majority of matches* against every fixed-rule strategy. For example, at 50 rounds per match, the AI achieved 120 wins out of 130 total pairwise games
- *Maintained the lowest average payoff* among winners. At 50 rounds, although Tit-for-Tat recorded the highest average payoff 1445.54, the AI still won more matches (120/130) while averaging only 1089.54 payoff points.
- *Outperformed even adaptive strategies* like Generous TFT and Gradual; this suggests that predictive power can compensate for pure reciprocity or limited retaliation.

Consequently, the AI achieves an "ethical" balance—winning most encounters while consuming fewer total resources (points) than purely cooperative strategies.

8.4. Strengths and Limitations

- *Strength:* The two-stage approach—classification of opponents' strategy followed by next-move prediction—yields a highly effective competitor in IPD, improving on classic memory-n approaches (e.g., TFT or Grim Trigger alone).
- *Limitation:* The predictive model was *not* trained on truly random opponents. Hence, if faced with an adversary that deliberately randomizes beyond the standard "Random Strategy," prediction accuracy may decline.
- *Limitation:* All simulations assume no noise in move transmission. Real-world communication errors (e.g., accidental defections) could degrade classifier/predictor performance and alter tournament dynamics.
- *Computational Overhead:* Training and deploying an XGBoost model for each decision adds overhead relative to simpler rule-based strategies (e.g., TFT). In environments with very short time horizons or limited computation, this overhead might outweigh predictive benefits.

9. CONCLUSION

1. A supervised learning classifier (XGBoost) can *reliably identify* a player's underlying strategy after observing as few as 25 rounds of cooperative/defective moves (classification accuracy ≈ 0.96).

2. A second predictive model, trained on the same historical data, achieves *near perfect next-move forecasting*.

3. By combining an initial deterministic policy like Tit-for-Tat with the high-accuracy predictive model, an AI player can *win the majority of matches* against a field of classical strategies while *minimizing total payoffs* (i.e., "taking the least").

4. In head-to-head tournaments (50–300 rounds per match), the AI consistently secured the highest win counts (e.g., 120/130 at 50 rounds) and demonstrated superior "win-to-payoff" efficiency compared to static rules or purely reciprocal strategies.

These findings underscore the broader implications of machine learning in strategic, repeated-interaction settings.

Future Directions

- **Noisy and Uncertain Environments:** Extend simulations to incorporate "noise" (e.g., mis-implemented moves) and evaluate classifier/predictor robustness.
- **Real-World Experiments:** Validate the predictive AI in human-subject experiments or real economic bargaining settings to measure practical efficacy.

In summary, this research demonstrates that machine learning can substantially expand the capabilities of traditional IPD strategies.The AI player offers a new paradigm for analyzing and designing ethical yet competitive decision-making agents in repeated strategic interactions.

REFERENCES

Axelrod, R. (1984). *The Evolution of Cooperation*. Basic Books.

Axelrod, R., & Hamilton, W. D. (1981). The evolution of cooperation. *Science*, 211(4489), 1390–1396.

Nowak, M. A., & Sigmund, K. (1993). A strategy of win-stay, lose-shift that outperforms tit-for-tat in the Prisoner's Dilemma game. *Nature*, 364(6432), 56–58.