Player Identification and Next-Move Prediction for Collectible Card Games with Imperfect Information

Logan Fields, John Licato

Advancing Machine and Human Reasoning (AMHR) Lab
Department of Computer Science and Engineering
University of South Florida
ldfields@usf.edu, licato@usf.edu

Abstract

Effectively identifying an individual and predicting their future actions is a material aspect of player analytics, with applications for player engagement and game security. Collectible card games are a fruitful test space for studying player identification, given that their large action spaces allow for flexibility in play styles, thereby facilitating behavioral analysis at the individual, rather than the aggregate, level. Further, once players are identified, modeling the differences between individuals may allow us to preemptively detect patterns that foretell future actions. As such, we use the virtual collectible card game "Legends of Code and Magic" to research both of these topics. Our main contributions to the task are the creation of a comprehensive dataset of Legends of Code and Magic game states and actions, extensive testing of the minimum information and computational methods necessary to identify an individual from their actions, and examination of the transferability of knowledge collected from a group to unknown individuals.

Introduction

The ability to identify a player within a game simply from their moves is deeply relevant to gaming research. By understanding the behavior of many different players, game developers can identify patterns in their actions and use them to create more engaging and personalized gameplay (Cowley and Charles 2016; Moura, el-Nasr, and Shaw 2011). Similarly, player identification allows developers to track individual player progress through games, identify potential areas of frustration, and alter the gameplay to maintain engagement (Xue et al. 2017; Zohaib 2018). Finally, player identification and action prediction are crucial for game security. By monitoring player moves and behavior, moderators can identify suspicious activity and intervene to prevent hacking, cheating, and other forms of exploitation (Parizi et al. 2019). Furthermore, identifying players from their behavior, as opposed to usernames or other user-provided identifiers, would allow developers to recognize players with multiple accounts and known cheaters in real-time (Conti and Tricomi 2020).

However, artificial intelligence struggles with identifying individuals of interest from their actions and predicting the future actions of unknown individuals (Sutrop 2020), potentially due to an inability to codify human judgment (Agrawal, Gans, and Goldfarb 2019). Functionally, such predictions would require AI systems to be capable of determining an actor's intentions and values (Sutrop 2020) or, at minimum, the acceptable values of a collective (Gabriel 2020). This *value alignment* problem is nontrivial given that individuals' judgments may be highly subject to variation (Sutrop 2020), and avoiding negative outcomes for an individual may produce different values than maximizing positive outcomes for a collective (Gabriel 2020).

Instead, we propose that observing an individual's behavior within specified environments, and with fixed resources, may provide valuable information about that individual's distinct decision-making habits. Furthermore, by analyzing variations in each individual's behavior, we may be able to ascertain their intent and goals, and subsequently predict their future actions. A system capable of modeling and predicting these individualized patterns would have numerous applications for player modeling, simulation of virtual agents, and various other gaming areas. For example, if an individual's future actions given a particular set of circumstances can be accurately predicted, it may be possible to adjust the circumstances to induce desired outcomes, similarly to how experienced chess players may lure novices into checks by predicting their upcoming moves and capitalizing on their vulnerabilities (Saariluoma 1995).

Playing collectible card games (CCGs) has been known to increase creativity, cognition, logical reasoning, and knowledge synthesis (Turkay, Adinolf, and Tirthali 2012). The nature of CCGs facilitates quick responses to attacks and incentivizes players to spend more resources for better combat power. As such, CCGs are a useful platform for examining resource allocation and rapid strategy development under pressure (Adinolf and Turkay 2011). However, their broad action spaces, imperfect information, and far-reaching planning structures also allow almost infinitely many possible decks and innovative multi-turn sequences, enabling the representation of individualized cognitive distinctions rather than generalized assumptions derived from the collective performance of numerous players (Miernik and Kowalski 2022; Yao et al. 2022). This makes CCGs an appropriate medium for researching the intersections between AI, gaming, and security.

Our Contribution. This paper will address four specific research questions:

RQ1 Given varying information about a player's environment, resources, and prior actions, what features of the information are relevant to player identification?

RQ2 What learning methods produce the best results, with respect to speed and accuracy, for identifying a player from their prior actions?

RQ3 Given a dataset of actions for a known player in a game, can we predict their next action?

RQ4 Given a dataset of actions for many players in a game, can we predict the next action for an unknown player?

We make three major contributions in addressing these questions:

- We introduce a dedicated dataset[1], specifically created for studying **RQ1–RQ4**, which captures the entire game state of a virtual CCG at each turn in the game and links that state to a unique player and their action, resulting in nearly two million tuples of gameplay data. This is notable as the limited amount of CCG play data has previously hindered AI and gaming research (Bertram, Fürnkranz, and Müller 2021). Although there are open-source CCGs for research, such as LoCM, to the best of our knowledge, no other comprehensive datasets of gameplay information exist.
- We train three transformer-based large language models (LMs) and a variety of simpler classification models with the purpose of determining the minimum environmental information and computational methods necessary to accurately identify an individual from aggregated data. This is also pivotal for CCG research in particular, as CCGs are games of imperfect information, and any research into play style identification will be inherently hindered as such.
- Finally, we examine the functional performance difference between individualized modeling and modeling aggregated data for predicting the next action of a previously unseen individual. Similarly to the concern about imperfect information, player modeling for games, in general, will always need to predict the actions of new or unknown players. If sufficient individualized information about a player is unavailable, understanding the applicability of aggregated data is crucial.

Background

Collectible Card Games. Collectible card games, also known as trading card games, typically focus on two aspects: deckbuilding and player-vs-player combat (Turkay, Adinolf, and Tirthali 2012). The specific mechanics vary between games, but in most CCGs players build their own card decks, either by selecting one card at a time from a set of pre-specified options (i.e., drafting) (Vieira, Tavares, and Chaimowicz 2020) or by creating synergistic groups to perform specific multi-turn effects from the cards that the player owns (Bertram, Fürnkranz, and Müller 2021; Bjørke and Fludal 2017). Players then randomly draw cards into their hands to engage in combat (Kowalski and Miernik 2020).

Most CCGs, including *Legends of Code and Magic* (LoCM) and *Hearthstone*, have asynchronous turn orders where each player performs all of their actions before the next player can begin their turn (Chen et al. 2018). Some games, such as *7 Wonders* and *Tybor the Builder*, are synchronous and all players play at the same time, but these typically include significant elements of board games and have fewer AI applications. Notably, *Magic: the Gathering* (MtG) is asynchronous but allows players to respond to attacks during their opponent's turns, creating the most reactive play structure. In all games, a single turn may consist of multiple actions, classified into three categories: a player playing cards from their hand onto the board (e.g., "summoning" creatures in LoCM), a player using cards either in their hand or on the board to increase their own health or deplete the health of their opponent (e.g., casting healing spells in MtG), or a player using cards to attack or augment other cards on the board (Kowalski and Miernik 2020). Games typically end when either player reaches zero health (Chen et al. 2018).

Behavioral Stylometry. Within games, the ability to identify a player from their actions alone is known as *behavioral stylometry* (McIlroy-Young et al. 2021). The first hurdle to overcome in designing an AI capable of behavioral stylometry is for the designers to effectively understand the game's rules and strategies. For any game, this requires significant domain knowledge, which in turn requires either human experts or a sufficiently simplistic action space. Consequently, behavioral stylometry has shown significant AI applications in such well-defined spaces as classic board games (de Mesentier Silva et al. 2017; Yannakakis and Togelius 2015), video games (Ferguson et al. 2020; Hsieh and Sun 2008; Shaker et al. 2011), and sports (Tuyls et al. 2021). In chess, in particular, few-shot classification frameworks have been able to correctly identify a player from among thousands of candidates with 98% accuracy given only 100 labeled games, and have been able to transfer that knowledge to previously unseen players (McIlroy-Young et al. 2021, 2022).

However, behavioral stylometry and other forms of play style identification remain a significant issue for CCGs (Bertram, Fürnkranz, and Müller 2021). Classifying sequences of common moves, as well as the possible play style strategies which apply them, based on data from large numbers of players is a continuing topic of research. We propose that identifying and analyzing an individual player's patterns can help codify their behavior and explain why they may be playing a particular way. While there has been significant research into pattern categorization across players (Drachen et al. 2012; Hoover et al. 2020), there is little work on using those patterns to identify a player, much less in CCGs. As such, this research will attempt behavioral stylometry in CCGs based only on a player's immediately preceding action, as well as the sum of their actions up to that point as we have seen in chess.

[1]The datasets for this work are available upon request.

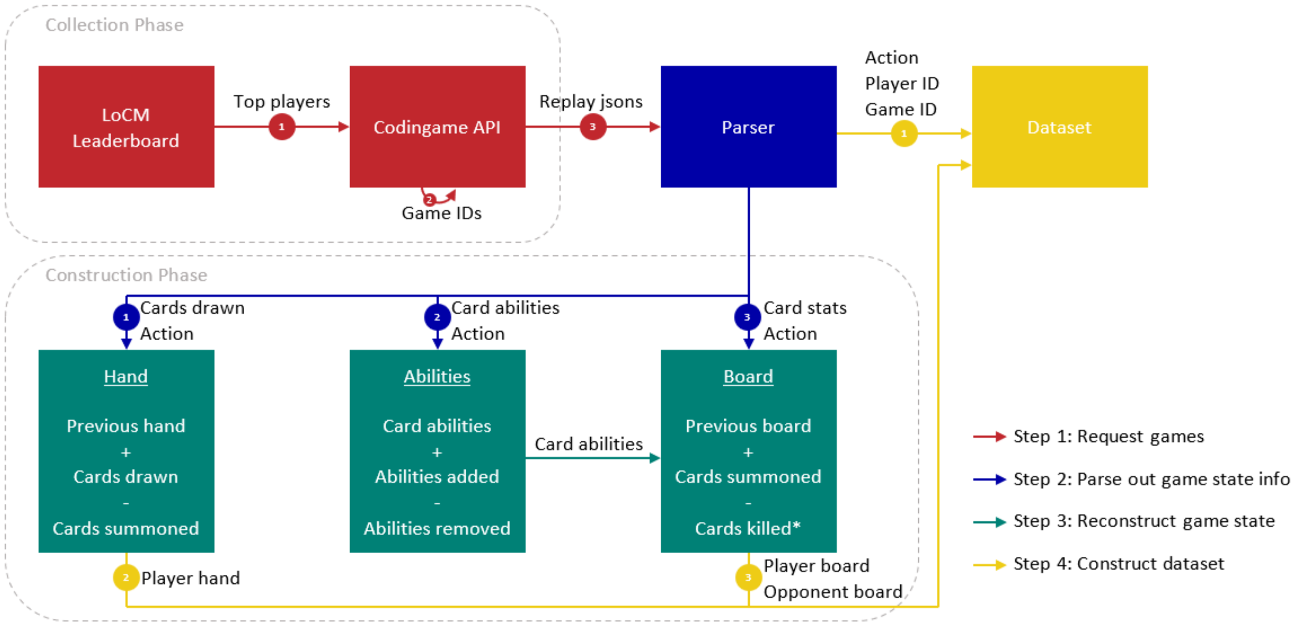

Figure 1: Dataset construction process. Creature cards are killed when attacked by another card with a Lethal ability or more attack points than the attacked card's defense points. All damage is negated for one turn if the attacked card has a Ward ability.

Play Style Identification in CCGs. Cards within CCGs typically have some amount of natural language information for players to parse through. Each card contains the name identifying the card, the cost to play it in combat, statistics indicating its strength and health, a description of the card's effect on the game state, etc. However, correctly interpreting a card's effect is a considerable issue, even for humans, and is often resolved in real time by expert judges. For example, MtG has entire corpora of decisions made by judges in tournament settings (Adler 2019). Interpreting these same effects in a human-like manner is an even more difficult task for AI, in part because humans may disagree considerably on their judgments (Sutrop 2020) and majority votes can be exceedingly difficult to replicate heuristically (Fields, Marji, and Licato 2022).

Therefore, for an AI system to interpret a card's effect and usability exactly the same way as an individual, it would need to understand that individual's intent, behavior patterns, and biases. Current play style identification models for CCGs are limited to recognizing pre-specified card combinations that are common to many players (Drachen et al. 2012; Hoover, Strobelt, and Gehrmann 2019), but they cannot handle arbitrary sequences which may be unique to individuals. We propose that this is because these systems are incapable of recognizing patterns in the cards' features which may indicate the player's intent and card substitutability.

As such, our approach explores two methods of determining a player's goals. First, we examine the usefulness of adding a feature for the game's state at a particular turn. This provides a snapshot of the player's environment and may give insight into the context under which they would make certain decisions. Second, we consider the application of transformer-based LMs. Transformers are distinguishable from other LMs due to their use of self-attention to weigh the significance of different portions of the input and focus more attention on more significant portions (Vaswani et al. 2017). This allows Transformers to process the entire input in parallel. We hypothesize that the self-attention will allow transformer-based models to better identify significant card patterns, and that the ability to process whole inputs will reduce the time and memory complexity for gaming purposes.

Legends of Code and Magic. We are initially studying these research questions using *Legends of Code and Magic*, an online CCG designed for testing AI capabilities. Full details on the game's mechanics can be found in (Kowalski and Miernik 2020), however we note a few relevant features and their implications here. First, because both players are presented with the same set of three cards during each round of the drafting phase, they may construct the same or highly similar decks. However, where the decks differ, the choice of cards may give insight into an individual player's strategy.

Additionally, all card effects in LoCM are deterministic (Kowalski and Miernik 2020), and the only probabilistic element of the game is the order in which cards are drawn from the deck into a player's hand. Finally, although the standard LoCM game board consists of two lanes — or board segmentations which specify the state a card is in and how it may be played — we utilized the simplified one-lane form which is hosted on the CodinGame platform[2]. This form was used in August 2018 for a platform-wide AI competition and, therefore, provided a robust set of player data for constructing our dataset.

[2]https://www.codingame.com

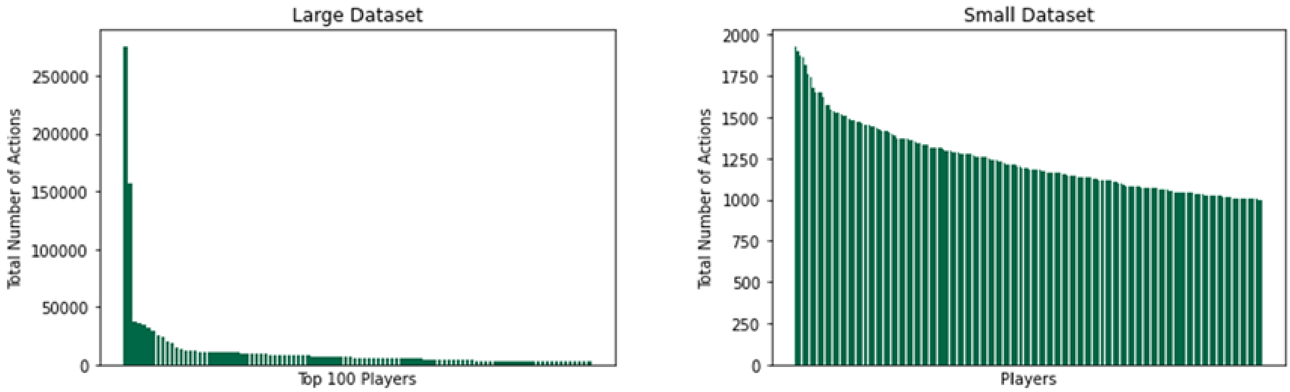

Figure 2: Representation of the difference in turn balancing between just the top 100 players in the full dataset (left) versus all 184 players in the limited training dataset (right).

Dataset Construction

To create a dataset with unique players, as well as robust game states and actions at each turn, we scraped the LoCM leaderboard for the top 1000 players, which was the maximum number allowed by the API. This constituted the majority of players who advanced past the tutorial, and the skill levels ranged from novice to advanced. We then used the CodinGame API[3], which hosts LoCM, to obtain the most recent games played by each player, formatted as JSON files. Games were removed if the opponent was the LoCM "Battlemage," a training AI designed to help players learn the game mechanics. Players cannot advance to the next league within the game without defeating the Battlemage, and doing so often requires specific strategies that the AI is attempting to teach. These games were removed because there was little variation in player styles against the opponent. Games were also removed if they ended within 64 turns. The first 62 game turns consist of drafting cards and dealing the initial hands, and turns 63 and 64 are the opening moves for each player. It is impossible to win a game within these two turns. Therefore a game lasting fewer than 64 turns indicates that one or both players encountered a fatal coding error during or immediately following the drafting phase. These games were removed for having no usable gameplay data.

The resulting games were then parsed to extract each player's starting hand, the subsequent cards they drew, and their actions. From this information, we could reconstruct the game state at each turn in the game. Each entry in the resulting dataset corresponds with one turn and contains the player ID for the player engaging in that turn, the game ID, the cards in the player's hand, the cards on the board, identified by whether they belong to the player or their opponent, and the player's action. This process is shown in Figure 1.

This produced nearly two million turns for 1,319 unique players (including the original 1000 players and their opponents) over 30,299 games. We then reduced this dataset to only include players who had between 1000 and 2000 overall turns to create the subset of aggregated players. The resulting set contains 228,487 turns for 184 unique players over 6,367 games. We chose to limit the aggregated set in this way for two reasons. First, the unaltered dataset is highly unbalanced to favor a few especially active players, which creates the potential for a model trained on the entire set to overpredict those players to arbitrarily improve accuracy. Second, because LoCM is a largely deterministic CCG, it is easily managed by hard-coded heuristics. In particular, bots can achieve high success by introducing ranked-choice preferences in the deckbuilding phase — selecting cards with strictly higher defense or attack points or strictly those cards with preferred abilities — and then randomly selecting actions from the allowable moves in the battle phase. As such, we chose to limit the aggregated data to mid-range players who foreseeably had a grasp of the game's rules and mechanics, but had yet to create heuristic methods to supersede opponents. We specifically chose the range from 1000 to 2000 turns as that was a relatively well-balanced section of the dataset that was reasonably close to the median (703 turns). A comparison of the dataset before and after pruning can be found in Figure 2. However, we used the top three most active players' data for individual modeling.

The datasets (aggregated and individual) were then split into train, validation, and test sets by randomly selecting 80% of the games for the train set and 10% each for the validation and test sets. The split was done to include entire games within a singular set, but split individual players across sets, so that the model could not predict a player's actions based exclusively on the game, as the state may remain relatively fixed for significant periods of time within a game.

Experiments

For the following experiments, we initially considered **RQ1** and **RQ2** in parallel and **RQ3** and **RQ4** in parallel. After initial experimentation, based on the results of **RQ2**, we ran additional experiments for **RQ1**, which will be discussed in detail. Across all four **RQ**s, our initial experiments used the State feature set (discussed in **RQ1**) unless we were explicitly testing feature sets and DistilBERT unless we were explicitly testing the model. We fixed the feature set and model to ensure that the results could be compared across **RQ**s and

[3]https://codingame.readthedocs.io/en/stable/api.html

Learning rate	Batch size	Weight decay
5×10^{-6}	16	0
1×10^{-5}	32	0.01
5×10^{-5}	64	0.1

Table 1: Tested hyperparameters.

to reduce statistical noise. DistilBERT was selected as our hypothesized most balanced model for speed and performance. For all experiments, we tested the hyperparameter values given in Table 1 and utilized the combination with the best performance on the validation set, averaged over four runs. We performed each training four times to ensure that any extremely good or poor results were not outliers.

RQ1: Varying Information to Identify Player

Feature Sets. To identify which features of the game state were most relevant to player identification, we performed an ablation study for **RQ1**. We started with the full information available to a player and removed aspects of that information to determine which features are meaningful from amongst the hand, board, and action. We first set the player ID as the gold label, then trained on four separate feature sets:

- State: The entire train dataset consisting of the player's hand, their board, and the opponent's board, augmented with the player's last action. As CCGs inherently have imperfect information, this is the most complete view of the game possible for a player and is equivalent to identifying an unknown player by watching a real-time game;
- Hand: A reduced train dataset consisting only of the player's hand, augmented with the player's last action. This is functionally equivalent to identifying an unknown player based on a known deck and action;
- Board: A reduced dataset consisting of the player's board and the opponent's board, augmented with the player's last action. This is functionally equivalent to identifying an unknown opponent based on their previous actions;
- No action: To examine the possibility that the models were simply learning to identify a player based on the cards they drafted, without considering their actions, this model selected a player using only the player's hand — which is randomly drawn from the drafted deck.

Model Parameters. For our initial experimentation, we trained DistilBERT-base-uncased[4] on all feature sets, using the aggregated dataset. We chose DistilBERT as opposed to larger LMs such as RoBERTa (Liu et al. 2019) for this task simply because it is faster and lighter (Sanh et al. 2020). Therefore it can be of more utility for games, which often can't run extremely large LMs due to speed or memory limitations. However, following the results of **RQ2**, particularly the unexpected performance of the random forest (RF) classifier, we ran additional experiments testing all feature sets for RoBERTa-base[5], RoBERTa-large[6], and the RF classifier.

[4]https://huggingface.co/distilbert-base-uncased

[5]https://huggingface.co/roberta-base

[6]https://huggingface.co/roberta-large

```
Hand: Scarlet Colossus, Worker Shellcrab, Slithering
Nightmare, Turta Knight, Venom Hedgehog, Gargoyle,
Board: Rootkin Drone, Steelplume Defender, Steelplume
Defender, Chameleskulk, Opponent board: Gargoyle
```

Figure 3: Example input string for State feature set.

The input to the model was the feature set being tested — with each feature labeled — conjoined into a single string. An example input string is given in Figure 3. For the RF classifier, the string was vectorized using TF-IDF vectorization. For the Transformer models, the string was tokenized using the model-specific Huggingface tokenizer. They were then finetuned for up to 30 epochs, but we stopped training when three consecutive epochs showed no F_1 improvement. All models used FP16 half-precision training for speed. We selected the output with the best F_1 score on the validation set for each hyperparameter combination and the hyperparameters with the best performance on accuracy and F_1, averaged over the four runs, to perform predictions on the test set. Wherever there was a discrepancy between F_1 and accuracy in selecting the final model, we prioritized accuracy. The specific hyperparameters for the Transformer models used in **RQ1** are given in Table 2.

Results. Results can be found in Table 3. The best overall performance was the State feature set, with approximately 17.35–25.74% accuracy. Additionally, regardless of model, all of the feature sets outperformed the random baseline in **RQ2**, indicating that they were actually learning about the input data instead of simply guessing, and the majority class baseline, indicating that they are capable of identifying unique features about all of the players, not just those play-

DistilBERT	Learning rate	Batch size	Weight decay
State	5×10^{-5}	16	0
Hand	5×10^{-5}	16	0.01
Board	1×10^{-5}	16	0
No action	5×10^{-5}	16	0.01

$RoBERTa_{base}$	Learning rate	Batch size	Weight decay
State	1×10^{-5}	16	0.01
Hand	1×10^{-5}	16	0.1
Board	1×10^{-5}	16	0.1
No action	1×10^{-5}	16	0

$RoBERTa_{large}$	Learning rate	Batch size	Weight decay
State	1×10^{-5}	16	0
Hand	1×10^{-5}	64	0
Board	5×10^{-6}	16	0.1
No action	1×10^{-5}	16	0.01

Table 2: Hyperparameters for feature sets.

Feature set	Model	Accuracy	F_1
State	DistilBERT	17.43%	0.1583
	RoBERTa$_{\text{base}}$	17.35%	0.1563
	RoBERTa$_{\text{large}}$	20.96%	0.1909
	RF ($d = 50$)	25.74%	0.2368
Hand	DistilBERT	9.91%	0.0924
	RoBERTa$_{\text{base}}$	9.40%	0.0872
	RoBERTa$_{\text{large}}$	10.73%	0.10034
	RF ($d = 50$)	11.85%	0.1127
Board	DistilBERT	6.98%	0.0629
	RoBERTa$_{\text{base}}$	6.83%	0.0614
	RoBERTa$_{\text{large}}$	7.93%	0.0723
	RF ($d = 50$)	10.52%	0.0990
No action	DistilBERT	7.95%	0.0758
	RoBERTa$_{\text{base}}$	7.50%	0.0707
	RoBERTa$_{\text{large}}$	8.95%	0.0848
	RF ($d = 50$)	9.23%	0.0900

Table 3: Identifying player with varying information.

ers that are most active. This implies that all features are relevant and the entire State feature set is preferred. However, in answer to **RQ1**, despite significantly outperforming arbitrary model baselines, which can be found in Table 4, even the State feature set does not show promising results in conclusively identifying a player from their actions, and other means of identification should be explored.

RQ2: Varying Models to Identify Player

Models. To create comparisons to our transformer-based approach, we first trained five simple classifier models using the aggregated dataset: Naïve Bayes, decision trees, random forest, and a linear support vector classifier (SVC) from the Scikit-learn Python library (Pedregosa et al. 2011), as well as XGBoost (Chen and Guestrin 2016). Despite being a binary classifier, we chose to utilize a linear SVC due to its potentially high performance and efficiency in classifying large numbers of samples. The linear version defaults to a one-versus-rest classification for multiclass problems. All of the remaining classifiers are multiclass by default. We were required to limit the maximum depth of the decision trees and random forest for practical memory purposes. We tested depths of 15, 25, and 50 and utilized the option with the best performance on each, averaged over four runs.

We then trained three transformer-based models on each task: RoBERTa$_{\text{base}}$, RoBERTa$_{\text{large}}$, and DistilBERT. RoBERTa was chosen for its strong capabilities in natural language understanding (NLU) (Liu et al. 2019). DistilBERT, while generally not as high-performing as RoBERTa, has considerably fewer parameters and increased computational speed (Sanh et al. 2020). The increased efficiency may be preferable depending on the functional reduction in performance, particularly because the input consists only of card names and LoCM has a relatively small card set.

Baselines. To evaluate whether our trained model was able to perform better than arbitrary models, we also used two baselines:

- Random: This model randomly selected a player from the entire space of players in the train set.
- Majority Class: To examine the possibility that the model was simply selecting the most active player in order to arbitrarily increase their accuracy, this model selected the player with the most turns in the entire set.

Model Parameters. The input to every model was the State input string from **RQ1**. For the baseline classifiers, the input string was vectorized using TF-IDF vectorization. For each transformer-based model, the input string was tokenized using the Huggingface tokenizer specific to that model. The Transformer models were then finetuned for up to 30 epochs, but we stopped training when three consecutive epochs showed no F_1 improvement. We selected the output with the best F_1 score on the validation set for each hyperparameter combination and the hyperparameters with the best performance on accuracy and F_1, averaged over the four runs, to perform predictions on the test set. Wherever there was a discrepancy between F_1 and accuracy in selecting the final model, we prioritized accuracy. The specific hyperparameters for the Transformer models used in **RQ2** are given in Table 2, under State.

Results. Results for **RQ2** can be found in Table 4. The best overall model was the random forest classifier with 25.74% accuracy and an F_1 of 0.2368, outperforming all three transformer-based models. However, all three Transformer models outperformed the random and the majority class baselines, as well as the other classifiers, suggesting that either a significant branching factor, using trees, or more complex computation, using LMs, is needed to differentiate between player features. Finally, although RoBERTa$_{\text{large}}$ outperforms DistilBERT, as expected, it took over 300% longer per epoch to train and 130% more epochs to converge to optimal performance. As such, we would not consider RoBERTa to be a significant improvement over DistilBERT, particularly when a simpler classifier (RF) performed better and faster than both. In answer to **RQ2**, a random forest classifier is most accurate at identifying an individual and performs significantly faster than its next-closest competitor.

Model	Accuracy	F_1
DistilBERT	17.43%	0.1583
RoBERTa$_{\text{base}}$	17.35%	0.1563
RoBERTa$_{\text{large}}$	20.96%	0.1909
Naïve Bayes	5.42%	0.0366
XGBoost	15.06%	0.1395
Decision tree ($d = 50$)	8.20%	0.0777
Random forest ($d = 50$)	25.74%	0.2368
LinearSVC	13.21%	0.1118
Random	0.551%	0.0050
Majority class	1.54%	0.0002

Table 4: Identifying player with varying models.

	Player 1	Player 2	Player 3
# of Games	6671	5030	978
# of Turns	208017	157134	36811

Table 5: Dataset information for players.

	Learning rate	Batch size	Weight decay
Player 1	5×10^{-5}	16	0
Player 2	5×10^{-5}	16	0.1
Player 3	1×10^{-5}	16	0.01

Table 6: Hyperparameters for individual models.

RQ3: Individual Modeling to Predict Action

Methodology. To predict the actions of an individual player, we took the datasets for the top three most active players, shown in Figure 2 and hereafter referred to as Players 1, 2, and 3, and set the action taken as the gold label. Despite its remarkable performance on **RQ2**, we chose not to train a random forest classifier for the action prediction tasks due to the significantly larger branching factor when classifying over 14,000 actions versus 184 players. For practical memory purposes, we would have had to limit the maximum depth to below 15, which would have considerably reduced the classifier's capabilities. We instead trained DistilBERT and compared the models' performances to the majority class baseline.

However, the random baseline has obvious shortcomings when assessing action prediction. In addition to the other data, the model is also presented with a set of cards in the game state at each turn, whether they be in the player's hand or on the board. The resulting action must be bounded by the cards that the player has available. As such, at each turn, we constructed a set of possible moves from the cards available in the game state. We then had a model, which we call the condensed random baseline, randomly predict an action from this reduced action space. As before, we performed each training four times. Player 1's "Pass" actions were also randomly downsampled by 67,612 instances to better balance the dataset. Dataset information regarding the individual players can be found in Table 5.

Model Parameters. The input to every model was the State input string from **RQ1**, sans the action label. The string was tokenized using the DistilBERT Huggingface tokenizer. Each model was then finetuned for up to 30 epochs, but we stopped training when three consecutive epochs showed no F_1 improvement. All three models used FP16 half-precision training for speed and tested the hyperparameters shown in Table 1. We selected the output with the best F_1 score on the validation set for each hyperparameter combination. However, given that players may have preferred moves, cards, or play styles, all three datasets were highly unbalanced to favor certain actions over others. We made every effort to ensure "Pass" was not an overwhelming option in any dataset, but true balance is infeasible. As such, for **RQ3**, we also accounted for the Matthews correlation coefficient (Matthews 1975), which is a preferred evaluation metric for imbalanced data (Chicco and Jurman 2020). For each dataset, we selected the hyperparameters with the best performance on at least two of our three evaluation metrics, averaged over the four runs, to perform predictions on the test set. The specific hyperparameters for **RQ3** are given in Table 6.

Results. Results for **RQ3** can be found in Table 7. All models outperform the condensed random and majority class baselines, suggesting that they are learning features of each individual's play style. Furthermore, all three models also outperform the model to predict actions from aggregate data, shown in Table 8. As expected, all models had relatively low F_1 values (a feature of data imbalance), but performed quite well when considering the Matthews coefficient. In answer to **RQ3**, all three models display relatively high performance at predicting an action for a known player, considering that there were over 14,000 possible actions.

RQ4: Group Modeling to Predict Action

Methodology. To predict the actions of a new or unknown individual based on the actions of an aggregated group of players, we trained DistilBERT using the aggregated train and validation datasets from **RQ1** and **RQ2**. We then performed predictions on the individual players' test sets. We also used the aggregated group's test set as a baseline.

Model Parameters. The input to every model was the State input string from **RQ1**, sans the action label. The string was tokenized using the DistilBERT Huggingface tokenizer. Each model was finetuned for up to 30 epochs, as before. The specific hyperparameters were determined by selecting the output with the best F_1 score on the validation set for each hyperparameter combination and the hyperparameters with the best performance on at least two of our three evaluation metrics, averaged over the four runs, to perform predictions on the test set. As such, every model had a learning rate of 5×10^{-5}, a batch size of 16, a weight decay of 0.1, and used FP16 half precision training for speed. This differs from **RQ1** because we are now predicting the action, not the player. We again report each model's accuracy, F_1, and Matthews coefficient.

		Accuracy	F_1	MCC
Player 1	Model	28.75%	0.0486	0.2855
	Random	5.57%	0.0302	0.0544
	Majority	0.7413%	0.000003	0.0
Player 2	Model	54.71%	0.2166	0.5449
	Random	7.22%	0.0360	0.0691
	Majority	2.42%	0.00002	0.0
Player 3	Model	28.37%	0.0350	0.2800
	Random	6.68%	0.0395	0.0645
	Majority	1.33%	0.00002	0.0

Table 7: Action prediction with individual modeling.

Model	Accuracy	F_1	MCC
Player 1	24.40%	0.0491	0.2423
Player 2	33.86%	0.1134	0.2859
Player 3	27.35%	0.0969	0.2711
Aggregate	26.95%	0.0595	0.2608

Table 8: Action prediction with group modeling.

Results. Results for **RQ4** can be found in Table 8. All models still outperform the condensed random and majority class baselines found in Table 7. This suggests that, even when a specific player has not been seen previously, some play style features can be effectively predicted from the common patterns of the group. However, only one individual model significantly outperforms the model to predict actions from aggregate data, and all three perform worse than models specifically trained to the individual, shown in Table 7. The models still had relatively low F_1 values compared to the Matthews coefficient. In answer to **RQ4**, all three models display strong performance at predicting an action for an unknown player with, at worst, approximately a 1 in 4 chance of correctly guessing the next move, but the performance would be improved by having some data specific to the player in advance.

General Discussion

For **RQ1**, it is clear that the entire game state is necessary to obtain the maximum possible accuracy in identifying a player. This is expected, given the small action space, as multiple players may have identical or highly similar decks and, therefore, similar hands at any given point. For these players, the only differentiating factor may be the moves they make when presented with a given environment.

As could also be expected, for **RQ2** large transformer-based LMs perform better than most other classifiers in identifying players from their CCG play styles. The notable exception is the random forest classifier, which shows overwhelming potential when provided with sufficient depth and performs faster than LMs, a necessary consideration for gaming. However, forests are functionally impractical for choosing between many actions, such as in **RQ3** and **RQ4**. Additionally, in relation to **RQ2**, larger Transformers with more parameters and layers identify players better at the cost of additional computational time. However, even the smallest Transformer model (DistilBERT) performed better than nearly all of the non-transformer methods it was compared against. Therefore, its performance loss against $\text{RoBERTa}_{\text{large}}$ is likely to be a justifiable tradeoff for speed and memory in many instances, particularly within gaming applications.

For **RQ3**, we can see that there is a strong potential to accurately predict a player's future actions, when trained on their common actions within a particular environment. In extreme cases, an action can be predicted with over 50% accuracy from a set of over 14,000 possible options. From **RQ4**, we also see that predicting a player's future actions using models tailored to that individual significantly outperform predictions based on aggregated data. However, the predictions made using such aggregated data still show a significant ability to predict moves.

However, in relation to **RQ4**, the models that predict an unknown player's actions from the common actions of a group perform the same or worse than predictions about the group's actions in most cases. This may be indicative of the higher-ranked individual players making creative and unconventional moves, which are not expected when considering only common actions. In this way, as we hypothesized, predicting an individual's future actions based on the majority's actions may cause unique features of that individual to be lost. Furthermore, the F_1 scores for group models were generally higher than the individual models. This substantiates our theory that the low scores were due to individual players having highly preferred and individualized actions (i.e., fewer true positives and more true negatives which were not captured by F_1), as opposed to the models being poor predictors (i.e., having more false positives and negatives which *were* captured by F_1). Further study is required to fully localize the causes of these phenomena.

The main drawback to using LoCM for this research, as well as its main advantage, is that the game has a significant player base with a relatively small action space, compared to other CCGs. This makes it simpler for AI agents to handle, allowing us to test complex algorithms and LMs easily. However, the limited action space also drastically diminishes one of the key benefits of CCGs, creativity and flexibility in play styles. Future work will focus on testing and adjusting these models to account for more complex CCGs, particularly those with natural language effects on the cards, such as Hearthstone or MtG. This would also drastically improve the model's robustness and generalizability to abstract resource allocation applications outside the realm of CCGs.

Finally, we plan to improve these results by exploring tree pruning strategies, to enhance the performance of the RF classifier, and introducing improved relative position embeddings, which have been shown to increase performance for LMs (Huang et al. 2020). An artifact of our dataset is that the state feature strings are ordered in the same way cards are presented to the player, from left to right. In CCGs other than LoCM, where players interact directly with the game, there may be biases toward playing certain cards based on their relative location in the player's hand, board, etc., which we would like to capture.

Acknowledgments

Part of this research was sponsored by the DEVCOM Analysis Center and was accomplished under Cooperative Agreement Number W911NF-22-2-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Adinolf, S.; and Turkay, S. 2011. Collection, Creation, and Community: A Discussion on Collectible Card Games. In *GLS'11: Proceedings of the 7th International Conference on Games + Learning + Society Conference*, 3–11. Association for Computing Machinery.

Adler, A. 2019. Keeping the Law of Magic: The Gathering. *Escapist Magazine*.

Agrawal, A.; Gans, J. S.; and Goldfarb, A. 2019. Exploring the Impact of Artificial Intelligence: Prediction versus Judgment. *Information Economics and Policy*, 47: 1–6.

Bertram, T.; Fürnkranz, J.; and Müller, M. 2021. Predicting Human Card Selection in Magic: the Gathering with Contextual Preference Ranking. In *Proceedings of the 2021 IEEE Conference on Games (CoG)*. Institute of Electrical and Electronic Engineers.

Bjørke, S. J.; and Fludal, K. A. 2017. *Deckbuilding in Magic: The Gathering Using a Genetic Algorithm*. Master's thesis, Norwegian University of Science and Technology.

Chen, T.; and Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. New York, NY: Association for Computing Machinery.

Chen, Z.; Amato, C.; Nguyen, T.-H. D.; Cooper, S.; Sun, Y.; and El-Nasr, M. S. 2018. Q-DeckRec: A Fast Deck Recommendation System for Collectible Card Games. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–8. Institute of Electrical and Electronic Engineers.

Chicco, D.; and Jurman, G. 2020. The Advantages of the Matthews Correlation Coefficient (MCC) Over F1 Score and Accuracy in Binary Classification evaluation. *BMC Genomics*, 21(6).

Conti, M.; and Tricomi, P. P. 2020. PvP: Profiling Versus Player! Exploiting Gaming Data for Player Recognition. In Susilo, W.; Deng, R. H.; Guo, F.; Li, Y.; and Intan, R., eds., *Information Security*, 393–408. Springer International Publishing.

Cowley, B.; and Charles, D. 2016. Behavlets: A Method for Practical Player Modelling Using Psychology-Based Player Traits and Domain Specific Features. *User Modeling and User-Adapted Interaction*, 26: 257–306.

de Mesentier Silva, F.; Lee, S.; Togelius, J.; and Nealen, A. 2017. AI-Based Playtesting of Contemporary Board Games. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, 13. Association for Computing Machinery.

Drachen, A.; Sifa, R.; Bauckhage, C.; and Thurau, C. 2012. Guns, Swords, and Data: Clustering of Player Behavior in Computer Games in the Wild. In *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, 163–170. Institute of Electrical and Electronic Engineers.

Ferguson, M.; Devlin, S.; Kudenko, D.; and Walker, J. A. 2020. Player Style Clustering without Game Variables. In *Proceedings of the 15th International Conference on the Foundations of Digital Games*, 66. Association for Computing Machinery.

Fields, L.; Marji, Z.; and Licato, J. 2022. Of a Different Persuasion: Perception of Minority Status and Persuasive Impact. In *Proceedings of CogSci 2022*.

Gabriel, I. 2020. Artificial Intelligence, Values, and Alignment. *Minds and Machines*, 30: 411–437.

Hoover, A. K.; Togelius, J.; Lee, S.; and de Mesentier Silva, F. 2020. The Many AI Challenges of Hearthstone. *KI - Künstliche Intelligenz (Artificial Intelligence)*, 34: 33–43.

Hoover, B.; Strobelt, H.; and Gehrmann, S. 2019. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformers Models. arXiv:1910.05276.

Hsieh, J.; and Sun, C. 2008. Building a Player Strategy Model by Analyzing Replays of Real-Time Strategy Games. In *Proceedings of the 2008 IEEE International Joint Conference on Neural Networks*, volume 10, 3106–3111. Institute of Electrical and Electronic Engineers.

Huang, Z.; Liang, D.; Xu, P.; and Xiang, B. 2020. Improve Transformer Models with Better Relative Position Embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 3327–3335. Association for Computational Linguistics.

Kowalski, J.; and Miernik, R. 2020. Evolutionary Approach to Collectible Arena Deckbuilding using Active Card Game Genes. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, 1–8. Institute of Electrical and Electronic Engineers.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Matthews, B. W. 1975. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure*, 405: 442–451.

McIlroy-Young, R.; Wang, R.; Sen, S.; Kleinberg, J.; and Anderson, A. 2021. Detecting Individual Decision-Making Style: Exploring Behavioral Stylometry in Chess. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 24482–24497.

McIlroy-Young, R.; Wang, R.; Sen, S.; Kleinberg, J.; and Anderson, A. 2022. Learning Models of Individual Behavior in Chess. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1253–1263. Association for Computing Machinery.

Miernik, R.; and Kowalski, J. 2022. Evolving Evaluation Functions for Collectible Card Game AI. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022)*, volume 3, 253–260.

Moura, D.; el-Nasr, M. S.; and Shaw, C. D. 2011. Visualizing and Understanding Players' Behavior in Video Games: Discovering Patterns and Supporting Aggregation and Comparison. In *ACM SIGGRAPH 2011 Game Papers*, SIGGRAPH '11. New York, NY, USA: Association for Computing Machinery.

Parizi, R. M.; Dehghantanha, A.; Choo, K.-K. R.; Hammoudeh, M.; and Epiphaniou, G. 2019. Security in Online Games: Current Implementations and Challenges. In Dehghantanha, A.; and Choo, K.-K. R., eds., *Handbook of Big Data and IoT Security*, 367–384. Springer International Publishing.

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

Saariluoma, P. 1995. *Chess Players' Thinking: A Cognitive Psychological Approach*. London: Routledge.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2020. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. *CoRR*, abs/1910.01108.

Shaker, N.; Togelius, J.; Yannakakis, G. N.; Weber, B.; Shimizu, T.; Hashiyama, T.; Sorenson, N.; Pasquier, P.; Mawhorter, P.; Takahashi, G.; Smith, G.; and Baumgarten, R. 2011. The 2010 Mario AI Championship: Level Generation Track. *IEEE Transactions on Computational Intelligence and AI in Games*, 3(4): 332–347.

Sutrop, M. 2020. Challenges of Aligning Artificial Intelligence with Human Values. *Acta Blatica Historiae et Philosophiae Scientiarum [Baltic Journal of the History and Philosophy of Science]*, 8(2): 54–72.

Turkay, S.; Adinolf, S.; and Tirthali, D. 2012. Collectible Card Games as Learning Tools. *Procedia - Social and Behavioral Sciences*, 46: 3701–3705.

Tuyls, K.; Omidshafiei, S.; Muller, P.; Wang, Z.; Connor, J.; Hennes, D.; Graham, I.; Spearman, W.; Waskett, T.; Steel, D.; Luc, P.; Recasens, A.; Galashov, A.; Thornton, G.; Elie, R.; Sprechmann, P.; Moreno, P.; Cao, K.; Garnelo, M.; Dutta, P.; Valko, M.; Heess, N.; Bridgland, A.; Pérolat, J.; De Vylder, B.; Eslami, S. M. A.; Rowland, M.; Jaegle, A.; Munos, R.; Back, T.; Ahamed, R.; Bouton, S.; Beauguerlange, N.; Broshear, J.; Graepel, T.; and Hassabis, D. 2021. Game Plan: What AI can do for Football, and What Football can do for AI. *Journal of Artificial Intelligence Research*, 71: 41–88.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Plosukhin, I. 2017. Attention is All You Need. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA, USA.

Vieira, R.; Tavares, A. R.; and Chaimowicz, L. 2020. Drafting in Collectible Card Games via Reinforcement Learning. In *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*, 54–61. Institute of Electrical and Electronic Engineers.

Xue, S.; Wu, M.; Kolen, J.; Aghdaie, N.; and Zaman, K. A. 2017. Dynamic Difficulty Adjustment for Maximized Engagement in Digital Games. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, 465–471. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee.

Yannakakis, G. N.; and Togelius, J. 2015. A Panorama of Artificial and Computational Intelligence in Games. *IEEE Transactions on Computational Intelligence and AI in Games*, 7(4): 317–335.

Yao, Z.; Shi, T.; Li, S.; Xie, Y.; Qin, Y.; Xie, X.; Lu, H.; and Zhang, Y. 2022. Towards Modern Card Games with Large-Scale Action Spaces through Action Representation. In *Proceedings of the 2022 IEEE Conference on Games (CoG)*, 576–579. Institute of Electrical and Electronic Engineers.

Zohaib, M. 2018. Dynamic Difficulty Adjustment (DDA) in Computer Games: A Review. *Advances in Human-Computer Interaction*, 2018: 5681652.