# I am a legend: hacking Hearthstone using statistical learning methods

Elie Bursztein
*Google*
*elieb@google.com*

***Abstract*—In this paper, we demonstrate the feasibility of a competitive player using statistical learning methods to gain an edge while playing a collectible card game (CCG) online. We showcase how our attacks work in practice against the most popular online CCG, *Hearthstone: Heroes of World of Warcraft*, which had over 50 million players as of April 2016. Like online poker, the large and regular cash prizes of *Hearthstone*'s online tournaments make it a prime target for cheaters in search of a quick score. As of 2016, over $3,000,000 in prize money has been distributed in tournaments, and the best players earned over $10,000 from purely online tournaments.**

**In this paper, we present the first algorithm that is able to learn and exploit the structure of card decks to predict with very high accuracy which cards an opponent will play in future turns. We evaluate it on real *Hearthstone* games and show that at its peak, between turns three and five of a game, this algorithm is able to predict the most probable future card with an accuracy above 95%. This attack was called "game breaking" by Blizzard, the creator of *Hearthstone*.**

## 1. Introduction

Over the last 20 years, since the inception of *Magic the Gathering* [34], collectible card games (CCGs) have emerged as one of the most pervasive forms of contemporary game play. In North America alone, sales were estimated to be above $800 million in 2008 [9]. Given the popularity of CCGs, it is not surprising that in the last few years, with the rise of mobile devices and tablets, computer-based CCGs have emerged as one of the largest parts of the e-sport scene. In particular, the most popular online CCG, *Hearthstone: Heroes of World of Warcraft* [11], which had over 50 million players as of April 2016 [26], is one of the most profitable games for e-sport players. As of 2016, over $3,000,000 in prize money has been distributed in Hearthstone tournaments since its creation in 2014 [12]. The ranking shows that the best players earned over $10,000 from purely online tournaments and over $200,000 in total [12]. Given *Hearthstone*'s popularity, the amount of money at stake in an online tournament and that most offline tournaments with prizes have an online qualification phase, it is important to understand how players can attack *Hearthstone* to gain an unfair edge.

**Contribution.** To address this need, this paper, to the best of our knowledge, describes the first security analysis of an online CCG and it is the first to develop statistical learning attacks [17] against CCG games. This paper presents the first algorithm that is able to learn and exploit the structure of card decks to predict with very high accuracy which cards an opponent will play in future turns. Using a dataset of 50,000 game replays collected in May 2014, we demonstrate that our statistical learning attack works in practice by successfully using it against *Hearthstone*. At its peak, between turns three and five of a game, our algorithm is able to predict the most probable future card with an accuracy above 95%. As recounted in the ethics section later in the introduction, the effectiveness of this attack was validated by the creator of *Hearthstone*, Blizzard Entertainment, which deemed it "game breaking."

**Ethics.** We reached out to Blizzard Entertainment to disclose our findings prior to publication but we did not get any response. However, after we presented some of the findings reported in this paper at a famous hacking conference, Blizzard reached out to us. During the conversation, one of the main game designers acknowledged that the deck prediction attack demonstrated was indeed game breaking and asked us to not publish our prediction tool or dataset. Following Blizzard's request, we agreed not to release the code for our prediction tool or the full dataset.

**Outline.** The remainder of the paper is organized as follows. In Section 2, we provide the background needed to understand the state of game security research, what a CCG is and how a *Hearthstone* game is played out in particular. In this section, we also discuss our threat model and how our dataset was created. In Section 3, we focus on predicting which cards the opposing player will play in future turns. We describe how our prediction attack works and report the results of our evaluation against our dataset. In Section 4, we briefly evaluate the game's most predictive metrics using a decision tree to shed light on the most important factors that players need to consider while playing *Hearthstone*. Finally, we conclude in Section 5.

## 2. Background

In this section, we provide the necessary background. We start by discussing the state of game security research. Then we briefly explain what collectible card games (CCGs) are. Next, we describe how a Hearthstone game is played, its win conditions and where the game complexity stems from. Next, we discuss the threat model and attack types considered in this paper. Finally, we explain how the dataset used in this paper was collected.

**Security research, machine learning and games.** Researching how to exploit games and detecting cheaters using machine learning has the long-standing tradition of being a very prolific security research topic. Detecting MMORPG bots, including *World of Warcraft* ones, was studied in [16], [22]. A technique to build a map hack and defend against them was presented in [5]. Using machine learning to detect aim bots for *Unreal 3* was presented in [15]. The research presented in [35], [1] extended the use of machine learning for aim-bot detection to other FPSs (First Person Shooters). Applying fuzzing to attack online games and in particular *League of Legends* was presented in [6].

**Collectible Cards Games.** Also known as trading card games (TCGs),[1] CCGs are broadly defined as turn-based strategy games that use a pool of cards (usually hundreds) that are collected by players. These games rose in popularity in 1993 with the release of *Magic The Gathering* [34] even though the origin of the genre can be traced back to a baseball card game published in 1904 [9]. What makes CCG different from traditional card games is that players cannot buy all the cards at once; they need to collect them by buying *boosters* to expand their collection. Cards have different rarity and a booster randomly includes a few common cards (the less rare ones) and at least one card that is rare or "better". *Hearthstone* has five levels of rarity: *sets* cards (which are given for free to players), *common* cards, *rare* cards, *epic* cards and *legendary* cards. Each *Hearthstone* booster comprises five cards selected at random with at least one card that is rare or better. *Hearthstone* is considered a "live" CCG, which means that its creator, Blizzard, keeps adding new cards to it in *expansion sets*, which are released regularly. During a game, players play with decks that they constructed from their personal collection of cards. The construction of a deck is strictly regulated. The number of cards a deck contains is usually restricted (30 exactly for *Hearthstone*) as is the number of copies of a given card that can be played (two of the same kind for *Hearthstone*, except for legendary cards, which are restricted to a single copy).

Beyond their recreational and competitive purposes, CCGs are also used for education, including teaching kids about diseases [28] or professionals about computer security [10].

1. The acronyms CCG and TCG are used interchangeably and they coexist due to licensing issues. In this paper, we will exclusively use the CCG acronym as *Hearthstone* is marketed as an online CCG by Blizzard.

From a theoretical standpoint, CCG games are viewed as imperfect information games with randomness. Previous research on CCGs focused on either optimizing how to collect cards [3] or taking a theoretical approach to determining the best way to play the game [8].

To the best of our knowledge, no prior work exists on predicting what cards the opposing player will play in future turns, on finding over-powered cards or on using machine learning to predict a game's outcome. However while there is no formal research on it, we note the existence of bots [2] that are designed to play the game instead of humans for leveling purposes. Those bots work by analyzing the current board state and attempting to make the best play possible. The only documented attempt to exploit a game design flaw using statistical learning to gain a competitive edge was the use of a genetic algorithm to find an optimal build order in *Starcraft 2*. This led to the discovery of a very efficient and yet counterintuitive build order called the "7 roach timing-attack" [4].

**Hearthstone.** *Hearthstone: Heroes of Warcraft* is a free-to-play digital CCG that was initially released in early 2014 by Blizzard entertainment [11]. As of May 2016, *Hearthstone* has more than 50 million player accounts registered worldwide [26], making it the most popular online CCG. As of 2016, over $3,000,000 in prize money has been distributed in Hearthstone tournaments since its creation [12]. The ranking shows that the best player earned over $10,000 from purely online tournaments and over $200,000 in total [12]. This makes Hearthstone one of the top ten most profitable games for pro-players and a target of choice for cheaters, given the large online cash prizes and that most offline tournaments will financially reward each player who qualifies to the offline phase through the online qualification phase.


Figure 1. Screenshot of the *Hearthstone* game board with the most important features illustrated

A *Hearthstone* game, as other CCGs, is a turn-based match between two opponents represented by their chosen "hero," an important character from Warcraft lore that has 30 health points, as depicted in Figure 1.

At the start of the game, each player draws three cards from their deck, which comprises 30 cards selected by the player from their card collection before the game started.At the start of their turn, the player draws a new card from their deck. While most cards are available to heroes of any class, a substantial portion is limited to a specific class, giving each hero its own strengths and unique possibilities. Each card or hero power requires the player to spend a specific amount of *mana* to play it, strategically limiting the player's options.

At the beginning of turns one through to ten, the player's mana pool is replenished and increased by one, allowing them to play more powerful cards as turns pass. After turn ten, the player's mana pool is replenished and capped to ten mana points. We note that for game balancing reasons, the player who starts second gets an extra card from their deck to equalize the draw count and a "bonus" card, called the coin, which gives them a free mana point when played. A Hearthstone match ends when one or both players have reached zero health, or choose to concede.

Building the best deck possible is an essential skill and many archetypes of decks exists. Deck archetypes are characterized by the distribution of the mana cost of the cards they contain, which is referred to as the *mana curve*. A low mana curve (having many cheap cards) is called an *aggro deck* and is usually very good early in a game. A balanced curve is referred to as a *mid-range deck* and a deck that contains mostly powerful cards with high mana cost is known as a *control deck*. Last but not least, the *combo deck* refers to a deck that leverages very powerful synergy between specific cards to kill the opposing player, usually in a single turn. Blizzard aggressively balances the game by *nerfing* decks (which means making them less efficient) that exhibit a too high win rate or "make the game not fun," which usually means combo decks that can kill an opponent in a single turn [31].

Hearthstone offers four types of card: *minions*, which are creatures that exist on the board as visible in Figure 1. A minion has a set of attack and health points and sometimes special abilities, which are described in the text of the card. A minion's current attack points are displayed in a yellow circle located on the bottom left of the card, whereas a minion's current health is represented as a red drop located on the bottom right of the card. The second type of card comprises *spells*, which are abilities directly played from the player's hand. The third type of card comprises *weapons*, which can be equipped one at a time by the player's hero to give it the ability to attack for a defined amount of attack points and time. Finally, *secrets* are cards that are played on the board but are hidden behind a question mark. They are triggered by specific conditions including if the opponent plays a minion or an opponent's creature tries to attack the player's hero.

**Threat model.** The threat model considered in this paper is the passive attacker model, which assumes that the attacker has access to the state of the game and the ability to observe all its network traffic. We also assume that the attacker has access to a vast set of previous game replays, which can be used to apply various statistical learning methods. This attacker model was chosen over the active attacker model because it matches what a cheater can realistically do during competitive Hearthstone games that are observed by referees, casters and a crowd that will immediately spot any kind of active tampering. In an online tournament, the referees see only the game screen via the in-game interface and the players' faces through their webcam, which leaves players full latitude to run the tools of their choice before and during games. Such passive attacks are far from being theoretic: for example, some players, due to screen reflections on the webcam feed, have been caught watching a casting video stream during a tournament to know their opponent hands [24].

**Attack types.** In this paper, we focus on how an attacker can use a machine-learning algorithm during the game to predict the most likely cards that the opponent will play in future turns. As in any strategy game, anticipating the other player's moves is essential for winning, as it informs the player's decision-making process. For example, a player routinely has to decide between playing an extra creature to further a tempo advantage or given the likelihood that the opponent will play a board-clear, to hold off to reduce the value that the opponent will gain from their board-clear. Such decisions made under time (a turn is 75 seconds) and tournament pressure are very difficult. This is even more true when the opponent plays secrets, which are cards that are in play but not revealed, as the type of secret played and its trigger condition need to be evaluated carefully. As recounted in the introduction, the effectiveness of our attacks was validated by *Hearthstone* game designers themselves to the point where they called the predicting attack "game breaking" and asked us to not publish our prediction tool.

**Attack generality.** In this paper, we analyze *Hearthstone* as it was in April and May 2014. Since then, many cards have been added and some cards have been changed (nerfed). While those changes affect the ranking of the most powerful cards currently available and the most popular decklists, our attacks and methods remain equally applicable, as they target the core mechanics of the game not specific cards or bugs. For the same reasons, our attacks work against any CCG.

**Dataset.** For the paper, we use a dataset of 50,000 anonymous replays that came from hundreds of players who used a game-recording tool. Players use this tool to keep track of and analyze their performance. As Blizzard does not provide a replay features, game replays were recorded from the game logs. As a result, our replays contain only partial information regarding the opposing player's deck: we only get a log when a card is drawn, played, activated or killed. If a card is drawn but never played, we know only that the opponent had an unknown card in their hand. However, this is not an issue because we are interested only in predicting the opponent's next move based on what the player knows. Replays were collected between April and May 2014.

## 3. Predicting an opponent's future plays


Figure 2. Our prediction tool in action. Predicted cards are displayed on the bottom left.

In this section, we demonstrate how we built a tool, as illustrated in Figure 2, that can predict what the opposing player will play in future turns based on the previous cards they played. These predictions give the player an edge because anticipating the other player's moves is essential for winning in *Hearthstone*, since this guesswork informs the player's decision-making process. For example, players routinely wonder if they should play an extra creature to further a tempo advantage or given the likelihood that the opponent will play a board-clear, whether they should hold off to reduce the value that the opponent will gain from their board-clear. Being able to figure out under time (a turn is 75 seconds) and tournament pressure the optimal play in these complex situations is what separates the best players from the rest.

The average win rate on the online ladder for the best decks for good players is around 53% while pro-players commonly reach 70% with the same decks [20]. One may think that a drawback of this attack is that it requires observing quite a few played cards before making good predictions. However, our evaluation shows that that is not the case: by turn two, our algorithm already makes accurate predictions and is able, for instance, to predict two cards that the opponent will play in the future with a success rate over 50%: 66.3% for the best prediction and 44.1% for the seond best prediction.

This section is structured as follows: first we discuss the underlying reasons that make Hearthstone predictable, then we explain how we are able to read game events programmatically. Next, we present our machine-learned ranking algorithm. Finally, we evaluate our tool on a dataset of 50,000 replays and show that it has up to a 96.2% success rate at predicting a card that will be played by the opponent in the future when its accuracy peaks between turns three and five.

### 3.1. Why is Hearthstone predictable?

*Hearthstone* in its original release, the one considered in this paper, had 465 cards that players could choose from to build their deck of 30 cards. It has a limit of two copies of a given card (see Section 2 for more background). This hypergeometric distribution [33] of choosing 30 cards out of 930 leads theoretically to a space of potential decks that is extremely large: roughly $4.7^{2358}$ possible decks. However, in practice the decks played have a lot of predictable structure due to the following reasons: First, some cards are restricted to a specific hero class. Second, by design many cards are meant to work best when played with other very specific cards. Third, a significant number of cards are just bad or under-powered, which means that they are almost never played in competitive games. Last but not least, a huge fraction of players rely on *netdecking* [25], which is the act of using a deck made by someone else (usually a pro-player) and found online.

As of 2015, the netdecking phenomenon has become so mainstream that weekly blog posts provide and rank the top decks currently played [30]. As a result, the distribution of cards that are used by players is heavily skewed. For example, the site HearthPwn, which has a very large set of decklists supplied by players, reports that the card *Keeper of the Groove* is used in 76.89% of druid decks [18]. Similarly, the *Northshire Cleric* is used in 84.16% of priest decks [19]. While self-reported decks may not accurately reflect the real distribution of what cards are played, the aforementioned statistics still convey that there is broad agreement among players on what are the must played cards. Our tool exploits this underlying structure by learning from game replays which cards are played significantly more often than others and which cards are often played together.

### 3.2. How to read the game state

Before delving into how our prediction algorithm works, it is worth explaining how our tool gets access to a player's actions, including which cards were played and how many cards were drawn. As Blizzard does not provide an API for this, over time, four main ways have been devised by the Hearthstone community. The first relies on DLL injection [13] and hooks the main functions of the game. This approach allows one not only to get a player's actions but also generates them (e.g., play a card). This makes it the method of choice for *Hearthstone* bots [2] that automate game playing for leveling purposes. Unsurprisingly, this method is also the one that Blizzard actively looks for and bans using its Warden [32], as it is used for these forbidden automation and map hacks [5]. Two alternative approaches were developed in the early days of Hearthstone to track games: the first relies on network traffic sniffing to capture players' actions at the network level. The second one uses image recognition techniques and continuually takes game screenshots.

Both approaches were deprecated in May 2014 when reverse engineering of the game client revealed it was possible to turn on debug logs that would provide enough information to track game actions [14]. Since then, every game tracker, including our prediction tool, moved to this method as it does not violate Blizzard's terms of service, like network sniffing, and is more reliable than image recognition. After setting *Hearthstone* in debug mode, our tool continually reads the logs and parses the entries to recreate an internal representation of the game state.

### 3.3. Machine-learned ranking algorithm

As outlined earlier, the goal of our tool is to provide a ranked list of cards that are in the opponent's deck and are yet to be played, based on cards previously played. To achieve this goal, we built a machine-learned ranking system [7] that given a set of previous cards returns a ranked list of cards by their likelihood of being played. An example of our tool output is visible on the bottom left of Figure 2.


Figure 3. *Hearthstone* card sequence as a bag of bigrams

Our machine-learned ranking system uses a modified version of Markov chains [23], where card co-occurrences are modeled as a bag of bigrams [27]. As illustrated in Figure 3, extracting a bag of bigrams from a card sequence involves extracting all possible combinations of card pairs regardless of the order they were played. While experimenting with various data models, we found that using a model requiring more memory (e.g., trigrams, 4-grams, ...) or even adding more structure to the model (e.g., using regular bigrams to learn the exact card sequence) yielded worse results. We provide an analysis of the reasons behind this counterintuitive result and a comparison between the performance of bigrams and trigrams in the evaluation part of this section. Until then, we focus on describing the approach that performs the best: modeling card sequences as a bag of bigrams.

**Learning phase.** To learn its model, our tool extracts for each game replay the sequence of cards played by the opponent and constructs a bag of bigrams. Those bags of bigrams are then used to construct the *occurrence table*, which for a given card returns the list of co-occurred cards with the number of times this co-occurrence happened. At prediction time, this allows it to find the popular pair/combos efficiently, as they are the ones that have the largest number of co-occurrences.


Figure 4. Prediction phase example

**Prediction phase.** During a game, as depicted in Figure 2, each time the opponent plays a card, the algorithm outputs a list of potential next cards ordered by their probability of being played. Under the hood, as exemplified in Figure 4, the tool leverages the occurrence table constructed in the learning phase to construct a ranked list of cards as follows: First, the algorithm uses the occurrence table to retrieve all cards that were co-played with the cards already played and the frequency of those co-occurrences. In our example, depicted in Figure 4, the algorithm looks up the cards that were co-played with *Deadly Poison* and *Shiv*. These lookups led it to retrieve the *Blade Flurry* card, which was co-played 350 times with *Deadly Poison* and 400 times with *Shiv*. It also retrieves the *Fan of Knives* card, which was co-played 500 times with *Deadly Poison*, and *Amani Berserker*, which was played 400 times with *Shiv*. Next, the algorithm sums up card appearances, removes cards already played and reverse sorts the remaining potential cards based on the frequency of their co-occurrences. In our example, *Blade Flurry* ends up being the most seen/probable card with a count of 750, *Fan of Knives* is second with a count of 500, and *Amani Berserker* is last with a count of 400. Cards with a frequency below a certain threshold, *Amani Berserker* in our example, are cut off to remove improbable or noisy cards. While not depicted in the example for clarity, before returning the list, the algorithm normalizes the card count by the total count, such that the returned list has a percentage associated with each card and not the raw count.

### 3.4. Evaluation

For the evaluation, we used 45,000 games from our dataset for training and 5000 games for testing. See Section 2 for the dataset collection methodology. For each test game, the algorithm was asked at the end of each turn to predict what would be the top ten cards that would be played in future turns.

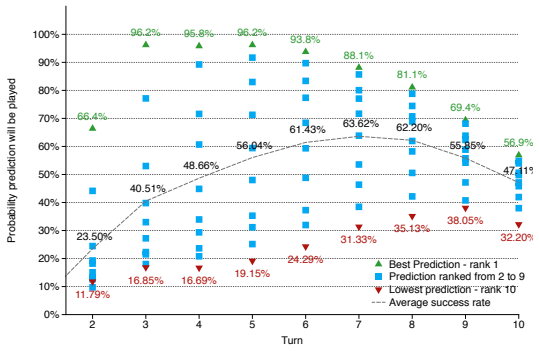


Figure 5. Success rate for top ten prediction bucketed by turn

We then compared each prediction with the cards that were effectively played in the upcoming turns and marked a prediction as correct if the card was effectively played. The result of our evaluation is summarized in Figure 5. To evaluate if our ranking algorithm is successful at ranking the cards with the highest chance of being played higher, we evaluated the success rate of each rank independently. Additionally, in our evaluation, to account for the fact that as turns pass, the amount of information (number of cards played) available to the algorithm increases and that the number of cards yet to be played decreases, we bucketed the result by turn. This allows us to understand how the tradeoff between more information and having a smaller set to predict from plays out.

The first observation we can make is that the ranking function performs as intended. In every case, the highest ranked prediction is the one with the highest success rate, as visible in Figure 5. Similarly, the prediction ranked number two always has a better success rate than the one ranked below it and so forth. The only misranking happens for the lowest prediction (rank tenth) at turn 2, where the prediction ranked tenth has a higher success rate than the one ranked ninth. This is likely due to the fact that the algorithm has seen very few cards at that stage (likely one) and, therefore, cannot separate the predictions.

Secondly, we observe that the algorithm is very successful at predicting what the opponent will play, with the best prediction having a *96.2%* success rate for turns three and five. Throughout the first ten turns of the game, the best prediction has a success rate above 56.9%, which shows that the algorithm successfully learned the hidden correlations between cards and is able to use it to make accurate to very accurate predictions reliably throughout the game.

Thirdly, we see that indeed the tradeoff between having more information and a smaller set of cards yet to play to predict from does affect the algorithm's performance. In the first seven turns, the average prediction success rate steadily climbs from 23.5% to 63.2%.

Then as the game progresses and the number of turns left and, therefore, the number of cards the opponent will play are both getting smaller and smaller, the accuracy drops back and reaches 47.5% by turn ten. This shows that past a certain point, more information cannot counterbalance the fact that the pool of cards yet to be played is getting too small.

Finally, we note that the delta between the best prediction success rate and the worst one decreases steadily as the turns pass. This is again expected because the amount of mana the player has has increased, which leads to a greater number of valid plays as turns pass.

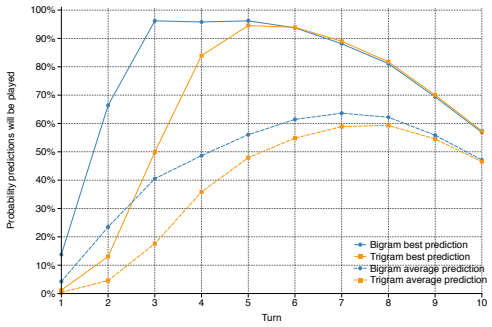


Figure 6. Bigram versus trigram success rate

During our experimentation, we attempted to use a model requiring more memory by using longer $n$-grams. However, none of these alternative models yielded better results than bigrams. Similarly, every attempt to use a more constrained model, such as strict bigrams, in the hope of exploiting further the relations between cards, yielded worse results than our bag of bigrams. This negative result is best illustrated by comparing the success rate of the algorithm when using bigrams and trigrams. As visible in Figure 6, the best trigram-based prediction is significantly worse for turns one to five and marginally better for turns six to ten. However, the average prediction success rate is worse for every turn and by quite a huge margin for the early turns.

This is a counterintuitive result because in other cases, including predicting words typed, spell-checking and predicting almost any type of sequence [21], using a model requiring more memory leads to increased accuracy. This result is partially explained because players draw cards at random, which adds baseline noise to each game. Also the order in which a player plays cards during a given turn may not influence the outcome, which makes all sequence permutations likely to see some play. Another potential reason that favors having a "laxer" model is that players run decks with slight variations, such as including one or two copies of a given card. That being said, it may be possible that a model requiring even more memory, such as LSTM/RNN [29], and using significantly more data, could outperform the current approach.

Given that our tool is already good enough at helping an attacker figure what their opponent will play in future turns, this investigation is left as an open question.

## 4. Predictive Features

In this section, we briefly explore the use of machine learning to predict the outcome of a *Hearthstone* game. Understanding the most predictive metrics sheds light on what are the optimal strategies for playing *Hearthstone* and can help players to make better decisions. Over the years, the Hearthstone community has come up with a few metrics that are routinely used to predict who will win a game. However, until this work, there was no formal analysis of how good those metrics are at predicting game outcomes. Drawing on our experience, forum discussions and game analysis by pro-players, we compiled and a formalized the following list of metrics:

- **Mana efficiency**: The mana efficiency metric is the difference between how much mana the player spent versus how much their opponent spent. So, if the player has spent 4 mana and their opponent has spent 2, then the mana efficiency will be $4-2=2$. Conversely, if the opponent spent 6 mana and the player spent only 2 mana, the mana advantage is $2-6=-4$. Given that according to our card model (Section 3), the power of a card is (almost) perfectly reflected by its mana price, we were expecting that this metric would be the most predictive. As reported earlier, this is the case, which supports even further the validity of our model and its assumptions.
- **Board mana advantage**: The board mana advantage is computed as the difference between the sum of the mana for the cards that the player has on the board versus the sum of the mana for the cards the opponent has on the board. According to our model, this metric should be a better predictor than the sheer number of creatures on the board, as our model implies that a 3-mana creature is more powerful than two creatures that cost 1 mana.
- **Board advantage**: The board advantage measures the difference between how many minions the player has versus how many minions the opponent has in play.
- **Draw advantage**: The draw advantage measures who drew the most cards by calculating the difference between how many cards the player has drawn so far and how many cards their opponent has drawn. For instance, if the player has drawn four cards and their opponent has drawn two, then the draw advantage is $4-2=2$. Our model predicts that the value of having a card is a fraction of the card's mana cost, so this metric should be a weak indicator.
- **Hand size advantage**: This metric refers to how many cards the player has in hand versus how many cards the opponent has. It is a somewhat strange metric, but many casters and players refers to it, so it was included. If the player has two cards in their hand and the opponent has three cards, then the hand size advantage is $2-3=-1$.

To evaluate which metric has the most predictive power, we used the standard random forest feature selection algorithm, as it outputs a ranked list of features with their

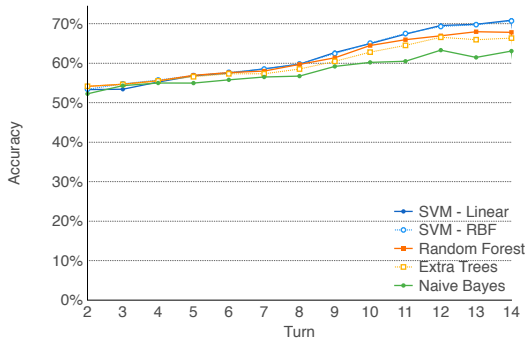


Figure 7. Various algorithms accuracy at predicting the winner

predictive power. To do this, we modeled games as a binary classification problem and used our metrics as features. The feature vectors are constructed by outputting the value of each metric at the end of each turn for the first 14 turns, the cumulative value of the features at the end of each turn for the first 14 turns of the game, and the value and cumulative value of the metrics at the end of the game. We then trained a random forest on our dataset using 45,000 games and tested it on 5000 games. To be thorough, on top of training a random forest algorithm, we also trained several classifiers, namely a naive Bayes, an extra tree, a support vector machine (SVM) with a linear kernel, and a SVM sigmoid kernel grid search. The accuracy of the various algorithms on our dataset is reported in Figure 7. All classifiers got better as turns passed, and the accuracy of each classifier is above the baseline of 50%. The accuracy of the random forest is very close to the best classifiers, which are based on SVM. This gave us confidence that using it to rank features will give meaningful results.

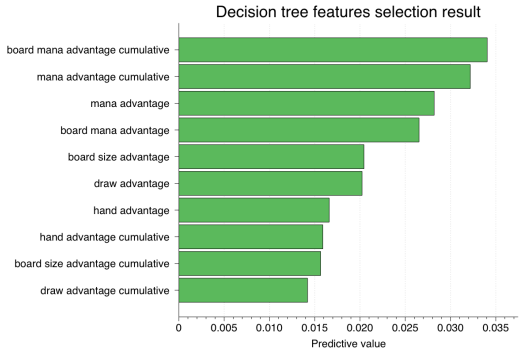


Figure 8. Feature prediction power

The results for the feature selection algorithm are reported in Figure 8. As visible in this chart, the cumulative metrics that track mana usage (mana advantage and board mana) are the most predictive. We expected that the mana advantage metric would be the best, instead of the board advantage, but with hindsight this makes sense.

A player's minions that are still in play after a turn gives them a lasting advantage. While the board size advantage, draw advantage and hand advantage are significantly less predictive, the results confirm the community's intuition that those metrics are important. We note that the metric selected by the algorithm as most relevant, was the metric at the end of the game not the metric at a particular turn, which suggests that there is no turn that is more important than another, unlike what some of the Hearthstone community believes, which is that turns two and three are critical.

## 5. Conclusion

In this paper, we demonstrated the feasibility for a competitive player to use statistical learning methods to gain an edge while playing collectible card games online. We showcased how our attacks work in practice against the most popular online CCG *Hearthstone: Heroes of World of Warcraft*, which had over 30 million players as of May 2015. We devised a statistical learning algorithm to attack *Hearthstone*. It learns and exploits the structure of card decks to predict with very high accuracy what an opponent will play in future turns. At its peak, between turns three and five of the game, this algorithm is able to predict the most probable next card with an accuracy above 95%.

## References


[1] Hashem Alayed, Fotos Frangoudes, and Clifford Neuman. Behavioral-based cheating detection in online first person shooters using machine learning techniques. In *Computational Intelligence in Games (CIG), 2013 IEEE Conference on*, pages 1–8. IEEE, 2013.

[2] Arstechnica. Hearthstone bot maker closes shop after blizzard crackdown. http://arstechnica.com/gaming/2014/11/hearthstone-bot-maker-closes-shop-after-blizzard-crackdown/, Nov 2014.

[3] Robert A Bosch. Optimal card-collecting strategies for magic: The gathering. *The College Mathematics Journal*, 31(1):15, 2000.

[4] Louis Brandy. Using genetic algorithms to find starcraft 2 build orders. http://lbrandy.com/blog/2010/11/using-genetic-algorithms-to-find-starcraft-2-build-orders/, Nov 2010.

[5] Elie Bursztein, Mike Hamburg, Jocelyn Lagarenne, and Dan Boneh. Openconflict: Preventing real time map hacks in online games. In *Security and Privacy (SP), 2011 IEEE Symposium on*, pages 506–520. IEEE, 2011.

[6] Elie Bursztein and Patrick Samy. Fuzzing online games. In *Defcon 20*, 2011.

[7] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136. ACM, 2007.

[8] Peter Cowling, Colin D Ward, Edward J Powley, et al. Ensemble determinization in monte carlo tree search for the imperfect information card game magic: The gathering. *Computational Intelligence and AI in Games, IEEE Transactions on*, 4(4):241–257, 2012.

[9] B David-Marshall, J v Dreunen, and M Wang. Trading card game industry-from the t to the c to the g. *Retrieved December*, 12:2013, 2010.

[10] Tamara Denning, Adam Lerner, Adam Shostack, and Tadayoshi Kohno. Control-alt-hack: the design and evaluation of a card game for computer security awareness and education. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 915–928. ACM, 2013.

[11] Blizzard Entertainement. Hearthstone: Heroes of warcraft. http://us.battle.net/hearthstone/en/, March 2014.

[12] esportearning. Hearthstone earning. http://www.esportsearnings.com/games/328-hearthstone-heroes-of-warcraft/top_players_online.

[13] Stephen Fewer. Reflective dll injection. *Harmony Security, Version*, 1, 2008.

[14] Flipperbw. Simple hearthstone logging - see your complete play history without tcp, screen capture, or violating the tos. https://www.reddit.com/r/hearthstone/comments/268fkk/simple_hearthstone_logging_see_your_complete_play, May 2014.

[15] Luca Galli, Daniele Loiacono, Luigi Cardamone, and Pier Luca Lanzi. A cheating detection framework for unreal tournament iii: A machine learning approach. In *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, pages 266–272. IEEE, 2011.

[16] Steven Gianvecchio, Zhenyu Wu, Mengjun Xie, and Haining Wang. Battle of botcraft: fighting bots in online games with human observational proofs. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 256–268. ACM, 2009.

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

[18] HearthPwn. Keeper of the groove statistics. http://www.hearthpwn.com/cards/459-keeper-of-the-grove, 2015.

[19] HearthPwn. Northshire cleric. http://www.hearthpwn.com/cards/600-northshire-cleric, 2015.

[20] Hearthstats. Ladder statistics. http://hearthstats.net/dashboards, Feb 2016.

[21] James H Martin and Daniel Jurafsky. Speech and language processing. *International Edition*, 2000.

[22] Stefan Mitterhofer, Christopher Kruegel, Engin Kirda, and Christian Platzer. Server-side bot detection in massively multiplayer online games. *IEEE Security & Privacy*, 2009.

[23] James R Norris. *Markov chains*. Cambridge university press, 1998.

[24] Pcgamer. Hearthstone's team archon releases hosty hours after cheating allegations. http://www.pcgamer.com/hearthstones-team-archon-releases-hosty-hours-after-cheating-allegations/, Jan 2015.

[25] Hearthstone players. The art of netdecking. http://hearthstoneplayers.com/art-netdecking/, Jun 2014.

[26] Polygon. Hearthstone now has 50 million players. http://www.polygon.com/2016/4/26/11511890/hearthstone-50-million-players-2016, April 2016.

[27] Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951.

[28] Richard A Steinman and Mary T Blastos. A trading-card game teaching about host defence. *Medical education*, 36(12):1201–1208, 2002.

[29] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *INTERSPEECH*, 2012.

[30] Tempostorm. The meta snapshot. https://tempostorm.com/hearthstone/meta-snapshot/meta-snapshot-36-blizzcon-hype.

[31] Steve Watts. Opinion: Why hearthstone's patron warrior nerf goes too far. http://www.shacknews.com/article/91731/opinion-why-hearthstones-patron-warrior-nerf-goes-too-far, Oct. 2015.

[32] Wikipedia. Blizzard warden. https://en.wikipedia.org/wiki/Warden_(software).

[33] Wikipedia. Hypergeometric distribution. https://en.wikipedia.org/wiki/Hypergeometric_distribution.

[34] Wikipedia. Magic: The gathering. https://en.wikipedia.org/wiki/Magic:_The_Gathering, 1993.

[35] Su-Yang Yu, Nils Hammerla, Jeff Yan, and Peter Andras. A statistical aimbot detection method for online fps games. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.