Deck Archetype Prediction in Hearthstone

Markus Eger
markus.eger@ucr.ac.cr
Universidad de Costa Rica

Pablo Sauma Chacón
pablo.saumachacon@ucr.ac.cr
Universidad de Costa Rica

ABSTRACT

Hearthstone is a competitive, online Collectible Card Game, in which players construct their own 30-card decks from hundreds of available cards. Different decks differ wildly in terms of their strategy, from very agressive decks that seek to attack the opponent early, to decks relying on certain combinations of cards, to decks that are focused on responding to the opponent's and ending the game slowly. The player community has therefore given names to different deck archetypes, depending on the strategy they pursue. When playing the game, knowing which archetype the opponent's deck is likely to have helps inform a player on how they should adapt their own strategy to best counter the opponent's. In this paper we introduce the problem of predicting a player's deck archetype from minimal information, in particular only from the actions they performed on their first turn. We discuss the relevance of this problem, and how it can help players adapt to the opponent's strategy, as well as information that can be learned from it. While the information was intentionally chosen to be minimal, due to the nature of the game it still varies in size from game to game, which presents an additional challenge. We describe different approaches to handle this information and their performance applied to this problem, comparing standard statistical methods with Recurrent Neural Networks, and their relative trade-offs, in particular with regards to training time.

CCS CONCEPTS

• **Information systems** → *Association rules*; • **Applied computing** → **Computer games**; • **Computing methodologies** → *Machine learning*.

KEYWORDS

analytics, machine learning, hearthstone

ACM Reference Format:
Markus Eger and Pablo Sauma Chacón. 2020. Deck Archetype Prediction in Hearthstone. In *International Conference on the Foundations of Digital Games (FDG '20), September 15–18, 2020, Bugibba, Malta.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3402942.3402959

1 INTRODUCTION

In 1993, Wizards of the Coast released the game Magic: The Gathering [13], designed by Richard Garfield, which was the first in the new genre of trading card games, also known as collectible card games. In contrast to more traditional games, players could not buy the entire game in one box, but would rather collect different cards, and choose from them to create personalized decks to play against other players. The success and popularity of Magic lead to the development of a vast number of different games using the same basic idea, in the form of physical games, as well as online-only games. The currently most popular online collectible card game is Hearthstone [6], which was released by Blizzard Entertainment in 2013, but is still actively developed with new expansions being released about three times per year. Each of these expansions typically adds between 130 and 140 new cards to the pool of available cards, leading to a growing and evolving game. To play, players have to construct their own decks by selecting 30 cards from among all cards that are available, with different play formats having different restrictions on which cards can be played. The so-called "Standard" format, for example, only encompasses roughly the last two years of released cards, plus some cards that are permanently available, resulting in a pool of around 1000 cards from which players can construct their decks. While this large number of available cards would allow for a very large number of different decks, the player community has identified several core strategies as being stronger than others, which are classified as deck archetypes. For example, the deck type "Secret Highlander Hunter" is a deck type containing no duplicates (called "highlander" after the tagline "There can be only one" of the movie with the same name) and cards that benefit from this, which utilizes "Secret" cards and the "Hunter" class, while "Hand Warlock" is a deck type based on getting a large number of cards in hand that uses the "Warlock" class. While these deck archetypes describe the general strategy followed by the decks, the cards contained within them may vary, with some cards appearing in almost every deck of a particular type, with others being more of a personal choice on the part of the player. On the flip side, many cards are actually useful across multiple different deck types, so neither the mapping from deck type to card, nor from card to deck type is fixed.

When playing the game, knowledge of which deck type the opponent utilizes presents a strategic advantage, because it allows a player to anticipate what the opponent is likely to do, and counteract accordingly. For example, Mike Flores' classic article "Who is the Beatdown?" posits that knowing which deck is the more aggressive one is essential in any match in Magic: The Gathering [7], and it has been argued that this is also important Hearthstone, due to its similarities with Magic [15]. However, in order to known which *deck* is more aggressive, a player needs to know which *deck type* their opponent is playing. Ideally, such information is available to the player as soon as possible, so that they can use the appropriate strategy from the start.

In this paper, we rigorously define the problem of deck archetype detection, including a description of which information is available

FDG '20, September 15–18, 2020, Bugibba, Malta

ACM ISBN 978-1-4503-8807-8/20/09…$15.00
https://doi.org/10.1145/3402942.3402959

to each player, and the desired result. We then present several ways to approach it, starting with a baseline based on deck popularity, as well as an approach based on Bayesian statistics. Since each game players have different amounts of information available to them, such an approach is limited in its applicability, and we therefore also show how Recurrent Neural Networks, which can learn information from variable-length sequences, can be applied to this problem. Finally, we compare the results we obtained, and discuss the trade-offs between the different approaches. In particular, we will also address other information players may have available to them, and how this information can be used by the different approaches. Before we describe our approach, though, we will provide a short summary of the rules and mechanics of Hearthstone.

2 HEARTHSTONE

Hearthstone is a competitive, online Collectible Card Game, in which players construct their own 30-card decks from hundreds of available cards. When constructing a deck, players must choose one of nine available *classes*: Mage, Warlock, Priest, Shaman, Druid, Warrior, Paladin, Rogue, or Hunter. The class a player chooses limits the cards which can be used in the construction of the new deck, with some cards being available to all classes, while others are restricted to a specific class. During game play, each class is represented by a *hero*, which defines the appearance of the player in game but has no effect on gameplay. Once players have constructed their deck, the game can match them automatically with an opponent, that the player then plays against[1].

The game is played in a turn-based manner, where each of the players begins the game with thirty *health* and three cards (or four for the second player). A player loses the game when their remaining health reaches zero, in which case the remaining player wins the game. The starting player is selected at random via a coin toss, the player who wins the coin toss begins and the player going second gets a special card called "The Coin".

During a players turn, that player may play cards from their hand, which can have one of four different types:

- **Minions**, which represent creatures under the players control, and which will enter the battlefield on a player's side, and can attack the opponent's minions or the opponent themself on the player's behalf.
- **Spells**, which have an immediate effect, such as destroying or damaging the opponent's minions, healing the player's own minions or the player themself, or providing other harmful or beneficial effects.
- **Weapons**, which are cards that are equipped by the player, and allow them to attack enemy minions or the enemy themself.
- **Hero cards**, which replace the player's current hero with a different hero, which usually comes with a special ability.

Each card has a *mana cost* associated with it that has to be paid in order to play that card. Mana is produced by *Mana Crystals*. Each turn each player gets one additional Mana Crystal, and then all Mana Crystals produce one mana each. This means that each player has one mana available to them on their first turn, two mana on their second turn, and so on, up to a maximum of 10 mana. Several cards generate mana or mana crystals, allowing players to play more expensive cards earlier in the game than they otherwise would be able to, but generally speaking cards with higher costs show up later in the game, and are more powerful, than cards with lower mana costs. The aforementioned special card "The Coin" is such a mana-producing card, which costs zero and produces a single mana to be used during one turn. In addition to their cards, each player's hero also has a special ability available to them (depending on the hero class) that can be used once per turn, which typically costs 2 mana and produces a minor effect.

One particular kind of spells are called *Secrets*, which, unlike normal spells, are not immediately revealed to the opponent, and instead are represented by a little question mark next to the player's in-game portrait. Each secret has a trigger condition associated with it that will cause it to be revealed. These trigger conditions are usually designed to interfere with the opponent's actions ("When a minion attacks your hero, destroy it"), and are therefore kept secret. As we will discuss below, Secrets have a direct effect on our work, because a player can only observe how much mana was spent to play a secret, and not the exact card. A related card type are "Quests", which, like Secrets, have a delayed effect caused by triggering some condition. Unlike Secrets, however, Quests are public information, and the trigger conditions are tasks the player has to achieve themself, such as "End 4 turns with any unspent mana".

In addition to playing cards, a player may order each of their minions to attack either an enemy minion or the opponent themself. Each minion has a name, a mana cost required to summon it, an attack value and health. Minions may also have a wide variety of different abilities which modify how they interact in game, as well as a minion type, which acts as a tag and may be used by other cards to identify a subset of minions (such as "Destroy target Murloc", where "Murloc" is a minion type). When a minion attacks another minion or the opponent, they deal damage equal to their attack value to their target, and, in the case of attacking a minion, also receive damage equal to the other minion's attack value. Any minion who's health is lowered to 0 dies and leaves the battlefield. As mentioned above, the goal of each player is to lower their opponent's health to 0, which can be achieved by attacking them with minions and playing spells that deal damage to the opponent.

Figure 1 shows a typical game of Hearthstone in progress, as it would appear to a player, who is playing the character at the bottom, a mage. The player has all 10 Mana Crystals, but only one mana left for the turn, as shown in the lower right corner. Over their portrait we can also see a question mark, indicating that that player has an active secret. The top left corner of each card in the player's hand shows its cost, with two cards costing 0 and one card costing 1 mana, each of which the player could play now. Next to the player's portrait we can also see their health, which is currently 1, and the effect over the opponent's player portrait means that the player just dealt 6 damage to them and won, as the opponent's player portrait does not display any health anymore.

Due to the variety of cards that exist, Hearthstone affords a several different strategies. Some decks are based around playing many cheap minions of a particular type and cards that have beneficial effects for minions of that type, and try to overrun the opponent. Others may focus their efforts on defending against early attacks,

[1]For a more detailed description see: https://playhearthstone.com/en-us/game-guide/

Figure 1: A screenshot of a game of Hearthstone in progress. Both players have several minions in play, with the bottom player just having dealt 6 damage to the top player. We can also see the cards in the hand of the bottom player with their respective mana costs.

and playing powerful expensive cards later in the game. Yet others rely on a certain combination of cards to be stronger than each card individually would be. Several cards, such as Quests, work better if a player constructs their deck in a way that maximizes the chance of meeting the Quest conditions. Other cards, such as minions with high attack values or defense for their cost are useful for a variety of different strategies. Due to the popularity of the game, players collectively identify stronger and weaker strategies from among all these possibilities over time and end up giving names to popular deck archetypes, depending on the strategy they pursue. When playing the game, knowing which archetype the opponent’s deck is likely to have helps inform a player how they should adapt their own strategy to best counter the opponent’s. Before we present this problem in more detail, we will first provide an overview over other work that has already been done with the game, as well as other relevant prior work.

3 RELATED WORK

Due to the popularity of Hearthstone, as well as the availability of a simulator [17], the game is used for several research projects around the different aspects of the game, many of which have been focusing on developing AI agents to play the game using different approaches [8, 19, 21, 22]. There is also a competition for different agents to compete [5] against each other, held at the IEEE Conference on Games. However, as Hoover et al. [10] note, there are many different aspects that make the game worth studying, including giving recommendations to players for the deck-building aspects of the game [3]. While our work may be used in the future to improve the playing capabilities of AI-agent, for example to improve the prediction capabilities of AI agents [4], our interest was in the analysis of existing game data, which has not been studied as extensively. Our work is most closely related to the work by Burszstein [2], in which he predicts future plays given the cards played in the first n turns of the game, reporting a correct prediction rate of over 95% after three turns of a card being played during the remainder of the game given which cards were observed to have been played in the past [2]. However, in contrast to this work, we predict *deck archetypes*, which may be used more broadly than just predicting individual cards, instead yielding the strategy used by the opponent. Additionally, and perhaps paradoxically, even though there are many more cards than archetypes, predicting potential future cards is easier in a sense, since cards that are particularly powerful show up in many different deck types, while some archetypes are similar to each other and only differ in a few cards. As the majority of predictions for individual cards will yield such universally powerful cards, its applicability to archetype prediction is likely limited. Nevertheless, since Burszstein’s work results in predictions of multiple possible

[2]Note that the exact evaluation is missing from the paper, but described in this blog post by the author: https://elie.net/blog/hearthstone/predicting-hearthstone-opponent-deck-using-machine-learning/

future cards, it may be possible to utilize some of these predictions to infer archetypes. We believe, however, that the problem benefits from a more targeted approach, such as the one presented in this article.

Beyond Hearthstone, detecting opponent strategy has been identified as an important problem in other games as well. Preuss et al. [16], for example, describe a rule-based system that adapts the behavior of their Starcraft bot to the game state, including their opponent's strategy. However, the rules in their system, such as "If the opponent has many air units, build anti-air units" have to be hand-coded in a strategy file, rather than being learned from existing data. Weber and Mateas [23], on the other hand, describe how to use machine learning to detect a player's strategy in Starcraft given their observed actions. In contrast to our work, there are a lot more observations that can be made in Starcraft, and our work therefore focuses on maximizing the information gained from what little information we have.

While the main analysis in our work was performed using basic Bayesian statistics [1], we will also discuss the application of Recurrent Neural Networks [9] to the observed action sequences. A Recurrent Neural Network is a variation of feedforward neural networks [20] which works by the addition of a hidden state, and which is particularly well suited for processing sequential data. The items of the sequence are passed to the network as input in order, and the hidden state acts as a memory. Practically, this can be realized by having a neural network which accepts one sequence item and the contents of the hidden state as input, and produces a new hidden state as output. When the first item is passed to the network, the hidden state is initialized to some values, typically all zeroes, and all subsequent calls for the other sequence items are passed the output of the previous call as additional input. After all input items have been processed this way, the hidden state is passed through another neural network layer which produces the actual network output. To train such a network, the temporal steps are "unrolled", and a variant of backpropagation, called backpropagation through time [12] is used. Recurrent Neural Networks have already been used in the past to analyze and learn from player behavior in games, such as for imitation learning in Starcraft [14], churn prediction [24], as well as activity recognition in general [18].

4 DECK ARCHETYPE PREDICTION

We now want to define the problem of Deck Archetype Prediction more rigorously, and which relevance it has for the game. First, we need to define what a *deck archetype* is. As described above, players can construct their deck by choosing thirty cards from among those that are legal to play. However, additionally, each deck is associated with one of the nine *character classes* that are available in the game. These classes serve two main purposes: One, about half of the available cards are restricted to a particular class and can not be played by any other. Two, each character class comes with a predefined *hero power*, a special ability that (only) that class can use once per turn, which guides the playstyle in a certain direction. The class-specific cards also often support the same theme. For example, the Warlock character class has a hero power that allows the player to spend two mana to draw a card and lose two life, while their class-specific cards are often very powerful for their cost, but damage the player themself in some way or another. Despite this seemingly narrow play style, there is a variety of different strategies that can be pursued using a Warlock deck. Undercosted, powerful cards lend themselves to aggressive strategies, but there are also decks that seek to negate the drawbacks built into the card and turn them into advantages via combinations of cards. Finally, having an ability that allows the player to draw extra cards every turn fits well into controlling decks that seek to gain a resource-advantage over the opponent in order to win the late game. One concrete deck archetype that falls into this last category is called "Hand Warlock", often shortened to "Handlock". It uses cards such as "Mountain Giant", which is a minion of cost 12 with 8 attack and 8 health, which costs one mana less for each other card in the player's hand. However, there is no exact, concrete list of cards that make up a "Hand Warlock" deck. Instead, there are several cards that almost every player that wishes to play that deck type will have in their deck, while other cards are a matter of personal preference, with some being more popular than others, but not present in every single deck of that archetype. The website https://hsreplay.net contains a database of different deck archetypes, and lists the "core" and "popular" cards for each. Figure 2 shows this data for the "Hand Warlock" archetype.

This means that a deck archetype can be interpreted as probabilities assigned to all cards, indicating the probability for each card being in a deck of that archetype. The core cards have a probability of 1, or close to 1, while popular cards have probabilities that vary depending on how often they are actually included in a deck of that archetype. Of course, these idealized probabilities could only be calculated with access to every single deck, and knowledge which archetype it is intended to belong to, i.e. labels for each deck. In order to then classify a *new* deck to determine which archetype it belongs to can be done using Bayes rule: The archetype gives us probabilities for observing particular cards given an archetype, and we can calculate sample probabilities for the occurrence of all cards and all archetypes, which then allows us to calculate the probability of the probability of the deck belonging to an archetype given the entire list of cards in the deck.

The problem we are proposing is related in that it requires us to determine the deck archetype of the deck of a given player. However, instead of having access to the entire list of cards, our input is what we observe during game play. The direct observations we can make are the class the opponent is using and the cards they play. We may also use other information that is implicitly conveyed during game play, such as the time the player takes to make a decision. Note, in particular, though, that secrets are played, as the name implies, secretly, and we can only observe how much mana the opponent spent, but not which specific secret was played.

> **Definition:** The *Deck Archetype Prediction Problem*: Given all observations of an opponent's public behavior, including their chosen class, cards played, time spent, etc., up to a point in the game, determine which deck archetype is used by the opponent.

As we will discuss below, there are several approaches to this problem. Since every deck archetype is strictly tied to one class, we can always restrict predictions to deck archetypes usable by the class being played. The simplest, which we use as a baseline,

Figure 2: Core cards and popular cards for the "Hand Warlock" deck archetype

is to assume that every player plays the most popular deck of their class, i.e. we ignore which cards they play. However, the two actual approaches we propose for this problem are one based in Bayesian statistics, and one using Recurrent Neural Networks. The statistical method follows the method outlined above: We calculated the marginal probabilities for the occurrence of each deck and each card, and the conditional probabilities for seeing a particular card *on the first turn* given a particular deck, and use Bayes' rule to calculate the probability of seeing a particular deck given a set of played cards, and predict the deck archetype according to the maximum among these probabilities. Finally, we also trained a Recurrent Neural Network that was passed the sequence of played cards with a target label for the deck archetypes.

5 RESULTS

For our work, we collected 16622 replays of Hearthstone games in the Standard format from https://hsreplay.net between September 25, 2019 and October 20, 2019. These dates were chosen as being over a month after the last released card set to give the metagame some time to settle, and before the next planned expansion. However, due to the necessary time to collect sufficiently many games without violating the website's terms of service, and the unpredictable update schedule of the game, it should be noted that a minor patch was released on October 8, 2019, that made 23 previously released cards available for standard play. We believe this minor change does not affect our results in a significant way, though. Replays on https://hsreplay.net were provided by interested players who also agreed to the website's terms of service, which allows the use of this replay information for future (non-commercial) analysis.

Before processing the data, we randomly split the data set into three sets: A training set (62%, 10306 replays), a validation set[3] and a holdout set (19%, 3158 replays each). While processing the data we ignored all games in which no cards were played on the first turn[4] or which were corrupted. The setup consisted of three different main classifiers: The static classifier, which predicts the most commonly played deck for each class, which we used as a baseline, the statistical classifier, which predicts decks based on the probabilities calculated using Bayes' rule, and a Recurrent Neural Network classifier. We trained each of these three classifiers using the training set, and used the validation set to tweak hyper-parameters and fine-tune. The results given below always refer to the classifiers' performance on the holdout set, which we performed after all fine tuning was done.

One important feature of the replays obtained from https://hsreplay.net is that they come with a label indicating which deck archetype was used by each of the two players. This label is assigned automatically based on the core- and popular cards based on the entire deck contents. In this sense, we are therefore attempting to reproduce the classification of this rule-based system using a vastly reduced amount of information. However, the website is very popular with the Hearthstone player community and we assume that their assigned labels reflect the truly intended deck archetype. Within the training set, the most popular deck type was "Quest Shaman", with almost 21% of players choosing to play this deck type, followed by "Quest Druid", which accounted for 9.5% of players, and "Combo Priest", which was played by 7.2% of players. Overall, we recorded 75 distinct deck types being played, with 18 of them being played by fewer than 10 players. Table 1 shows the play percentage of each of the 15 most played decks, with the most played deck for each class highlighted in bold. This deck breakdown also corresponds to one published by the group Vicious Syndicate, which regularly posts articles with a Hearthstone metagame analysis [25].

As we observe *every* player's first turn, they may be going first or second, which determines how many cards they have in hand, as well as if they have "The Coin" or not. In addition to analyzing which decks are played, we also looked at which cards are played most often. Of 191 unique cards being played on the first turn, "Corrupt the Waters" was the most frequently played card, accounting for 15% of all cards being played on the first turn. This is not surprising, as "Corrupt the Waters" is the name-giving quest of "Quest Shaman", the most frequently played deck. The second most frequently played card was "The Coin", accounting for 14.2% of cards played on the first turn, followed by "Untapped Potential", with a percentage of 9.8%. Like "Corrupt the Waters", "Untapped Potential" is also the name-giving/key card in its deck, "Quest Druid", as well as "Malygos Quest Druid". Note that players may play more than one card on their first time, either utilizing "The Coin", other cards that produce mana, like "Innervate", or simply by playing cards with cost 0, but this was not as common. The breakdown of the 15 most played cards with their percentages can be found in table 2. In our

[3]Due to the data set size we opted for a single, large validation set, instead of cross-validation to avoid any possibility of meta-parameter tuning leading to overfitting
[4]While some decks are more likely to not play any card on the first turn, the almost complete lack of information made these games less interesting to analyze. Often, the lack of cards was also a result of inactivity of a player, which would have required further distinction regarding the exact cause.

Deck name	Percentage
Quest Shaman	20.79%
Quest Druid	9.53%
Combo Priest	7.23%
Quest Rogue	5.44%
Highlander Hunter	4.49%
Murloc Shaman	4.18%
Control Warrior	3.98%
Quest Resurrect Priest	3.6%
Zoo Warlock	3.22%
Tempo Rogue	2.79%
Aggro Warrior	2.77%
Malygos Quest Druid	2.69%
Murloc Paladin	2.11%
Quest Hunter	1.91%
Holy Wrath Paladin	1.83%

Table 1: Breakdown of the 15 most played deck types with their corresponding percentage among all training data.

Card name	Percentage
Corrupt the Waters	15.01%
The Coin	14.22%
Untapped Potential	9.8%
Bazaar Burglary	4.45%
Activate the Obelisk	3.93%
Eternium Rover	3.21%
Town Crier	3.09%
Northshire Cleric	2.98%
Secretkeeper	2.52%
Lightwarden	2.44%
EVIL Totem	2.36%
Murloc Tidecaller	2.35%
Crystology	2.28%
Murmy	2.11%
Pharaoh Cat	2.09%

Table 2: Breakdown of the 15 cards most often played on the first turn, with their corresponding percentages among all cards played on the first turn in the training data.

training set, 11794 players played only 1 card on their first turn, 2136 two, 547 three, and 9 more than three cards, with one player managing to play 6 cards on their first turn.

5.1 Static Classification

Given that the decks used by different players are heavily skewed towards certain deck types, we implemented a baseline classifier that predicts that each player will play the most popular deck for their class. The deck breakdown shown in table 1 highlights the most played deck for each class in bold. If a player plays, for example, a Rogue, we would predict that they play a "Quest Rogue" deck. Note that the most commonly played deck for the Mage class, "Quest Mage" does not even appear in the table, as it was only played by 0.75% of players in our training set, appearing as the 26th most popular deck type, behind 4 different Shaman decks and 5 different Warrior decks. Because of how biased the data is towards these more popular deck types, the classifier actually performed reasonably well, with an accuracy of 57.7%(±0.48%), and a weighted average F1-score of 0.44 (weighted average precision 0.36, weighted average recall 0.58). Of course, for classes which have multiple popular deck archetypes, such as "Priest", where "Combo Priest" (7.23%), and "Quest Resurrect Priest" (3.6%) are both pretty popular, this classifier is not well suited. However, it provides a much more suitable baseline than a random classifier.

5.2 Statistical Classification

A more appropriate model to predict the opponent's deck than simply using the most common deck is to utilize the cards being played by the opponent to determine the probability for each deck type being played, and using the archetype with the maximum among these probabilities as the prediction. The probability of a deck given a sequence of observed cards can be calculated using Bayes' rule from the probability of seeing a sequence of cards given a deck, and the marginal probabilities of the deck and the sequence of cards, respectively:

$$P(D|C_1, \ldots, C_n) = \frac{P(C_1, \ldots, C_n|D) \cdot P(D)}{P(C_1, \ldots, C_n)}$$

In order to be able to apply this formula, we therefore need the probability distributions $P(C_1, \ldots, C_n|D)$, $P(D)$, and $P(C_1, \ldots, C_n)$, or at least the sample distributions as estimates for them calculated from our training data. However, one challenge that appears are the joint probabilities $P(C_1, \ldots, C_n|D)$ and $P(C_1, \ldots, C_n)$ for sequences of multiple cards. While, as mentioned above, our training data includes many players that played more than one card on their turn, the number of samples for combinations of three cards is very small (556 in total), which means that there are three-card sequences that may appear in the future that are not present in our training data, or, even if they are present the probability given by our sample distribution would differ wildly from the true probability. Note that even for two-card sequences, even though there are more samples of them, we have the same problem in practice due to the sheer number of cards. To avoid this issue, instead of using the true joint probability, we multiply the individual marginal (sample) probabilities for each card, as if the cards were independent:

$$P(D|C_1, \ldots, C_n) = \frac{P(C_1|D) \cdots P(C_n|D) \cdot P(D)}{P(C_1) \cdots P(C_n)}$$

Even with this approximation we still run into situations in which cards are played that were not part of our training set, and for which therefore would have a marginal probability of 0. Our classifier ignores such cards, and in the case that *none* of the cards that were played were present in the training set, falls back to the most popular deck type for the player's class.

Our classifier implemented using these formulas strongly outperformed our baseline with an accuracy of 78.3%(±0.4%), and a weighted average F1-score of 0.72 (weighted average precision 0.71, weighted average recall 0.78). Table 3 shows the precision, recall and F1-scores for each of the 15 most popular deck types, and figure

Deck type	Precision	Recall	F1-score
Quest Shaman	0.90	1.00	0.95
Quest Druid	0.76	1.00	0.86
Combo Priest	0.99	0.99	0.99
Quest Rogue	0.93	0.99	0.96
Highlander Hunter	0.58	0.98	0.73
Murloc Shaman	0.86	0.87	0.87
Control Warrior	0.51	0.97	0.67
Quest Resurrect Priest	0.78	1.00	0.87
Zoo Warlock	0.79	0.98	0.88
Tempo Rogue	0.72	0.96	0.82
Aggro Warrior	0.61	0.26	0.37
Malygos Quest Druid	0.00	0.00	0.00
Murloc Paladin	0.98	0.99	0.98
Quest Hunter	0.86	1.00	0.92
Holy Wrath Paladin	0.68	0.98	0.80

Table 3: Classification results for the 15 most popular deck types using a basic statistical classifier.

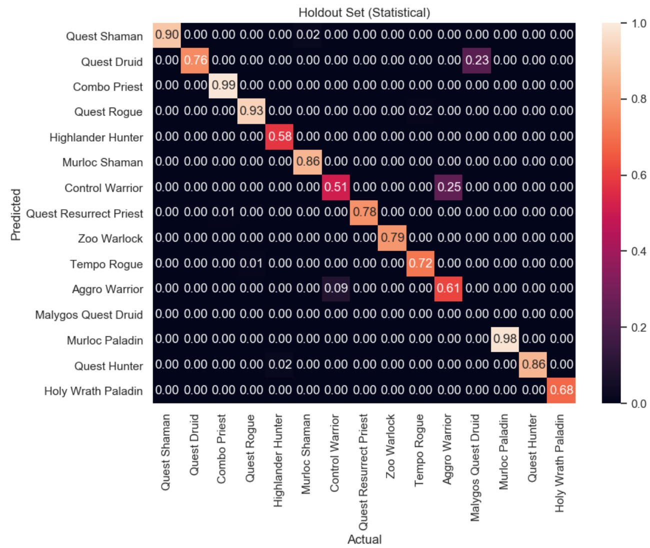

Figure 3: Confusion matrix for the deck archetype classification problem using the basic statistical classifier.

3 shows the corresponding confusion matrix. This approach was particularly successful at identifying the most popular decks, with nine of the ten most popular deck types, having a recall of at least 0.96, with a precision to match in most cases, unlike the static predictor. However, the model still has difficulty distinguishing certain decks, such as "Control Warrior", and the 11th most popular deck "Aggro Warrior", two deck types that have very different playstyles. However, the challenge with distinguishing between these two decks is that the low-cost cards are either the same, or are not typically cast on the first turn, such as "Inner Rage" or "Shield Slam", which require minions to already be in play. Another challenging distinction is between "Malygos Quest Druid" and regular "Quest Druid": As the names imply, share the two deck types many cards, and even the general strategy. The main difference is which cards they use towards the end of the game, with "Malygos Quest Druid" using the eponymous "Malygos", which increases the damage done by all the player's spells by 5, while the regular "Quest Druid" opts to use "Ysera, Unleashed", which produces several powerful dragon minions. During the early game, however, both decks use basically the same strategy, which is meant to let them survive until they can cast their respective game-ending spells. Due to this virtual equality in play patterns, the statistical classifier never predicted the "Malygos Quest Druid" archetype for any player, since the prior probability of regular "Quest Druid" is much higher, and almost all cards that were seen could be played in either deck. A big advantage of this statistical classifier is that training it only involves calculating the sample probabilities over the training set, which needs less than 1 second on a modern PC. The main limitation is that, by assuming independence for all cases, it does not capture any correlations between different cards, nor the order in which the cards were played. We will now describe how a Recurrent Neural Network can be used to overcome this limitation.

5.3 Recurrent Neural Network Classification

To fully capture all subtle interactions between different cards, we designed a Recurrent Neural Network (RNN) to process the sequence of cards played and predict the deck archetype. In our network, we use a single recurrent layer, with a hidden state of size of 54 using the tanh activation function, followed by a dropout layer with $p = 0.2$. After the entire sequence is processed we pass the hidden state through a single softmax output layer to produce the classification as a one-hot-encoding of all 75 available deck archetypes. For training, we used the Adam optimizer [11] with an initial learning rate of 0.01 until the accuracy on the validation set started to decrease, up to a maximum number of epochs. During training, we split up the training data into minibatches of at most 100 samples each, where each minibatch only consisted of samples with the same sequence length to avoid the need for padding and input masking. The meta-parameters that are the size of the hidden state, the dropout-probability, the initial learning rate, and maximum number of epochs, were tuned using a grid search approach using the accuracy on the validation set as the optimization metric, resulting in the parameter values presented here. For the grid search, the size of the hidden state was varied from 2 to 512, spaced exponentially; the dropout-probability was varied between $p = 0$ and $p = 0.6$, spaced linearly, the initial learning rate was varied between 0.0001 and 0.1, spaced exponentially, and the maximum number of epochs was ranged from 10 and 2000, spaced exponentially. Once we found the optimal parameters in this grid, we also explored nearby alternative values, in a refined grid, which only changed the size of the hidden state. Note that, due to the relatively short sequence lengths, we opted for a vanilla RNN, instead of more advanced techniques like LSTMs, which would be able to support longer sequences of varied interconnectivity between sequence items.

The use of neural networks also opened another opportunity for input data: While we are interested in using only the information available to the player on the first turn, this includes some information that is not directly part of the game state, in particular the time the other player spends considering their action. This information would be challenging to integrate with the statistical model presented above, but it can simply be used as another input to a Neural Network. We therefore created two versions of our network: One which processes exactly the same data as the statistical model, i.e. the cards being played, and another that also receives the time that has passed since the player's last action for each card that is being played. For technical reasons, this timing information is only available for one player for each replay, which means we only had half the training, validation and holdout data compared to the other case.

After we determined all parameters using the meta-optimization described above we applied the trained models to the holdout set. First, for the RNN that only used the cards that were played the accuracy on the holdout set was 78.9%(±0.4%) with a weighted average F1-score of 0.73 (weighted average precision 0.71, weighted average recall 0.79). Figure 4 shows the confusion matrix for the classification results of the most popular 15 deck types. While the overall performance did not improve much from the statistical classifier, the precision on the most popular 5 decks improved in all cases, with the precision on recognizing "Quest Shaman" being 0.97, compared to the 0.9 when using the statistical classifier (the recall is 1.0 in both cases). The precision for the next 10 most popular decks are mostly comparable, but in some cases the RNN produces a slightly worse recognition than the statistical classifier. Also similar to the statistical classifier, the classifier has trouble distinguishing between the "Aggro Warrior" and "Control Warrior" archetypes, and completely fails to recognize the "Malygos Quest Druid" archetype. Additionally, training this model took 1925 seconds on an NVidia GeForce RTX 2070. However, we would like to note that while the statistical classifier was prevented outright from predicting a deck from a class differing from the other player's class, the neural network received the played class as an additional input and learned this relationship automatically, slightly reducing engineering overhead.

Before performing this experiment we hypothesized that thinking times can be used as an indirect indicator of how many options a player has at their disposal, which differs between deck types, and may therefore be useful to distinguish between certain deck archetypes that are otherwise hard to distinguish. During our experimentation, the network actually managed to distinguish "Quest Druid" and "Malygos Quest Druid" on the validation set when using times, but these results did not generalize to the holdout set. The classifier actually performed slightly worse than the RNN that did not have access to the time information due to slight overfitting of the timing information, with an accuracy of 78.1%(±0.55%), and a weighted average F1-score of 0.72 (weighted average precision 0.72, weighted average recall 0.78[5]). As for the basic RNN, figure 5 shows the confusion matrix for the classification results of the most popular 15 deck types. When comparing the two confusion matrices, we can see that the RNN that uses timing performs slightly worse for most of the 15 most popular decks, but it actually manages to outperform the regular RNN when it comes to distinguishing between "Combo Warrior" and "Aggro Warrior", providing some support for our hypothesis about thinking times. Another noteworthy improvement is the recall for the "Quest Resurrect Priest" deck, which is 0.83 compared to 0.78 from before.

[5] While it may seem counterintuitive for the F1-score to be the same as the precision and less than the recall, the results are correct, since the weighted average is calculated for each of the three measures separately. Each individual F1-score is the harmonic mean of its respective precision and recall, but this relationship is not preserved by the weighted average.

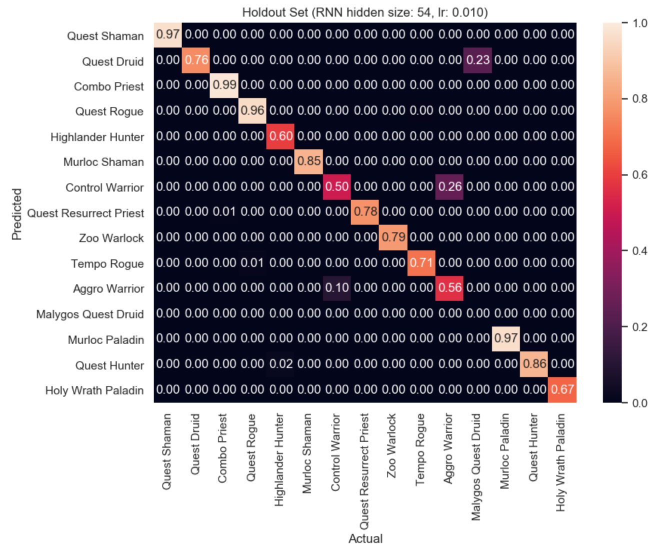

Figure 4: Confusion matrix for the deck archetype classification problem using a Recurrent Neural Network.

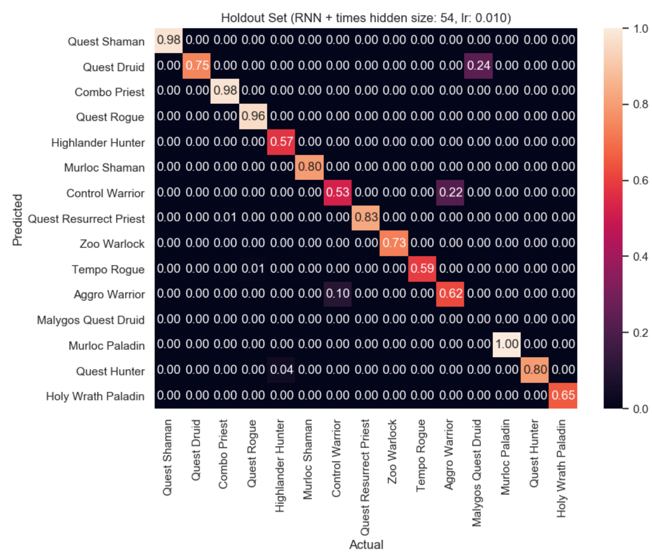

Figure 5: Confusion matrix for the deck archetype classification problem using a Recurrent Neural Network that has access to the timing of the actions performed by the player.

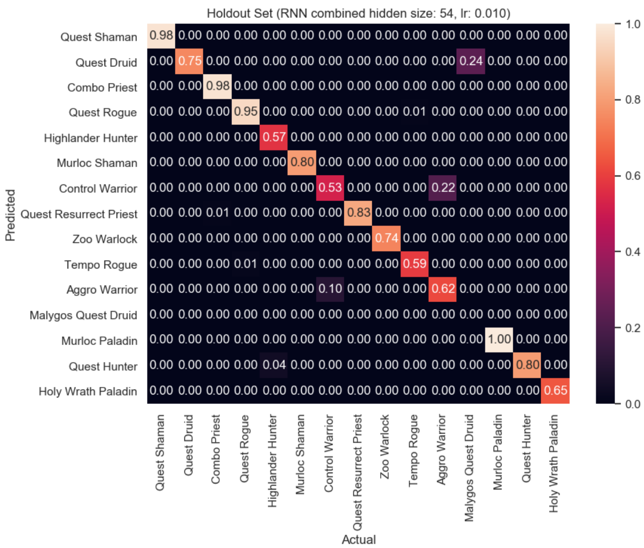

Figure 6: Confusion matrix for the deck archetype classification problem using a combination of the previous two Recurrent Neural Network, depending on the opponent's chosen class.

While the RNN using temporal information has failed to provide improvements in the general case, the fact that it is able to distinguish between certain deck types more precisely than the first RNN gives us another option: We can use the other player's class as a decision variable for which of our two RNNs to use. If the other player is playing as Warrior or Priest, we use the RNN that uses action times as an input, and otherwise we use the first RNN. Figure 6 shows the confusion matrix for the 15 most popular decks for this combined classifier. As can be seen, the confusion matrix is almost an exact combination of the previous two, where it provides a better distinction between the two Warrior deck types, and better recall for the "Quest Resurrect Priest", and almost the same performance as the regular RNN on the other deck types. However, because only half of all input data comes with temporal input, the results are not exactly the same, and the overall accuracy for this classifier is 78.3%(±0.55%), with a weighted average F1-score of 0.73 (weighted average precision 0.71, weighted average recall 0.78). Overall, this classifier therefore does not improve classification accuracy, but we believe it is still useful to distinguish between certain deck types, and we will discuss potential future developments in this area below.

5.4 Discussion

Compared to the baseline classifier, all three other classifiers, using basic statistics, using an RNN, and using an RNN that also takes action timing into account, provide clear improvements, and a respectable accuracy of just under 80%, given very little information about the game state. While the basic RNN-based classifiers performed very slightly better than the statistical classifier in our experiment, but this additional accuracy comes at the expense of additional training time, while the RNN that also uses actions times

Classifier	100	250	500	5000	16622
Static	53.6%	56.3%	57%	57.7%	57.7%
Statistical	67.4%	72.7%	74.7%	77.7%	78.3%
RNN	63.5%	71%	75.2%	78.1%	78.8%
RNN + times	50.9%	63.1%	70.7%	76.9%	78.1%
RNN combined	61.5%	70.3%	74.3%	77.9%	78.3%

Table 4: Prediction accuracy on the holdout set when each of the classifiers is trained on a limited number of samples, as listed in the columns (all accuracies are within ±1.8% or less in the 95$ confidence interval).

actually performs slightly worse (but statistically indistinguishable from the statistical classifier), while still requiring this additional GPU time. Even though this training time is very short by Deep Learning standards, one challenge for this application is that the deck distribution changes very rapidly, so that the actual results presented in this article no longer reflect the reality of the game, requiring the models to be retrained. A bigger challenge is therefore the need to frequently re-train the classifiers, and to collect sufficient data to do so. As mentioned above, we collected over 16000 replays over a period of four weeks. Such a long data collection period was necessary, because the replays available on https://hsreplay.net are provided by players signed up for this service, resulting in at most 100 unique games per hour in our experience. To determine how robust our classifiers are in the presence of less training data, we therefore repeated all aforementioned experiments and only used 100, 250, 500, and 5000 replays selected at random from the training set. Table 4 summarizes the accuracy on the holdout set obtained from these experiments, and compares them to the classifiers' accuracy when trained on the full training set. As can be seen, the statistical classifier actually performs better with fewer examples, but even at only 500 training samples the standard RNN model already performs comparable to the statistical model. It should be noted that the RNN model that includes times basically only operates on half the training samples, since, timing information is only available for one player, as mentioned above.

5.5 Ethical Considerations

Finally, we also briefly want to discuss potential ethical issues arising from our work. First, we want to reiterate that the data used for our work was provided by the users voluntarily, and under terms of service that explicitly mention the use of this data for analysis purposes, and https://hsreplay.net has kindly given use permission to use this data for research purposes. Additionally, since our work was actually player-agnostic, we removed player-identification before any further processing. However, there are several other considerations that must be taken into account. Our work originated from the desire to ultimately improve AI agents playing the game, which can utilize the archetype-information to adapt their own play strategy, but there are other applications that may be seen as problematic by some. For example, another application we could see is as a tool that assists players during game play, which some may view as "cheating". On the other hand, we also believe that the general public benefits from knowing that such a tool is possible and what its capabilities are, as other parties may already be using something similar.

Another potential concern is the use of this information to "bully" certain archetypes out of the meta-game, either by players, or by the developers. For example, if a player realizes that their opponent is playing an "undesirable" archetype they could immediately quit the game. If a sufficient number of players does the same, this would result in a greatly reduced enjoyment of the game for the player with the "undesirable" archetype. We would argue that if there was an archetype that elicits such a strong reaction from a significant part of the player base, that there would be a larger underlying problem with player enjoyment of particular matchups. The developer, on the other hand, does not gain much from the techniques described in our data, since they have access to the entire game information, and can perform a much more accurate archetype analysis using all cards present in a player's deck.

6 CONCLUSION

In this paper, we have presented the problem of identifying the opponent's deck archetype in Hearthstone based on the cards they play on their first turn. This problem is of interest because the archetype defines the playstyle, and this allows a player to adapt their own strategy to what they expect from their opponent. We present several different classifiers to address this challenge, from a baseline that returns the majority deck type for a given player class, to a classifier based on statistical modeling of card and deck type probabilities, to a model using a Recurrent Neural Network (RNN) to automatically find all relevant relations between different cards played and the deck types. The RNN model also allows us to easily include other information, such as the time taken to perform an action, which can help to distinguish between certain deck types.

We showed results of the different classifiers based on a data set of over 16000 game replays that were collected over a period of four weeks in September and October 2019. Comparing our different classifiers, the baseline achieves an accuracy of 58%, with all other classifiers reaching an accuracy between 78% and 79%. While the RNN model achieved a slightly higher accuracy than the statistical model, this minor improvement comes at the expense of significantly longer training times (less than 1 second compared with over 1500 seconds). However, it does allow us to easily add other features to our classifier, which we demonstrate by adding opponent thinking time as an input. While this additional input does not improve the performance in the general case, it can improve the distinction between certain deck types that are otherwise harder to distinguish. We presented one way of combining the two approaches, based on observations of on which classes the timing information makes a difference, but the exact relationship to use could also be learned using e.g. a decision tree. Another topic for future work would also be to refine the performance metrics. In our present work, we assumed all misclassification to be equally bad, but in practice a misclassification of a "Quest Highlander Hunter" as "Quest Hunter" may be less severe than a misclassification of "Aggro Warrior" as "Control Warrior". We have alluded to such differences in our discussion, but in future work we want to define the severity of classification errors based on the target application more rigorously to distinguish such cases.

We also discussed how resilient the different classifiers are to having less training data available to them. While the classification performance improves with a greater number of training examples, as expected, our RNN-based model achieved an accuracy of over 75% even when only trained on 500 game logs, which can feasibly be collected in an afternoon. If even fewer training examples are available , the statistical model provides a reasonable starting point, reaching a prediction accuracy of 67.4% when trained on only 100 game logs. Another difference between the classifiers that we discussed is the required training time. Training of the RNN-based classifiers needed 500 seconds and more, even with only 100 training examples, and this training time increased with more samples, even though training converged in fewer iterations in these cases.

We believe that the problem of identifying the opponent's deck archetype is of relevance in many scenarios, be it to assist a human player, or to improve AI agents that play Hearthstone. In the future, we would like this prediction to be provided in the form of a tool that can assist players to determine what they are up against. Our present work only focused on the analysis of the first turn, but future work could explore how the estimated probabilities and predictions should be updated with incoming card information. While it is possible to extend our models to more turns, the fact that the opponent may also adapt their strategy to what they believe our deck archetype to be may lead to different play patterns. We are also planning to applying RNNs and the lessons learned from this analysis to other strategy games in the future.

ACKNOWLEDGMENTS

The authors would like to thank hsreplay.net for their kind permission to use their Hearthstone replay files. This article is not otherwise endorsed by or represents hsreplay.net in any way, and its content is the sole responsibility of the authors.

REFERENCES

[1] Thomas Bayes. 1763. An essay towards solving a problem in the doctrine of chances.

[2] Elie Bursztein. 2016. I am a legend: Hacking hearthstone using statistical learning methods. In *2016 IEEE conference on computational intelligence and games (CIG)*. IEEE, 1–8.

[3] Zhengxing Chen, Christopher Amato, Truong-Huy D Nguyen, Seth Cooper, Yizhou Sun, and Magy Seif El-Nasr. 2018. Q-deckrec: A fast deck recommendation system for collectible card games. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 1–8.

[4] Alexander Dockhorn, Max Frick, Ünal Akkaya, and Rudolf Kruse. 2018. Predicting Opponent Moves for Improving Hearthstone AI. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*. Springer, 621–632.

[5] Alexander Dockhorn and Sanaz Mostaghim. 2019. Introducing the Hearthstone-AI Competition. *arXiv preprint arXiv:1906.04238* (2019).

[6] Blizzard Entertainment. 2014. Hearthstone: Heroes of Warcraft. https://playhearthstone.com.

[7] Mike Flores. 1999. Who's the beatdown? https://articles.starcitygames.com/premium/whos-the-beatdown/. accessed 2020-04-15.

[8] Łukasz Grad. 2017. Helping AI to play hearthstone using neural networks. In *2017 federated conference on computer science and information systems (FedCSIS)*. IEEE, 131–134.

[9] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[10] Amy K Hoover, Julian Togelius, Scott Lee, and Fernando de Mesentier Silva. 2019. The Many AI Challenges of Hearthstone. *KI-Künstliche Intelligenz* (2019), 1–11.

[11] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. http://arxiv.org/abs/1412.6980

[12] Michael C Mozer. 1995. A focused backpropagation algorithm for temporal pattern recognition. *Backpropagation: Theory, architectures, and applications* 137 (1995).

[13] Wizards of the Coast. 1993. Magic: The Gathering. https://magic.wizards.com/.

[14] Nathan Partlan, Abdelrahman Madkour, Chaima Jemmali, Josh Aaron Miller, Christoffer Holmgård, and Magy Seif El-Nasr. 2019. Player Imitation for Build Actions in a Real-Time Strategy Game. (2019).

[15] PendHS. 2018. *Who's the beatdown? Understanding and evaluating our decks role.* https://redd.it/899tpl accessed 2020-04-15.

[16] Mike Preuss, Daniel Kozakowski, Johan Hagelbäck, and Heike Trautmann. 2013. Reactive strategy choice in StarCraft by means of Fuzzy Control. In *2013 IEEE Conference on Computational Inteligence in Games (CIG)*. IEEE, 1–8.

[17] HearthSim Project. 2014. HearthSim: Hearthstone Simulation and AI. https://hearthsim.info/.

[18] Mengshi Qi, Jie Qin, Annan Li, Yunhong Wang, Jiebo Luo, and Luc Van Gool. 2018. stagnet: An attentive semantic RNN for group activity recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 101–117.

[19] André Santos, Pedro A Santos, and Francisco S Melo. 2017. Monte carlo tree search experiments in hearthstone. In *2017 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 272–279.

[20] Jürgen Schmidhuber. 2015. Deep learning in neural networks: An overview. *Neural networks* 61 (2015), 85–117.

[21] Andreas Stiegler, Keshav P Dahal, Johannes Maucher, and Daniel Livingstone. 2017. Symbolic reasoning for Hearthstone. *IEEE Transactions on Games* 10, 2 (2017), 113–127.

[22] Maciej Świechowski, Tomasz Tajmajer, and Andrzej Janusz. 2018. Improving hearthstone ai by combining mcts and supervised learning algorithms. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 1–8.

[23] Ben G Weber and Michael Mateas. 2009. A data mining approach to strategy prediction. In *2009 IEEE Symposium on Computational Intelligence and Games*. IEEE, 140–147.

[24] Meng Xi, Zhiling Luo, Naibo Wang, and Jianwei Yin. 2019. A Latent Feelings-aware RNN Model for User Churn Prediction with Behavioral Data. *arXiv preprint arXiv:1911.02224* (2019).

[25] Zacho, WWLos, Cheesee10, dib, SoloJazz, Winston, Ginko, Spacemonkey, Ridiculoushat, Prozach, Fenom, Lookitzjoe, Naramo, and Sentenza. 2019. vS Data Reaper Report #143. https://articles.starcitygames.com/premium/whos-the-beatdown/. accessed 2020-04-15.