Predicting Player Churn in Destiny: A Hidden Markov Models Approach to Predicting Player Departure in a Major Online Game

Marco Tamassia*, William Raffe†, Rafet Sifa‡, Anders Drachen§, Fabio Zambetta¶, Michael Hitchens‖
* † ¶ School of Science, RMIT, Melbourne, Australia
Email: {marco.tamassia,william.raffe,fabio.zambetta}@rmit.edu.au
‡Fraunhofer IAIS, Sankt Augustin, Germany, Email: rafet.sifa@iais.fraunhofer.de
§Aalborg University, The Pagonis Network, Copenhagen, Denmark
Email: andersdrachen@gmail.com
‖ Department of Computing, Macquarie University, Sydney, Australia
Email: michael.hitchens@mq.edu.au

***Abstract*—Destiny is, to date, the most expensive digital game ever released with a total operating budget of over half a billion US dollars. It stands as one of the main examples of AAA titles, the term used for the largest and most heavily marketed game productions in the games industry. Destiny is a blend of a shooter game and massively multi-player online game, and has attracted dozens of millions of players. As a persistent game title, predicting retention and churn in Destiny is crucial to the running operations of the game, but prediction has not been attempted for this type of game in the past. In this paper, we present a discussion of the challenge of predicting churn in Destiny, evaluate the area under curve (ROC) of behavioral features, and use Hidden Markov Models to develop a churn prediction model for the game.**

I. Introduction

In this paper the problem of behavioral prediction in digital games is brought into the context of the most expensive game developed and released to date, the online shooter *Destiny*. Developed and released by Bungie in September 2014, the game cost over half a billion US dollars to develop, but sold more than that on its first day of retail, making it the biggest new franchise launch of all time in the game industry. According to Bungie, the game reached a billion US dollars in revenue around May 2015 [1].

Destiny is not only an example of the most expensive to develop and heavily marketed titles today, also referred to as ”AAA” games, but also defines a new form of online game, mixing traditional individual/team-based shooter games with elements from Massively Multi-Player Online Games (MMOGs), and includes both Player-vs-Environment (PvE) and Player-vs-Player (PvP) elements, combining competition and collaboration. In terms of gameplay, *Destiny* is a complex title with a variety of different game modes and wide player freedom in terms of navigation and how to spend time in the game, wrapped in a traditional class-based progression system with missions and instances reminiscent of MMOGs such as *World of Warcraft*.

Destiny as a popular hybrid game title is in its own right worthy of study, but it is also of interest in a broader context: *Destiny* is representative of a trend among major commercial game titles, where publishers are moving away from the traditional retail fire-and-forget business model, and navigating towards a hybrid model which tries to take advantage of the revenue streams offered by persistent online games in the form of, for example, Downloadable Content (DLC), micro-transactions (In-App Purchases, IAPs) and similar tools. *Destinys* developers have experimented with a variety of different revenue opportunities already, including DLC, the sale of emotes and cosmetic items, weapon packs and more. However, with the change in business model comes also new requirements for analytics support. In a retail business model, there is no direct need to monitor the population of the players, but in a persistent game situation, the monitoring and forecasting of player behaviour becomes important to ensuring the revenue stream and operations of a game [2]–[5].

While smaller commercial games for mobile platforms, notably Free-to-Play (F2P) games have been the subject of prediction modelling recently [3]–[5], major commercial titles have received less attention. This is possibly partly due to a lack of accessible data, and partly due to the traditional non-persistent nature of such titles. In the types of AAA titles that are based on persistent game design, the situation is different however. For MMOGs, monitoring of the player community and prediction of their behaviour have been topics of considerable interest, notably from network balancing perspectives as these games have to balance a large population of players across multiple servers [6]–[8]. Similarly, within the domains of eSports – when computer games are played competitively – analytics support has received substantial interest, with behavioral analysis playing a similar role as in physical sports analytics [9].

The situation in *Destiny* compares with all of these related domains of inquiry but along different trajectories. Similar to F2P games, *Destiny* adopts micro-transactions as a source of revenue and feature numerous in-game currencies and reward vectors (equipment, reputation, appearance, etc.). Similar to MMOGs, the game is highly persistent in nature, in essence you are never finished with the game, and there is a running stream of updates and new content being released. The game also supports a huge population of players operating within the same virtual environment. Finally, similar to eSports games, there is a substantial competitive and team-based play element in *Destiny*, exemplified by the *Crucible*, which is the framework for PvP play in the game. In essence, while *Destiny* as a title has not seen previous attention from game analytics, there is some related – but also very recent – work available which can form the basis for investigating churn prediction in the game.

In this paper the focus is on exploring the potential for predicting player churn in AAA titles like *Destiny*, with the game being used as the test case. Given the recent shifts in the revenue generation strategies in AAA games, the game with its hybrid design and large user base forms an ideal platform for investigating behavioral prediction. This is augmented by the availability of high-dimensional, time-series datasets about player behavior thanks to the telemetry tracking of Bungie. Access to data collected since the beginning of the history of the game is available through an API that is already used by the player community to inform the players, for example through services such as destinytracker.com. A similar pattern is observable for eSports games, where the importance of feeding behavioral data back to the community is essential to drive engagement in e.g. tournaments [9].

II. Contribution

In this paper time-series behavioral data from 10,000 randomly selected Destiny players are used as the basis for investigating churn in *Destiny*. It is the first time this kind of hybrid online game has formed the basis for behavioral analysis.

The contribution of this paper is threefold: a) we present the first behavioral analysis of Destiny, the to date most expensive AAA-level commercial title produced in the world; b) We present a churn prediction model for the game based on Hidden Markov Models (HMMs), chosen due to the time series nature of the data, and further due to their successful application in F2P mobile game contexts [3]–[5], [10], [11]. HMM results are benchmarked against other ML models; c) We present a thorough discussion of the kinds of behavioral features typically tracked of player performance in online AAA titles, and their relative usefulness in connection with churn prediction.

III. Related work

Behavioral prediction in games is a relatively recent topic, but has a strong tradition outside games. One of the earliest successful churn models was presented by Mozer et al. [12], in the area of wireless communication, and churn has been investigated in e.g. in retail banking and insurance.

Churn prediction work in digital games has primarily taken place across four vectors: a) F2P mobile games [3], [4]; b) MMOGs [2], [8]; c) other games [13]–[15] and d) Game AI in general [16]. The latter approach is the least directly applicable to the current problem as the focus here – generally – is on artificial agents and mimicking the behavior of players, as compared to analyzing the behavior of the players. Due to space constraints the focus in this section will be on work directly related to the challenge of behavioral prediction in online persistent games.

A. F2P mobile games

Recent work on prediction in F2P mobile games has covered a variety of machine learning models, and is focused on either predicting players leaving the game [4], [5], or conversely which players that will make a purchase in the game [3], [10], [11]. The churn problem in games was formally defined by Hadiji et al. [5], who also identified a number of behavioral features which are applicable across F2P game titles, including several temporally-bound features such as Number of Sessions, Avg. Time Between Sessions, Total Days Played, Current Absence Time, and Average Playtime per Session. The features identified by Hadiji et al. [5] were later used by Sifa et al. [3] and others, and several similar features occur in F2P prediction work such as Xie et al. [11], Rothenbuehler et al. [10] and others who focused on predicting IAPs. In general, evaluating the usefulness of these features in predicting churn across five F2P titles, the work in F2P games has highlighted the importance of behavioral features associated with playtime as a function of real-world time, e.g. the Number of Sessions, Number of Days Played and Avg. Playtime per Session were found to be the most important features. Interestingly, the duration of the time between play sessions have also been found to be important to churn prediction in MMOGs, see e.g. [8].

The methods applied range from pattern recognition and historical analysis, simple forecasting and multiple regression, to machine learning techniques. The latter notably includes Decision Trees [DTs] and variants such as Random Forest [3], [5], Support Vector Machines [11] and Hidden Markov Models [HMMs] [4], [10].

As yet deep learning methods have not been applied in behavioral prediction in games but forms a potentially interesting addition to the current arsenal of game analysts due to the ability of deep learning methods to handle sparse and imbalanced data, which are typical in behavioral telemetry situations [2], [3], [5]. There is as yet not enough publicly available knowledge to draw conclusions about which behavioral features provide the best result across different games, or which ML models work best for predicting player behavior, but commonly reported accuracies lie above 0.8, meaning that predicting player behavior in F2P games is definitely possible. It should be noted that these games are generally also much more restrictive in their design in terms of player agency than MMOGs and AAA-level titles such as *Destiny*. In essence, they are simpler games.

B. MMOGs

The focus of behavioral modeling in MMOGs has from the onset had a quite different focus than work in the F2P space, namely that these lines of research originate in network science. Some of the earliest work includes Kawale and Srivastava [17] who investigated churn in the MMOG *EverQuest II* using social network analysis as the basis, proposing a churn model based on social influence among players. However, the precision and recall rates obtained were approximately 50%, which led Borbora et al. [18] to try out other classifiers, similarly using *EverQuest II* data and hybrid methods with a binary decision approach, defining churners as players who cancelled their subscription or been inactive for 2 months. Also with the focus on MMOGs, Nozhnin [19] focused on the first few minutes of gameplay in the game *Aion*, investigating triggers for churn. The author highlighted the challenge of feature selection in behavioral prediction in games.

Focusing on the two MMOGS *World of Warcraft* and *Warhammer Online*, Pittman and Gauthier [7] mined client-server streams from client servers measuring player distributions using data such as session length and high-level movements of the players, with the focus on informing MMOG server architecture. Feng et al. [8] applied traffic analysis to a three-year dataset from the MMOG *EVE Online*. The results indicated that churn rates in the game varied across the lifespan of the game, generally increasing with the age of the game. Furthermore, Thawonmas et al. [6] analyzed player revisitation in terms of returning to play the game, as well as returning to specific in-game areas, in the MMOG *Shen Zhou Online*. The primary behavioral features used were login time and login frequency.

C. Other games

Outside the confines of MMOGs and F2P mobile games, behavioral prediction has been the topic of a few publications, across a variety of games. For example, one of the earliest investigations into behavioral prediction in AAA-level commercial games was performed by Mahlman et al. [20], who used Decision Trees to predict retention in the action-adventure game *Tomb Raider: Underworld*. Sifa et al. [21] built a tensor factorization based representation learning framework to incorporate the movement information of numerous players into the retention prediction process for the sandbox game Just Cause 2.

Within the genre of Multi-player Online Battle Arena (MOBA) games, Yang et al. [22] presented an approach for discovering and defining patterns in combat tactics among winning teams in the eSports title *DOTA 2*, based on graph representation. Schubert et al. [9] developed an algorithm for dividing eSports matches into encounters, which were then used as the basis for prediction models focusing on match outcome in *DOTA 2*.

Operating with data from the Steam game distribution and -hosting client, Bauckhage et al. [23] modelled players engagement to games using lifetime analysis across five major commercial titles. Modeling the players interest as a hidden variable the authors extracted playtime information and showed how the interest can be represented in terms of lifetime distributions and their corresponding processes. This work formed the first attempt at an explanation of the power law pattern evident in many studies of playtime and retention in games to that point. The work was followed up by Sifa et al. [15] who found similar patterns across more than 3000 game titles, working with over five billion hours of play across more than six million players.

In summary, prediction has in the field of game analytics been focused on F2P games, with a deeper history in online games such as MMOGs. However, prediction is also finding uses within 3D navigation, progress prediction, or predicting what kind of problems specific players will encounter, etc. The vast majority of the current knowledge about these application areas rests within the industry, where the combined resources for behavioral research outranks academic research by at least a factor of ten, and only small glimpses of such business-sensitive knowledge is available through industry talks and presentations.

IV. Destiny: gameplay

Destiny is formally a science fiction-themed online first-person shooter game which has been blended with a number of features reminiscent of MMOGs, notably a persistent online world, as well as with Role-Playing Games (RPGs), notably character development along a number of trajectories, which also occur in many MMORPGs (Massively Multi-player Online Role-Playing Games). The game was developed by Bungie, and published by Activision in September 2014. The game is only available on major gaming consoles and requires the player to be always-online.

In terms of comparisons with earlier work on behavioral prediction in games, *Destiny* forms a unique case in that it shares design elements and mechanics that are found within the types of games this earlier work has focused on, without being similar to any previous game that has formed the basis for predictive analytics. For example, similar to F2P games, *Destiny* features microtransactions and purchasable content, but also has other features as noted above. Similar to prediction work on other types of games, *Destiny* features team-based combat reminiscent of MOBAs [9].

To understand these differences, and the impact this has on feature selection, it is necessary to explain how *Destiny* operates as a game.

The core mechanics of *Destiny* are those of a traditional FPS, and include run, jump, crouch and shoot as well as simple melee combat. The interface provides information such as ammunition, health, a mini-map and floating information text over enemies. Other mechanics more resemble RPGs, with character classes, attributes and levels based on earnt experience points, all feeding into the complex damage system. Enemies typically take multiple hits to kill, although some weak enemies can be easily dispatched and headshots provide additional damage. *Destiny* also features an inventory system, a range of collectibles and crafting.

The setting is an extensive persistent game world, exploration driven by quests and available activities in the form of player-versus-environment (PvE) and player-versus-player (PvP) content. Single and multi-player elements both feature heavily, in the style of MMORPGs such as *Everquest* and *World of Warcraft*. There are also clear relationships to team-based FPS games, such as *Call of Duty* and *Team Fortress*, although the persistent game world sets it apart from those games. The overall game experience is one of fast paced combat action with players presented with multiple options in a large persistent world. One unusual feature for an online multi-player game, and one which has caused considerable controversy within the player base, is the limited in-game support for player to player communication.

Story missions direct the player through settings across the solar system, from Earth to the moon and out to the other planets and may be completed with other players but can typically be completed alone. Group activities in *Destiny* are based around a fireteam of three players. Strikes and the larger and more involved Raids are instanced co-operative group content for three and six players, respectively. Other co-operative group-based content, known as public events, take place in the persistent world, where all players that can reach the site of the event can participate.

PvP content (known as Crucible matches) takes place in instanced environments and involve one or two fireteams a side, for a maximum of twelve players. PvP modes include team and individual deathmatch, area control and some less well-known forms, although other familiar modes, such as capture the flag, are absent. The size of the teams gives the matches more of a feel of other small scale PvP content, such as *World of Warcraft* arena battles and small team battles possible in games such as *Counter-strike*, rather than the larger teams possible in many online FPS games that do not have a persistent world.

The restricted communication options, particular the lack of any text based chat channels, produces a different experience to many other MMO games, particularly those played on a PC. Voice communication was, at initial release, only possible between members of pre-formed fireteams, usually consisting of players who know each other outside the game. Recently added is the option of voice communication to players who are randomly put into teams by the matchmaking service in both PvE and PvP content. These voice-chat features are opt-in.

V. Definitions

The term *churn* here refers to the process of a player leaving the game indefinitely and discontinuing to be a customer. While it is natural for most players to eventually turn away from a game over time, for games with hybrid revenue models such as Destiny holding onto players for as long as possible not only increases revenue but maintains a higher density of player interaction in the MMOG setting. Therefore, retaining players by either encouraging churned players to return or preventing current players from churning in the near future has a vast impact on the success of the game overall.

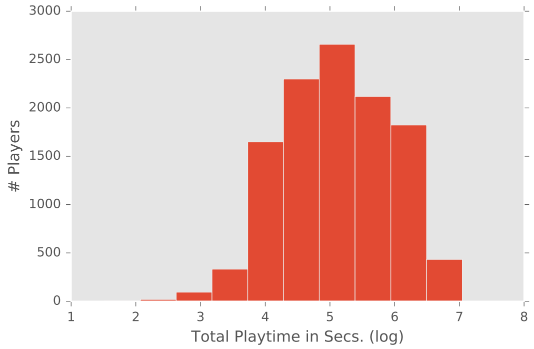

Fig. 1: Total playtime distribution in the log scale.

More specifically, the goal in this work is to identify players who are *about to churn* rather than those that have *already churned*. Enticing a churned player back to the game is less likely to be successful than encouraging a currently active player to continue playing [4]. By analyzing a player's time series data, we seek to predict whether that player is about to churn by identifying patterns of disengagement in in-game activity and how often they play. Additionally, we are predicting late churn of players who have been playing the game for at least a month; this is opposed to predicting early churn in players who have only recently begun playing.

VI. Modeling

This section starts with an outline of the Destiny dataset that was used in the experiments in Section VII. It then gives a detailed account of how the data is pre-processed and labeled in preparation for classification. This pre-processing is focused on preparing the data for use with a HMM classifier. This section then concludes with the specifics of the features used, the setup of the HMM classifier, and the non-temporal classifiers that were tested for comparison purposes.

A. Dataset

The dataset used in this study contains detailed daily behavioral information of more than 10000 Destiny players with 24118 characters that have been randomly sampled from all of the players that played the game at least for two hours. Grouped by the game modes, the dataset contains general metrics about in-game activities such as average scores per kills and number of deaths as well as very detailed information about the gameplay such as the suicides and the performed resurrections. The data covers 17 months of activity starting from September 2014 to January 2016 and the total playtime played by the players is 1,809,564 hours and the average per player is 158 hours. Figure 1 1 shows a log scaled histogram of total playtime.

B. Data Pre-processing and Labeling

A subset of players was randomly sampled from all players of the game who played for at least two hours. This threshold is set to eliminate bias imposed by people who never migrate

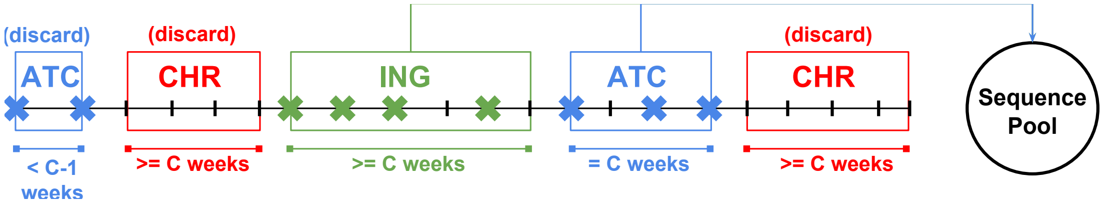

Fig. 2: Example of labeling sequences in time-series data of a player as churned (CHR), about-to-churn (ATC) or in-game (ING) with $C = 4$. Black notches on the timeline are weeks with no play data and each colored X marks a week where at least 120 seconds of play time was recorded. Labels below the timeline show the rule that determined the class assignment.

to becoming actual players of the game, e.g. people who install the game but never play it, or only play it very briefly. The exact placement of the threshold can of course be debated, but was in the current instance based on an investigation of the approximate playtime it takes to navigate the earliest step of the tutorial elements in the game.

The dataset provides daily snapshots of each players activities, performance, and achievements for every day that they played during the sample period. However, this data was further aggregated into weekly snapshots for each sampled player, where days that the player had no activity were filled with null values. If playtime was less than 120 seconds in a week, that week is considered to have no activity and is zeroed out. This threshold value was chosen through intuition but could be further explored through sensitivity analysis.

Aggregation is done because of irregular play behaviour between players within a week. For example, because Destiny play sessions require players to be heavily engaged with the game (unlike say with ”casual” games), it is likely that a busy schedule will prevent them from playing for many days in a row. If a player is not active for a few days, it is unlikely that they have churned therefore leading to incorrect predictions.

For the same reason, we define a *churn window* (C) such that if there is no data for a player during at least a C weeks period, then that player is considered to have churned during that time. Note that this means that it is possible for a player to churn for a period of time and then return. In all of our experiments, we set $C = 4$. The reason for this can be seen in Figure 3 where each point represents a sampled player, plotted against their playtime (in seconds) and their average absence from the game (in days) during the sample period. The density of the plotted player data is much higher for values of *average absence* less or equal to 28 (i.e. 4 weeks).

All of the classification techniques used to predict churn in this paper are trained through supervised learning and so requires the data to contain class labels. Sequences of weeks are labeled as either *churned* (CHR), *about-to-churn* (ATC), or *in-game* (ING). However, all CHR data is discarded and instead we use ATC for positive samples and ING for negative samples for classifier training and testing purposes.

The data is processed backwards through time. If there is no player activity for C weeks or more, then those weeks are grouped into a sequence and that sequence is given a label of CHR. Any weeks with activity that are within C weeks before the CHR sequence starts are again grouped together and given the label ATC. All other sequences of weeks with activity are given the label ING. Additionally, because players may return after C weeks of inactivity, it is possible to have more than one of each sequence type for each player. It is also possible for sequences to be less than C weeks long if, for example, an ATC period is surrounded by CHR periods and doesn’t have activity that is at least C weeks apart. If these sequences are shorter than $C - 1$ weeks long, then they are discarded. This means that both CHR and ING sequences can span $C - 1$ or more weeks, while ATC sequences are either $C - 1$ or C weeks long. Once these sequences are identified, they are added to a pool of sequences from all players. It is from this pool that training and test set data is drawn to be used with the classifiers.

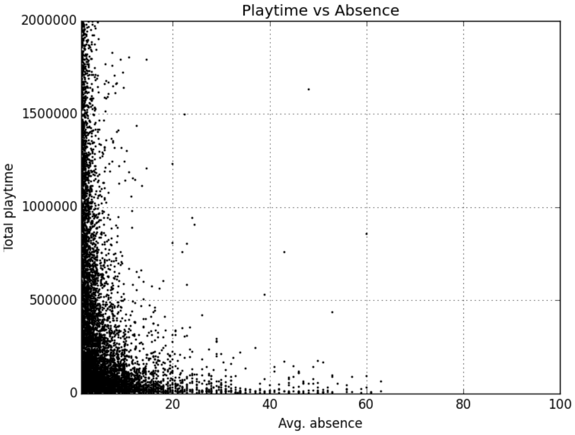

Fig. 3: Play time (in seconds) vs. average absence (in days) for each sampled player during the sampled time.

This process can be seen in the example in Figure 2. The center black line indicates the timeline while each X marks a week where the player played the game. Here, the player plays for just two weeks before churning but then returns, plays consistently for a number of weeks, their play time becomes more sparse, and then they churn again. As the first two weeks of play are less than our $C - 1$ threshold, this sequence is discarded along with the CHR sequences. The two central ING and ATC sequences are extracted and added to the global pool of sequences.

C. Features

All the features below are discretised by comparing the current feature value with that of a rolling average and assigning a value of $(0, 1, 2)$ corresponding to $(less, same, more)$ tags. The rolling average for feature x at week $t+1$ is calculated as $\sigma_{x,t+1} = \sigma_{x,t} + \alpha * (x_t - \sigma_{x,t})$, where $\alpha = 0.4$ is used as a weighting parameter for the experiments below. If the value for the current week is within 20% of the rolling average, it is given the discretised value for being the *same* as the average. Otherwise, it is given the value of *more* or *less* depending on whether it was greater or less than that of the rolling average beyond the 20% threshold.

In choosing features to use, we tested many combinations using experiments similar to those found in the experimental results (Section VII) below to find those that yielded the best HMM performance. The Destiny dataset provides 35 player performance statistics regarding what they did in the game and how well they did it. However, the best combination that we could find (and that we use in the experimental results below) only utilizes three of these, as well as three engineered temporal statistics regarding how often they played. Each week of recorded play is represented by a vector of the following feature values:

- Mean Lifespan: Total number of seconds played divided by the number of times the player has died since they started playing.
- Kill-Death Ratio: Total number of kills the player has divided by the total number of deaths since they started playing.
- Activities Completed Ratio: The ratio of activities that the player completed to the number of activities that they entered.
- Current Absence: Number of weeks the since the player last played. A week with less than 120 seconds of activity has a value of 1, the second consecutive week of no activity will have a value of 2, and so on. If a week contains at least 120 seconds of play time, then this value is 0.
- Current Absence to Mean Absence Ratio: Mean Absence takes into account all absence periods, excluding the current one.
- Weeks Present Ratio: Number of weeks this player has been active divided by the total number of weeks since they first registered to the game.

D. Classifiers

As we are dealing with time series data, the main classifier that we examine is a multinomial Hidden Markov Model (HMM). The HMM models are learned using the *hmmlearn* Python library (https://github.com/hmmlearn/hmmlearn). Sequences of data are extracted in the pre-processing step as each constitutes a sequence of observations to be used by the HMM, with the features of each week making up a single observation.

Two HMM models were trained, one for the ATC class and one for the ING class. By passing the respective data to each model, they were trained to recognize patterns corresponding to the specified class. During testing, the models take a sequence of observations and it returns the log likelihood that the sequence was produced by the given model, thus giving a probability of the sequence belonging to that class.

If an observation (a single week) in a test set sequence has not been seen by a multinomial HMM during the training phase, then that entire sequence will be unclassifiable. If the sequence is classifiable by one model but not the other, then the predicted class is that of the model that can classify it, even if the log likelihood is low. If the sequence is unclassifiable by both models, then the prediction defaults to ATC.

We also compare the performance of this HMM with that of several other classifiers, utilizing the *scikit-learn* Python library (http://scikit-learn.org/stable/). As these are non-time series models, each week in the dataset is given the class label of the sequence that it is a part of. Each week is then joined with $C-2$ other neighboring active weeks to create a single sample with $n*(C-1)$ features. For example, each sample will have three separate Kill-Deaths-Ratio features, one for each week. In our case, we use the $n = 6$ features listed earlier and $C = 4$, giving 18 features per sample. This is done in order to provide at least some measure of temporal data to the classifiers. We also tested the classifiers by treating each week as a single sample with $n = 6$ features but this performed worse than the $n*(C-1)$ setup. It is also worth noting that using the original feature values performed better than the pre-processed discretised values for all of these non-temporal classifiers and so these original features are used in the results shown below.

VII. Experimental Setup and Results

This section discusses the results of the HMM classifier versus the other non-temporal classifiers across the temporal data. As the HMM training process is stochastic in nature and can lead to different models even when provided with the same training data, the results for the HMM show the average performance of 15 training and testing cases using the same training and set sets. Meanwhile, all other classifiers use stratified 10-fold cross validation on the same training set as the HMM. The results for these classifiers show the performance of the best model identified by cross fold validation running on the same test set as the HMM.

The training set and test set were formed by splitting the sampled data on the date of the 20th of October, 2015. Any sequences with weeks entirely before or on this date were added to the training set, while those with weeks entirely after or spanning both sides of this date were added to the test set. In total there were 12086 ATC sequences and 3996 ING sequences in the training set. There were 3065 ATC and 927 ING sequences in the test set. In terms of individual weeks, there were 69287 ATC weeks and 49962 ING weeks in the training set and 33497 ATC weeks and 35235 ING weeks in the test set. Only 0.9% of sequences were unclassifiable by the ING HMM model while 0.4% were unclassifiable by the ATC model.

TABLE I: Results of the best models found for non-temporal classifiers and the mean HMM performance on the same training and test data sets. Bold values are best in column.

Classifier	**Prec**	**Acc**	**Recall**	**F1**	**AUC**
Theoretical Random Classifier	0.75	0.5	0.5	0.6	0.5
Bagging	0.54	0.55	0.49	0.51	0.56
Naive Bayes	0.55	0.57	0.70	0.61	0.61
Nearest Neighbor	0.52	0.53	0.57	0.54	0.54
Gradient Boosting	0.56	**0.58**	0.63	0.59	0.61
Decision Tree	0.55	0.57	0.60	0.58	0.59
Discriminant Analysis (Quadratic)	0.54	0.56	0.73	0.62	0.60
Discriminant Analysis (Linear)	0.54	0.56	0.75	**0.63**	0.61
Ada Boost	0.56	**0.58**	0.66	0.60	0.61
Logistic Regression	0.54	0.57	0.75	**0.63**	0.61
Random Forest	0.54	0.56	**0.76**	**0.63**	0.60
Hidden Markov Model	**0.92**	0.53	0.43	0.57	**0.77**

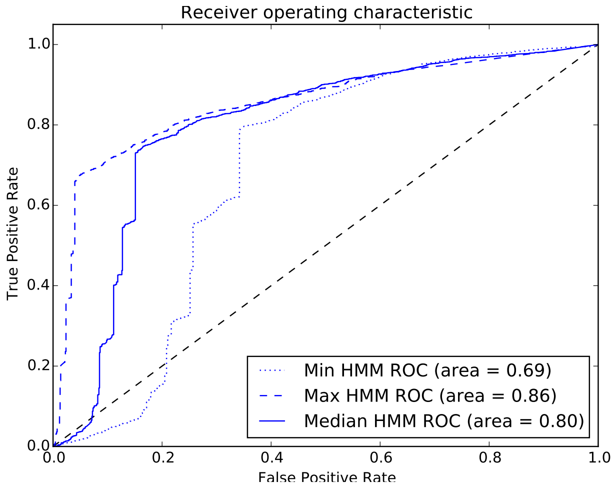

Fig. 4: The ROC curves for the best, worst, and median pair of HMM models found after 15 separate stochastic training runs.

A. Results

Table I shows the results of these tests, given as precision, accuracy, recall, F1 score (all with no bias in the binary classification threshold), and area under ROC curve (AUC). Bold values indicate the classifier that performed the best for the given metric. Theoretical results of a random classifier are also provided, calculate from the class distributions in the previous section. From these results we can see that the HMM far outperforms the other classifiers on precision, meaning that when it predicts that a player is about-to-churn (ATC), then it is highly likely that the player will churn. However, the HMM recall is the worst, which means the HMM is conservative in its predictions and is failing to identify at least half of ATC players. The fewer true positive predictions also gives the HMM the worst accuracy.

This result is opposite to many of the non-temporal classifiers, such as the Random Forest classifier, that has a high recall but a low precision. These classifiers are acting quite liberally in predicting that players are ATC. Overall the HMM performs best with respect to the AUC, suggesting a better balance between precision and recall overall with various prediction thresholds. Figure 4 shows the ROC curves of the HMM model pairs with the minimum, maximum, and median AUC values. This figure highlights that, especially for the best performing HMM, the recall (true positive rate) climbs rapidly as false alarm rate (false positive rate) increases. This suggests that by using an unbalanced binary prediction threshold to be slightly more generous with positive predictions, the HMM could increase recall to a reasonable level while not sacrificing too much precision.

B. Discussion

There is a choice to be made in which classifier would be best to deploy to the real-world, based upon the results shown here. Let us assume, for example, that players identified as ATC at any given time are offered an incentive by Bungie to continue playing Destiny. If this incentive is costly to Bungie or could hurt the in-game economy (such as giving players a rare in-game item), then precision is more important as the company would only want to give such an expensive incentive to those who are truly about to churn. However, if say the incentive is just a common in-game item corresponding to the players level then a higher recall is more important: it is less important if the non-churning player receives this bonus, as long as more players are encouraged to stay in the game. Overall, the HMM will also provide the best balance between these two simply by adjusting the binary classification threshold.

HMMs were chosen as the focus here as a representative of classifiers that have the ability to factor in additional temporal information from the in-game behavior of the players, in this case performance-based features. However, similar to work in F2P and MMOGs, the best performance relies on a mixture of in-game and temporal features, the latter describing how often the player was active. Of interest is that the combination of these features provided similar performance to that of [10] (based upon their provided ROC curve) who operated in a F2P game context, even though the authors only use specifically temporal features. There is a vast difference in the relative complexity of the design and mechanics between *Destiny* and F2P games. It is also interesting to see the importance of temporal features across these diverse game situations, which was speculated on in the work of Bauckhage et al. [23] and Sifa et al. [15], who hypothesized an underlying model of player interest in games from observing playtime distributions across more than 3000 games.

It may then be argued that the player performance features offer little benefit for churn prediction and that both the HMM and the non-temporal classifiers would be better served using only temporal features. Temporal features have previously shown promise as a genre agnostic feature set for a wide variety of F2P games [5] and a single-player sandbox game [21]. However, when testing feature combinations similar to those used by [5], [21], we witnessed significantly poorer performance in both types of classifiers. This suggests that features used for successful churn prediction in F2P mobile and single player games may not be applicable in the context of MMO console games. This highlights the need for more research into potentially generalizable feature sets for game analytics. In the absence of those general features though, it remains important to consider the genre of the game and the context of the data when addressing the churn prediction problem.

VIII. Conclusions and Future Work

In this paper detailed time-series behavioral feature data from 10,000 randomly selected players from the hybrid FPS online game *Destiny* have been analyzed for the purpose of building a churn prediction model. Based on an application of Hidden Markov Models, a churn prediction model is presented. The HMM model has been benchmarked against an array of classifiers, and the relative performance of different approaches described and discussed. The results presented highlight the differences in the demands on the behavioral features used for prediction across game types, as is clear when comparing work across F2P mobile games, MMOGs, single-player games and now hybrid online games.

The work included here represents a first step towards building behavioral prediction models in *Destiny* and similar games. The results form a step on the way to developing robust predictive models for AAA-level commercial game titles, similar to the models currently available in F2P mobile games. With the increasing focus within major commercial game titles towards extending the interaction period, churn prediction models are an important first step in developing games that are user-responsive and able to adapt to prevent player disengagement, as is currently being explored in F2P mobile games [4] and Game AI [16].

Future work will focus on improving the precision and recall rates, for example by further feature engineering, moving beyond performance-based and temporal metrics, such as progression metrics (e.g. character level, faction reputation, missions accomplished). Future work will also investigate the potential for predicting other aspects of player behavior, notably related to monetization, social behavior and game content absorption. The latter forms an example of game-based behavioral predictions that directly target informing design, as compared to the monetization and retention focus common in prediction work in games, and forms a venue in game analytics that has not been well explored.

Acknowledgment

The authors would like to extend their sincere gratitude to Bungie for making telemetry from Destiny available.

References

[1] "Activision blizzard announces better-than-expected first quarter 2015 financial results," http://bit.ly/1rLD9uK, accessed: 01-05-2016.

[2] M. Seif El-Nasr, A. Drachen, and A. Canossa, Eds., *Game Analytics – Maximizing the Value of Player Data*. Springer, 2013.

[3] R. Sifa, F. Hadiji, J. Runge, A. Drachen, K. Kersting, and C. Bauckhage, "Predicting Purchase Decisions in Mobile Free-to-Play Games," in *Proc. of AAAI AIIDE*, 2015.

[4] J. Runge, P. Gao, F. Garcin, and B. Faltings, "Churn Prediction for High-value Players in Casual Social Games," in *Proc. of IEEE CIG*, 2014.

[5] F. Hadiji, R. Sifa, A. Drachen, C. Thurau, K. Kersting, and C. Bauckhage, "Predicting Player Churn in the Wild," in *Proc. of IEEE CIG*, 2014.

[6] R. Thawonmas, K. Yoshida, J.-K. Lou, and K.-T. Chen, "Analysis of Revisitations in Online Games," *Entertainment Computing*, vol. 2, no. 4, pp. 215–221, 2011.

[7] D. Pittman and C. GauthierDickey, "Characterizing Virtual Populations in Massively Multiplayer Oline Role-playing Games," in *Proc. of the 16th Int. Conf. on Advances in Multimedia Modeling*, 2010, pp. 87–97.

[8] W. Feng, D. Brandt, and D. Saha, "A Long-term Study of a Popular MMORPG," in *Proc. of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*, 2007.

[9] A. Schubert, M.; Drachen and T. Mahlman, "Esports Analytics Through Encounter Detection," in *Proc. of the 10th MIT Sloan Sports Analytics Conference*, 2016.

[10] P. Rothenbuehler, J. Runge, F. Garcin, and B. Faltings, "Hidden Markov Models for Churn Prediction," in *Proc. of SAI IntelliSys*, 2015.

[11] H. Xie, S. Devlin, D. Kudenko, and P. Cowling, "Predicting player disengagement and first purchase with event-frequency based data representation," in *Proc. IEEE Conference on Computational Intelligence and Games*, 2015, pp. 230–237.

[12] M. Mozer, R. Wolniewicz, D. Grimes, E. Johnson, and H. Kaushansky, "Predicting Subscriber Dissatisfaction and Improving Retention in the Wireless Telecommunications Industry," *IEEE Trans. on Neural Networks*, vol. 11, no. 3, pp. 690–696, 2000.

[13] B. Medler, "Play with Data – An Exploration of Play Analytics and Its Effect on Player Experiences," Ph.D. dissertation, Georgia Institute of Technology, 2012.

[14] T. Mahlmann, A. Drachen, J. Togelius, A. Canossa, and G. N. Yannakakis, "Predicting Player Behavior in Tomb Raider: Underworld," in *Proc. of IEEE CIG*, 2010.

[15] R. Sifa, C. Bauckhage, and A. Drachen, "The Playtime Principle. Large-Scale Cross-Games Interest Modeling," in *Proc. of the IEEE Computational Intelligence in Games*, 2014, pp. 139–146.

[16] G. N. Yannakakis and J. Togelius, "A Panorama of Artificial and Computational Intelligence in Games," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 7, no. 4, 2015.

[17] J. Kawale, A. P., and J. Srivastava, "Churn Prediction in MMORPGs: A Social Influence Based Approach," in *Proc. of CSE*, 2009.

[18] J. S. Z. Borbora, K. Hsu, and D. Williams, "Churn Prediction in MMORPGS Using Player Motivation Theories and an Ensemble Approach," in *Proc. of IEEE International Coriference on Social Computing*, 2011.

[19] D. Nozhnin, "Predicting Churn: Data-Mining Your Game," *Gamasutra*, 2012.

[20] T. Mahlman, A. Drachen, A. Canossa, J. Togelius, and G. N. Yannakakis, "Predicting Player Behavior in Tomb Raider Underworld," in *Proc. Conference on Computational Intelligence in Games*, 2010, pp. 178–185.

[21] R. Sifa, S. Srikanth, A. Drachen, C. Ojeda, and C. Bauckhage, "Predicting Retention in Sandbox Games with Tensor Factorization-based Representation Learning," in *Proc. of IEEE CIG*, 2016.

[22] P. Yang, B. Harrison, and D. L. Roberts, "Identifying Patterns in Combat that are Predictive of Success in MOBA Games," in *Proc. of the Foundations of Digital Games*. FDG, 2014.

[23] C. Bauckhage, K. Kersting, R. Sifa, C. Thurau, A. Drachen, and A. Canossa, "How Players Lose Interest in Playing a Game: An Empirical Study Based on Distributions of Total Playing Times," in *Proc. of the IEEE Computational Intelligence in Games*, 2012.