

# Predicting Player Churn in the Wild

Fabian Hadiji<sup>\*§</sup>, Rafet Sifa<sup>†</sup>, Anders Drachen<sup>‡</sup>, Christian Thureau<sup>§</sup>, Kristian Kersting<sup>\*†</sup>, Christian Bauckhage<sup>†¶</sup>

<sup>§</sup>Game Analytics, Berlin, Germany

<sup>\*</sup>Technical University Dortmund, Dortmund, Germany

<sup>†</sup>Fraunhofer IAIS, St. Augustin, Germany

<sup>‡</sup>Aalborg University, Aalborg, Denmark

<sup>¶</sup>B-IT, University of Bonn, Germany

**Abstract**—Free-to-Play or “freemium” games represent a fundamental shift in the business models of the game industry, facilitated by the increasing use of online distribution platforms and the introduction of increasingly powerful mobile platforms. The ability of a game development company to analyze and derive insights from behavioral telemetry is crucial to the success of these games which rely on in-game purchases and in-game advertising to generate revenue, and for the company to remain competitive in a global marketplace. The ability to model, understand and predict future player behavior has a crucial value, allowing developers to obtain data-driven insights to inform design, development and marketing strategies. One of the key challenges is modeling and predicting player churn. This paper presents the first cross-game study of churn prediction in Free-to-Play games. Churn in games is discussed and thoroughly defined as a formal problem, aligning with industry standards. Furthermore, a range of features which are generic to games are defined and evaluated for their usefulness in predicting player churn, e.g. playtime, session length and session intervals. Using these behavioral features, combined with the individual retention model for each game in the dataset used, we develop a broadly applicable churn prediction model, which does not rely on game-design specific features. The presented classifiers are applied on a dataset covering five free-to-play games resulting in high accuracy churn prediction.

**Index Terms**—game analytics, churn, games, game data mining, churn prediction, free-to-play, freemium, behavior, behavior modeling

## I. INTRODUCTION

The application of behavioral analytics for the purpose of evaluating and understanding player behavior has within the past five years emerged from the sidelines to become a core component of commercial and academic game development. There are several reasons for this development, but generally they relate to a combination of technological advancement with respect to mobile devices; changing business models that have increasingly seen games be brought online, and broken with the traditional retail-based revenue models. In Free-to-Play (F2P) games, revenue is mainly driven by advertisements and in-game purchases. Thus, the ability to monitor, analyze and predict the behavior of the players is crucial to building a sustainable business [1]–[3].

Considering the market and business intelligence problems associated with contemporary game development, notably for online and mobile platforms, one of the most important factors for success is the ability to detect and define subscribers of a system (e.g. players of a game), that will leave sometime

in the future. This type of analysis is called *churn analysis* or *churn prediction*. Any subscriber, user or - in the case of games, *player* - leaving a service are generally referred to as “churners”, and the ratio of churners over non-churners as a function of time determines the churn rate [4]. In F2P games, churn rates are generally high with strongly skewed frequency distributions, typically seeing the vast majority of players leaving during the first minutes of play [1]–[3], [5].

This emphasizes the interest in predicting player churn, not only to detect leaving or increasingly uninterested players - which are increasingly expensive to recruit in the first place - in order to activate protocols for incentivizing them to remain in the game, but also in order to maximize revenue, notably in the context of in-app purchases (IAPs). Retention and monetization are two different, but in practice often closely interlinked, measures of the success of a game [2], [3], [6].

In this study we concentrate on predicting churn in five mobile/social, F2P games. We outline the challenges in performing churn prediction in games, and define the behavioral features which are useful to inform churn prediction [7], [8]. Having proposed different models of churn, we use a combination of statistical models of the retention likelihood and non-linear behavioral functions that capture the player’s engagement behavior across five different games.

## A. Contribution

As significant numbers of games are shifting to the F2P, i.e. “freemium”, revenue model, predicting churners has become a key challenge in game data mining and game analytics. To the best of our knowledge, together with [9] this work is a worlds-first in several ways: 1) It is the first time a multi-game telemetry dataset from F2P games has been used for academic research; 2) It is the first time churn prediction has been performed across multiple games; 3) It is the first time only game-agnostic features are used for prediction; 4) It is the first time churn in games is defined formally and different definitions of churn are used for evaluating churn, with both of these definitions grounded in game design; 5) The work presented is one of the first to consider churn prediction, as compared to churn description, in games. Apart from the novelty of the work, there are three main contributions in this paper, as follows: 1) we identify and formalize the problem of churn prediction for interactive systems that need generic models that can be parametrized efficiently. This includes two

data extraction methods and two distinct problem definitions that align with industry standards in mobile/F2P or social games. The two different definitions of churn - tight and relaxed window - are evaluated rather than simply considering churn to be the last time a player is seen. This helps with situations where churn profiles vary across games or where the nature of game design or the way value is assigned to a player (e.g. value based on total purchases vs. social value) means different definitions of churn are relevant. The focus here is on churn prediction, not how to incentivize players towards retaining them. Relaxed churn, or soft churn, for example, considers churn to be a gradual process taking many days or even weeks. Using soft churn has the advantage that it permits monitoring of the processes leading to player churn. 2) we introduce and define a set of generic behavioral features that allow churn prediction for a broad range of F2P (and other) games. Notably, the most important features center around user engagement and thus are independent of the actual game content. It should be noted that previous works achieved high accuracy predictions mostly by relying on behavioral features relating to game-specific features (see related work below). 3) we evaluate trained models on five different F2P games, achieving high prediction accuracy. Prediction accuracy matches or exceeds that of the few solutions for specific games published previously. It should be noted that we could not verify our models on the data of related work as the datasets used for such research are usually proprietary.

## B. Related Work

In F2P games, analytics is foundational for managing and optimizing design and monetization [1]–[3]. Within academia, behavioral data mining has a long history, but within the context of digital games specifically this venue of investigation has historically been the focus of game AI [10]–[12] or network analysis [13], [14]. However, in parallel with the rapid uptake of analytics, user research and business intelligence practices in the game industry, player behavior has also become the subject of more widespread research, covering e.g. learning, impact assessment, decision making, behavioral profiling, prediction, retention, churn and engagement [15], [16], integration in game development process, management of analytics processes, and so forth; all with the overall goal of informing game development, whether in order to optimize user experience, engagement, monetization, learning, etc. [1].

We focus here on publications that focus on using behavioral analysis directly or indirectly for churn analysis. Churn analysis was only recently adopted in game contexts, but has been studied in a variety of disciplines for decades. This notably includes retail banking [4], insurance [17], telecommunication [7], [18], [19], on-line advertisement [20] and Community Based Question Answering platforms [8].

Within the specific context of digital games, most of the work on churn prediction and related topics such as retention, generally use accumulated information about the players and predicted churn in a single setting, and do not define churners in general. Furthermore, the datasets used are generally from

single games, rather than building models applicable across games. An important reason for this is the confidential and valuable nature of behavioral telemetry, which secondarily results in a general lack of available industrial data for use by academic researchers [6].

Important exceptions include [14], who investigated game popularity across the top 50 games from the now de-funct GameSpy service, using average number of players per day and noted a power law distribution. Bauckhage *et al.* [21], who modeled players' engagement to the games using life-time analysis across five major commercial titles. Modeling the player's interest as a hidden variable, Bauckhage *et al.* extracted playtime information and showed how the interest can be represented in terms of lifetime distributions and their corresponding processes. As also noted by [14], the results indicate how the interest of the players decrease over time in a power-law fashion, although Bauckhage *et al.* [21] go a step further in explaining why this pattern emerges.

Another example is Weber *et al.* [22] who analyzed the correlation between retention and in-game features for a football game, by using regression modeling to identify the most influential features on player retention. Kawale *et al.* [16] focused on the MMORPG EverQuest II, using diffusion models for social networks combined together with playtime distribution modeling to investigate the relationship between the social impact of a player in a game and churn. However, the precision and recall rates obtained are around 50% at their best, which could indicate that different classifiers should have been explored. Also using a dataset from EverQuest II, Borbora *et al.* [15] used hybrid (supervised and unsupervised) methods to predict churn. Defining churners as the people who canceled their subscription to the game, or had otherwise not been active for 2 months. Different gameplay features were extracted along with meta data information such as how many characters the player posses and etc., to learn a binary decision function, that determines whether the player is going to churn or not.

A substantial part of the work on retention and by extension churn in games stems from network science, where the goal has been to understand the impact of network conditions on quality of service and player satisfaction. The typical game type investigated is Massively Multi-Player Online Role-Playing Games (MMORPGs), where data can be obtained via mining the client-server stream. To outline a few examples relevant to the current work, Pittman and Gauthier [23] mined client-server streams from one server from each of the two MMORPGs World of Warcraft and Warhammer Online, focusing on measuring and modeling player distributions, session lengths, and player movements in the virtual worlds in order to inform MMORPG server architectures. They examined session lengths and arrival/departure rates, concluding that both MMORPGs experienced significant churn, with churn understood as players joining and leaving the servers, not necessarily the game per se. Feng *et al.* [13] applied traffic analysis to telemetry data from the MMORPG EVE Online. The dataset used covered the early history of the game from

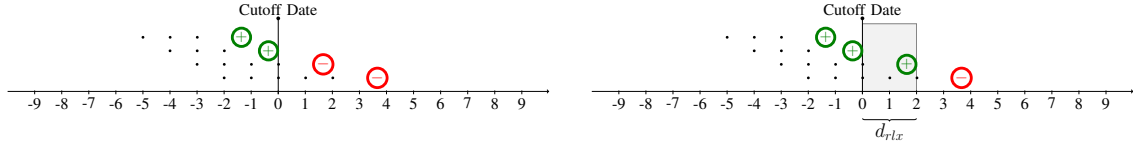


Fig. 1: **(left)** **P1** considers all users as churners that do not return after a specific cutoff date. Here, the upper two players are churners (green +), while the lower two players do return after the cutoff date (red -). **(right)** Compared to **P1**, **P2** considers the third user from the top also as being churned because he will churn in the next  $d_{rlx}$  days. (Best viewed in color)

2003-2006. Several conclusions were reached. For example, the authors examined the rates of players joining and quitting the game on a month-by-month basis, finding that for EVE Online, rates of joining and leaving followed each other closely, with however a gradually increasing player number from a few thousand in mid-2003 to 150,000 in late 2005. The authors concluded that player churn increases over time, speculating that the reasons for this could be that new players come into the game at a disadvantage compared to the existing players - and this disadvantage increases over time.

## II. CHURN PREDICTION

In the following, we treat churn prediction as a binary classification task, i.e. classifying each player as churned or returning. The classifiers are trained on labeled data of already observed players but of course require actual data of a target player for the prediction. Thus, given the information of a player up to a certain point in time, from now on referred to as *cutoff date*, the classifier decides if the player has churned or will return. In the first and most straightforward interpretation of the churn prediction task, we consider a player without a single session after the cutoff date as churned. We will refer to this problem definition as **P1**. See Fig. 1(left) where these players are marked green while red is used for returning players. **P1** only allows a very harsh decision on churning players, which is not very useful for real-world applications as with no play session left, the chances of reactivating the player in question is very low (at least from within the game). To relax this definition, and thus make it more applicable for real-world settings, we can label players as churners that have a low number of sessions or days to play after the cutoff date (soft churn). The remaining days of play have to fall into the range of a sliding window of size  $d_{rlx}$ . This relaxation can be interpreted as a separation of engaged and no longer engaged players, as opposed to the harsher separation of churned and non-churned players. From an industry point of view, this distinction is valuable as the players who are likely to quit playing soon are easier to reach and potentially incentivize to stay in the game, whereas players who have already left are harder to reach and incentivize. We will refer to this relaxed problem definition as **P2**. Fig. 1(right) shows an example of **P2** and also allows for a comparison of **P1** and **P2**.

## III. DATA PREPROCESSING

The data we consider in this paper contain observations of player behavior in five different games provided by GameAnalytics ([www.gameanalytics.com](http://www.gameanalytics.com)). The data was gathered

over a five month timespan in 2013. In total, we analyzed about twenty million play sessions. The data is normalized so that game content dependent features are discarded. The primary feature representation is centered around sessions. If applicable, additional information is added (e.g. in-game purchases). Please note that this does not violate the basic idea of game content independent churn prediction, as in-game monetization is common nowadays. We exercised some standard data cleaning procedures (e.g. we discard possibly wrong data such as one user starting thousands of sessions in a few seconds). Note that not all five games stretch over the same time period, or the same lifecycle of games. For example, we cannot observe an increase in the user-base for every game and instead these games already start with a high number in daily active users. Moreover, some games still have a loyal user base and as such a noticeable reduction in users has not been observed yet. While this certainly makes the classification task more difficult (and also makes the results more difficult to compare), it is a rather common scenario. Due to confidentiality reasons, we cannot give more detailed information about the games, nor the titles of the games.

### A. Data Generation

In the following, we denote classes of churned players as TRUE and returning players as FALSE. As we consider supervised classification, the task now is to build models that correctly predict the corresponding class labels for a set of test data. However, it is crucial to decide on the absence time  $d_{chrn}$  (in days) after which a player is considered to have left the game. For practical applications, the generation of training and test data is critical and directly influences classifier accuracy when used for live data. We employ two methods for generating data. The first method is a more classical/straight-forward data generation method, whereas the second method is more optimized towards real-world applications, e.g. to be used by game developers to predict player churn on a regular basis with updated models.

**(M1)** The simplest way of generating training examples is to look at all churned players and consider these as positive examples. To generate negative examples, we sample from all players that have played more than a single session. We start with the first session and randomly select a cutoff session. This subset of sessions then defines a negative example because there will be future sessions of that player. This process is depicted in Fig. 2(left). This method allows to generate a large dataset quickly and the positive and negative classes can be

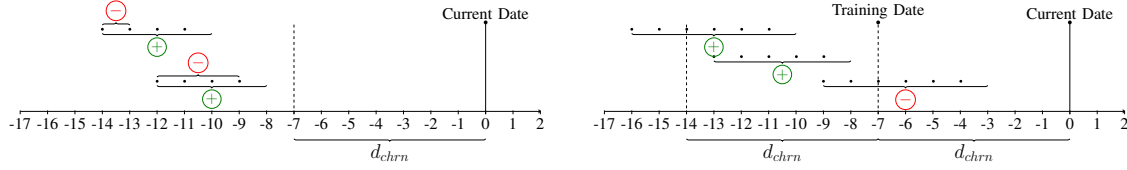


Fig. 2: (left) **M1** creates examples of churned players from all players in the database that have not been active in the past  $d_{chrn}$  days. Negative examples are generated by sampling a subset from these sessions. (right) Data generation method **M2** depends on a specific point in time and generates positive and negative examples based on this point. (Best viewed in color)

balanced by discarding data. However, as we will also see in the empirical evaluation, many users play a game only once, and hence cannot be used to generate a negative example. Please note that this way of generating training data is difficult to apply in real-world settings as all data of a player is assumed to be available regardless of the current date. However, as it is straight-forward and more common in the scientific literature in general, we decided to include it.

(**M2**) Another way of generating data is to take only sessions within a certain range into account. This approach takes a look at players at a chosen day and a time window of  $d_{chrn}$  days. This training day is depicted in Fig. 2(right) with the dashed line left of the current date. FALSE examples are all players that have logged in at this point and will return in the future, i.e., in the time between the current date and the dashed line left of it. However, generating an adequate number of TRUE examples is slightly more complex. A naive approach would consider all players within the window up to the training day that do not return. Unfortunately, this can result in highly unbalanced datasets, as the number of positive examples increases much quicker than the number of negative examples (assuming that we advance the window over time). Thus, we define an additional sliding-window for the positive examples and only data within this window is considered. Again, we use  $d_{chrn}$  days and this point is depicted with the outer left dashed line in Fig. 2(right). We use this window size because all players that have only been active before that date are considered as churners on the training day already. In contrast to **M1**, we can easily incorporate the absence time, i.e. the time since the last session, as a feature describing player behavior (note that also for **M1** this would be possible but it requires another sampling of dates as the reference absence day and thereby adding additional complexity). As we are only considering players in the sliding window, this absence time is bounded by  $d_{chrn}$  days. For the evaluation presented later, we define training and test data by randomly choosing sliding window center dates, using the data up to that point in time. For example, a possible training date is  $d_{chrn}$  days before the test date. In a live system, the most up-to-date classifiers would be trained with data of training day  $d_{chrn}$  days back in time. Thus, classifiers would change over time and thereby reflect changes of the game content or the user base.

### B. Data Statistics

We generated for each game one dataset based on method **M1** that contains 50,000 randomly selected users. Using

TABLE I: Dataset Statistics for P1

Game	M1		M2	
	TRUE	FALSE	TRUE	FALSE
1	0.707	0.293	0.495 0.824	0.505 0.176
2	0.623	0.377	0.849 0.806	0.151 0.194
3	0.690	0.310	0.775 0.748	0.225 0.252
4	0.896	0.104	0.978 0.979	0.022 0.021
5	0.648	0.352	0.378 0.373	0.622 0.627

TABLE II: Dataset Statistics for P2

Game	$d_{rlx} = 1$		$d_{rlx} = 2$	
	TRUE	FALSE	TRUE	FALSE
1	0.618	0.382	0.672	0.328
	0.914	0.086	0.948	0.052
2	0.910	0.09	0.932	0.068
	0.939	0.061	0.953	0.047
3	0.893	0.107	0.925	0.075
	0.883	0.117	0.916	0.084
4	0.995	0.005	0.998	0.002
	0.995	0.005	0.998	0.002
5	0.564	0.436	0.645	0.355
	0.584	0.416	0.671	0.329

method **M2**, we generated a training and a test dataset for each game for different dates. As mentioned above, **M2** works around a specific point in time and we chose the dates in such a way that there are  $d_{chrn}$  days between the training and test date. In all of our experiments, we set  $d_{chrn} = 7$ . We chose a value of 7 because we observed in the data, that more than 95% percent of the absence times lie in this windows and it contains the full cycle of a week with every weekday. The size of the resulting datasets for training and testing depends on the number of active players in the used timespan of each game and hence varies. However, we ensured that each test dataset had several ten thousand players for evaluation.

Tbl. I and Tbl. II show the distribution of TRUE (churned) and FALSE (non-churned) labeled players among the five games. For each game, the first row presents statistics of the training dataset and the second row of the test data. We can see

that (overall) **M1** leads to more churned than returning players. This can be explained by the fact that all games contain a large number of players that play only once. Unfortunately, these data points only allow to construct negative examples and thus can bias the class distribution. Looking at the statistics of the datasets generated by method **M2**, the first row depicts statistics from the training dataset and the second row the test data. The class imbalance for games 1-4 is rather large compared to game 5. Overall, game 5 has a higher retention (especially day-1 retention is higher).

As **M2** is the arguably more important data generation method due to its better applicability in real-world settings, we evaluated **P2** only using method **M2** (first results also showed that the behavior/performance is similar to **P1**). Instead, we modify the newly introduced parameter of the sliding window size. The two datasets differ in the selection of the sliding time window  $d_{rlx}$ . For the first dataset we set  $d_{rlx} = 1$  and for the second we set  $d_{rlx} = 2$ . We observe similar statistics as in the datasets for **P1** with the exception that the **TRUE** class also dominates especially in game 5. Also note that with an increase of the window size, the **TRUE** class becomes more and more dominant, where eventually, all players have churned.

#### IV. FEATURES

One of the most important steps towards successfully predicting churn is the proper selection of features that capture the behavior of the players and allow prediction of player churn. As we are building a game-content independent churn prediction model, we extracted a set of features that are *universal* among all of the games in our platform. In this section we explain the features we used in our model and how they are related to churning behavior of the players. In Sec. V we show the relative influence of the above features by evaluating the importance of different features in the classifiers trained for each game.

Starting with basic temporal features that have an obvious influence to churning behavior, we use **Number of Sessions**, i.e. the number of past sessions a player was actively involved. In addition, we use the **Number of Days**, which represents the number of days since the player has signed up for the game. These are presumably important features as rapid churn is a common feature in F2P games. We also observe for some of the games in the dataset that there are players churning right after their first or second day and some churning after only one or two play-sessions. Additionally, for data generation method **M2** we incorporate the **Current Absence Time** which represents the elapsed time since player’s most recent activity.

Session-wise temporal information has, for MMORPGs, been shown to be useful in modeling player departure [13]. For session level granularity we therefore consider **Average Playtime per Session** and **Average Time Between Sessions**, where the former represents the total playtime of the player divided by number of sessions and the latter represents the average time between the sessions, i.e. intersession time.

For capturing playtime, we use two temporal models of the user’s time spent in the game. We model the playtime

over sessions by fitting a power-law function for each player’s individual observations, following past research on the influence of playtime for predicting decrease of interest in playing a game [21]. In this work, we use the **Playtime Model Parameters**, i.e. parameters of the player-based power law function (fitted to past observations), to represent the player’s playtime history till the day of prediction. Additionally, to add the influence of the average loss of interest in a particular game, we incorporated a **Retention Value**. That is, having built a retention model based on the average player retention for a game (by fitting a function), we obtain a player-based retention value by their respective day of play.

Given the importance of in-app purchases to F2P games, we also integrate 4 virtual economy related features: **Premium User Flag**, **Predefined Spending Category**, **Number of Purchases** and **Average Spending per Session**. We use **Premium User Flag** for denoting if a particular player has purchased at all. In order to categorize the overall purchase experience of the players we feed a set of predefined spending categories of the players to our model as well. The spending categories segment player according to their relative spending into one of three different groups, the top spenders in the first group, average spenders in the second group, and light spenders in the last group. This is followed by the total **Number of Purchases** a player has made. The final virtual economy related feature is the **Average Spending per Session** which we consider as another important indicator of the purchase frequency where we measure the average amount spent per played sessions.

#### V. EXPERIMENTS

In this section we present an experimental evaluation of the introduced churn prediction method. The evaluation presented considers both problem definitions **P1** and **P2**, as well as both data generation methods **M1** and **M2**, applied to each of the five games. Combined, the dataset used spans several hundred thousand players, spanning millions of sessions. For details on the data generation and dataset statistics please refer to Section III. For **M1**, all data sets and combinations are evaluated based on a 10-fold cross validation scheme.

For the purposes of predicting player churn, we compared a range of different classifiers, including Neural Networks (NNs), Logistic Regression (LR), Naive Bayes (NB) and Decision Trees (DTs). As an in-depth comparison of classifiers is beyond the scope of this work, we only report on key results here. That being said, as can be seen in Tbl. III, we found that decision trees [24] performed (overall) the best in terms of weighted averaged F1-score for one of our games. Hence, in the remainder of this section we only show the results of decision trees. A concrete advantage of decision trees is that they provide the means for relatively simple and intuitive explanations of the observed behaviors, which assists with describing the most important and influential features in churn prediction.

The retention curves of the five games show substantial churn rates, but a note should be made for game 4, which loses players at a very high rate, and has the lowest number

TABLE III: Results for Different Classifiers

Method	F1-Score TRUE	F1-Score FALSE	WAvg. F1-Score
DTs	0.945	0.782	0.916
LR	0.928	0.710	0.889
NNs	0.910	0.705	0.874
NB	0.887	0.418	0.804

TABLE IV: Results for P1

	M1		M2	
1	0.617	0.916	0.824	0.119
	0.824	0.119	0.945	0.782
2	0.584	0.860	0.759	0.295
	0.759	0.295	0.903	0.682
3	0.570	0.925	0.816	0.020
	0.816	0.020	0.950	0.852
4	0.846	0.994	0.945	0.000
	0.945	0.000	0.997	0.842
5	0.591	0.851	0.780	0.244
	0.780	0.244	0.804	0.880

TABLE V: Results for P2 with M2

Game	$d_{rlx} = 1$		$d_{rlx} = 2$	
1	0.883	0.910	0.932	0.293
	0.932	0.367	0.944	0.293
2	0.921	0.940	0.963	0.271
	0.963	0.271	0.973	0.257
3	0.918	0.945	0.957	0.620
	0.957	0.620	0.972	0.646
4	0.992	0.997	0.997	0.000
	0.997	0.000	0.999	0.000
5	0.821	0.844	0.850	0.781
	0.850	0.781	0.885	0.760

of returning players. Thus, for the task of classification, having a minority class can result in bad prediction for this particular class while still having an overall high prediction accuracy. By using f-scores we try to alleviate these effects and make the class imbalance more obvious.

#### A. Results for P1

In Tbl. IV, we show results for experiments conducted for player churn definition **P1**. For each game, the upper row shows the weighted averaged F1-score and the lower row shows the individual F1-scores for churners and non-churners.

For **M1**, the weighted F1-scores are close to .60 and it can be seen that classifying non-churning users has a worse accuracy, possibly due to the unbalanced datasets and hence fewer positive learning examples. Compared to the other games, the overall performance for game 4 is surprisingly good. However, looking at the individual scores in more detail, one can see that the classifier completely fails on the non-churners. For this game, the group of returning users is negligible and the DT learner fails to find a good model for that class.

For **M2** the results look different. As it can be seen in the right column of Tbl. IV, the F1-scores are rather high and the scores for the class of non-churners is much higher too. Overall, these F1-scores come close to or exceed the results from the literature, however, it is difficult to compare directly with related work as different games and behavioral features are used here. Also note that the games considered here are F2P/freemium games, and can be considered more “casual” than the major commercial MMOGs used in the related work. Seeing that a class imbalance, as the consequence of less popular games, leads to worse classification accuracy, we gain confidence in the ability to reproduce the results presented here for other games and game genres.

Besides game 4 with **M1**, the classification performance is similar for all games for a specific data generation method. The observed differences in scores are due to the fact that the user behavior is not equally well captured in all games by the means of the given features. Overall, for data generation method **M2** the classification performance is higher and especially more stable for both classes. As mentioned before, **M2** has the capability of using the additional feature “current absence time”. As we will see in the more detailed discussion of

the features below, this particular feature is one of the most important features for predicting player churn.

#### B. Results for P2

In Tbl. V, we show results for the experiments conducted for player churn definition **P2** in combination with data generation method **M2**. We present here only results for **M2** because it returns favorable results over **M1** as we have already seen in the previous section. Here, we varied the window size  $d_{rlx}$ . Similar to the previous experimental results for **M2**, the overall F1-scores are high and interestingly, the combined (weighted average) F1-scores are similar for both window sizes. However, the prediction of the FALSE class becomes often slightly worse with an increasing window size  $d_{rlx}$  as the class imbalance becomes heavier. Thus showing again, that it becomes more difficult for the classifier to find the returning players as this group vanishes.

For example, for game 4 the amount of non-churners is negligible even for  $d_{rlx} = 1$ . It is very important to note that, the results show that the choice of  $d_{rlx}$  is crucial in real world applications and its value needs to be adjusted to match the average user engagement for the respective game. Clearly, game 5 shows the best retention and the adjustment of  $d_{rlx}$  slowly changes the distribution of the players. The number of non-churners in a two day time window is still sufficiently large that the developers could think about approaches influencing the churn behavior of these players.

The parameter  $d_{rlx}$  offers an easy and intuitive way of adapting the otherwise general data generation procedure **M2** to the requirements of each game. The experiments show that it has indeed a certain influence on the classification accuracy and it allows to tailor the classifier system to different games or game genres. Thus, there is still room (but no need) for optimizing churn prediction using the proposed classifier system, once additional (or historic) data for a particular game becomes available.

From the experiments we can see that for some games the problem of predicting player churn might be ill posed. In particular, if the vast majority of players churns after only one or two play sessions, the task of predicting player churn seems to become a *random process* in which it is difficult to establish rules about why players are leaving the game (for example



consider game 4). For most other cases, the proposed classifier system leads to high accuracy player churn prediction which is independent of the actual game content or game genre, as the used features are universal.

### C. Feature Importance

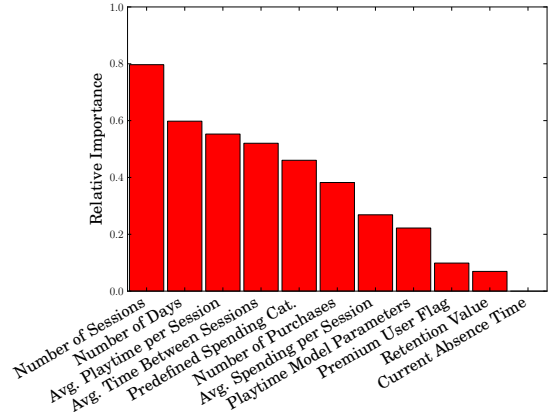
Knowing that players are about to leave a game is useful in its own right. However, taking a step beyond pure churn prediction, gaining an understanding about why a player is about to leave is even more useful as it allows developers to try different actions and incentivize the user to remain. There is, obviously, an infinite or near infinite amount of reasons a player might cease to play a particular game. We cannot capture all of these, but we can analyze which features, of the ones we introduced, are most important w.r.t. their information gain and classifier accuracy.

To analyze relative importance of the behavioral features, we closely inspect the learned decision trees. In the first step, we counted how often each decision tree makes use of every feature. For **M1**, the features **Number of Sessions** and **Number of Days** occur most often. In fact, these two features were the most decisive features across all games. A full picture of the relative importance of each feature for **M1** is given in Fig. 3a. Interestingly, the root node was identical for all trees and branches based on the feature **Number of Days**. Besides these two features, **Average Time Between Sessions** was also observed frequently apart from game 1. This aligns well with the results reported in [13] where it was shown that the final inter-session time is significantly longer than the early ones. Thus, an increase in the time between two sessions indicates that a player loses interest which is captured by the learned decision tree rules. However, our results go even beyond the results by Feng et al. as we report these findings for variety of games.

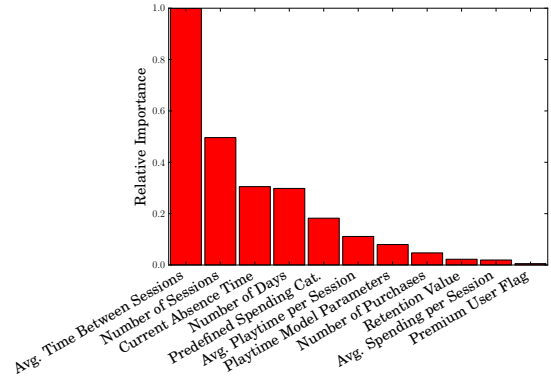
For **M2**, we observe a change in feature importance (see Fig. 3b for the full details). Interestingly, the **Average Time Between Sessions** now becomes the most important feature and appears in every decision tree. This further underlines the connections to the findings by Feng et al. as discussed in the case of **M1**. Technically, the **Average Time Between Sessions** is usually the root node of each learned tree which further supports its importance. Additionally, we can now observe that the **Current Absence Time** is always among the top five features. This also partly explains why the prediction made by the trees learned based on **M2** are more stable. **M2** adapts over time and thus focuses more on the current user base and due to its access to real-time features, such as **Current Absence Time**, the classification accuracy is more stable w.r.t. the class imbalance issues.

## VI. CONCLUSION AND FUTURE WORK

*Predictive Analytics* has become an important part of business intelligence and a core requirement of data-driven enterprises. This is also the case in game development, where the increasing use of “freemium” business models, or hybrid retail-freemium models, emphasize the need for analyzing



(a) Feature Importance for M1



(b) Feature Importance for M2

Fig. 3: The importance of the features.

the behavior of game players in order to understand their behavior, improve design and optimize monetization. As in all subscription-based systems, for games taking steps to keeping a customer is usually cheaper than obtaining a new customer and this makes churn prediction very important for F2P games that generate revenue via in-game purchases and in-game advertising [2], [3], [6], [21], [25].

In this paper, the challenge of predicting player churn in freemium games, is addressed, and a machine learning approach presented which can be applied across games, under real-life conditions, i.e. in the wild. The approach is tested using data from five commercial games across mobile- and web-based social-online platforms. This research presented here is together with [9], to the best knowledge of the authors, the first study to churn prediction in F2P games, operates across multiple games, and uses solely games-agnostic behavioral features such as playtime and session time. It is the first time churn prediction has been performed across multiple games, and it is the first time that churn in games has been formally defined. We summarize the contributions as follows:

- By analyzing the churn prediction problem specifically for digital games, we formally introduce two data extraction methods to analyze and provide predictive models for churning behavior in games. The first data extraction

method defined, **M1**, is based on analyzing only churned players by creating negative examples by selecting player based random cut-off points. In comparison, by setting an arbitrary time in the past as a cut-off point, we defined a data extraction methodology **M2** more suitable for the real-world setting, which uses all of the player information (i.e. the churning and non-churning players).

- Similarly, two formal models of predicting churn are defined, that depend on whether we are detecting churners for a specific cutoff point or in a time range (“hard” vs. “relaxed” churn windows). In the first problem definition **P1** we detected the players if they have not returned after a specific date. The second problem definition proposes a relaxed definition, **P2**, where we analyzed the churners for an arbitrary time window, which is flexible and can be adjusted for example according to game-specific requirements. Specifically, we predict whether a player will return to play the game within a certain amount of time after the cutoff date.
- Rather than building a prediction model based on game specific features, we present *universal* basic and composite F2P/freemium behavioral features that can be used when predicting churn in games, for example playtime, session time and intersession time.
- Finally, we employ different prediction classifiers (e.g. decision trees, naive bayes) on the different combinations of sampling and churn definition, across five different commercial F2P games (online/mobile), obtaining high accuracy scores using decision trees. Furthermore, we show which behavioral features are important when determining churners and non-churners in games and how the importance of the features depends on the analyzed games.

Given the young age of the domain of game analytics, and behavioral prediction in games in general, there are a great number of potential areas for follow-up work. In brief, we plan to extend the presented model to include an *adaptive* approach to determine the length of the relaxed churn window (or soft churn window), which is informed by the behavioral telemetry, so that the *window length* is game specific, or even genre specific. We will also use different cutoff dates for **M2** and seek to combine classifiers from different dates. Furthermore, we are aiming to investigate the young area of cross-game analytics to make inferences about a game using data from similar games. Namely, we will investigate similarity measures between freemium games to be able to make early predictions about a game that, for example, has just been released and does not have enough users for prediction analysis. Finally, having built the prediction models based on full trees here, we are also aiming to investigate how the prediction results will behave when we build our models based on pruned trees.

## VII. ACKNOWLEDGMENTS

We thank GameAnalytics for providing the datasets. The work in this paper was carried out within the Fraunhofer and

University of Southampton research project SoFWiReD which is funded by the Fraunhofer ICON initiative.

## REFERENCES

- [1] El-Nasr, M.S. and Drachen, A. and Canossa, A., *Game Analytics: Maximizing the Value of Player Data*. Springer, 2013.
- [2] T. Fields and B. Cotton, *Social Game Design: Monetization Methods and Mechanics*. Morgan Kaufmann, 2011.
- [3] W. Luton, *Free-to-Play: Making Money From Games You Give Away*. New Riders, 2013.
- [4] T. Mutanen, J. Ahola, and S. Nousiainen, “Customer Churn Prediction—A Case Study in Retail Banking,” in *Proc. of ECML/PKDD Workshop on Practical Data Mining*, 2006, pp. 13–19.
- [5] “The playtime principle: Large-scale cross-games interest modeling,” in *Proc. IEEE CIG*, 2014.
- [6] A. Drachen, C. Thureau, J. Togelius, G. Yannakakis, and C. Bauckhage, “Game Data Mining,” in *Game Analytics: Maximizing the Value of Player Data*, M. El-Nasr, A. Drachen, and A. Canossa, Eds. Springer, 2013.
- [7] 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.
- [8] G. Dror, D. Pelleg, O. Rokhlenko, and I. Szepkter, “Churn Prediction in New Users of Yahoo! Answers,” in *Proc. of the 21st international conference companion on World Wide Web*, 2012.
- [9] “Churn prediction for high-value players in casual social games,” in *Proc. IEEE CIG*, 2014.
- [10] G. Yannakakis, “Game AI Revisited,” in *Proc. of ACM Computing Frontiers Conference*, 2012, pp. 285–292.
- [11] C. Thureau, T. Paczian, and C. Bauckhage, “Is Bayesian Imitation Learning the Route to Believable Gamebots?” in *Proc. GAME-ON NA*, 2005.
- [12] R. Sifa and C. Bauckhage, “Archetypical Motion: Supervised Behavior Learning Using Archetypal Analysis,” in *Proc. IEEE CIG*, 2013.
- [13] 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.
- [14] C. Chambers, W. Feng, S. Sahu, and D. Saha, “Measurement-based Characterization of a Collection of On-line Games,” in *Proc. of ACM SIGCOMM Conf. on Internet Measurement*, 2005.
- [15] Z. Borbora, J. Srivastava, K.-W. Hsu, and D. Williams, “Churn Prediction in MMORPGS Using Player Motivation Theories and an Ensemble Approach,” in *Proc. of IEEE International Conference on Social Computing*, 2011, pp. 157–164.
- [16] J. Kawale, A. Pal, and J. Srivastava, “Churn Prediction in MMORPGs: A Social Influence Based Approach,” in *Proc. of the 2009 International Conference on Computational Science and Engineering*, 2009.
- [17] K. Morik and H. Köpcke, “Analysing Customer Churn in Insurance Data—A Case Study,” in *PKDD*, 2004, pp. 325–336.
- [18] J. Ferreira, M. Vellasco, M. Pacheco, R. Carlos, and H. Barbosa, “Data Mining Techniques on the Evaluation of Wireless Churn,” in *Proc. of European Sym. on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2004, pp. 483–488.
- [19] H. Hwang, T. Jung, and E. Suh, “An LTV Model and Customer Segmentation Based on Customer Value: A Case Study on the Wireless Telecommunication Industry,” *Expert systems with applications*, vol. 26, no. 2, pp. 181–188, 2004.
- [20] S. Yoon, J. Koehler, and A. Ghobarah, “Prediction of Advertiser Churn for Google Adwords,” in *Proc. of JSM*, 2010.
- [21] C. Bauckhage, K. Kersting, R. Sifa, C. Thureau, 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. IEEE CIG*, 2012.
- [22] B. Weber, M. John, M. Mateas, and A. Jhala, “Modeling Player Retention in Madden NFL 11,” in *IAAI*, 2011.
- [23] D. Pittman and C. GauthierDickey, “Characterizing Virtual Populations in Massively Multiplayer Online Role-playing Games,” in *Proc. of the 16th Int. Conf. on Advances in Multimedia Modeling*, 2010, pp. 87–97.
- [24] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.
- [25] E. B. Seufert, *Freemium Economics: Leveraging Analytics and User Segmentation to Drive Revenue*. Elsevier, 2014.