Prediction of Player Churn and Disengagement Based on User Activity Data of a Freemium Online Strategy Game

Karsten Rothmeier
Deloitte
krothmeier@deloitte.de

Nicolas Pflanzl, Joschka A. Hüllmann
Department of Information Systems
Westf. Wilhelms-Universität Münster, Germany

Mike Preuss
LIACS, Universiteit Leiden, The Netherlands
m.preuss@liacs.leidenuniv.nl

Churn describes customer defection from a service provider. This can be observed in online freemium games, where users can leave without further notice. Game companies are looking for methods to detect and predict churn to enable management reaction. The recorded data of games can be analyzed for this purpose. We conducted a case study based on data from the freemium game The Settlers Online. Churn detection was achieved by application of four different labeling approaches, based on common churn and disengagement definitions within the game analytics literature. In order to model predictive classifiers, features were computed from the raw game data. Eight different machine learning algorithms returning binary classifications were applied. The results were compared for all algorithms regarding all labeling approaches. Random forests with sliding windows were the best solution in our case, returning AUC values higher than 0.99, thereby enabling prediction accuracies of 97% in our data set. The results were confirmed by tests on an independent data set and in our discussion, we offer guidance on the interplay of feature engineering, labeling approaches—in particular disengagement—and machine learning algorithms for churn prediction. Our recommendations are valuable for game companies and academics, who pursue similar studies.

***Index Terms*—Player churn, Disengagement, Classification, Machine Learning**

I. Introduction

In the last decade, *freemium games* (also free-to-play or F2P games) have emerged within the sector of online games [1]. The typical payment model of such games allows players to play the core game for free, but imposes certain restrictions on gameplay that are intended to move players towards making a payment, for instance for faster progression. Consequently, the financial success of an F2P game requires a design that is able to retain players while making them interested in spending money for the game. However, retaining the players is an ongoing challenge as the reasons for leaving a game are manifold. Previous studies show that the decision to leave the game can be related to social features and the level of collaborative play, game progress and achievements, or adverse in- and out-of-game experiences such as cheating, griefing, or account theft [2, 3, 4, 5, 6, 7]. A player's decision manifests itself in changing gaming behavior, which we anticipate by a thorough analysis of such behavior in our study. The act of leaving the game, i.e., to stop playing, is called the customer *churn*. Churning in freemium games can be regarded as customer churn in noncontractual settings, a case seen as problematic in the more general customer churn literature [8], and thus requiring advanced methods for its prediction.

While customer churn is a research topic with a long history in the domain of service providers, such as banking, insurance, and telecommunications and is especially prevalent in marketing research [9], churn analyses in game companies often lack sophisticated techniques and rely on simplistic methods instead (e.g., just defining a threshold of days without login) [10]. To examine the use of more advanced methods for the games industry, this paper focuses the recent surge of machine learning (ML) techniques, which are nowadays used in a wide array of different areas, such as spam filtering, face recognition, and self-driving cars. The term machine learning refers to computational methods that aim to optimize the performance of a system by using past example data (or "training data") [11, 12]. These techniques can be used to predict player behavior in games, for example, whether players are going to churn.

In this paper, we report on the results of an exploratory case study and examine the possibilities for player churn prediction in a freemium game. The study is a cooperation with the German video game developer and publisher Blue Byte, a subsidiary of Ubisoft Entertainment SA. Blue Byte creates freemium online games, the most popular being The Settlers Online[1] (TSO), which is the subject of this work. Before our study, there was no churn prediction method in place for the game and the publisher did not know whether it was feasible to detect player churn before it occurs and what kind of behavior is important. Our study closes this gap by analyzing event log data generated through TSO players using ML techniques.

A challenge in the study was that TSO users can withdraw from the game without any prior notice. Thus, the detection of churn strongly depends on the utilized definition of a user having left the game. Furthermore, several days, weeks, or months may elapse before a decision about whether a particular user has defected can be made, thereby causing long reaction times for a publisher. Various class labeling approaches exist to determine which time frame constitutes a defection from a game, or in other words, the approach defines the label that constitutes churn. We identify the class labeling approach as well as the ML model, which together perform best on our given data.

This study contributes to the current state of research about churn prediction in games in various ways:

[1] See: www.thesettlersonline.com/. Last accessed: 2018-06-07.

- We directly compare four different labeling approaches, eight different classification algorithms, and use more than 1000 features, providing a comprehensive overview. Our approach is not to show that a specific method is good, but to evaluate the available methods in all reasonable combinations with labeling methods in order to detect what yields the best results.
- Most works are limited to the concept of churn and disregard *disengagement*. However, we distinguish between both (see Section III) and give a comprehensive recommendation for the labeling approaches.
- This work is in line with most recent publications and uses the ROC (receiver operating characteristic) curve and the AUC (area under curve) measure for evaluation, which are harder to bias than the metrics often used in the past such as precision and recall. Previous works regularly reach an accuracy between 60% and 80% (see e.g., [13, 14, 15]) with some reaching 90% [16]. Our methods reach an accuracy of more than 97%, and AUC values of more than 0.99. However, the results are limited to our game and data set, meaning that other applications may yield varying results.

Many inquiries employ a selected user base with their models being limited to distinct user groups. This increases the difficulty for prediction, because one-time users are not considered. We deliberately include this user type to produce a classification model that has a high practical applicability. Blue Byte provided us with a large, real-world data set. Usually, the industry is confidential and companies are unlikely to hand out their data to researchers [17]. It should be noted that most data sets used in publications in this domain are proprietary, and thus it is usually not possible for third parties to validate the created classifiers. While this holds true for our work, Section VIII includes a validation using an independent data set, provided by Blue Byte only after the first part of the study was finished, and which confirms the robustness of the recommended models.

While our approach and our experiences will be useful for other games as well, we are aware that as games are different, the reasons for churn are also different. We target a rather complex and slow strategy game with little social interaction compared to massively multiplayer online role-playing games (MMORPG), or multiplayer battle arena games (MOBA). Hence, malicious behaviors that appear in other games and may lead to churn such as real money trade [18], fully automated bots [19] or cyberbullying [20] are very uncommon for our game according to the publisher. We thus presume that the main reasons for churning in these games are either lack of time, money, or interest.

The work is structured as follows: We start by explaining TSO and the associated challenge of predicting churn in TSO. Then, the theoretical background is established in section III. It focuses on churn and disengagement, including an overview of important concepts within the game analytics domain and the current state of the art regarding ML techniques and class labeling approaches. Section IV summarizes related work, whereas in section V, we describe the context of the case study and how the data is pre-processed to get the final set of features. Section VI depicts our experimental setup including the selected class labeling approach and the applied methods to train and validate the models, and to evaluate and visualize the results. Section VII presents the results of the analysis for each of the ML techniques and labeling approaches. These are further discussed in section VIII, followed by conclusions, which include recommendations for the different steps of applying churn prediction to other freemium games.

Fig. 1. The Settlers Online (TSO). A strategy freemium game by Blue Byte.

II. The Settlers Online

The Settlers Online (TSO) is a strategy game developed by Blue Byte that implements the freemium payment model. Figure 1 shows a screen shot of the game. Players build towns consisting of mines, production facilities, and military buildings, and gain experience by performing tasks such as defeating non-player enemies, going on adventures and trading with friends. It is possible to create various units, e.g., workers or soldiers. While the game progresses, more buildings and units can be obtained. However, creating such resources takes a certain amount of time, which can be reduced through so-called buffs that players can apply to their own buildings or the buildings of friends. To obtain an advantage over others, players may choose to spend real-world money on an in-game, premium currency that can be used to buy resources or special items. An economy overview enables users to review the performance of their production facilities in order to optimize them.

III. Measuring Churn and Disengagement

Generally speaking, the term *churn* describes the defection of a customer from a service provider [8]. In the context of this paper, churn refers to players having stopped playing TSO. The detection of churn depends on the definition of churn that is employed, and thus different *labeling approaches* for churn yield different results. In the following, we present two different labeling approaches for churn using our ternary definition adapted from Hadiji et al. [21] and Runge et al. [22], as well as two approaches based on churn being defined as maximal disengagement [23, 24].

A. Naive Approach

The first approach is based on a subdivision of the data set into two partitions of roughly equal size, which are created by choosing a specific timestamp as a cutoff point. This timestamp may either be an arbitrary date, or represent a particular event with high relevance to the focal game, such as the release of a new content update. Based on these partitions, we distinguish three types of players:

- *Churners*: Users with actions in the first partition of the data set, but not the second one.
- *Beginners*: Users with actions in only the second partition of the data set that have not progressed beyond a certain point of the game.
- *Non-churners*: Users with actions in (at least) the second partition of the data set that are not labeled as beginners.

The distinction of beginners is intended to improve class balancing by filtering them out from the class of non-churners, thereby preventing the latter from becoming too dominant.

The naive labeling approach serves as a reference for more sophisticated approaches and presents a number of drawbacks. First, applying it to a specific data set can be difficult if there is no clear timestamp for its subdivision. Second, the exclusion of beginners from the set of non-churners may prevent gaining certain insights. Third, the method of subdivision may create a situation in which large amounts of data exist for certain users, and very small amounts for others who only recently started playing but cannot be seen as beginners anymore, which complicates classification. Fourth, users with strongly declining activity or long phases of absence may still be classified as non-churners if they occur at least once in the second partition of the data set. Lastly, training only on the first partition by cutting away the second and more recent partition means that the user behavior that shall be represented may have changed in the meantime.

B. Sliding Windows Approach

The second approach is based on sliding windows and is frequently employed in academic literature [13, 21, 16, 14, 22]. Similarly to the naive approach, it consists of splitting the data set into two non-overlapping parts by defining two windows, the *training window* and the *labeling window* (see Figure 2). Consequently, a user is considered a churner if he appears in the former, but not the latter, a non-churner if he appears in both, and a beginner if he only occurs in the latter.

To define both windows, three dates must be selected. First, an *end date* for the analysis must be chosen either deliberately or randomly. Counting back n days from this date yields the so-called *training date*, and counting back another m days from the latter then yields the *starting date*. Naturally, the training window extends from the starting date to the training date, and the labeling window from the training date to the end date. Since m and n do not need to be of equal size, both windows may have different sizes to influence the number of days a user must be inactive before being considered a churner, to balance the recentness of analysis and the amount of data used for prediction, and in order to counter class imbalance [21].

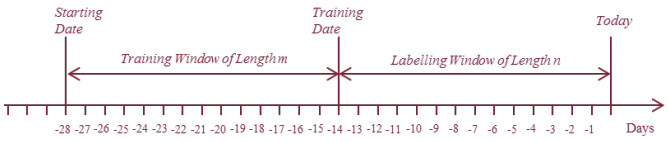

Fig. 2. Users playing between starting date and training date are churners if they do not occur between training date and today as well. Otherwise they are non-churners.

For cross-validation, multiple cutoff points are selected from the available data set, thus the windows *slide* through the data set. While this approach does not depend on the selection of a single meaningful cutoff point as the naive approach, determining appropriate values for m and n is a challenge. However, this allows individual tuning for various games and class distributions.

C. Disengagement

In contrast to churn, disengagement is a condition that is characterized by a significant reduction of player activity in a game [24]. Thus, a player who displays the maximal disengagement is seen as churned, although not all disengaged players are churners (yet). In the following, two different labeling approaches are presented based on XIE ET AL. [24, 23].

1) Quartile Approach

This approach extends the sliding windows approach, but disregards beginners who only appear in the labeling window [24]. In both windows separately, players are ordered according to a measure of user activity, such as the number of in-game actions, playtime, or rounds played. Users are ranked per window according to the quartile they appear in. On this basis, players are considered *engaged* if their rank in the labeling window is at least as high as their rank in the training window, and *disengaged* otherwise, i.e., if their quartile has decreased.

2) Trend Over Varying Dates Approach

XIE ET AL. propose a second method for disengagement labeling [23], which forms a more balanced class distribution, as every sample in the data set is utilized, including beginners. Because this improves the classifier, we adopt their method for our work.

The essence of their labeling approach is the assumption, that at some point in time, every user will become disengaged and churn afterwards. Therefore, it is useful to have a look at the 50% most disengaged users, regardless of their absoute engagement.

As before with the quartile approach, a time frame of length $m + n$ days is selected and split into two respective sections. Only users occurring in both sections are considered in order to eliminate (most of) the beginners. This change to the original approach is made to prevent that beginners make up a large share of the 50% most disengaged users, i.e., they supersede the non-beginner disengaged users. Another change includes the generalization of the game rounds feature. While the original implementation uses rounds to measure disengagement, the present study transfers this to logins. A login is transferable to a broader spectrum of games than a

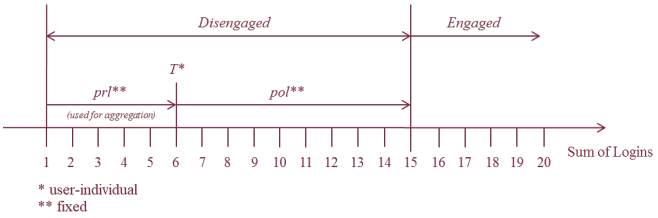

Fig. 3. Definition of prl and pol with an example of $prl = 6$ and $pol = 9$. The scale refers to the sum of logins of a user for a distinct time frame. The sum of prl and pol marks the boundary between disengagement and engagement. prl and pol are the same for all users (fixed). T is determined by prl and user-individual. Only user data until T are used to train a prediction model.

round. The number of logins is summed up over the whole time frame for each remaining user. Two parameters describe the user engagement: prior logins (prl) and post logins (pol). prl is a fixed parameter, which is game-dependent and found experimentally, that describes the number of logins before a splitting date T, while pol contains the number of logins after this date. T is individual for each user (hence the *varying dates*) and results from prl. The users are labeled as disengaging if they are not able to reach a number of logins that exceeds $prl + pol$. However, the data aggregation for training data only considers data until T. Thus, a classifier based on this data learns the specific behavior of users before the trend of disengagement starts. Figure 3 visualizes this labeling concept.

This approach requires a game to have a measure of activity similar to logins or game rounds. Depending on the genre, this can include measures like races, battles or competitions.

IV. Related Work

KAWALE ET AL. [2] use a diffusion model with social networks to identify how churn spreads across the player-base of a massively multiplayer online role-playing game (MMORPG). BORBORA ET AL. [25] perform a cluster analysis on subscription-based MMORPG to identify different types of churning users. Both studies make use of the number of sessions and session length. Several studies perform churn prediction on freemium games [21, 13, 22, 26, 7, 24] and retail box[2] games [27] by binary classification of players into churned or not churned. Except for the last two studies, they define churn as a player absence from the game for 7, 10, 14 or 28 days, or 13 weeks, respectively, while the last two define churn as a declining playtime trend over varying dates. In the studies, important features are the number of sessions, days played, score, last purchase, playtime, and the number of social relationships. The study by LEE ET AL. [7] first filters for loyal and valuable users by clustering the data before applying the classification model to increase prediction quality for the valuable users, which are underrepresented compared to casual players. Otherwise, these valuable users would be represented as outliers and not adequately predicted. Used models include simplified decision trees (as a heuristic for rapid churn prediction after the first session), logistic regression, support vector machines (SVM), random forests, neural networks, such as convolutional and deep neural networks (DNN), and long short-term memory (LSTM), gradient boosting and hidden markov models (HMM). Based on the AUC, the best models are logistic regression (0.99) and random forests (0.94)—respecting that different data sets are used.

[2]Retail box refers to games that only require a one-time fee to be played and can be purchased in retail stores.

TABLE I
RELATED CHURN PREDICTION STUDIES

Lit.	Ap.	Gametype	Churn Def.	Best Algorithm (AUC)
[2]	O	MMORPG	Subscription	
[25]	O	MMORPG	Subscription	
[21]	B	Freemium	7 Days	decision tree
[13]	B	Freemium	7 Days	
[24]	B	Freemium	Trend/time	logistic regression (0.99)
[22]	B	Freemium	2 Weeks	neural network (0.930)
[27]	B	Retail box	4 Weeks	HMM (0.77)
[26]	B	Freemium	10 Days	gradient boosting (0.824)
[7]	B	MMORPG	13 Weeks	gr. boosting, r. forest (0.94)
[28]	B,S	MMORPG	Trend/time	C4.5
[29]	B,S	MMORPG	5 Weeks	LSTM+DNN, trees classif.
[16]	S	Freemium	10 Days	mult. cond. inf. trees (0.96)
[30]	O	MOBA	7 Days	k-nearest neighbors

Beyond binary classification as churned or not churned, studies approach the problem as a survival analysis and predict churn as a function of time. These studies describe survival analyses on subscription-based MMORPG [28, 29] and freemium games [16], utilizing a churn definition of trend over varying dates, player absence for 5 weeks or 10 days (in the same order as cited). The studies identified relevant features to be player score, actions, time of actions, actions per day, sessions per day, last purchase, and days played per total days. They include a variety of methods such as LSTM and deep neural networks, extra-trees classifiers, logistic regression, SVM, ensemble of conditional inferences trees and naive Bayes. Based on AUC, the best model is the ensemble of conditional inference trees (0.96). While our experimental study at first incorporated survival analysis in terms of survival trees and survival ensembles, this paper does not include it any more as we found it difficult to handle and adapt to perform well on our data so that its performance was unsatisfactory.

CASTRO ET AL. [30] train a probabilistic classifier based on features derived from a frequency and time-frequency representation of the users' login times. They gained these features through a wavelet transformation of the login time series, an approach that generalizes the Fourier transformation and that is well known from signal processing. Their idiosyncratic approach reaches an AUC of 0.79.

Table I gives an overview of the studies and their approach (Lit.=literature, Ap.=problem approach, O=other, B=binary classification, S=survival analysis; Def.=definition), of which some do not report the AUC. In general, the described features are game-agnostic and related to sessions, playtime, score and purchases. The difference in AUC between the described models is small. Instead, the applied churn definition and the choice for training and labeling windows affect the performance of the prediction.

V. Feature Engineering

In the following, we describe the data set and the derived features that are later on used for training classification methods. Blue Byte provided us with user activity data of all users who had registered on or after 1st January 2016 in Czech Republic or the Netherlands for a 91-day period from 1 September until 30th November 2016. In total, the data set comprises 7,014 users and 90,356 observed events. Besides the event information (two tables), the data set includes session information (one table). All three tables include a game ID (the country), a user ID, a date and an average user level. The user ID is unique across countries. Blue Byte applies event-based tracking in TSO, which means that users cause the tracking by triggering events, or *actions*, (e.g., a click on a button) [10]. The independent data set includes activity data from 1st December 2016 until 28th February 2017 with 7,439 users and 113,643 events. The sample log format is a table with the columns: games ID, user ID, date, item type, item type amount, item amount.

A. Selection and Aggregation

A merged table of 1040 columns is considered as basis for the features. The features are separated into 1034 *action features* (usually game-specific), e.g., added or removed item types and item amounts, and six *non-action features* (usually not game-specific). We compute more features from the existing non-action features. For example, the inactive days of each user are determined from the days, on which a user does not play. The sum of all actions of a user is calculated per day. Although this depends on the game actions of a user, it is not seen as an action feature here, because it does not refer to a single action. These features represent the data set on user-day aggregation level, which consists of 1042 features.

Aggregation. Since most analyses do not require user data on the day level, the data is further aggregated depending on a chosen time-frame. This time frame is determined by the chosen class labeling approach. During aggregation, we compute more features from the descriptive statistics of the data set. For premium currency, logins, actions sum, and inactive days, the statistical features mean, sum, maximum value, standard deviation, and correlation on time (cf. [31]) are calculated. For the computation of the latter, the dates are type-casted into epochs since 1st January 1970.

Complete data set. Following Runge et al. [22], we add the last consecutive sequence of inactive days as the new feature missed days. We also use the inactive days to calculate the number and average length of day streaks. A day streak is defined as the number of consecutive days a user played the game, with the minimal case being a day streak of length one. For the average length, all day streaks of a user are considered and the mean of their lengths is computed. Combining games ID, user ID, the features added by aggregation (24) and the three additionally added features generates a group of 29 non-action features and a combined data set with overall 1063 columns.

B. Reduction

We only consider action features for reduction, because they constitute a major part of the feature set. At first, features with a standard deviation of zero in the non-aggregated feature set are removed, which reduces the data set from 1034 to 749 columns. In a next step, a correlation matrix is built and features with a correlation of at least 0.7 are removed. This lowers the number of relevant columns to 475. We further reduce the number of features using random ferns[3] on our labeled data. Random ferns are non-hierarchical structures that contain a small set of binary tests to determine the probability of the label assignments [32]. We set the number of ferns to the default of 1000, which is assumed to be a comparably robust value [33]. The depth is raised from the default value of six to eight, because this value offers a good trade-off between accuracy and speed.

We apply cross-validation and implement test and training sets for our feature reduction approach to overcome bias in the accuracy of the error prediction [34]. The intersection of ten iterations builds the feature set for each data set. For the final feature set, the intersection of the reduced sets over all labeling approaches is selected, which consists of 25 action features. Combined with the 29 non-action features, the data set is composed of 54 features. The two identifiers games ID and user ID, however, are excluded from the experimental analysis, meaning that the algorithms can not match data from different folds for the same user.[4]

VI. Approach and Experimental Analysis

As stated in the introduction, we do a "combinatorial" exploration of classification methods and labeling methods with parameter testing in order to see what works best. Our basic approach is therefore a performance comparison.

In our experimental analysis, we compare the following classification methods: decision trees, random forests, gradient boosting trees, support vector machines, naive Bayes, logistic regression, neural networks, and k-nearest neighbors (KNN). We identify the class labeling approach as well as the machine learning algorithm that show the best performance on the given data and we give recommendations based on our findings. In order to achieve this, we determine parameters for the labeling approaches (e.g., window sizes) that best suit our data (e.g., in terms of class balancing), apply training as well as validation, configure and apply the above mentioned classification methods and discuss their results.

Time frames: Training window and labeling window. All labeling approaches, except the naive approach, depend on adaptable time frames specified by the number of days they span. Especially the sliding windows approach depends on a proper adjustment of its time frames, which comprise the training window with length m and the labeling window with length n. The determination of a suitable training window length is a trade-off between several risks. If the training window is too short, the prediction accuracy suffers. If the

[3] We use the R package `Boruta`.

[4] Full and reduced feature sets are available at https://github.com/johuellm/churn-prediction-settlers/.

TABLE II
MOST IMPORTANT PARAMETER VALUES OF THE DIFFERENT ALGORITHMS, FOUND BY MANUAL TUNING

Method	Parameter	Value
Decision Trees (small)	Leaf nodes	6
Decision Trees (large)	Leaf nodes	17
Random Forests	Number of trees	500
Gradient Boosting Trees	Number of trees	20
SVM	Gamma	1/54
Single Layer Neural Network	No. of hidden nodes	12
Multi Layer Neural Network	No. of hidden nodes	28;12

training window is too long, the algorithm might be biased by data recorded long ago. The length of the training window influences the proportions of the classes within the data set as well. A short length leads to the dominance of non-churners, while the number of churners increases with the length. Thus, the selection of m influences class balancing.

If the labeling window is too short, users who at first behaved as if they were churners, but return to the game later would be falsely labeled as churners. If the labeling window is too long, the training data can be very old and the user behavior might have changed in the meantime. The length of the labeling window influences class balancing. A long window causes a low amount of churners, while a short window causes a high amount. In order to find a trade-off between the mentioned risks, to prevent unnecessary complexity in the labeling approaches, and to obtain feature values in a similar range for both windows, m is set to the same value as n in this study.

Summarizing, the selection of a proper value for m and n is a trade-off between the two mentioned risks under consideration of class balancing. In order to determine m and n for the data set at hand, we apply multiple statistics. The statistics are based on the work of DING ET AL. [35], HADIJI ET AL. [21] and RUNGE ET AL. [22] and include the total number of active players per month, absolute and relative frequencies of the inactive days, the maximum length of consecutive inactive days per user as well as the maximum frequency of logins per user and day. Concluding from the statistics of the data set, we set m and n to 14. The analyses showed that this value includes more than 90% of the player breaks (consisting of several days without login) and about 60% of the users, comprising the users who return exactly after a fortnight. We see 14 days as an appropriate trade-off between the risk of falsely labeling a user and the risk of changing user behavior in the meantime. Additionally, first experiments with data sets based on this value showed an even balance of churners and non-churners.

Class labeling. The middle of the overall time frame, i.e., the 15th October is used as split date for the naive approach. For TSO, the users having an average level below twelve are considered as beginners, while the rest is assigned to the group of non-churners. This is due to the reason that the tutorial ends at level twelve, which is likely to be a point of churning according to Blue Byte. Users who return after at least six weeks (this equals the length of the first partition) of not playing the game and who are not within the tutorial anymore, are not considered as beginners. All further labeling approaches require a total time frame of $n + m = 28$ days. The measure we apply for the quartile approach is the sum of the actions, because this feature describes the total activity of a user in TSO. In the trend over varying dates approach, the change to use logins instead of rounds simplifies the search for optimal values of the parameters. XIE ET AL. [23] apply an evolutionary algorithm for parameter search. In contrast to this, the present study uses a search over the complete search space with constraints, because the low absolute number of logins makes it computationally feasible.

Training and Validation. For each labeling approach a data set is generated. We apply K-fold cross-validation with $K = 10$, which is a recommended value and serves as a good compromise between variance and bias in most situations [36]. The creation of folds for the naive data set is conducted by an algorithm that belongs to the R package `caret` and that considers the class balance of each fold (cf. [37]). Since the labeling approaches of the remaining data sets contain a short-termed time frame, the caret algorithm is not suited. Instead, the creation of folds considers multiple training dates. Thus, ten iterations which are created based on randomly picked dates are taken as fold, respectively. The exact dates are (day followed by month): 05/10 20/10 18/10 09/11 10/10 05/11 30/11 29/10 30/09 07/10. These stand for the end of each used time frame. The subtraction of m days (which equals 14, in this case) from a date yields the beginning. The resulting models of the study are validated on an independent test data set provided by Blue Byte with 23,000 observations that was not available at model creation time. The test data set starts one month after the end of the training data set and there is no overlap of windows. The trained models are taken as-is without further training and directly applied and measured.

Method Parameters The most important parameter values of all used methods are displayed in table II. We determine the parameters by performing manual tuning, which varies across methods as well as data sets and is informed by related works.

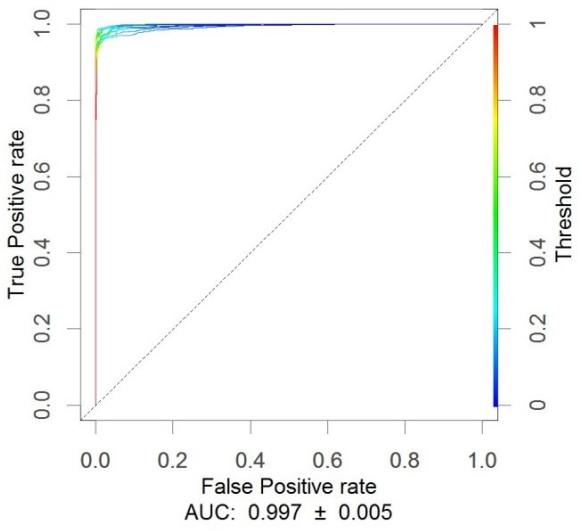

Fig. 4. ROC plot of a random forest for the sliding windows data set

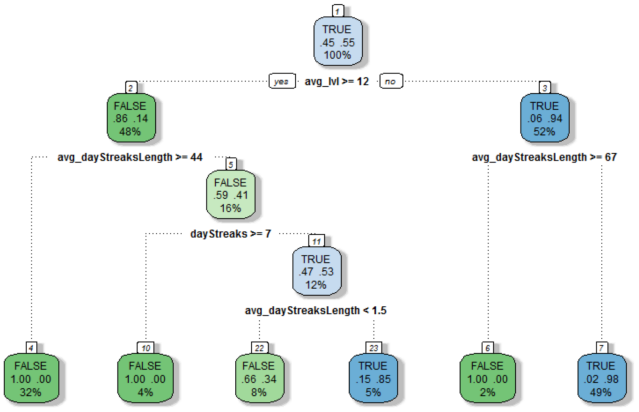

Fig. 5. Exemplary Decision Tree for the Naive Data Set. Blue nodes contain more churners than non-churners and green nodes vice versa. The numbers within a node consecutively mean: non-churning probability, churning-probability and share in the data.

Parametrizing k-nearest neighbors, we choose $k = 499$ for naive and sliding windows approach, $k = 100$ for the quartile approach and $k = 300$ for trend over varying dates approach. The kernel for the SVM is the euclidian distance function with $\gamma = 1/54$. The number of input nodes of the neural networks as well as the table width for all other methods is 54, the number of outputs is always 1 for a binary decision. Whenever possible (e.g. for the single layer neural networks, or k-nearest neighbors) the decision on a specific parameter value (number of neurons in this case, or k) is determined by performing a grid search or exhaustive search over a wide interval of possible values. The leading principles were that all methods shall be in the range of decent performance and the amount of effort for tuning the methods shall be comparable.

VII. Results

The performance of the different algorithms can be compared in table III, and the validation on the independent data set for the best performing algorithms is shown in table IV. In the following, we summarize the results of the

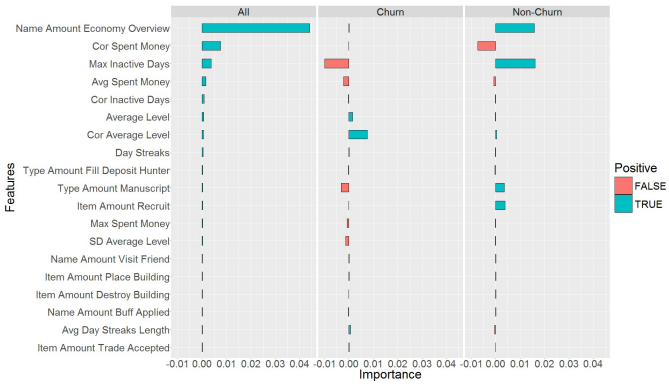

Fig. 6. VIMP (variable importance) for random forest features with positive values regarding overall importance. The three panels depict overall importance (left-hand side), importance for class churn (middle) and importance for class non-churn (right-hand side). Cor means correlation with time.

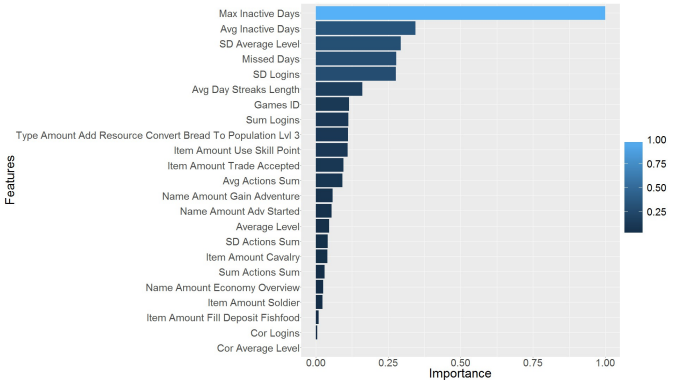

Fig. 7. Most important features for single-layer neural network on sliding windows data set with positive effect on churning. The importance values are relative and range from -1 to 1. Only positive values are depicted here.

different algorithms and labeling approaches, discussing their implications in the subsequent section VIII.

Classification Algorithms. The random forest models are among the two best performing algorithms for our data sets and provide accuracy values of 97%. Figure 4 shows the ROC curve on the sliding windows data set. Large decision trees are mostly on par with the results of random forests. Single-layer and multi-layer neural networks provide similar performance for churn prediction. However, in their current form they do not seem to be well-suited for disengagement prediction.

Small decision trees and gradient boosting trees provide accuracy values right below those of large decision trees. SVM and logistic regression models show mixed results for churn prediction and deliver low performance values for disengagement prediction. They do not reach one of the top two results in either of the two predictions. Naive Bayes and KNN belong to the worst performing algorithms in our study. While they still provide fair results for churn prediction, most of their disengagement prediction results are poor.

Labeling Approaches. From a labeling perspective, the applied algorithms perform best on the naive approach. Examining the best classification models for each labeling approach shows the best results for the sliding windows labeling approach, followed by the quartile labeling and the trend over varying dates with the worst results. The sliding windows labeling approach enables a prediction of whether a user churns within the next 14 days with a very high accuracy.

Feature Importance. It would require too much space to provide an overview for every method, and depending on the method this cannot always be detected easily. However, we provide feature importance information for decision trees, random forests, and neural networks. A sample decision tree is depicted in Figure 5, and the most important features according to impurity reduction were (in this order): day streaks, missed days, max inactive days, sum inactive days, max average level, avg inactive days, sum logins. Figure 6 shows the most important features of the random forest. Remarkably, "name amount economy overview", a feature that describes the number of times the economy overview was opened, clearly works as non-churn marker. In Figure 7, the most important features of the neural network are displayed,

TABLE III
DIFFERENT MACHINE LEARNING ALGORITHMS COMPARED OVER THE DIFFERENT APPROACHES

	Naive	Two Windows	Quartile	Trend Over Varying Dates
Decision Trees (small)	0.991 ± 0.006	0.977 ± 0.011	0.836 ± 0.060	0.676 ± 0.065
Decision Trees (large)	0.992 ± 0.004	**0.987 ± 0.005**	**0.852 ± 0.036**	**0.710 ± 0.061**
Random Forests	**1.000 ± 0.000**	**0.997 ± 0.005**	**0.888 ± 0.031**	**0.700 ± 0.053**
Gradient Boosting Trees	0.990 ± 0.000	0.984 ± 0.005	0.805 ± 0.037	0.694 ± 0.061
SVMs	0.990 ± 0.005	0.837 ± 0.076	0.687 ± 0.065	0.533 ± 0.028
Naive Bayes	0.887 ± 0.019	0.792 ± 0.017	0.649 ± 0.038	0.492 ± 0.019
Logistic Regression	0.967 ± 0.056	0.910 ± 0.023	0.676 ± 0.037	0.546 ± 0.034
Neural Networks (Single-Layer)	0.983 ± 0.012	**0.987 ± 0.005**	0.810 ± 0.037	0.583 ± 0.036
Neural Networks (Multi-Layer)	**0.994 ± 0.005**	0.980 ± 0.008	0.818 ± 0.041	0.584 ± 0.034
K-Nearest Neighbors	0.810 ± 0.000	0.840 ± 0.000	0.720 ± 0.000	0.520 ± 0.000

TABLE IV
VALIDATION RESULTS OF THE BEST PERFORMING METHODS ON THE SECOND DATA SET USING THE ALREADY TRAINED MODELS OF TABLE III

	Naive	Two Windows	Quartile	Trend Over Varying Dates
Decision Trees (small)	0.948 ± 0.008	0.972 ± 0.009	0.787 ± 0.025	0.711 ± 0.050
Decision Trees (large)	0.974 ± 0.007	0.984 ± 0.008	0.804 ± 0.024	0.724 ± 0.046
Random Forests	0.980 ± 0.000	0.999 ± 0.003	0.842 ± 0.036	0.705 ± 0.046
Neural Networks (Single-Layer)	0.962 ± 0.018	0.977 ± 0.005	0.747 ± 0.032	0.515 ± 0.015
Neural Networks (Multi-Layer)	0.975 ± 0.005	0.972 ± 0.008	0.701 ± 0.021	0.519 ± 0.017

being quite similar to the features selected by the decision trees.

VIII. DISCUSSION

We discuss the advantages and disadvantages of the methods, the labeling approaches, and the used metrics and features, resulting in suggestions on which methods to use and which ones to avoid, and how to attempt further improvement. In order to strengthen the obtained evidence, we have performed a second test by using the trained models on new test data that was not available at the time of the first experiment. As reported, this validation experiment largely supports the findings of the first experiment.

Classification Algorithms. We conclude that random forests are the best choice for building predictors on the data set at hand. The validation experiment yields similar values, and confirms their robustness, performance, and—partially almost perfect—prediction quality. This is an important finding, as random forests for churn prediction perform better than those for disengagement prediction. The random forest models may offer unused potential in the number of trees or the applied impurity measure. In case of issues regarding the time and resources needed for training in a productive scenario, large decision trees can serve as an excellent alternative, in particular due to their comparably short training time. During independent tests, large decision trees prove robust and do not reveal signs of overfitting. Single-layer and multi-layer neural networks provide excellent results. However, the validation on an independent test set reveals a comparably high loss in performance for several networks. Even though in absolute values the deviation is small (meaning this is not a clear indicator for overfitting) it can be stated that certain neural networks depend on the training on current data, and thus have a comparably strong tendency for model staleness. Neural networks can be subject to further research, because their wide range of configurations offers untapped potential. We do not recommend to pursue the algorithms of gradient boosting trees, SVM, naive Bayes, logistic regression or KNN due to their low performance.

Labeling Approaches. The naive labeling approach yields very good results. However, drawbacks such as the long time frame bear the threat that several users might not be labeled correctly. While the approach serves well as a reference to demonstrate possible prediction quality and to set the goals for the labeling, we do not recommend it in practice. Instead, we recommend the sliding windows approach, because it provides similar results without the same drawbacks. This approach is designed with practical applicability in mind and produces valuable and actionable results for a production environment. It enables the flexible adaption of the considered time frames for input as well as prediction, which implies straightforward transferability to different games.

The prediction of disengagement presents a harder task for the classification algorithms than predicting churn. A specificity of the quartile labeling approach, which we noticed during the case study, is that in case users do not play in the second time frame, they have a value of zero total activities in that time frame. If this holds true for more than 25% of the users, this group occupies the lower quarters and can switch them randomly. For example—in an extreme case—if users are assigned to the bottom quarter within the first time frame and do not play afterwards, they can rise to the quarter above within the second time frame. This is due to the random tie-breaking, i.e., users with equal values are assigned to a quarter randomly. In other words, these users are labeled as engaged although they show disengaging behavior. Increasing the number of segments (quantiles) the users are assigned to may enhance this labeling approach. In combination with improving the random forest model, this can boost prediction quality. We recommend to disregard the trend over varying dates data labeling. Although we assume that further algorithm and labeling approach improvement is possible, it is hard to

achieve good or even excellent performance. However, the general idea can be transferred to the creation of new labeling approaches. For example, it should boost the performance to base the disengagement definition on the total sum of user activities instead of logins. The increased effort for finding the user-individual parameters due to the vast number range of the sum of activities can be encountered by a well-configured evolutionary algorithm. For all labeling approaches, a more detailed subdivision into more than two classes may be meaningful. For example, the finding that new users behave significantly different from long-term users implies that the application of separate classes for these groups can benefit the prediction.

Metrics and Features. Due to the size of the analysis data set in terms of features (1063), their selection is a crucial step in pre-processing. We recommend the reduction to a group size roughly in the two-digit area (54 in our case) as it simplifies processing with regard to time and interpretability. The relevant features according to multiple importance measures are based on inactive days (see section VI). This makes sense, as the predicted behavior directly relates to the inactive days, i.e., after the point of churning users have at least 14 of them. Another approach can explore the feasibility of a classification algorithm, only relying on these features in order to handle large amount of data (e.g., greater time spans). Discriminating action features include various types of user movement in the game and the frequency of displaying the economy overview is one of the strongest indicators. The more often this window is opened, the smaller is the probability to churn. This implies that passionate users are interested in understanding and optimizing the overall economy of their settlers, while less motivated users disregard this. The number of collected pickups indicates motivation in the game. Less passionate users collect fewer pickups, while the number rises with a decreasing probability to churn. The amount of produced recruits is relevant, because these become available after level 15 and they are the first military unit, indicating game progress.

IX. Conclusion and Outlook

This exploratory case study of user churn prediction in a freemium online game overcomes the two challenges: to find a well suited ML technique as well as to determine an appropriate class labeling approach. Eight different binary classification algorithms are compared and four different labeling approaches are developed. Each ML technique is evaluated on all the labeling approaches. The best model (achieved with random forests) reaches AUC values of more than 0.99 in our game and data set. This enables an accuracy of more than 97% in predicting whether a user returns to the game within the next 14 days. This high predictive power has usually not been reached in the game analytics research domain. Nevertheless, the application of our proposed model to other data sets requires careful consideration of the parametrization, and may yield varying results, in particular for different game types.

Decision trees and neural networks perform excellent, while the remaining ML algorithms perform considerably worse. The simplest labeling approach yields the highest quality scores. However, we recommend a slightly more complex approach using sliding windows as it returns only marginally worse results and increases practical value by an enhanced applicability and flexibility. The results for the disengagement labeling approaches are less robust than those for the churn labeling. A selection of models, including the best-performing ones, are tested on an independent data set. This test data set starts one month after the training data and does not overlap with the training data. Its analysis shows the robustness of our models concerning overfitting and model staleness. None of the models reveals a drastic quality loss on the independent test data. However, neural networks show indications of model staleness. A detailed analysis of feature importance for the best model on the recommended labeling approach produces deeper insights of the relevant data structures. Features based on the inactive days of users are very important. The countries of origin do not play into the prediction—with the assumption that this is valid for Central Europe. We have not detected big differences between results for players from the Netherlands or Czech Republic.

Our study provides novel aspects regarding the comprehensiveness of class labeling approaches and classification algorithms. The disengagement concept is a recent approach in game analytics with little literature coverage. The achieved predictive power in terms of AUC values and accuracy has hardly been reached previously in this domain. The results of the case study imply that the prediction models are valuable from a business perspective and transferable to other games as well as other software.

Our study serves as a starting point for further research in the domain of predictive game analytics. The next task for the methods used in this work should be to select distinct users for data analyses, e.g., to remove users with few data or to focus users contributing to the revenue (see e.g., [16, 22]). Another analysis approach might be to pursue the identification of possible clusters and to assign meaning to them [25]. These clusters can then be used as input for specialized models to predict other user aspects in game analytics, e.g., the amount and value of future purchases.

The selected time frames offer untapped potential. For example, Zimmermann [38] demonstrates that the data generated within the first hour of play time in a game is more insightful than later data in discriminating user types. Our methods can be used to build a model that predicts user behavior based on detailed data of the first hour of interaction. An analysis whether longer time frames benefit the quality of the models will prove useful, e.g., for the disengagement prediction.

The implementation of this case study can be further improved, for example in regard to the applied parameters and parameter search. With sufficient computational resources, the application of computationally expensive deep learning algorithms becomes feasible. A comprehensive grid search for the recommended classification algorithms enables further optimization of the model parameters. However, it bears the danger of overfitting the methods to a specific data set and point in time.

In order to apply churn prediction methods to a new problem, we recommend to do the following, based on our findings:

- start with applying decision trees and if they prove valuable, compare their performance to random forests,
- start with few game-agnostic features related to activity,
- carefully adjust the time frame to your game data by analyzing it thoroughly,
- apply n-fold cross validation with each fold stemming from different dates,
- choose the same time frame length for training and for test data,
- consider class balancing in your data set, and
- use an independent test set stemming from another time frame with the trained models for verification.

ACKNOWLEDGMENT

We would like to express our gratitude to our colleagues at Blue Byte, who generously provided us with the two data sets as well as background information and insights.

REFERENCES

[1] Tim Fields and Brandon Cotton. *Social game design: monetization methods and mechanics*. Boca Raton, FL, USA: CRC Press, 2012.

[2] Jaya Kawale, Aditya Pal, and Jaideep Srivastava. "Churn prediction in MMORPGs: A social influence based approach". In: *Proceedings of the 2009 International Conference on Computational Science and Engineering*. Vol. 4. IEEE. 2009, pp. 423–428.

[3] Jiyoung Woo, Hwa Jae Choi, and Huy Kang Kim. "An automatic and proactive identity theft detection model in MMORPGs". In: *Appl. Math* 6.1S (2012), 291S–302S.

[4] Kenneth B. Shores, Yilin He, Kristina L. Swanenburg, Robert Kraut, and John Riedl. "The identification of deviance and its impact on retention in a multiplayer game". In: *Proceedings of the 17th ACM conference on Computer supported cooperative work and social computing*. ACM. 2014, pp. 1356–1365.

[5] Haewoon Kwak, Jeremy Blackburn, and Seungyeop Han. "Exploring cyberbullying and other toxic behavior in team competition online games". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM. 2015, pp. 3739–3748.

[6] Kunwoo Park, Meeyoung Cha, Haewoon Kwak, and Kuan-Ta Chen. "Achievement and friends: Key factors of player retention vary across player levels in online multiplayer games". In: *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee. 2017, pp. 445–453.

[7] Eunjo Lee, Boram Kim, Sungwook Kang, Byungsoo Kang, Yoonjae Jang, and Huy Kang Kim. "Profit Optimizing Churn Prediction for Long-term Loyal Customer in Online games". In: *IEEE Transactions on Games* (2018).

[8] Ali Tamaddoni, Stanislav Stakhovych, and Michael Ewing. "Comparing Churn Prediction Techniques and Assessing Their Performance: A Contingent Perspective". In: *Journal of Service Research* 19.2 (2016), pp. 123–141.

[9] Susan M. Keaveney and Madhavan Parthasarathy. "Customer switching behavior in online services: An exploratory study of the role of selected attitudinal, behavioral, and demographic factors". In: *Journal of the Academy of Marketing Science* 29.4 (2001), pp. 374–390.

[10] Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa. *Game Analytics*. London, UK: Springer, 2013.

[11] Ethem Alpaydin. *Introduction to Machine Learning*. 2nd ed. Cambridge, MA, USA: MIT Press, 2010.

[12] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.

[13] Anders Drachen, Eric T. Lundquist, Yungjen Kung, Pranav Rao, Rafet Sifa, Julian Runge, and Diego Klabjan. "Rapid Prediction of Player Retention in Free-to-Play Mobile Games". In: *Proceedings of the Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2016. arXiv: 1607.03202.

[14] Pierangelo Rothenbuehler, Julian Runge, Florent Garcin, and Boi Faltings. "Hidden Markov models for churn prediction". In: *Proceedings of the 2015 SAI Intelligent Systems Conference* (2015), pp. 723–730.

[15] Rafet Sifa, Sridev Srikanth, Anders Drachen, Cesar Ojeda, and Christian Bauckhage. "Predicting Retention in Sandbox Games with Tensor Factorization-based Representation Learning". In: *Proceedings of the 2016 IEEE Conference on Computational Intelligence in Games*. October. 2016.

[16] Africa Perianez, Alain Saas, Anna Guitart, and Colin Magne. "Churn Prediction in Mobile Social Games: Towards a Complete Assessment Using Survival Ensembles". In: *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics*. IEEE, 2016, pp. 564–573.

[17] Günter Wallner, Simone Kriglstein, Florian Gnadlinger, Michael Heiml, and Jochen Kranzer. "Game User Telemetry in Practice: A Case Study". In: *Proceedings of the 11th Conference on Advances in Computer Entertainment Technology* (2014), pp. 1–4.

[18] Eunjo Lee, Jiyoung Woo, Hyoungshick Kim, and Huy Kang Kim. "No Silk Road for Online Gamers!: Using Social Network Analysis to Unveil Black Markets in Online Games". In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web*. International World Wide Web Conferences Steering Committee. 2018, pp. 1825–1834.

[19] Eunjo Lee, Jiyoung Woo, Hyoungshick Kim, Aziz Mohaisen, and Huy Kang Kim. "You are a Game Bot!: Uncovering Game Bots in MMORPGs via Self-similarity in the Wild." In: *NDSS*. 2016.

[20] Haewoon Kwak, Jeremy Blackburn, and Seungyeop Han. "Exploring cyberbullying and other toxic behavior in team competition online games". In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM. 2015, pp. 3739–3748.

[21] Fabian Hadiji, Rafet Sifa, Anders Drachen, Christian Thurau, Kristian Kersting, and Christian Bauckhage. "Predicting player churn in the wild". In: *Proceedings of the 2014 IEEE Conference on Computational Intelligence and Games*. IEEE. 2014.

[22] Julian Runge, Peng Gao, Florent Garcin, and Boi Faltings. "Churn prediction for high-value players in casual social games". In: *Proceedings of the 2014 IEEE Conference on Computational Intelligence and Games*. 2014.

[23] Hanting Xie, Sam Devlin, and Daniel Kudenko. "Predicting Disengagement in Free-To-Play Games with Highly Biased Data". In: *Proceedings of the Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*. 2016, pp. 143–150.

[24] Hanting Xie, Sam Devlin, Daniel Kudenko, and Peter Cowling. "Predicting player disengagement and first purchase with event-frequency based data representation". In: *Proceedings of the 2015 IEEE Conference on Computational Intelligence and Games* (2015), pp. 230–237.

[25] Zoheb Borbora and Jaideep Srivastava. "User behavior modelling approach for churn prediction in online games". In: *Proceedings of the 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing*. IEEE. 2012, pp. 51–60.

[26] Seungwook Kim, Daeyoung Choi, Eunjung Lee, and Wonjong Rhee. “Churn prediction of mobile and online casual games using play log data”. In: *PloS ONE* 12.7 (2017), e0180735.
[27] Marco Tamassia, William Raffe, Rafet Sifa, Anders Drachen, Fabio Zambetta, and Michael Hitchens. “Predicting player churn in destiny: A Hidden Markov models approach to predicting player departure in a major online game”. In: *Proceedings of the 2016 IEEE Conference on Computational Intelligence and Games*. IEEE. 2016, pp. 1–8.
[28] Luiz Bernardo Martins Kummer, Julio Cesar Nievola, and Emerson Cabrear Paraiso. “Applying Commitment to Churn and Remaining Players Lifetime Prediction”. In: *Proceedings of the 2018 IEEE Conference on Computational Intelligence and Games*. IEEE. 2018, pp. 1–8.
[29] Eunjo Lee et al. “Game Data Mining Competition on Churn Prediction and Survival Analysis using Commercial Game Log Data”. In: *IEEE Transactions on Games* (2018).
[30] E. G. Castro and M. G. Tsuzuki. “Churn prediction in online games using players’ login records: a frequency analysis approach”. In: *IEEE Transactions on Computational Intelligence and AI in Games* 7.03 (2015), pp. 255–265.
[31] Rafet Sifa, César Ojeda, and Christian Bauckhage. “User Churn Migration Analysis with DEDICOM”. In: *Proceedings of the 2015 ACM Conference on Recommender Systems*. 2015, pp. 321–324.
[32] Mustafa Özuysal, Michael Calonder, Vincent Lepetit, Pascal Fua, and Mustafa Oezuysal. “Fast Keypoint Recognition Using Random Ferns”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.3 (2009), pp. 448–461.
[33] Miron B. Kursa. *Package Boruta*. Tech. rep. 2017.
[34] Christophe Ambroise and Geoffrey J. McLachlan. “Selection bias in gene extraction on the basis of microarray gene-expression data.” In: *Proceedings of the National Academy of Sciences of the United States of America*. Vol. 99. 10. 2002, pp. 6562–6566.
[35] Jun Ding, Daqi Gao, and Xiaohong Chen. “Alone in the game: Dynamic spread of churn behavior in a large social network a longitudinal study in MMORPG”. In: *International Journal of Smart Home* 9.3 (2015), pp. 35–44.
[36] Ron Kohavi. “A study of crossvalidation and bootstrap for accuracy estimation and model selection”. In: *Proceedings of the International Joint Conference on Artificial Intelligence*. IEEE. 1995, pp. 338–345.
[37] Max Kuhn. *A Short Introduction to the caret Package*. Tech. rep. 2016.
[38] Thomas Zimmermann. “Software Analytics for Digital Games”. In: *Proceedings of the 2015 IEEE/ACM 4th International Workshop on Games and Software Engineering* (2015), pp. 1–2.

Karsten Rothmeier is a consultant in the Analytics & Information Management practice of Deloitte Consulting. He works hands-on at the nexus of Big Data Engineering, Data Science and Cloud in various industries.

Nicolas Pflanzl is a former research associate at the University of Münster. He obtained his PhD there in 2018 on “Gamification for Business Process Modeling”.

Joschka Hüllmann is a research associate at the University of Münster and part of the competence Center for Smarter Work. His research focuses on the analysis of digital traces from communication and collaboration systems to faciliate productive and sustainble work practices in organisations.

Mike Preuss is assistant professor at Leiden University, The Netherlands, and member of the ERCIS network. Previously, he was with the group of *Information Systems and Statistics* at the University of Münster, Germany. In 2013, he received his PhD at TU Dortmund University, Germany.

His research interests focus on the fields of game AI, of evolutionary algorithms for real-valued problems, especially niching methods, and on social media computing.