

# Game Data Mining Competition on Churn Prediction and Survival Analysis Using Commercial Game Log Data

Eunjo Lee<sup>1</sup>, Yoonjae Jang, Du-Mim Yoon, Jihoon Jeon, Seong-il Yang<sup>2</sup>, Sang-Kwang Lee, Dae-Wook Kim, Pei Pei Chen, Anna Guitart, Paul Bertens, África Perriáñez, Fabian Hadiji<sup>3</sup>, Marc Müller, Youngjun Joo<sup>4</sup>, Jiyeon Lee, Inchon Hwang, and Kyung-Joong Kim<sup>5</sup>

**Abstract**—Game companies avoid sharing their game data with external researchers. Only a few research groups have been granted limited access to game data so far. The reluctance of these companies to make data publicly available limits the wide use and development of data mining techniques and artificial intelligence research specific to the game industry. In this paper, we developed and implemented an international competition on game data mining using commercial game log data from one of the major game companies in South Korea: NCSoft. Our approach enabled researchers to develop and apply state-of-the-art data mining techniques to game log data by making the data open. For the competition, data were collected from *Blade & Soul*, an action role-playing game, from NCSoft. The data comprised approximately 100 GB of game logs from 10 000 players. The main aim of the competition was to predict whether a player would churn and when the player would churn during two periods between which the business model was

changed to a free-to-play model from a monthly subscription. The results of the competition revealed that highly ranked competitors used deep learning, tree boosting, and linear regression.

**Index Terms**—Churn prediction, competition, data mining, game log, machine learning, survival analysis.

## I. INTRODUCTION

GAME artificial intelligence (AI) competition platforms help researcher's access well-defined benchmarking problems to evaluate different algorithms, test new approaches, and educate students [1]. Since the early 2000s, considerable effort has focused on designing and running new game AI competitions using mathematical, board, video, and physical games [2]. Despite a few exceptions, most of the research has concentrated on building AI players to play challenging games such as *StarCraft* and simulated car racing and fighting games; these competitions commonly rank AI players based on the results of numerous game plays, using, for example, final scores and win ratios. Recently, there have been special competitions that target human-likeness [3], general game playing [4], and the learning ability of AI players. However, there have been few competitions on content creation [5], game player modeling, or game data mining.

There have been many attempts to analyze game players. Bartle proposed analyzing multiuser dungeon game players (MUDs) into four types in 1996 [13] and expanded the model to eight types [14]. Quantic Foundry proposed six gamer motivation models based on 12 motivation factors from 5-min surveys with 250 000 gamers [15]. Others have attempted to model game players based on data analysis [16]–[20]. In recent years, game data mining has become increasingly popular. Data mining techniques extract useful information from large databases and are widely adopted in practical data analysis [21], [22].

Game companies generate a large amount of game player data based on their actions, progress, and purchases. From such data, it is possible to model the users' patterns [6] and attain useful information, including in-game dynamics, a user's likelihood of churning, lifetime, and user clusters. For example, Lee *et al.* discovered a "real money trading" pattern from the MMORPG *Lineage* game [8] and detected Bots and socio-economy patterns from the MMORPG *Aion* game [7], [9]. Despite its potential benefits to the game industry, collaboration between academics

Manuscript received January 8, 2018; revised June 21, 2018 and September 18, 2018; accepted November 21, 2018. Date of publication December 20, 2018; date of current version September 13, 2019. This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Science, ICT and Future Planning under Grant 2017R1A2B4002164, in part by the Ministry of Culture, Sports and Tourism and Korea Creative Content Agency in the Culture Technology Research and Development Program 2018, and in part by the European Commission under Grant agreement 731900—ENVISAGE. (Corresponding author: Kyung-Joong Kim.)

E. Lee and Y. Jang are with NCSoft, Seongnam 13494, South Korea (e-mail: gimmesilver@ncsoft.com; yoonjaej@ncsoft.com).

D.-M. Yoon and J. Jeon are with the Computer Science and Engineering Department, Sejong University, Seoul 05006, South Korea (e-mail: krad@hanmir.com; rhd00513@gmail.com).

S.-i. Yang, S.-K. Lee, and D.-W. Kim are with the Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea (e-mail: siyang@etri.re.kr; sklee@etri.re.kr; doormie@etri.re.kr).

P. P. Chen, A. Guitart, P. Bertens, and Á. Perriáñez are with the Game Data Science Department, Yokozuna Data, Silicon Studio, Tokyo 150-0013, Japan (e-mail: peipei.chen@siliconstudio.co.jp; anna.guitart@siliconstudio.co.jp; paul.bertens@siliconstudio.co.jp; africa.perrianez@siliconstudio.co.jp).

F. Hadiji and M. Müller are with Goedle.io GmbH, Cologne 50672, Germany (e-mail: fabian@goedle.io; marc@goedle.io).

Y. Joo, J. Lee, and I. Hwang are with the Department of Computer Engineering, Yonsei University, Seoul 03722, South Korea (e-mail: chrisjoo12@gmail.com; jitamin21@gmail.com; ich0103@gmail.com).

K. J. Kim was with the Computer Science and Engineering Department, Sejong University, Seoul 05006, South Korea. He is now with the School of Integrated Technology, Gwangju Institute of Science and Technology (GIST), Gwangju 61005, South Korea (e-mail: kjkim@gist.ac.kr).

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the author.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TG.2018.2888863

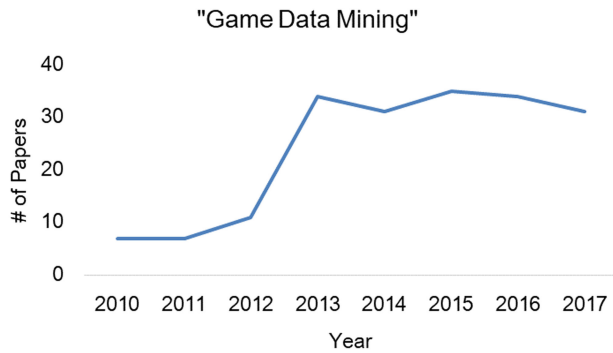


Fig. 1. Number of papers on game data mining in 2010–2017 (from Google Scholar using “game data mining” keywords).

and game companies to develop and apply advanced techniques to game log data has not flourished to the extent of AI game player development or content generation. Fig. 1 shows that the number of studies on game data mining plateaued after a sudden increase in 2013.

The purpose of our game data mining competition was to promote the research of game data mining by providing commercial game logs to the public. In coordination with NCSOFT, one of the largest game companies in South Korea, approximately 100 GB of game log data from Blade & Soul were made available.

We hosted the competition for five months from March 28, 2017, to August 25, 2017. During this period, we had 300 registrations on the competition’s Google Groups that were given access to the log data. Finally, we received 13 submissions for Track 1 (churn prediction) and 5 submissions for Track 2 (survival analysis). The participants predicted the future behavior of users by applying different sets of state-of-the-art techniques such as deep learning, ensemble tree classifiers, and logistic regression (LR).

The contributions of the game data mining competition are listed below.

- 1) The competition opened commercial game log data from an active game to the public for benchmarking purposes. After the competition, NCSOFT allowed users to copy and retain data for educational, scholarly, and research applications. In addition, labels for the test dataset were made publicly available.
- 2) Similar to the ImageNet competition, the competition provided a test server for participants to submit their intermediate performance and benchmark predictions using 10% of the test data.
- 3) The competition problems applied practical containments. For example, a much longer time span (e.g., three weeks) was given between the training data and prediction window, reflecting the minimum time required to develop and execute churn prevention strategies to retain potential churners. This enhanced the difficulty of the competition problem compared with the conventional time span of one to two weeks.
- 4) The competition was designed to incorporate concept drift, specifically, a change in the business model, to



Fig. 2. Screenshot from Blade & Soul.

measure the robustness of the participants’ models when applied to constantly evolving conditions. Consequently, the competition comprised two test datasets, each from different periods. Between the two periods, the aforementioned business model change took place. The final standings of entries were determined based on the harmonic average of final scores from both test datasets.

## II. COMPETITION PROBLEMS: CHALLENGES IN GAME DATA MINING COMPETITION

Blade & Soul, launched in June 2012, features a combination of epic martial-arts actions with highly customizable characters<sup>1</sup> (see Fig. 2). Along with the solo-play experience with numerous quests and dungeons, Blade & Soul incorporates party dungeons and quests, as well as group combination attacks cultivating true camaraderie among party members. Additionally, fighting in an intense player-versus-player arena mode has allowed the game to gain global popularity, as evidenced by the participation of hundreds of teams from nine regions in the most recent 2017 World Championship held in Seoul, South Korea.

In this paper, the competition consisted of two tracks. Participants could choose to participate in one or both of these tracks. Track 1 aimed to predict whether target users had churned, and Track 2 required participants to predict the survival time of target users. Both tracks used the same dataset consisting of one training set and two test sets. Each set comprised in-game activity logs of sampled players from different periods. To maximize the benefits of the churn prevention model, the competition problem was structured using data from an actively serviced game, Blade & Soul, and considered the following factors.

### A. Prediction Targets

To benefit the most from churn prediction and prevention, prediction targets should be those that provide the most profit if retained. Naturally, not all users provide game companies with the same profit; in fact, most users are casual game players accounting for a small proportion of sales, and there are even users who undermine the game services [23]. These light, casual, and malicious players were excluded from the scope of this competition, as our focus was on predicting the churn of

<sup>1</sup><http://www.bladeandsoul.com/en/>

TABLE I  
SUMMARY OF CHURN PREDICTION STUDY

Year	References	Games (Publisher)	Genre (Platform)	Payment System	Target Customers	Churn Definition (Inactivity Period)	Prediction Point
2017	GDMC 2017 Competition	Blade & Soul (NCSOFT)	MMORPG (PC)	Monthly Charge, Free-to-Play	Highly Loyal Users (cumulative purchase above the certain threshold)	five weeks	Churn after three weeks Survival Analysis
2017	Kim <i>et al.</i> , [12]	Dodge the Mud, Undisclosed, TagPro	Causal Game (Online/Mobile)	Free-to-Play	All gamers	ten days	Churn within 10 days
2016	Tamassia <i>et al.</i> , [26]	Destiny (Bungie)	MMOG (Online)	Package Price	Randomly sampled players (play time > 2 hours)	four weeks	Churn after four weeks
2016	Perianez <i>et al.</i> , [11]	Age of Ishtaria (Silicon Studio)	Social RPG (Mobile)	Free-to-Play	High value players (whales)	ten days	Survival Analysis
2014	J. Runge <i>et al.</i> , [10]	Diamond Dash and Monster World Flash (Wooga)	Social Game (Mobile)	Free-to-Play	High Value Player (top 10% of all paying players)	two weeks	Churn within the Week

loyal users only, namely, highly loyal users with a cumulative purchase above a certain threshold and many in-game activities. Given that highly loyal users seldom churn or churn on occasion due to external factors, we expected that the participants' churn prediction performance would not be comparable to prior churn prediction work. Table I shows that several previous works included all or most game player types in their churn prediction.

### B. Definition of Player Churn

Unlike telecommunication services in which user churn can be easily defined and identified by the user unsubscribing [24], such is not the case for online game services. Online game players seldom delete their accounts or unsubscribe, although they have no intention of resuming game play. In fact, according to our analysis, only 1% of the players inactive for over one year explicitly leave the service by deleting their accounts.

Consequently, we decided player churn using consecutive periods of inactivity. However, what length of inactivity should be considered as churn? This is a difficult question to answer, given that there are various reasons behind a player's inactivity. For example, some players may be inactive for several days because they only play on the weekends. Some may appear inactive for a few weeks because they went on a trip or because they had an important exam that month. If the period to decide player churn is too short, the misclassification rate will be high. If the period is too long, on the other hand, the misclassification rate would be lower but it would take longer to determine whether a player had churned or not; consequently, by the time a player is identified as a churning, there would not be sufficient time to persuade him or her to return.

To resolve this dilemma, we defined a churning as a user who does not play the game for more than five weeks. Fig. 3 shows the weekly and daily play patterns of players in concordance with their life patterns and weekly server maintenance every Wednesday morning. Thus, in our analysis, a week is defined from one Wednesday to the next Wednesday. Table I shows that previous works used ten days or two weeks for the decision of churn. However, in a recent work for the AAA title, Destiny used four weeks for the churn decision, similar to our definition.

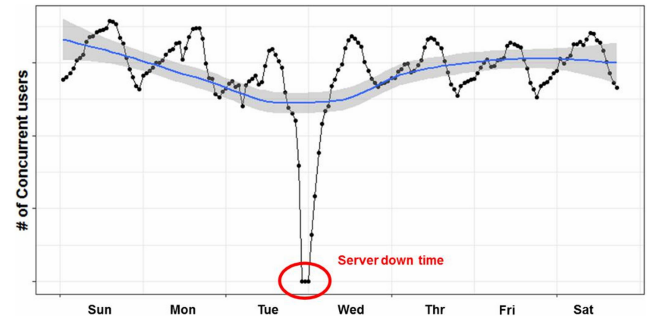


Fig. 3. Time series of concurrent users for one week. Daily and weekly cyclic patterns are shown.

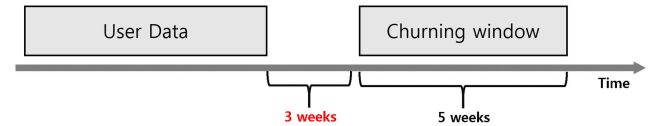


Fig. 4. User data and churning window were used to predict and determine churn, respectively.

### C. Prediction Point

It is likely that user behaviors leading up to the point of churn (e.g., deleting the game character) would play a vital role in churn prediction. However, predicting churn just before the point of leaving would be pointless, as at that point not much can be done to persuade users to stay loyal. Predictions should be made in a timely manner before the churning point so that a churn prevention strategy, namely, a new promotion campaign, content update, and so on, can be implemented to retain potential churners. If the prediction is made too early, the result will not be sufficiently accurate, whereas if the prediction is made too late, not much can be done to retain the user even if the prediction is correct. For our competition, making predictions three weeks ahead of the point of actual churn was deemed effective. Consequently, as shown in Fig. 4, we examined user data accounts up to three weeks before the initiation of the five-week churning window.



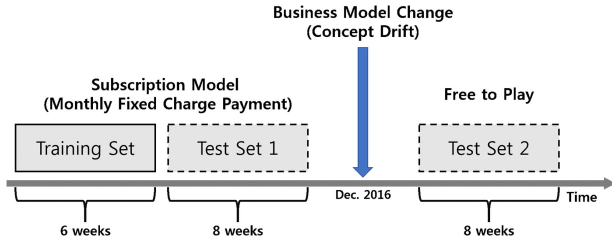


Fig. 5. Test sets 1 and 2 were constructed with data from different periods to reflect the business model change.

#### D. Survival Analysis

While churn prediction itself is well worthwhile, predicting the specific churn point would increase the value of the model, so this is the second focus of the competition. In addition to churn prediction, we asked participants to perform a survival analysis to predict the survival time. The survival time was then added to the labeled data for the training set. Survival time is defined as the period between the date of last activity from the provided data and the date of most recent activity, which was not provided but instead calculated when all predictions were submitted. Because we can only check for observable periods, this survival analysis is regarded as a censoring problem. We added a "+" sign after survival periods for right-censoring data to distinguish censoring data from churning data.

#### E. Concept Drift

Various operational issues arise when applying the prediction model to actively serviced and continuously evolving games; concept drift epitomizes how such change can affect a prediction model [25]. Compared to a model that had high performance during the modeling period but showed declining performance over time, a robust model that maintains performance, despite externalities, often proves to be more valuable considering update and maintenance costs, even if the prediction performance of the robust model during the modeling period was not as accurate. To encourage the participants to generate a model that is robust enough to withstand changing conditions over time, we created two test sets each with data from different periods to evaluate model performance. The first test set consisted of data from the period two months after the training data period, and the second consisted of data seven months after the training data period.

Between the first and second test set periods, there was a significant change in the business model of Blade & Soul, as its subscription-based model was changed to a free-to-play model in December 2016. The chronological positioning of training and test sets and the business model change are shown in Fig. 5. In an effort to encourage the participants to form a robust model addressing business model change, we placed no restrictions on using test sets (without data labels) for training.

### III. RUNNING GAME DATA MINING COMPETITION

#### A. Dataset Preparation

The competition consisted of two tracks. Three datasets were provided: a training set and two testing sets. Participants were

TABLE II  
GENERAL INFORMATION OF THE PROVIDED DATASET

Data Set	Time Period	Weeks	Number of Gamers	Data Size*
Training	APR-1-2016 ~ MAY-11-2016	6	4000 (30% churn)	48G (175m Events)
Test Set 1	JULY-27-2016 ~ SEP-21-2016	8	3000 (30% churn)	30G
Test Set 2	DEC-14-2016 ~ FEB-08-2017	8	3000 (30% churn)	30G

asked to perform churn prediction (Track 1) and survival time prediction (Track 2). Table II summarizes the basic dataset information.

For the competition, participants were given access to raw log data of actual Blade & Soul users that captured all of their in-game activities. Among hundreds of log types, 82 main log types, which captured information on connection, character, item, skills, quest, and guild, were used to create the competition dataset. Each log type consisted of 77 fields categorized into common, actor, object, and target fields, as described below.

- 1) Common fields captured information common to all log types such as log type, log creation time, and in-game location of the corresponding action.
- 2) Actor fields depicted information regarding the actor of the action with fields such as actor identification (ID), actor level, and actor race.
- 3) Object fields included information on the object of the action, ranging from item ID, item grade, and item quantity to skill name.
- 4) Target fields contained information regarding the target of the action, such as the target character ID, gold, and damage received by the target.

Each field held different information depending on the log type; detailed log schema and descriptions were provided to the participants, along with the data, through the competition website.<sup>2</sup>

As mentioned in the competition problem section, all predictions were made three weeks ahead of the five-week-long churning window, during which users' activities were observed to determine whether they had churned. User churn was determined strictly during the five-week churning window; activities during the three weeks of the "no data" period had no impact on determining user status. Despite activities over other periods, if a user did not exhibit activity during the churning window, the user was considered to have churned. Several examples of users' activities and corresponding churning decisions are shown in Fig. 6.

For the second track of the competition, which required participants to predict the survival time of users, any prediction of survival time longer than the actual observed time was considered to be correct given the censoring nature of the problem. For example, if a user's survival time was 103 days by the time submission date, and a participant predicted the survival time of the user to be 110 days, such a prediction was accepted as the correct prediction.

<sup>2</sup><https://cilab.sejong.ac.kr/gdmc2017/>

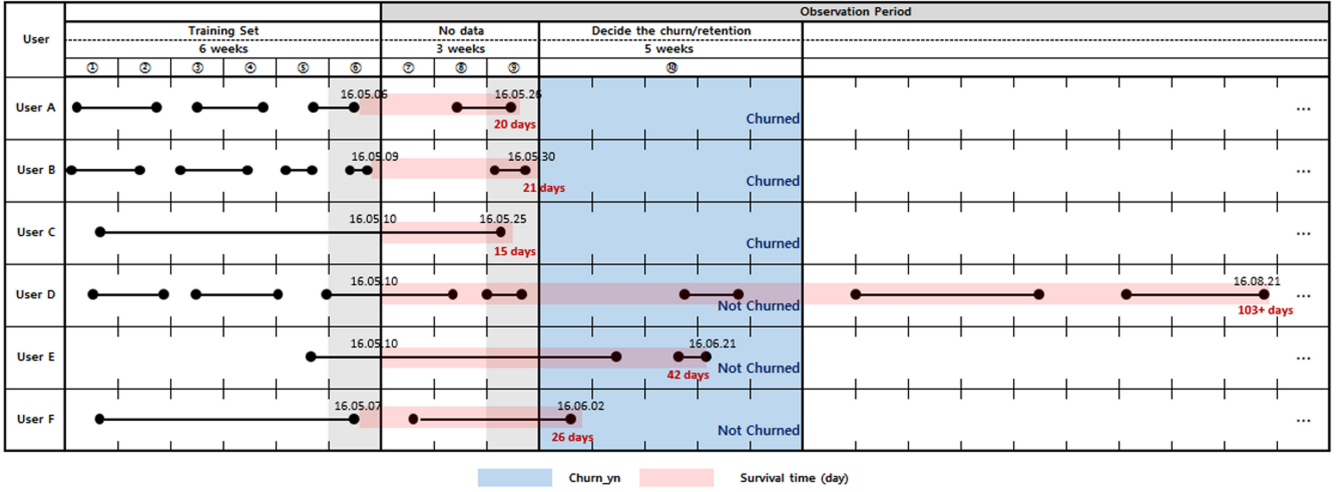


Fig. 6. User activities, churning decision, and survival period.

### B. Definition of Loyal Users

NCSOFT is constantly detecting malicious users who have an adverse effect on game services. The detection system consists of two steps: One is the step of restricting game service because of collecting concrete evidence for malicious activity, and the other step is the step of follow-up observation because it is a circumstantial suspicion. We excluded the users, who have circumstantial evidence for malicious activities, for the dataset. We note that the detailed rules for the detection cannot be disclosed here, since the information is confidential.

Besides, NCSOFT assigns all users into fourteen grades—the lower grade means a higher loyalty—for customer relationship management. Loyal grades are determined using k-means clustering and take info features such as payment amount, play time, and in-game contents usage rate.

Based on this information, we have selected only users who have been assigned at least grade nine more than once in the last three months. The proportion of the selected group is less than 30% of the total users.

### C. Performance Measure Definition

Participants' performances were measured using the average of the F1 score and root mean squared logarithmic error (RMSLE) on the two test sets for Track 1 and Track 2, respectively

$$F_1 = 2 \cdot \frac{1}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

$$\epsilon = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(p_i + 1) - \log(a_i + 1))^2} \quad (2)$$

where  $p_i$  and  $a_i$  are the predicted and actual values of the  $i$ th data in the test set, respectively.

### D. Participants

On April 15, 2017, sample data and the data schema, as well as competition details, were announced. The entire dataset was

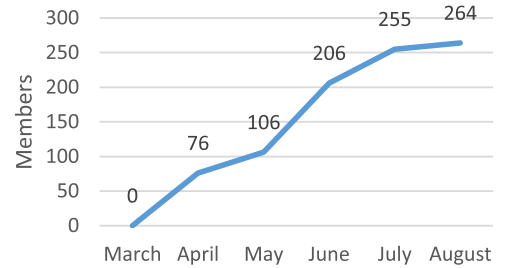


Fig. 7. Number of registrants in the Google Groups site.

made available on April 30, 2017, and the final submission date, originally July 31, 2017, was postponed to August 10, 2017. Starting on May 19, all participants were able to access a test server to validate their work using 10% of the test data.

For efficient communication with the participants, we opened a Google Groups page<sup>3</sup> in which participants were required to register for access to the log data. As shown in Fig. 7, the number of registrants who joined the Google Groups increased steadily every month from the end of March 2017 when the group was created. For the competition, we received 13 submissions from South Korea, Germany, Finland, and Japan for Track 1 and 5 submissions for Track 2. The details of all participants are summarized in the supplementary materials.<sup>4</sup>

## IV. COMPETITION RESULTS

### A. Track 1

This track aimed to predict the churn of the gamer. A total of 13 teams participated in this track. The Yokozuna Data team won, with a final total score of approximately 0.62. To facilitate an easy start and provide a reference point, we provided a tutorial with a baseline model on the competition page<sup>5</sup> In the tutorial, we used 22 simple count-based features and lasso regression.

<sup>3</sup><https://groups.google.com/d/forum/gdmc2017>

<sup>4</sup><https://arxiv.org/abs/1802.02301>

<sup>5</sup><https://cilab.sejong.ac.kr/gdmc2017/index.php/tutorial/>

An F1 score of 0.48 was achieved, indicating that the eight submissions among the 13 entries outperformed the baseline model.

The winning performance with an F1 score of 0.62 was similar to the predictability of models introduced by previous churn prediction work [11]. Considering that there were two constraints that greatly hindered prediction accuracy: targeting loyal users only and predicting based on data from specific time periods, the fact that the winning team showed comparable or even better results underscores the novelty of the work provided by the participants.

Interestingly, participants with lower ranks generally performed better on the first test set than on the second set, reflecting data after the change of business model from a monthly subscription model to a free-to-play model. On the other hand, those with higher ranks performed better on predicting user churn during the second set.

The final standing differed significantly from the ranking on the test server, as given in Table III. We believe such a difference in standing is due to the fact that the test server measured performance using only 10% of the test data, whereas the final performance measurement was conducted with the remaining 90% of the test data. Consequently, we suspect that many models that did not perform relatively well suffered from overfitting to the test server results. In contrast, the Yokozuna Data team who won the competition specifically attempted to avoid overfitting using features with the same distribution throughout a different test set. Given its superior performance, we believe that Yokozuna Data invented a robust model that was properly generalized.

### B. Track 2

Although predicting the survival time of users carries more benefits for game companies, it is more difficult to produce accurate prediction models compared with predicting a simple classification as Track 1. Consequently, only five of the teams that participated in Track 1 also participated in Track 2. Yokozuna Data won this track as well, with a total score of 0.72.

Unlike Track 1, performance was measured using RMSLE, in which a lower score is better. As with the results for Track 1, the test server results differed from those at the final outcome. The highest score from the test server was 0.41, whereas that of the final performance measure was around 0.72.

## V. METHODS USED BY PARTICIPANTS

### A. Overview

Throughout the competition, various methods were applied and tested on the game log data of Blade & Soul of NCSOFT. For Track 1, there was no overriding technique. All groups were evenly distributed in the rankings. This means that the combination of features, learning models, and feature selection is more important than what learning model is used to improve churn prediction accuracy.

Proper data preprocessing was highly important with regard to achieving high performance in the prediction. The winner,

TABLE III  
COMPETITION RESULTS (THE ARROWS INDICATE THE CHANGE OF RANKS FROM THE TEST SERVER)

(a) Track 1

Rank	Team		Precision	Recall	F1 score	Final score
1	<b>Yokozuna Data</b> (▲3)	Test1	<b>0.55</b>	<b>0.69</b>	<b>0.61</b>	<b>0.62</b>
		Test2	<b>0.54</b>	<b>0.76</b>	<b>0.63</b>	
2	UTU (▲1)	Test1	0.53	0.71	0.60	0.60
		Test2	0.60	0.60	0.60	
3	TripleS (▼1)	Test1	0.54	0.62	0.57	0.60
		Test2	0.56	0.71	0.62	
4	TheCowKing (▲1)	Test1	0.55	0.64	0.59	0.60
		Test2	0.56	0.67	0.60	
5	goedleio (▼4)	Test1	0.55	0.60	0.57	0.58
		Test2	0.58	0.62	0.60	
6	MNDS (▲2)	Test1	0.51	0.62	0.55	0.56
		Test2	0.51	0.62	0.56	
7	DTND	Test1	0.51	0.49	0.49	0.53
		Test2	0.50	0.72	0.58	
8	IISLABSKKU (▼2)	Test1	0.55	0.58	0.56	0.52
		Test2	0.72	0.37	0.48	
9	suya (▲1)	Test1	0.50	0.40	0.44	0.42
		Test2	0.38	0.44	0.40	
10	YK (▲2)	Test1	0.63	0.40	0.49	0.39
		Test2	0.64	0.22	0.33	
11	GoAlone	Test1	0.29	0.85	0.42	0.35
		Test2	0.31	0.31	0.31	
12	NoJam (▲1)	Test1	0.31	0.30	0.30	0.30
		Test2	0.31	0.31	0.31	
13	Lessang (▼4)	Test1	0.30	0.29	0.29	0.29
		Test2	0.29	0.29	0.29	

(b) Track 2

Rank	Team	Test1 score	Test2 score	Total score
1	<b>Yokozuna Data</b> (▲3)	<b>0.88</b>	<b>0.61</b>	<b>0.72</b>
2	IISLABSKKU (▲3)	1.03	0.67	0.81
3	UTU (▼2)	0.92	0.89	0.91
4	TripleS (▼1)	0.95	0.89	0.92
5	DTND (▼3)	1.03	0.93	0.97

Yokozuna Data used the most various features among participants. They used daily features, entire period features, time-weighted features, and statistical features. It is also interesting that Yokozuna Data used different algorithms for each of the test sets. In addition, deep learning appeared to be less popular than computer vision, speech analysis, and other engineering domains in the field of game data mining, as there were many participants who implemented tree-based ensemble classifiers [e.g., extra-tree classifiers and Random Forest (RF)] and LR. These prediction techniques were classified into three major groups: neural network, tree-based, and linear model.

For Track 2, no entries with deep learning techniques were submitted; all of the submissions used either tree regression or linear models. The winner, Yokozuna Data, combined 900 trees for the regression task.

Those who performed well for Track 1 also excelled in Track 2, as the rankings among Track 2 participants were the same as those of Track 1, except for team IISLABSKKU. Although the rankings for Track 2 were somewhat similar to those for Track 1, the same was not the case for the usage of techniques, as no one implemented a neural network model to solve the problem for Track 2. Even team Yokozuna Data, which showed impressive performance for Track 1 using a neural network, chose not to apply neural network techniques to Track 2.

Now, we describe each method submitted to the competition, beginning with Yokozuna Data, the winner of the competition.

### B. Yokozuna Data (Winner of Track 1 and Track 2)

Here are summarized methods of the Yokozuna Data feature engineering process, which is based on previous works [11], [27].

Yokozuna Data used two types of features: daily features and overall features. Daily features are the features which are aggregated the user's in-game activities by day, while overall features are calculated over the whole data period.

There were more than 3000 extracted features after the data preparation. Besides adopting all features, the following three feature selection methods were also tested: long short-term memory (LSTM), auto encoder, feature value distribution, and feature importance.

Multiple models were evaluated for both tracks. Additionally, different models were used for the two test datasets. The models that produced the best results—and that led us to win both tracks—were LSTM, extremely randomized trees and conditional inference trees. Parameters were adjusted through cross validation.

#### 1) Binary Churn Prediction (Track 1):

*a) LSTM:* A model combining an LSTM network with a deep neural network (DNN) was used for test set 1. First, a multilayer LSTM network was employed to process the time-series data and learn a single vector representation describing the time-series behavior. Then, this vector was merged with the output of a multilayer DNN that was trained on the static features. After merging, an additional layer was trained on top of the combined representation to get the final output layer predicting the binary result. In order to prevent overfitting, dropout [28] was used at every layer. Additionally, dropout was also applied to the input to perform random feature selection and further reduce overfitting. As there were many correlated features, selecting only a subset of the time-series and static features of the input prevents the model from depending too much on a single feature and improves generalization.

*b) Extremely randomized trees:* This technique provided better prediction results for the test set 2. After parameter tuning, 50 sample trees were selected and the minimum number of samples required to split an internal node was set to 50. Yokozuna Data used a splitting criterion based on the Gini impurity.

#### 2) Survival Time Analysis (Track 2):

*a) Conditional inference survival ensembles:* For both tests of the survival track, the best results were obtained with a censoring approach, using conditional inference survival en-

sembles. The final parameter selection was performed using 900 selected unbiased trees and subsampling without replacement. At each node of the trees, 30 input features were randomly selected from the total input sample. The stopping criteria employed was based on univariate p-values. For more details please check [11].

### C. UTU (2nd Place in Track 1 and 3rd Place in Track 2)

UTU used various features that represented activities such as availability, playing probability, message rate, session lengths, and entry time. These features were split into three features of overall, last week, and change measure. In addition, these activity features were extended by smart features of essentially previous activity (total experience, maximum experience, current experience, rating, money, etc.), whereby “smart” means that they reverse-engineered how the experience accumulation worked with regard to “previous activity” measurement.

For classification, UTU used simple LR, with some regularization. They did not attempt to use the cross-validated “optimal settings” for parameters because they were unable to find any differences. Moreover, because of a covariate shift, they concluded that it would not necessarily be an optimal setting. Feature selection based on the basic L1-norm seemed useful in test set 2.

For regression, UTU used ridge regression. To linearize the features for both submissions, they used a feature transformation based on quantiles.

### D. TripleS (3rd Place in Track 1 and 4th Place in Track 2)

TripleS used play time (referring to how long the user was connected to the game), level and mastery level, the sum of mastery experience, log count, and specific log ID count as features. All features were extracted on a weekly basis. In addition, they calculated the coefficient of variance and first-order and quadratic functions of the above-mentioned features and added them as features only for active players.

After preprocessing the features (normalizing and scaling), TripleS used an ensemble method, such as RF, for Tracks 1 and 2. Moreover, to obtain the best result, the parameters for RF were optimized. All of the results of Tracks 1 and 2 were validated by five-fold cross validation with a training dataset.

### E. IISLABSKKU (8th Place in Track 1 and 2nd Place in Track 2)

IISLABSKKU created 1639 features through feature engineering. They calculated the feature's importance through RF and used xgboost to calculate the final results for the 100 critical features.

### F. TheCowKing (4th Place in Track 1)

TheCowKing used log counts on a weekly basis. Log ID and actor level were used as features. The main learning model of TheCowKing was LightGBM.<sup>6</sup>

<sup>6</sup><https://github.com/Microsoft/LightGBM>



### G. Goedle.io (5th Place in Track 1)

1) *Feature Engineering*: Goedle.io transformed the events for each player into an event-based format. Additional information, such as an event identifier (e.g., to differentiate a kill of a PC versus NPC) or an event value (e.g., to track the amount of money spent), can be added to each event. While not all events provide meaningful identifiers or values, they added to roughly a third of the events an identifier, value, or both. Sometimes, more than one identifier or value is possible. In such cases, they duplicated the event and set the fields accordingly.

Many of the features used were initially inspired by [29] and [30]. Over the past years, they have added numerous additional features to their toolbox. The features can be categorized into different buckets.

- 1) *Basic activity*: Measuring basic activity such as the current absence time of a player, the average playtime of a player, the number of days a player has been active, or the number of sessions.
- 2) *Counts*: Counting the number of times an event or an event-identifier combination occurs.
- 3) *Values*: Different mathematical operations applied to the event values, e.g., sum, mean, max, min, etc. (i.e., to the statistic features).
- 4) *Timings*: Timing between events to detect recurring patterns, e.g., a slowly decreasing retention.
- 5) *Frequencies*: A player's activity can be transformed from a time series into the frequency domain. Now the strongest recurring frequency of a player can be estimated and used as a feature.
- 6) *Curve fitting*: As described in [29], curve fitting can be applied to time series data. Parameters such as a positive slope of the fit suggest that the interest in the game is increasing.

Based on these features, datasets for training and testing sets generated. This resulted in almost 600 features for each player. But with only 4000 players in the training dataset, one has to be careful with too many dimensions. Depending on the algorithm, feature selection and regularization are not only helpful but necessary. Additionally, they applied the outlier detection to the training dataset. Outlier detection helps to remove noise from the raw data and to exclude misleading data introduced by malicious players or bots. The outlier detection removed 0.5% of players in the training dataset.

2) *Modeling*: Goedle.io evaluated a variety of algorithms for each prediction problem. Some algorithms handle raw datasets quite well, e.g., tree-based algorithms, but other algorithms strongly benefit from a scaling data. For that reason, they apply a simple scaler to the datasets.

To find the best performing algorithm including features, pre-processing steps, and parameters, they iteratively test different combinations. To evaluate algorithms and to measure the improvements due to the optimization, they used a five-fold cross validation based on the F1-score. Goedle.io selected the best algorithm among different ones including: LR, Voted Perceptron, Decision Trees, RF, Gradient Tree Boosting, Support Vector Machines, and Artificial Neural Networks.

### H. MNDS (6th Place in Track 1)

MNDS encoded the time-series data into image pixels, user variables were mapped to the  $x$ -axis of the image, and the game-use period (eight weeks) was linearly mapped to the  $y$ -axis of the image. The features were composed of 13 input variables, obtained by compressing 20 variables according to the order of variable importance provided from RF model. The process was repeated, in which similar variables were removed. Ultimately, the image pixels per user were composed of  $13 \times 56$  pixels. Its based on the work on tiled convolutional neural networks [31].

In addition, since the behavior occurring near the eighth week affects customer churn as opposed to the behavior of the first week, a distance-weighted coefficient  $W_i$  is assigned to the reciprocal of the time "distance" to give a weight for each day, as follows:

$$W_i = \frac{1}{d(t_q, t_i)} \quad (3)$$

where  $t_q$  is the date of  $q$  (query) point;  $t_i$  is the last date + 1; and  $d(t_q, t_i)$  is the distance between two dates.

Next, to convert the weighted time-series data into image pixels using a Gramian Angular Field (GAF), min-max normalization was performed with a value between  $[-1, 1]$ , according to the following:

$$\tilde{x}^{(i)} = \frac{(x^{(i)} - \max(X)) + (x^{(i)} - \min(X))}{\max(X) - \min(X)}. \quad (4)$$

The next step of the GAF is to represent the value of the normalized value  $\tilde{X}$  in the coordinates of the complex plane, where the angle is expressed by the input value and the radius  $i$  is expressed by the time axis

$$\begin{aligned} \emptyset &= \arccos(\tilde{x}^{(i)}), -1 \leq \tilde{x}^{(i)} \leq 1, \tilde{x}^{(i)} \in \tilde{X} \\ r &= \frac{i}{N}, i \in N. \end{aligned} \quad (5)$$

Finally, the GAF is defined ahead, and the  $G$  matrix is encoded as an image

$$G = \begin{bmatrix} \cos(\emptyset_1 + \emptyset_1) & \cdots & \cos(\emptyset_1 + \emptyset_n) \\ \vdots & \ddots & \vdots \\ \cos(\emptyset_n + \emptyset_1) & \cdots & \cos(\emptyset_n + \emptyset_n) \end{bmatrix}. \quad (6)$$

The GAF provides a way to preserve the temporal dependence, since time increases as the position moves from top-left to bottom-right. The GAF contains temporal correlations, as the  $G$  matrix represents the relative correlation by superposition of directions with respect to the time interval  $k$ .

MNDS then applied the modified model of Inception-V3 [32] into image pixel data for churn prediction.

### I. DTND (7th Place in Track 1 and 5th Place in Track 2)

DTND supposed that the data were noisy. Hence, GLM and Elastic-net were applied to the learning model. The feature set



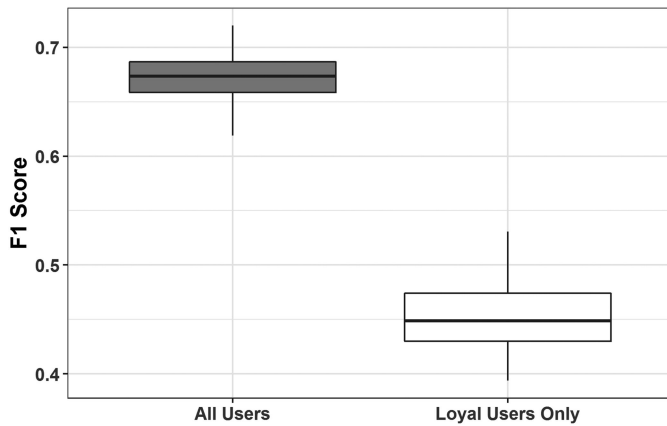


Fig. 8. F1 score distribution resulting from a comparative experiment of churn prediction using five machine learning algorithms. Higher predictive performance was achieved when training and evaluation were conducted using data with all users (left) compared to when the same process was applied to a dataset comprising only loyal users (right).

included days played, action counts, action type diversity, and file size. None of the features had a negative correlation. With vanilla GLM, owing to offset overloaded features, negative correlation factors exist. In addition, they predicted nonchurn instead of churn.

## VI. DISCUSSION: POSTCONFERENCE ANALYSIS

### A. Difficulty of the Competition Problems

Churn prediction of only loyal users differed considerably from that based on all user types. When conducting churn prediction targeting all users, the task becomes relatively straightforward as the majority of users show obvious churning signals. On the other hand, loyal users seldom churn or their churn is often due to out-of-game-related issues, which hinders churn prediction based on in-game activity data. Our comparison of performance between predicting churn of all users and that for loyal users only confirmed that the performance differed significantly with respect to prediction group targets.

Two experiments were conducted to compare prediction performance regarding churn of all users and of loyal users only. For the first experiment, both training and evaluation sets were created with all user data, whereas for the second experiment, the training and evaluation sets were created using only data from loyal users. Each experiment was trained on data of 3000 users and was evaluated using a dataset consisting of 1400 users' data.

Five machine learning algorithms, RF, LR, Extra Gradient Boosting, Generalized Boosting Model, and Conditional Inference Tree, were applied to churn prediction of all users and loyal users only. When each machine learning algorithm was trained and evaluated on a dataset with all users, the F1 score ranged approximately from 0.6 to 0.72. When the same procedure was repeated with a dataset of only loyal users, the F1 score ranged from 0.39 to 0.53, showing a significant drop in overall predictive performance (see Fig. 8).

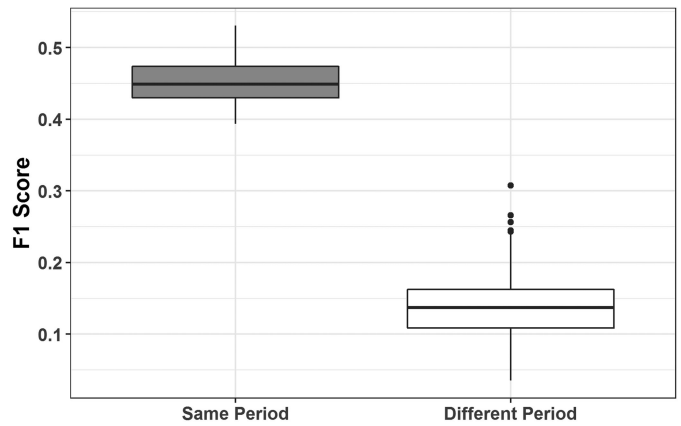


Fig. 9. Performing churn prediction with concurrent training and evaluation data (left) shows more accurate prediction compared with performing the same task with training and evaluation data created from different periods (right).

Owing to the emphasis on chronological robustness, we expected that imposing a time difference between training and test sets would lower the performance of participants compared to other churn prediction work, as confirmed by a comparative experiment of churn prediction of users. The experiment was conducted using two evaluation sets where one was created with data from the same period as the training set and the other of data from periods two months after. The results revealed much lower performance by predicting the churn of users from different periods.

Using the same five machine learning algorithms, prediction models were trained on user data from November 1, 2017, to November 30, 2017. Performance was measured using a test set containing data of concurrent yet different users, yielding a mean F1 score of approximately 0.45. The same models were used to predict the churn of different users two months later—from January 1, 2018, to January 31, 2018—yielding a significantly lower F1 score ranging from 0.04 to 0.3 (see Fig. 9).

Based on these experimental results, we expected the participants' prediction performances to be lacking compared with those of previous studies.

### B. Right Censoring Issue in Track 2

For any survival analysis problem, the exact survival time may be incalculable due to censoring, in which some data are only partially known due to actual constraints, as opposed to model design. The most common type of censoring is right censoring, where survival time is at least as long as the observation period, as shown in Fig. 10. Owing to the censoring nature of survival analysis, we predicted that employing a method that correctly incorporates both censored and uncensored data would be crucial. Consequently, it was not surprising to find that Yokozuna Data, the only group to consider the censoring nature of the problem explicitly and implement an effective method, namely, a conditional inference survival ensemble, exhibited superior performance and won the track.

Contestants also had to consider that the performance assessments via a test server and the final results were performed

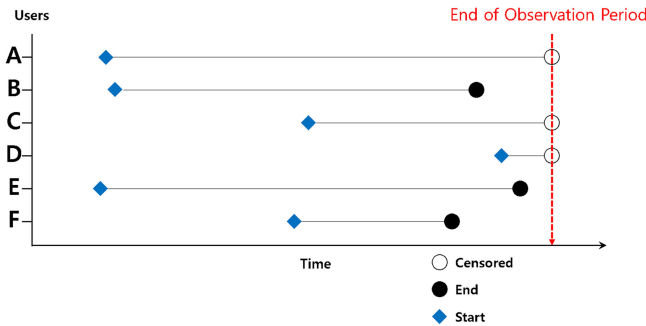


Fig. 10. Data censoring (right censoring).

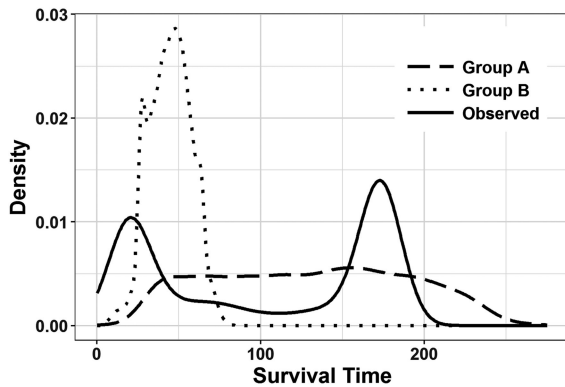


Fig. 11. Distribution of survival time for Test 2 compared among the prediction results of Groups A and B and observed survival times.

through calculation using different censoring points. The final evaluation was made based on the survival time calculated on July 31, 2017, whereas the survival time for the test server evaluation was calculated based on survival time by March 28, 2017. We expect that the consideration of such different censoring points between the test server and final evaluation would be crucial to prevent overfitting to the test server and attain accurate predictions; such expectations coincided with the final results. Yokozuna Data and IISLABSKKU were the only teams that not only reached the top places—first and second—but also improved their final standings compared with test server standings; they also appeared to be the only contestants who considered the different censoring points.

When the distributions of survival time of each team for the second test set were compared, there was a significant difference between the survival time prediction of Yokozuna Data and IISLABSKKU (Group A), whose final standing improved greatly compared with the standings from the test server results and the other participants (Group B) whose standing deteriorated.

As shown in Fig. 11, the survival time predicted by Group A had a wider spread, which indicated that Group A predicted users to survive far beyond the test server and final evaluation censoring point. In contrast, Group B's prediction did not seem to consider survival beyond the censoring point of the test server. Because Group B predicted significantly shorter survival times, we suspect that their results were over-fitted to the test server results.

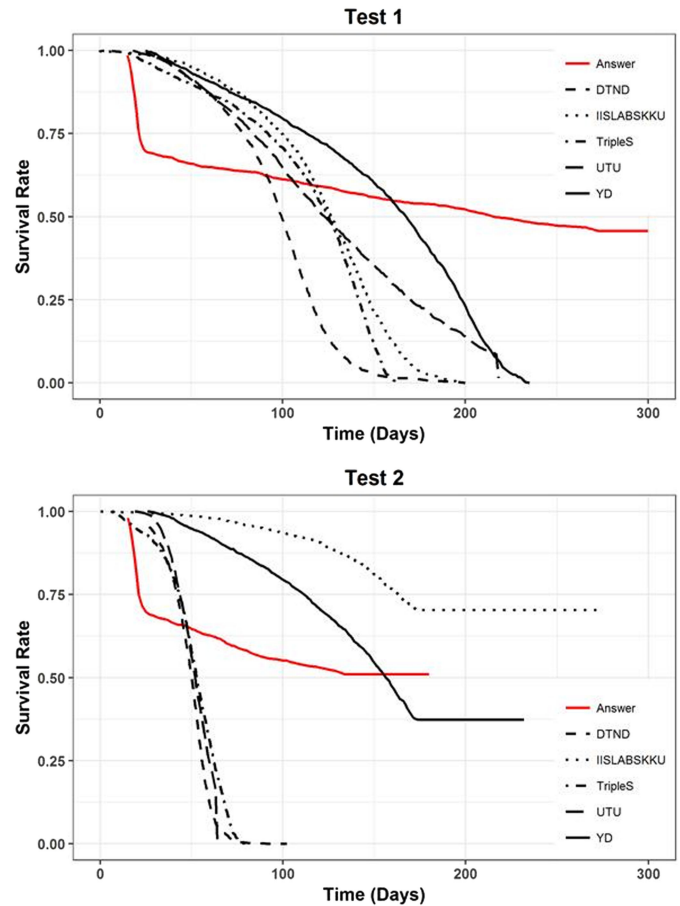


Fig. 12. Survival curves of Test set 1 and 2 in Track 2.

Furthermore, our performance measurement for survival time prediction should be improved in the future. Fig. 12 shows survival curves of participants' prediction result for test set 1 and test set 2. According to the Fig. 12, we suspect that RMSLE tends to benefit from overestimating survival time. We think time-dependent ROC curve can be a good alternative for measuring a performance of survival time prediction [33].

### C. Future Works to Improve the Competition

To run the competition, we provided training and test datasets (without labels) and allowed the participants to build models. This means that they could use the test data for unsupervised learning tasks to build more generalized models. It is important to measure the progress of the participants and encourage them to compete with each other. For that purpose, we supported a test server, which return test accuracy on 10% of datasets (sampled from the test sets). Participants used the test server to monitor their model's progress and their current position on the leaderboard of the server. Although the test server is a useful tool for competition, there are several issues to solve for future game data mining competitions.

- 1) For the competition organizers, it is important to run the test server securely. For example, it must not be possible to get the ground truth of the test sets used in the test

server. In the ImageNet competition, for example, there was a case in which the participants used the test server illegally by opening multiple accounts.

- 2) For the participants, it is important to provide useful information from the test server to enable the best performance. However, it is challenging to run the competition server properly while gamers on the test dataset are still playing the Blade & Soul game. In Track 2 (survival analysis), the ground truth of the test sets was not determined before the final evaluation of the entries because the gamers on the test log data still played the games and their ground truth value (survival weeks) changed, up until the last evaluation.

Compared with other game AI competitions, the number of participants in this game data mining competition was not small; however, future efforts should attempt to attract more data science researchers to the competition. In terms of participant diversity, the competition should attempt to attract participants from many different countries (e.g., this year, South Korea was the dominant country in the competition). This requires the use of various advertisement venues to attract a more diverse group of international participants. This year, participants used the test server implemented by our team; however, well-known data competition platforms such as Kaggle (<https://www.kaggle.com/>) may help the competition promote.

## VII. CONCLUSION

In this paper, we propose a competition framework for game data mining using commercial game log data. The goal of the competition was very different from other types of game AI competitions that targeted strong or humanlike AI players and content generators. Here, the goal of the game data mining competition was to build an accurate predictor trained using game log data. The first track focused on the classification problem to predict churn or no churn as a binary decision task. The second task involved solving a regression problem to predict the number of weeks that the gamer would survive in the future. From a practical perspective, the second task is more desirable for the game industry; however, it is more difficult to make an accurate prediction for this.

Furthermore, we designed the competition problem in consideration of the various practical issues in live game. First, a much longer time span was given between the training data and prediction window, reflecting the minimum time required to execute churn prevention strategies. Second, test sets include a change of business model to drive concept drift issue. Third, we provided only log data of loyal users for the competition. According to our experiments, they tend to be more difficult to predict churn than others, however they are more valuable in business.

Throughout the competition, various methods were applied and tested on the game log data of Blade & Soul of NCSOFT. A total of 13 teams participated in Track I and 5 teams in Track II. Yokozuna Data won the both.

Our approach marks the first step in opening a public dataset for game data mining research and provides useful benchmark-

ing tools for measuring progress in the field. For example, Kummer *et al.*, recently reported the use of commitment features using the benchmarking dataset [34]. Although we limited the competition's goal to predict players' churning behavior and number of surviving weeks, another purpose of our game data mining competition is to promote the research of game data mining by providing commercial game logs to the public.

## ACKNOWLEDGMENT

The authors would like to express great thanks to the all contributors. All the data used in this competition are freely available through GDMC 2017 homepage (<https://cilab.sejong.ac.kr/gdmc2017/>) and google groups (<https://groups.google.com/d/forum/gdmc2017>).

## REFERENCES

- [1] K.-J. Kim and S.-B. Cho, "Game AI competitions: An open platform for computational intelligence education," *IEEE Comput. Intell. Mag.*, vol. 8, no. 3, pp. 64–68, Aug. 2013.
- [2] J. Togelius, "How to run a successful game-based AI competition," *IEEE Trans. Comput. Intell. AI Games*, vol. 8, no. 1, pp. 95–100, Mar. 2016.
- [3] P. Hingston, "A Turing test for computer game bots," *IEEE Trans. Comput. Intell. AI Games*, vol. 1, no. 3, pp. 169–186, Sep. 2009.
- [4] D. Perez *et al.*, "The 2014 general video game playing competition," *IEEE Trans. Comput. Intell. AI Games*, vol. 8, no. 3, pp. 229–243, 2016.
- [5] N. Shaker *et al.*, "The 2010 Mario AI championship: Level generation track," *IEEE Trans. Comput. Intell. Games*, vol. 3, no. 4, pp. 332–347, Dec. 2011.
- [6] M. S. El-Nasr, A. Drachen, and A. Canossa, *Game Analytics: Maximizing the Value of Player Data*. Berlin, Germany: Springer, 2013.
- [7] J. Lee, S.-W. Kang, and H.-K. Kim, "Hard-core user and bot user classification using game character's growth types," in *Proc. Int. Workshop Netw. Syst. Support Games*, 2015, pp. 1–3.
- [8] E.-J. Lee, J.-Y. Woo, H.-S. Kim, and H.-K. Kim, "No silk road for online gamers!: Using social network analysis to unveil black markets in online gamers," in *Proc. 2018 World Wide Web Conf.*, 2018, pp. 1825–1834.
- [9] S. Chun, D. Choi, J. Han, H. K. Kim, and T. K. Kwon, "Unveiling a socio-economic system in a virtual world: A case study of an MMORPG," in *Proc. World Wide Web Conf.*, 2018, pp. 1929–1938.
- [10] J. Runge, P. Gao, F. Garcin, and B. Faltings, "Churn prediction for high-value players in casual social games," in *Proc. IEEE Conf. Comput. Intell. Games*, 2014, pp. 1–8.
- [11] A. Perianez, A. Saas, A. Guitart, and C. Magne, "Churn prediction in mobile social games: Towards a complete assessment using survival ensembles," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics*, 2016, pp. 564–573.
- [12] S. Kim, D. Choi, E. Lee, and W. Rhee, "Churn prediction of mobile and online casual games using play log data," *PLoS One*, vol. 12, no. 7, pp. 1–9, 2017.
- [13] R. Bartle, "Hearts, clubs, diamonds, spades: Players who suit MUDs," *J. MUD Res.*, vol. 1, no. 1, p. 19, 1996.
- [14] R. Bartle, "A self of sense," *Multi User Entertainment*, pp. 1–13, 2003.
- [15] N. Yee, "The gamer motivation profile: What we learned from 250,000 Gamers," in *Proc. Annu. Symp. Comput. Human Interact. Play*, New York, NY, USA, 2016, p. 2.
- [16] Artificial and Computational Intelligence in Games, *Artificial and Computational Intelligence in Games: A Follow-up to Dagstuhl Seminar 12191*. Red Hook, NY, USA: Curran Associates, Inc, 2015.
- [17] S. S. Farooq and K.-J. Kim, "Game player modeling," in *Encyclopedia of Computer Graphics and Games*, N. Lee, Ed. Berlin, Germany: Springer, 2015, pp. 1–5.
- [18] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player modeling using self-organization in Tomb Raider: Underworld," in *Proc. IEEE Symp. Comput. Intell. Games*, 2009, pp. 1–8.
- [19] J. M. Bindewald, G. L. Peterson, and M. E. Miller, "Clustering-based online player modeling," in *Computer Games*. Cham, Switzerland: Springer, 2016, pp. 86–100.
- [20] M. S. El-Nasr, A. Drachen, and A. Canossa, *Game Analytics: Maximizing the Value of Player Data*. Berlin, Germany: Springer, 2013.



- [21] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*. New York, NY, USA: Elsevier, 2011.
- [22] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, vol. 21. Menlo Park, CA, USA: AAAI Press, 1996.
- [23] E. Lee *et al.*, "You are a game bot!: Uncovering game bots in MMORPGs via self-similarity in the wild," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2016, pp. 1–15.
- [24] M. C. Mozer, R. Wolniewicz, D. B. Grimes, E. Johnson, and H. Kaushansky, "Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 690–696, May 2000.
- [25] A. Tsymbal, "The problem of concept drift: Definitions and related work," *Comput. Sci. Dept., Trinity College Dublin, Dublin, Ireland*, vol. 106, p. 2, 2004.
- [26] M. Tamassia, W. Raffae, R. Sifa, A. Drachen, F. Zambetta, and M. Hitchens, "Predicting player churn in destiny: A hidden Markov models approach to predicting player departure in a major online game," in *Proc. IEEE Conf. Comput. Intell. Games*, 2016, pp. 1–8.
- [27] P. Bertens, A. Guitart, and Á. Periañez, "Games and big data: A scalable multi-dimensional churn prediction model," in *Proc. IEEE Conf. Comput. Intell. Games*, New York, NY, USA, 2017, pp. 33–36, doi: [10.1109/CIG.2017.8080412](https://doi.org/10.1109/CIG.2017.8080412).
- [28] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, arXiv:1207.0580.
- [29] F. Hadiji, R. Sifa, A. Drachen, C. Thureau, K. Kersting, and C. Bauckhage, "Predicting player churn in the wild," in *Proc. IEEE Conf. Comput. Intell. Games*, 2014, pp. 1–8.
- [30] R. Sifa, F. Hadiji, J. Runge, A. Drachen, K. Kersting, and C. Bauckhage, "Predicting purchase decisions in mobile free-to-play games," in *Proc. 11th AAAI Conf. Artif. Intell. Interactive Digit. Entertainment*, 2015, pp. 79–85.
- [31] Z. Wang and T. Oates, "Encoding time series as images for visual inspection and classification using tiled convolutional neural networks," in *Proc. Workshops 29th AAAI Conf. Artif. Intell.*, 2015, pp. 40–46.
- [32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2818–2826, 2016.
- [33] P. J. Heagerty, T. Lumley, and M. S. Pepe, "Time-dependent ROC curves for censored survival data and a diagnostic marker," *Biometrics*, vol. 56, no. 2, pp. 337–344, 2000.
- [34] L. B. M. Kummer, J. C. Nievola, and E. Paraiso, "Applying commitment to churn and remaining players lifetime prediction," in *Proc. IEEE Conf. Comput. Intell. Games*, 2018, pp. 1–8.