# Social networks for enhanced player churn prediction in mobile free-to-play games

1. Introduction

* Social networks in online and mobile games – important for the players and their experience
* E.g. communities and clans of players in multiplayer online and battle games and social network games
* Affect in-game performance, frequency, length of play and engagement
* Information from social networks – provides alternative and significant info when predicting churn in the telecommunication industry (previous research)
* This paper: study the importance of networks on player churn in a f2p mobile game with 1v1 matches
* 2 types of networks based on how the players are matched
  + Explicit connections (friend matches)
  + Implicit connections (match based on players’ similarity)
* Extract features from the networks and train churn prediction models with combinations of behavioral and network features
* Evaluate whether info form the 2 social networks enhances the predictive performance of player churn prediction models
* Focus on players’ behavior in the first day of game playing -> use such info to predict player churn
* Quantify the burstiness of players in terms of the model’s performance (how quickly they churn)

1. Related work

* Social networks in gaming
* Churn prediction in online and mobile gaming

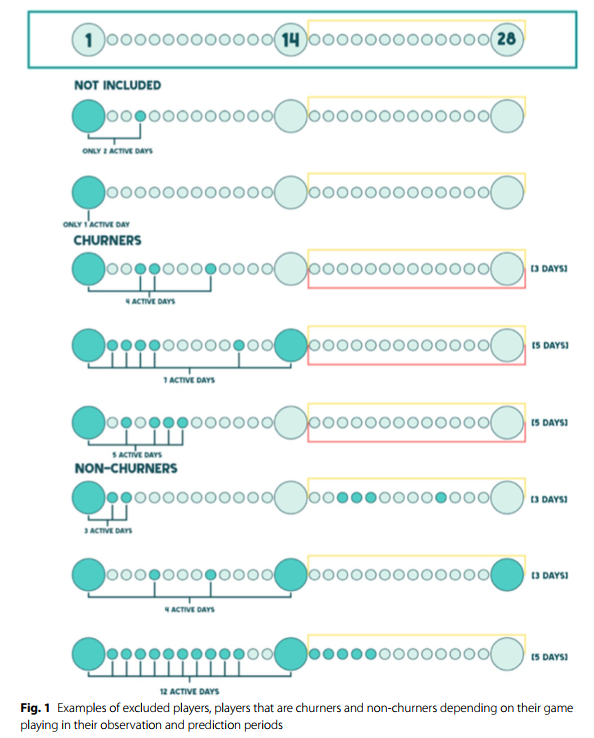
1. Methodology
2. Player data

* Dataset of players of a mobile game
* Sample of players: 10 consecutive days -> random sample of 1000 players that installed the game on each of these 10 days -> 10000 players
* Included players’ gaming behavior: all in-game activity for 1 year after installing the game
* Consider when the players become active in the game as the first active day
* Features generated from in-game activity data – Benchmark features, representing information used for churn prediction without any network info

|  |  |
| --- | --- |
| Type | Description |
| Per day | Features relating to how often per day do certain events: how many matches a player plays, wins and losses, occur on average, median, minimum, and maximum for each player |
| Per match | Features relating to how often per match do certain match related events occur on average, minimum, maximum for each player (how often a player wins or loses a point and misses a stroke) |
| Over all | Features relating to how often match related events occur for each player during the 14 day observation period |
| Match types | Features relating to how many Bet-, Friend-, and Tournament matches a player plays |
| Playing period | Features relating to how many days players play the game during the observation period and number of days between first and last played days within observation period |
| Streaks | Features relating to how many win and loss streaks a player has, the number of matches within their streak and how long a streak is. The number of matches within a streak = the number of matches won / lost in a row |
| Currency | Features relating to the hard and soft currency of players |
| Purchases | Features relating to the purchases made by players, with both hard and soft currency |
| Friends | Features relating to matches played against friends |
| Player | Features relating to the statistics of a player |
| Win-loss types | Features relating to different types of wins/losses (e.g., player automatically wins if opponent abandons a match) |

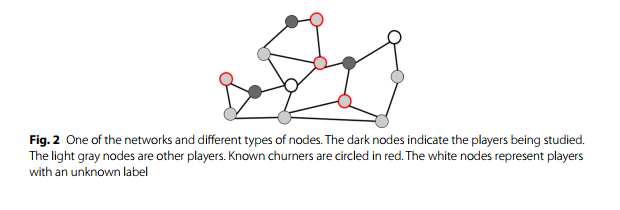
1. Defining churn

* Churn prediction in f2p mobile games – challenging for 2 main reasons:
* No subscription model -> players can stop playing at any time without notice -> no clear or formal churn event
* Player behavior is bursty – new players ten to play very intensively for a short period of time and then quit / churn -> data distribution is very skewed
* The first week or 2 are the most crucial time when trying to predict and prevent churn
* Method to define churn:
* Observation period: first 14 active days of a player
* Prediction period: subsequent 14 days
* If player played no games in the prediction period -> churners
* Require players to have played for at least a fixed number of days in their observation period. Consider 2 cases
  + At least 3 days -> 50.04% churn rate
  + At least 5 days -> 45.87% churn rate
* Note: 5-day data is a subset of the 3-day data

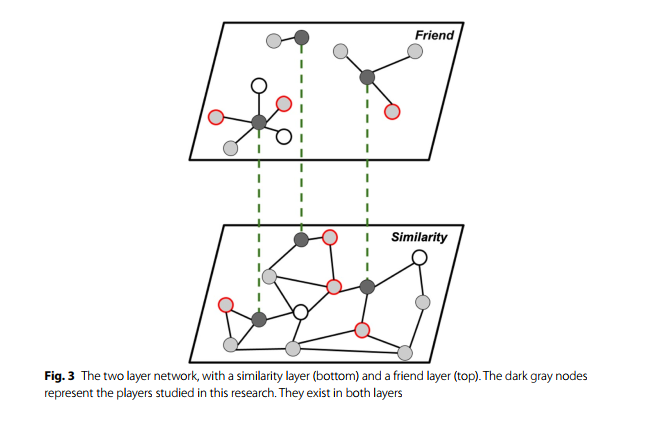


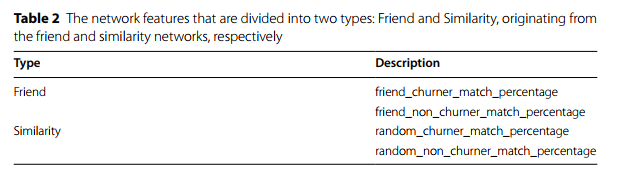
1. Networks and network features

* 2 types of matches: friend and similarity
* 2 types of networks: friend network and similarity network
* Friend network: explicit social network
* Similarity network: implicit social network – nodes are connected based on similar properties (e.g., skill level and geographic location)
* when 2 players play a match, and edge is created between them. The nodes are the 2 players
* use 10 cohorts of players to create the 2 networks
* The networks include: players under study + players who did not install the game during the 10 days mentioned
* Dark gray: players under consideration
* Light gray and white: their neighbors
* Red circle: churners
* White: label unknown



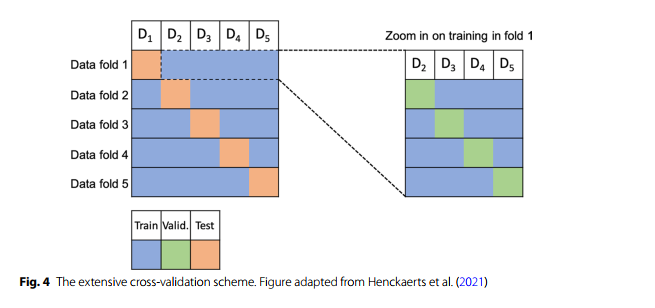
* 2 layer network (dashed lines connecting the same player between 2 layers)
* Extract features on the nodes’ connectivity and created 2 types of features: friend and similarity
* Further analysis into how friends affect game play



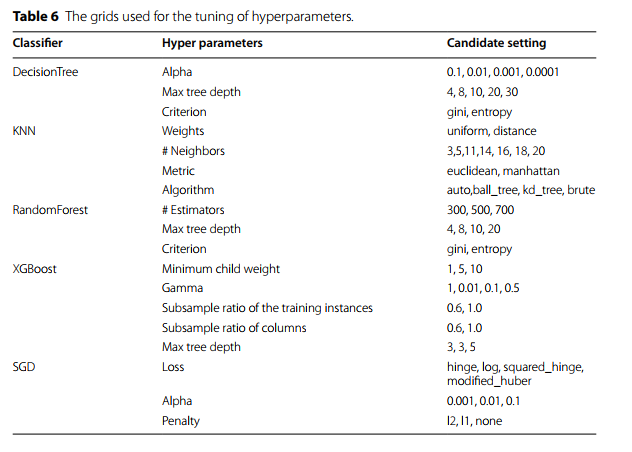


1. Experimental setup

* Using the 2 datasets, 3-day and 5-day, train and test a range of ML algorithms to predict churn and evaluate their performance
* Each algorithm was trained on both datasets, using the following combinations of feature types:
* Benchmark
* Benchmark + friends
* Benchmark + similarity
* Benchmark + friends + similarity
* Algorithms: decision trees, random forests, stochastic gradient descent (SGD), KNN, XGBoost
* Cross validation scheme (Henchkaerts et al, 2021)



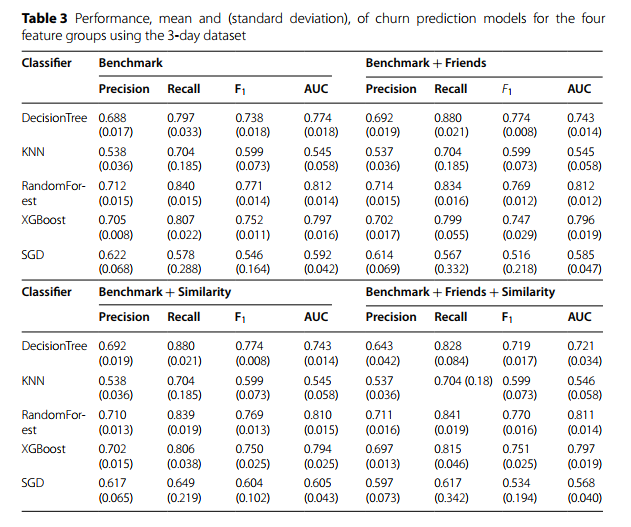
* Split the full dataset D into 5 disjoint subsets
* 5-fold cross-validation where each of the 5 subsets served as a test set in one fold
* In each training fold, we further performed a four-fold cross validation to tune the hyperparams for highest performance
* Hyperparams that gave the highest performance were then used to train the model in the original fold and the model validated on the test set of each respective fold
* Serves 2 purposes:
  + Allow hyperparam tuning with a 4-fold cross validation approach
  + As the models are trained several times, we could evaluate their predictive performance on multiple datasets, instead of a single test set
* Obtain multiple performance measures per model -> more accurate performance assessment
* Enable sensitivity checks to assess the stability of different algorithms
* Reduce the risk of bias in the results
* Grids used for tuning hyperparams

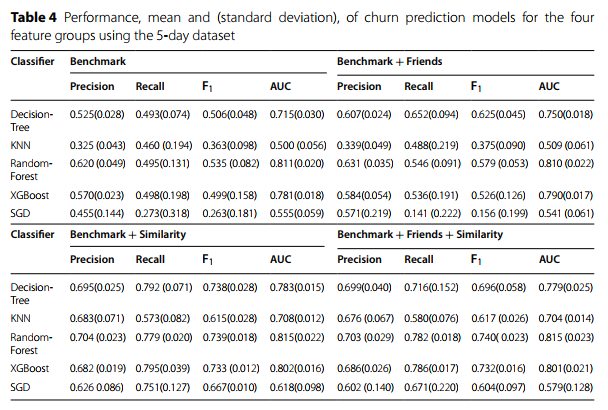


* Performance evaluated using: precision, recall, F1 score and AUC

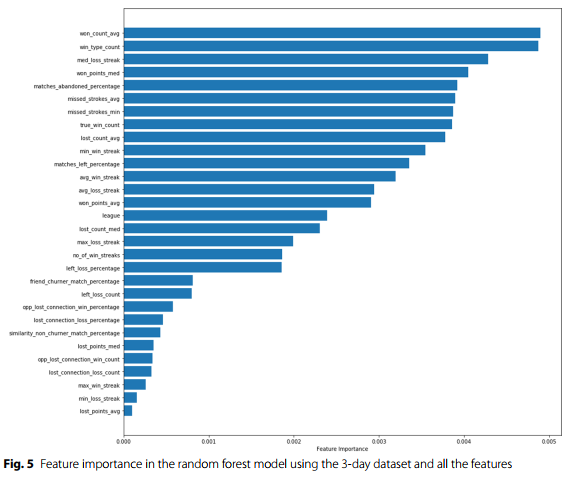
1. Results
2. Impact of network features

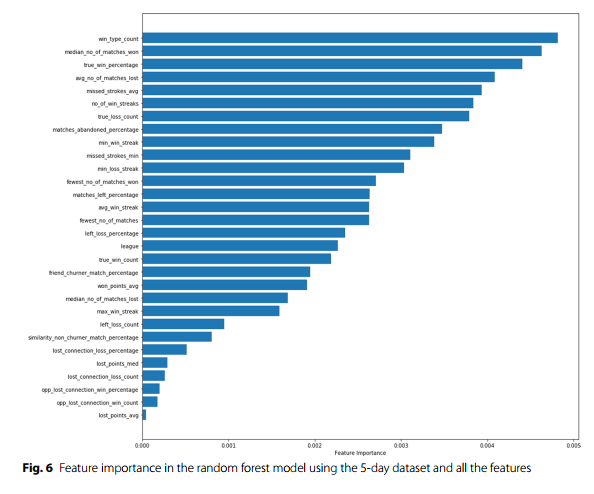
* Reflect on the results of including network features in the churn prediction models





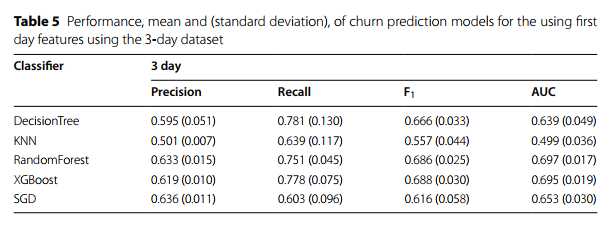
* Benchmark features: RF and XGBoost performed the best for both datasets, followed by decision trees
* Benchmark + features from the friend network: RF and decision trees performed the best for both datasets
* Benchmark + features from similarity network: XGBoost, RF and decision trees performed the best for both datasets
* All features included: RF and XGBoost performed the best
* Comparing predictive performance of feature groups:
* Adding friend features -> slight increase in the model’s performance
* Adding similarity features -> greater boost in the model’s performance (especially in the 5-day dataset)
* All features included -> highest performance (especially recall)
* Performance on the 3-day dataset is slightly higher than for the 5-day dataset. Notably this is due to the higher recall in the 3-day dataset, where the model are better at detecting churners
* Inspecting feature importance for the random forests models when using all the features
* Features representing wins and losses are important
* As well as streaks
* In-game activity
* Friend and similarity network features appear among the top 30 most important features
* Friend network feature is the percentage of games played against churners -> if a player’s friends churn, then the player is more likely to churn
* Similarity network feature is the percentage of games played against non-churners





1. First day features

* The less the player played the game the more likely they were to be churners
* Examine whether it would be possible to predict a player’s churn label after their first day of game play
* Same algorithms applied to the features for the players first day of game play
* Model showed that it is possible
* Features for the first day were different form the features used for modeling of the main data
* Many of the original features were, for e.g., based on the number of times an event occurred during the observation period
* Some features were represented with a Boolean value instead of a quantitative value (e.g., currency)



1. Discussion and conclusion