Performance of Machine Learning Algorithms in Predicting Game Outcome from Drafts in Dota 2

Aleksandr Semenov[1], Peter Romov[2,3], Sergey Korolev[4,5], Daniil Yashkov[2,3], and Kirill Neklyudov[2,3]

[1] International Laboratory for Applied Network Research,
National Research University Higher School of Economics, Moscow, Russia
[2] Yandex Data Factory, Moscow, Russia
[3] Moscow Institute of Physics and Technology, Moscow, Russia
[4] National Research University Higher School of Economics, Moscow, Russia
[5] Moscow State University, Moscow, Russia
avsemenov@hse.ru, peter@romov.ru, sokorolev@edu.hse.ru,
daniil.yashkov@phystech.edu, k.necludov@gmail.com

Abstract. In this paper we suggest the first systematic review and compare performance of most frequently used machine learning algorithms for prediction of the match winner from the teams' drafts in DotA 2 computer game. Although previous research attempted this task with simple models, weve made several improvements in our approach aiming to take into account interactions among heroes in the draft. For that purpose we've tested the following machine learning algorithms: Naive Bayes classifier, Logistic Regression and Gradient Boosted Decision Trees. We also introduced Factorization Machines for that task and got our best results from them. Besides that, we found that model's prediction accuracy depends on skill level of the players. We've prepared publicly available dataset which takes into account shortcomings of data used in previous research and can be used further for algorithms development, testing and benchmarking.

Keywords: online games, predictive models, DotA 2, factorization machines, MOBA

1 Introduction

Cybersport or eSports has became really popular in recent years, turning into a spectacular entertainment and sport discipline where professional teams regularly participate in tournaments with huge money prizes [1]. Due to such popularity it generates huge amounts of behavioral data on players and matches which is often accessible via API (Application Programming Interface). This data allows researchers to apply different machine learning algorithms to enhance gaming AI [2], develop optimal strategies [3], detect game balance issues [4] and identify players with multiple aliases [5].

DotA 2 is an online multiplayer video game and its first part, DotA (after "Defense of the Ancients") created a new genre, called Multiplayer Online Battle Arena (MOBA). It is played by two teams, called Radiant and Dire which consist of five players each. The main goal of the game is to destroy other team's "Ancient", located at the opposite corners of the map after destroying all the towers on three different lanes that lead to that Ancient.

Each of the players choose one hero to play with from a pool of 113 heroes. A hero has a set of particular features that define his role in the team and playstyle. Among these features there are his basic attribute (Strength, Agility or Intelligence), unique set of 4 (or for some heroes even more) skills which serve for a wide variety of purposes from healing and increasing stats of friendly units to different types of damage, stun and slow down of enemy heroes. Besides that, there is a lot of items a player can buy for his hero, which increase some of his stats, skills or add new effects or spells. Skills and items allow each hero to fill several roles in the team, such as "damage dealer" (hero, whose role is to attack the enemies in the fight), "healer" (hero, who mostly heals and otherwise helps his teammates), "caster" (hero, who mostly relies on his spells) etc. However, besides roles there is another way of classification the heroes in the teams which is based on their functions and is believed to be the most efficient and balanced. According to that classification, the optimal composition of the team is the following: "Mid" (player, who starts from the middle lane and responsible for ganking attempts on the other lanes), "Carry" (damage dealer who is supposed to kill enemy heroes), "Offlaner" or "Hardlaner" (hero who either starts on the "hard lane" which is bottom lane for Dire and top lane for the Radiant, or roams between lanes and in the jungle) and two "Supports" (heroes who are responsible for buying items for the team, like observer wards, smokes, couriers etc.).

The game has several modes: All Pick, Captains Draft, Random Pick etc. which define the way and order the players choose heroes. In All Pick, for example, players can choose from all the pool of heroes without well defined order of pick which means that the one, who was quicker gets the hero he wants to play. Captains Draft on the other hand make only one person in the team responsible for picks and bans which are made in consecutive order. This two opposite regimes of play create different strategies for players in both choosing a hero and playing it. For example in All Pick player tries to take the hero he likes the most which sometimes isn't good for the balance in the team. Moreover, some heroes are over represented in All Pick while this stats is smoothed in other modes (like Random Draft, where each player gets his hero randomly).

There is a Matchmaking Rating system in Dota 2 called MMR, which allows the algorithm to put players with similar skill level into the same match for more balanced gameplay. It is also possible to make your rating visible to others via participating in such modes as Ranked All Pick, Ranked Random Draft and Ranked Captain's Mode which give all the members of the winning team some MMR points and removes the same amount of points from the lost team. Currently only a dozen of players have MMR $\geq$ 8,000, while everything below 4,000 is considered as low-level, and 6,000 $-$ 7,000 is a level of the professional player who participates in esport championships.

Different gaming modes affect on the hero pick popularity, win rate and other parameters. Moreover, the same hero's stats for kills, deaths, win rate etc. may differ dramatically between gamers from different skill brackets, measured by MMR system.

From this brief introduction to the game's mechanics it must already be clear that this system leads to a huge number of possible combinations of heroes and their spells, items and roles. For example the amount of different teams for 5 heroes each is equal to 140,364,532. In the next section we'll review how researches dealt with this level of complexity in prediction of winning team.

2 Previous Research

DotA 2 got attention from researchers only recently. First articles were mostly descriptive, general and theoretical, investigating, for example, rules and fair play maintenance of the games [6] or correlation of leadership styles of players with roles in the game (carry, support, jungler etc.) they choose to play [7]. In the first quantitative research of Dota 2, authors analyzed cooperation withing teams, national compositions of players, role distribution of heroes and some other stats based on information from its web forums [8].

Later researchers discovered the potential of the data provided by the game itself soon after that and started using to test hypothesis, detect patterns and make predictions. For example Rioult et al. [9] analyzed topological patterns of DotA teams based on area, inertia, diameter, distance and other features derived from their positions and movements of the players around the map to identify which of them are related with winning or loosing the game. Drachen et al. used Neural Networks and Genetic Algorithms to analyze and optimize patterns of heroes movements on the map in DotA [10]. Eggert et al. [11] applied classification algorithms to heroes game statistics to identify their roles. However most researched topic was win prediction from heroe drafts.

Conley & Perry were the first to demonstrate the importance of information from draft stage of the game with Logistic Regression and k-Nearest Neighbors (kNN) [6]. They got 69.8% test accuracy on 18,000 training dataset for Logistic Regression, but it could not capture the synergistic and antagonistic relationships between heroes inside and between teams. To address that issue authors used kNN with custom weights for neighbors and distance metrics with 2-fold cross-validation on 20,000 matches to choose d dimension parameter for kNN. For optimal d-dimension = 4 they got 67.43% accuracy on cross-validation and 70% accuracy on 50,000 test datasets. Based on that results authors built a recommendation engine with web interface. However one of its drawbacks was it's slow speed: for $k = 5$ kNN took 4 hours and 12 hours for cross-validation.

Although their work was the first to show the importance of draft alone, the interaction among heroes within and between teams were hard to capture with such a simplistic approach. Agarwala & Pearce tried to take that into account including the interactions among heroes into the logistic regression model [12]. To define a role of each hero and model their interactions they used PCA analysis of the heroes' statistics (kills, deaths, gold per minute etc.). However, their results showed inefficiency of such approach, because it got them only 57% accuracy while the model without interactions got 62% accuracy. But its worth noticing that although the PCA-based models couldn't match predictive accuracy of logistic regression, the composition of teams they suggested looked more balanced and reasonable from the game's point of view. Another caveat of their approach was that they took data from different sources: the data on match statistics was taken from public games while stats on heroes were based on professional games. This might bias the results because public games are completely different from professional ones and match stats from the former should not be mixed with heroes stats from the letter. In short, they didn't use heroes roles directly and replaced them with PCA components to model the balance of teams. Besides that, they tried to find some meaningful strategies with K-Means clustering on end-game statistics but couldn't find clusters which means that no patterns of gameplay could be detected on their data.

Another approach to that problem of modeling heroes' interactions was proposed by Kuangyan Song, Tianyi Zhang, Chao Ma [13]. They took 6,000 matches and manually added 50 combinations of 2 heroes to the features set and used forward stepwise regression for feature selection. They chose data from the "All Pick", "Ranked All Pick" and "Random Draft" without leavers and zero kills. 10-fold CV logreg: 3,000 matches total: 2,700 vs 300. Training error 28%, test error – 46% . They concluded that only addition of particular heroes improves the model while the others might cause the prediction go wrong.

Kalyanaraman was the first one to implicitly introduced the roles of the heroes as a feature in the model of win prediction [14]. Author took 30,426 matches from the "All Pick", "Random Draft", "Single Draft", "All Random", "Least Played" and "Captain Draft" game types because in theory it should represent all the heroes in the best way since appearance of any particular hero depends on game type. They filtered the matches by MMR to select only skilled players and took the games which were at least 900 seconds and used ensemble of Genetic Algorithms and logistic regression on 220 matches. Logistic regression alone return 69.42% and ensemble with Genetic Algorighm and logistic regression approached 74.1% accuracy on the test set. Although it's the highest result among all the articles in the review, lack of ROC AUC information and small sample of matches, chosen for the Genetic Algorithm, hampers its reliability.

Another attempt to include interaction among heroes was done by Kinkade & Lim, who took 62,000 matches with "very high" skill level without leavers and game duration at least 10 minutes [15]. 52,000 training, 5,000 testing and 5,000 validation. Tried Logistic Regression and Random Forest with such feature of a pairwise winrate for Radiant and Dire. The feature could capture such relationships as matchup, synergy and countering and each of them increased the quality of the model up to 72.9%. Logistic Regression and Random Forest on picks data only. Got 72.9% test accuracy for Logistic Regression and overfitted Random Forest which gave them after tuning only 67% test accuracy. It is worth mentioning that their baseline, which included highest combined individual win rate for the heroes, had 63% accuracy.

Some authors expanded the scope of win prediction from draft information to other data from the game. Johansson & Wikstrom wrote a thesis where they trained Random Forest on the information from the game (such as amount of gold for each hero, his kills, deaths assists for each minute etc.) which had 82.23% accuracy at the five minute point [16]. Although such accuracy seem to be very high, that fact that it's based on data from the game events makes its use very limited, because it demand real-time data to be practically useful.

From the previous research we've found the following shortcomings:

- vague data acquisition strategies (for example, its not clear why authors filter players by their skill level and only use games with high MMR);
- not enough details on the the quality of results (for example, most papers reported only precision, without ROC AUC, recall, etc.);
- small or incorrect samples of data (sometimes datasets were gathered during periods when some changes in the game mechanics were introduced or the samples were mere thousands of matches).

Hence our contribution is:

- mining and preparing of large and consistent dataset of DotA 2 matches for prediction modeling tasks;
- test the methods suggested previously on this dataset with standard performance metrics;
- introduce Factorization Machines algorithm for match outcome prediction based on interactions among heroes;
- make this dataset publicly available[1];

The rest of the article is organized as follows. In the next part we describe our dataset and our approach to the representations of hero drafts. Then we introduce machine learning algorithms we chose to test on this data. And finally we demonstrate the results of our comparison and their implication for the future research.

3 Game Outcome Prediction

We've set out to estimate the quality of a range of machine learning algorithms for prediction of the match outcome given each team hero drafts. Given this subset of 5 heroes per team we try to predict the result of the match, assuming that there is no ties, so $P(radiant\ wins) = 1 - P(dire\ wins)$.

4 Dataset

We have collected dataset using Steam API. It contains 5,071,858 matches from Captains Mode, Random Draft and Ranked All Pick modes, played between 11^{th}

[1] `http://dotascience.com/papers/aist2016`

February 2016 10:50:04 GMT and 2^{nd} March 2016 14:07:10 GMT, including skill levels of players. During this period there were no changes to the core mechanics of the game, such as major patches, which makes this dataset especially appropriate for algorithm development and testing. Another key feature of this data is augmenting it with players' MMR for ranking the games into several brackets depending on the players skills.

The distributions of number of matches for skill levels and game modes in the dataset are:

	Normal Skill	High Skill	Very High Skill	Total
Captains Mode	33,037	5,599	8,840	47,476
Random Draft	86,472	15,560	39,407	141,439
Ranked All Pick	2,937,087	917,001	1,028,855	4,882,943
Total	3,056,596	938,160	1,077,102	5,071,858

Table 1: Data distribution

In the end we used three representations of hero drafts as input of the algorithms. First is just "bag of heroes" technique, where each draft is encoded as a binary vector of length 2 × N where N is the size of hero pool with

$$x_i = \begin{cases} 1, & \text{if } i \leq N \text{ and hero } i \text{ was in the radiant team} \\ & \text{or if } i > N \text{ and hero } i - N \text{ was in the dire team} \\ 0, & \text{otherwise} \end{cases}$$

We used it as input to the naive bayes, since we wanted to use this result as the baseline for the most straightforward type of draft encoding. We also used it with logistic regression to compare it's performance between this encoding and the second one. Also, factorization machines require binary vectors as input for the model, so we used it with this data as well.

Second is "bag of heroes" with team symmetry for equal weights of Logistic Regression where

$$x_i = \begin{cases} 1, & \text{if hero } i \text{ was in radiant team} \\ -1, & \text{if hero } i \text{ was in dire team} \\ 0, & \text{otherwise} \end{cases}$$

This way of data representation allows logistic regression to use same weights for the same hero picks on radiant or dire side so as to force the symmetry of the game mechanics.

Third is the same "bag of heroes" as in first one, but with added features for number of carries, pushers, supports and other roles in the radiant and dire team. Our hope was that these features would provide tree-based model with explicit information about the strong and weak sides of the given draft based on the composition of heroes' roles in the team.

5 Methods

For our tests we've chosen the following models: Naive Bayes classifier, Logistic Regression, Factorization Machines and Gradient Boosting of Decision Trees. The quality of prediction was measured by AUC and Log-Loss (Cross Entropy) on 10-fold cross-validation.

Naive Bayes and Logistic Regression were chosen to replicate results of previous works on our dataset and to set the baseline performance. Since we assume that combinations and interactions between different heroes and their roles should increase the quality of the model, we chose Factorization Machines and Decision Trees for their ability to model complex interactions on the sparse data.

5.1 Naive Bayes

This model assumes the independence of variables (picking particular heroes), compute univariate probability estimates from training set $P(x_j|y)$ and then use bayesian rule to infer win probability of the draft:

$$P(y|x) \propto P(x_1|y) \ldots P(x_p|y)P(y)$$

The final decision rule for the model is

$$\hat{y} = \underset{k \in \{0,1\}}{\operatorname{argmax}}\, p(C_k) \prod_{i=1}^{n} p(x_i|C_k)$$

Where C_k is the possible outcome, n is the number of features.

5.2 Logistic Regression

Logistic Regression is a linear model that tries to estimate the probabilities for given classes using a logistic function:

$$P(win) = \sigma(w_0 + \sum_{i=1}^{p} w_i x_i),$$

where $\sigma(a) = (1 + \exp(-a))^{-1}$ is an activation function.

Similar to the Naive Bayes approach this model can not distinguish interactions between heroes and estimate possible combinations and their significance for the match outcome. As such it can only estimate individual picks importance for the result of the match.

We tested Logistic Regression on both first and second type of draft encodings and compared the results.

5.3 Factorization Machines

Factorization Machines proposed in [17] models some real-valued target as:

$$\hat{y}(x) = w_0 + \sum_{j=1}^{P} w_j x_j + \sum_{j=1}^{P} \sum_{j'=p+1} P x_j x_{j'} \sum_{f=1}^{k} v_{j,f} v_{j',f}$$

where $\Theta = (w_0, w, V)$ — set of model parameters. For binary probability prediction, bayesian inference is used.

In other words, Factorization Machines compute predicted probability using pairwise interactions of the second order between chosen heroes.

We have used bayesian Factorization Machines [18] implemented in FastFM library [19].

5.4 Gradient Boosting of Decision Trees

We have used XGBoost library [20] for implementation of gradient boosting algorithm. It minimizes the following regularized objective:

$$\mathcal{L}(\phi) = \sum_{i} l(\hat{y}_i, y_i) + \sum_{k} \Omega(f_k)$$

$$\text{where } \Omega(f) = \gamma T + \frac{1}{2}\lambda ||w||^2,$$

$\hat{y}_i$ is model prediction, y is the true value, l is the loss function, T is the number of leaves in the tree, each f_k corresponds to an independent tree structure and leaf weights w and λ is an L_2 regularization parameter.

The prediction $\hat{y}_i$ is the sum of predictions of trees $f_k(x_i)$:

$$\hat{y}_i = \sum_{k=1}^{k} f_k(x_i)$$

6 Results

We've ran all of the models described above on 10-fold cross-validation for ROC AUC and log-loss estimation on the respective datasets and achieved following results.

Skill	Normal		High		Very High	
Method	auc	log_loss	auc	log_loss	auc	log_loss
libFM	0.706	0.898	0.670	0.933	0.660	0.940
XGBoost	0.701	0.903	0.664	0.937	0.654	0.944
XGBoost_roles	0.702	0.902	0.663	0.938	0.653	0.945
LogReg	0.687	0.916	0.656	0.943	0.643	0.952
LogReg_BoW	0.688	0.915	0.656	0.943	0.643	0.952
Naive Bayes	0.685	0.917	0.653	0.945	0.641	0.954
Naive Bayes AP	0.684	0.918	0.654	0.944	0.641	0.954
Dummy	0.500	0.996	0.500	0.999	0.500	0.999

Table 2: Results

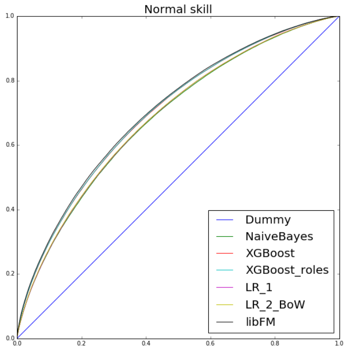

Fig. 1: Normal skill.

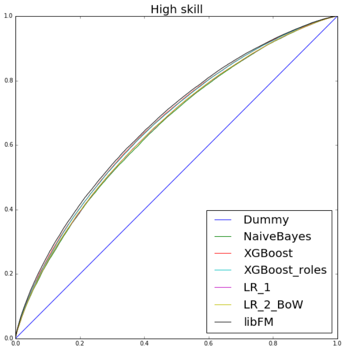

Fig. 2: High skill.

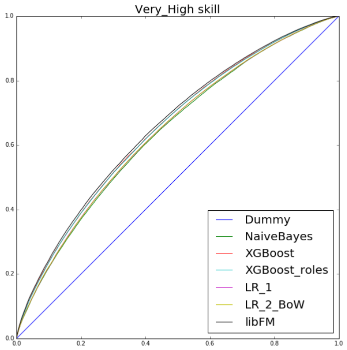

Fig. 3: Very high skill.

In the Table 2 libFM is Factorization Machines classifier, XGBoost is boosting classifier on simple encoding, XGBoost_roles is boosting classifier on enhanced dataset with roles of heroes, LogReg_BoW is the logistic regression classifier on the second type of "bag of heroes" encoding.

We've decided to omit the confidence intervals from this table, since for each of the models the standard deviation was less than 0.0008.

First of all, we decided to check if mixing of different game modes affect the modeling quality. For that purposes we ran Naive Bayes classification algorithm on 'All Pick' mode only (model Naive Bayes AP in the table) and compared the results with the same algorithm applied to the full dataset with all game modes. As we can see, the difference is at 0.001 level which means that mixing different game modes into one dataset almost doesn't affect prediction quality.

Besides that there is no difference between two logistic regression models performance, which means that there is already symmetry in the data so the model is able to infer the individual hero importance without the dataset augmentation. It suggests that side selection doesn't affect the performance of a player on the hero.

Final ranking of ROC AUC scores of the models are representative of the models ability to account for interactions among heroes from the data. Good performance of both Factorization Machines and XGBoost confirmed our assumptions that the ability to include interactions into the model results in significant increase of its performance. It's noteworthy that although gradient boosting decision trees can include interactions of more than second order while factorization machines work only with interactions of 2 heroes, the latter demonstrated better prediction quality than the former. The reasons behind that is unclear for as and might be the subject of research of its own.

It is also important to point out that the addition of role information to dataset held no improvement over "bag of heroes" data for XGBoost classifier. This finding seems counterintuitive since team's composition in terms of heroes' roles is considered to be important and common sense of the game suggests that having no carries in the game or having 5 of them is similarly bad for a team. It might be explained by the hypothesis that decision trees somehow found roles-like features by themselves, but we didn't investigate that hypothesis further.

Besides that we found out that the performance of the classifiers varies between different skill levels of players. More specifically, the higher the skill of the players is the harder it is to predict the outcome of a match. That might mean that low-skilled players depend on the pick more because they can only play on a limited amount of heroes. However for the more skilled player that's not the case since they know how to play and counter more heroes and interact with other teammates.

7 Discussion

In this article we introduced Factorization Machines and Gradient Boosted Decision Trees for the prediction of DotA 2 match winner. With them we managed to increase the accuracy of the winner prediction compared to the models, previously used to solve the problem of hero interactions in the drafts of DotA 2. We also mined the largest dataset of all studied before, for all levels of players' skills, demonstrated several data transformation techniques and used standard and consistent prediction quality measurements on it. Finally, we published the dataset and the code for our analysis online for reproducibility, algorithm development and benchmarking purposes. systems Although our results can be applied for prediction purposes it's still challenging to convert it into recommendation system which would suggest teams optimal pick in real time based on the currently picked/banned heroes because optimization of factorization machines is a very complex computational and analytic problem which demands further investigation.

Such work might be promising and useful for the game developers to access the balance of the game, and for the professional teams, because it will allow them to make data driven decisions for the drafts during the training and preparation for the tournaments. Besides that building applications, based on recommender systems for casual players also looks promising both from scientific and commercial point of view.

Acknowledgements

This paper was prepared within the framework of a subsidy granted to HSE by the Government of Russian Federation for implementation of the Global Competitiveness Program.

References

1. T. Taylor, *Raising the Stakes: E-sports and the Professionalization of Computer Gaming.* Mit Press, 2012.
2. S. Ontanón, G. Synnaeve, A. Uriarte, F. Richoux, D. Churchill, and M. Preuss, "A survey of real-time strategy game ai research and competition in starcraft," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 5, no. 4, pp. 293–311, 2013.
3. G. Synnaeve and P. Bessiere, "A bayesian model for opening prediction in rts games with application to starcraft," in *Computational Intelligence and Games (CIG), 2011 IEEE Conference on*, pp. 281–288, IEEE, 2011.
4. G. Bosc, P. Tan, J.-F. Boulicaut, C. Raissi, and M. Kaytoue, "A Pattern Mining Approach to Study Strategy Balance in RTS Games," *IEEE Transactions on Computational Intelligence and AI in Games*, pp. 1–1, 2015.
5. O. Cavadenti, V. Codocedo, J.-F. Boulicaut, and M. Kaytoue, "When cyberathletes conceal their game: Clustering confusion matrices to identify avatar aliases," in *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, IEEE, oct 2015.
6. K. Conley and D. Perry, "How Does He Saw Me? A Recommendation Engine for Picking Heroes in Dota 2," tech. rep., 2013.
7. T. Nuangjumnonga and H. Mitomo, "Leadership development through online gaming," in *19th ITS Biennial Conference: : Moving Forward with Future Technologies: Opening a Platform for All*, (Bangkok), pp. 1–24, 2012.

8. N. Pobiedina and J. Neidhardt, "On successful team formation," tech. rep., 2013.
9. F. Rioult, J.-P. Métivier, B. Helleu, N. Scelles, and C. Durand, "Mining Tracks of Competitive Video Games," *AASRI Procedia*, vol. 8, no. Secs, pp. 82–87, 2014.
10. A. Drachen, M. Yancey, J. Maguire, D. Chu, I. Y. Wang, T. Mahlmann, M. Schubert, and D. Klabajan, "Skill-based differences in spatio-temporal team behaviour in defence of the Ancients 2 (DotA 2)," *Games Media Entertainment (GEM), 2014 IEEE*, vol. 2, no. DotA 2, pp. 1–8, 2014.
11. C. Eggert, M. Herrlich, J. Smeddinck, and R. Malaka, "Classification of Player Roles in the Team-Based Multi-player Game Dota 2," vol. 9353 of *Lecture Notes in Computer Science*, pp. 112–125, Cham: Springer International Publishing, 2015.
12. A. Agarwala and M. Pearce, "Learning Dota 2 Team Compositions," tech. rep., Stanford University, 2014.
13. K. Song, T. Zhang, and C. Ma, "Predicting the winning side of DotA2," tech. rep., Stanford University, 2015.
14. K. Kalyanaraman, "To win or not to win? A prediction model to determine the outcome of a DotA2 match," tech. rep., University of California San Diego, 2014.
15. N. Kinkade, L. Jolla, and K. Lim, "DOTA 2 Win Prediction," tech. rep., University of California San Diego, 2015.
16. F. Johansson, J. Wikström, and F. Johansson, *Result Prediction by Mining Replays in Dota 2.* PhD thesis, Blekinge Institute of Technology, 2015.
17. S. Rendle, "Factorization machines," in *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, (Washington, DC, USA), pp. 995–1000, IEEE Computer Society, 2010.
18. C. Freudenthaler, L. Schmidt-Thieme, and S. Rendle, "Bayesian factorization machines," in *Workshop on Sparse Representation and Low-rank Approximation*, Neural Information Processing Systems (NIPS-WS), 2011.
19. I. Bayer, "fastfm: A library for factorization machines," *CoRR*, vol. abs/1505.00641, 2015.
20. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *ArXiv e-prints*, Mar. 2016.