Evaluation of expert knowledge in LSTM

Win Probability Prediction of Dota 2 using LSTM

Oscar Alsing, Simon Westerlind, Marcus Larsson
{*oalsing, swestrl, mlar5*}*@kth.se*

KTH Royal Institute of Technology
School of Computer Science and Communication
Stockholm, Sweden

Abstract. Highly-tuned expert systems have long been the state-of-the-art approach for data predictions. As of the introduction of Deep Neural Networks in Computer Science, research has indicated improved prediction accuracy using generic data representations without hand-crafted features from domain experts. In order to further investigate this claim, we use a Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) to predict the winner of Defense of the Ancients 2 matches using time-series with generic data representation and features extracted with expert knowledge. Our results shows improved model performance when using the generic data representation in favor of the hand-crafted features, and promising results in the prediction of Dota 2 matches using LSTM.

Keywords: Artificial Neural Networks, Deep Learning, Time-Series, LSTM, Recurrent Neural Networks

1 Introduction

Expert knowledge and manually crafted features have been the state-of-the-art model in machine learning for decades. However, modern research has indicated drastic improvements in predictions using generic data representations and Deep Convolutional Neural Networks (CNN) in comparison to historical networks with hand-crafted features [1]. Furthermore, CNNs have been proven to be able to learn very specific classification tasks using raw data on huge general-purpose data sets [2].

1.1 Problem statement and hypothesis

The purpose of this paper is to evaluate the win probability prediction of Defense of the Ancients 2 (Dota 2) [3] matches, using a Long Short-Term Memory Recurrent Neural Network (LSTM) to analyze the match data from the first five minutes, as LSTMs have previously been suggested for analyzing time series data [4]. More specifically, we are to evaluate how well an LSTM performs on generic temporal time-series data in comparison to processed data containing hand-crafted features originating from domain expertise.

Hopefully, our LSTM will be able to quite reliably predict the game outcome based on the actions and the information from the states of the first 5 minutes of the game, where a typical game is approximately 45 minutes.

1.2 Scope and objectives

In this paper we have evaluated the Dota 2 game outcome prediction performance of an LSTM using generic data, and data containing hand-crafted features in order to compare the difference in performance. We've limited our input data to the data available through the data set as described in *Section 2.3*, and the hand-crafted features are limited to the scope of items data, as described in *Section 3.2*.

2 Background

2.1 Related work

Previous research have proven that a CNN is capable of generating input data to an SVM using generic data instead of generating input data using hand-crafted features, and therefore outperforming state-of-the-art systems [5]. This have also been the case in research on fully automated LSTM models [6] [7].

Predicting Dota 2 match outcomes using Machine Learning algorithms have been the subject of earlier researcher, with promising results. Most research have focused on the selection of heroes [8, 9, 10] and the impact of the game outcome analyzing hero compositions, but there also exists research on temporal data [11].

2.2 Defense of the Ancients 2 data

Dota 2 is an multiplayer online battle arena (MOBA) video game developed by Valve Corporation. The game is played in teams of five, where two teams are matched to defend their occupied area against the matched team, and the primary objective is to destroy the core enemy base known as the *Ancient*.

Each player chooses one of the 113 unique virtual character, known as a *hero*, in the beginning of the game, where each hero possesses the three shared basic attributes *Strength*, *Agility* and *Intelligence*, and at least four unique skills. Players are not able to choose a hero already chosen by another player. Due to the uniqueness of each hero, the style of play for optimal hero performance varies greatly between different heroes, and is the origin of the immense difficulty and complexity of high-end competitive gameplay.

During the game, each player collects *experience* and *gold*, where the later is used to purchase *items*. The experience collected increases the heroes general experience level, thereby unlocking new skills and increasing the power of existing skills of the hero. Gold obtained through the game is used to purchase items,

Fig. 1. Dota 2 gameplay [12].

that are used to either increase hero attributes or give access to unique item skills. As of today, there are 215 unique items available. Procurement of the right items and item combinations as well as the accumulation of general hero experience and the gold required to procure these items are crucial to increase the general power of the hero.

2.3 Data

The *Dota 2: Win Probability Prediction* [12] competition was presented in Kaggle in early 2016, where Data Scientists were asked to predict the winner of Dota 2 matches by analyzing the first 5 minutes of the game. As the competition was launched, another data set [13] almost as comprehensive as the competition data set was also presented. This data set contains multiple tables and consists of in-game events such as hero kills, item purchases and group fight outcomes from 50000 *ladder* (ranked).

In this data set, some events such as item purchases and player health status after a team fight are recorded with time stamp, and some tables are represented as minute wise snapshots covering certain aspects of the game, such as experience gained, gold acquired, enemy kills and player deaths for each player of the game. There is also non-temporal data in the data set, such as the hero selection of each player.

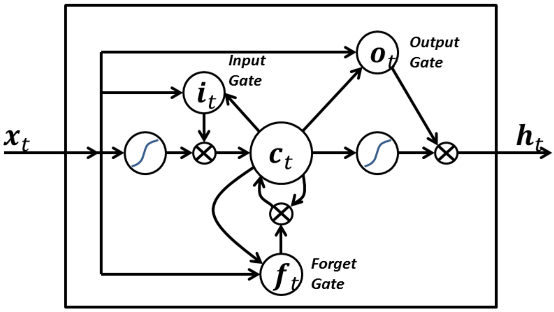
In this paper we aim to predict the win probability for the teams by using data from the first five minutes of the game.

2.4 Long Short-Term Memory Recurrent Neural Networks

LSTM Neural Networks are of Recurrent Neural Networks architecture, and are well-suited to learn from experience and to predict time-series. Their key advantage to traditional RNN architectures is the fact that they don't use an activation function within their recurrent layers, inherently not squashing stored values over time and therefore not suffering from vanishing gradient. As of this, LSTM units excel at remembering values for shorter and longer periods of time [14].

Multiple LSTM units are most often implemented together, forming an LSTM block. Each block contains gates, each controlling a specific flow of information - the *input*, *target* and *output* gate, with their corresponding activation functions. The input gate controls the extent to which new values flows into the memory, the target gate controls how much of the stored value remains in the memory, and the output gate controls to which extend values are used to compute the output activation of the block. See *Figure 2* for a graphical representation.

Fig. 2. Long Short-Term Memory [15].



3 Approach

3.1 Data preprocessing

Due to the curse of dimensionality, using all of the available data without preprocessing would not be an reasonable approach to our presented problem. In order to generalize and reduce the input space of our model, we decided not to represent the hero choice of each player as unique to that player slot, but rather represent the hero choices of the team as a 5-hot representation. It would be unrealistic to assume that the random slot a player is assigned to within the team has any correlation to the probability of that team winning. In order to enhance the training process, both the generic data representation and the data that was manually hand-crafted was normalized with $\mu = 0$ and $\sigma = 1$.

As the data origins from various files, all relevant attributes have been manually extracted, normalized and merged together to one tensor.

3.2 Expert knowledge

As earlier mentioned, Dota 2 is a video game with immense complexity, and expert knowledge would have been of utter importance in order to achieve predictions with decent accuracy prior to the wide use of Deep Learning algorithms.

Only considering the choice of heroes, and the number of unique items purchasable by the players in total, and not the fact that players can buy most items an unlimited number of times, the number of unique hero lineups and items purchases are as described in *Equation 1* and *2*.

$$\frac{113!}{103!} \approx 2.25 \cdot 10^{20} \tag{1}$$

$$215^{10} \approx 2.11 \cdot 10^{23} \tag{2}$$

The features to be manually-crafted are not intuitive by nature, and therefore we've consulted a Dota 2 expert, who've guided us through the most important aspects of the game in terms of extracting game events and compose these to hand-crafted features.

Manually hand-crafting expert features, we've decided to focus on the various items available for purchase, as item choices are more difficult to represent efficiently in our data, and the fact that the choice of heroes is limited to five in each team and therefore do not suffer from the curse of dimensionality to the same extent. When creating the hand-crafted labels, all of the 215 unique items have been summarized in 36 different feature categories, as seen in *Table 1*. Please see [16][17][18] for further explanation of hero and items attributes.

Table 1. Hand-crafted expert knowledge Dota 2 features

Category	Description	Example Attributes
1-3	Primary Attributes	Strength, Agility & Intelligence
4-15	Extra Attributes	Health Points, Mana, Health Regeneration, Damage, Damage Block
16 - 28	Special Attributes	Crowd Control, Stealth, Immunity, Blink, Active Buffs
29 - 36	Consumables	Wards, Town Portal Scroll, Gems, Courier

The summarized features provides a compact representation of the various items available for purchase, and their underlying game mechanics features are represented together. Furthermore, the hand-crafted item data compare each item to a base item within it's subtype, and represents each item with it's percentage difference in item *powerfulness* versus the base item.

3.3 Success determination

The success determination is split into two categories. Firstly, we are to evaluate the performance of the LSTM classifier with generic data representation, and the performance of the classifier using the hand-crafted features.

In order to evaluate the difference between the data representations, both of the data sets will be used for training on a variety of LSTM architectures, and their performance will be compared accordingly.

Secondly, the overall performance of the best performing LSTM classifier as a general mean to predict the outcome of Dota 2 matches, based on temporal data from the first five minutes of the match has to be evaluated. Naturally, there are multiple ways to explore this evaluation, and we've decided to take the following benchmarks as evaluation methods:

- Kaggle Dota 2: Win Probability Prediction Leaderboard [19].
- LSTM performance on temporal data in comparison to MLP performance on the game state of the fifth minute.

The leaderboard for the official competition serves as an indicator for the overall success of our predictions, but is not directly comparable as the data available in the official competition is more comprehensive and with a higher degree of temporality than the data available to us. For example, the competition data includes hero pick & ban order, hero kill log time and where objectives were completed - with corresponding time stamps.

Furthermore, if we make use of temporal data and therefore outperform the analysis of the last time frame only, our LSTM implementation successfully analyzes Dota 2 matches utilizing the data temporality.

3.4 LSTM implementation

In order to implement our LSTM with ease, and to try out various configurations, we implemented our network using Keras [20] running a CUDA [21]-powered Tensorflow [22] backend.

The 50000 matches where split into training, validation and test data sets using optimal ratios from earlier research [23]. The final data sets consists of 28000 matches for training, 10000 matches for validation and 12000 matches for testing.

The search for an optimal LSTM architecture began with an initial 1-layer LSTM model. Thereafter, we continuously incremented the complexity of the model while carefully monitoring the accuracy and loss values on the validation set. During this process, SGD, Adagrad, RMSprop were used as optimizers interchangeably and independently evaluated.

As the the complexity of the initial LSTM model increased, experiments with multiple stacked LSTM blocks and fully connected ReLU layers between the LSTM layer and the output layer were evaluated.

The optimal network architecture and hyper-parameter settings were found using random search [24] and training the network models for only 10 epochs.

3.5 Input space

In order to determine how to utilize the raw data, numerous different versions of the input space were tested on the various models. Our experiments concluded that the generic input as described in *Equation 3* was optimal.

$$input = \underbrace{\overbrace{xp,\ gold\ and\ last\ hits}^{\text{per player}}}_{\text{normalized}} + \overbrace{items\ purchased}^{\text{per team}} + \overbrace{objectives}^{\text{per team}} + \overbrace{heroes\ chosen}^{\text{5-hot per team}} \tag{3}$$

where the addition symbolizes concatenation. The total number of features in the generic data representation was 704. Similarly the input space with hand-crafted item features (expert data) was total 386 features and structured as described in *Equation 4*.

$$input = \underbrace{\overbrace{xp,\ gold\ and\ last\ hits}^{\text{per player}} + \overbrace{item\ attributes}^{\text{per team}}}_{\text{normalized}} + \overbrace{objectives}^{\text{per team}} + \overbrace{heroes\ chosen}^{\text{5-hot per team}} \tag{4}$$

Whereas the input for training a MLP was only the state at the end of the fifth minute with no temporal data (static data), residing in a total of 256 features, as described in *Equation 5*.

$$input = \underbrace{\overbrace{xp,\ gold\ and\ last\ hits}^{\text{per player}}}_{\text{normalized}} + \overbrace{heroes\ chosen}^{\text{5-hot per team}} \tag{5}$$

An explanation of the summands can be found in *Table 2*. For the LSTM, the input tensors were fed into the network at each time step. Ideally, all of the data would have been per player, but such an input space is approximately 5 times larger and also inappropriately sparse. If such a per-player input space were to be used, it would however be possible be to synthesize additional matches by shuffling the player slots within the teams. By exploiting this possibility, we could increase the amount of matches by a factor of $5! \times 5! = 14400$, and therefore acquire additional training data. This was not implemented in this report, in order to limit our research.

4 Experiments

In order to evaluate the impact of expert knowledge in our prediction problem, and to evaluate and optimize our models, we conducted a series of experiments as presented in *Section 4.1*. Initially, the focus of our experiments was to optimize our LSTM models and inherently achieving as good results as possible. Later, the models were evaluated on the generic data representation and the data representation generated through expert knowledge, as presented in *Section 4.2*.

4.1 Random hyper-parameter search

Model optimizations were conducted as a series of coarse random hyper-parameter searches. The networks were trained for 10 epochs during the coarse search and

Table 2. Input features

Name	Description	Representation
xp	Experience gained by a player	A number normalized over each time frame.
gold	Gold earned by a player	A number normalized over each time frame.
last hits	Number of non-player characters killed by a player	A number normalized over each time frame.
items purchased	The items purchased by a player this time step	A list of same length as the number of unique items. When an item is purchased, the corresponding list position of that item id is incremented.
objectives	First blood, Tower kills, Tower denies & Roshan kills	A list of length four for each team, where each position represents one of the four main objectives. If an objective is completed during the time step, the corresponding position of that objective in the list is incremented.
heroes chosen	The heroes the different teams have chosen	A list of same length as the number of unique heroes, where each hero represents as unique position in the list. All numbers in the list are set to zero but for the positions corresponding to the chosen heroes of that team, which are set to one.
item attributes	The attributes gained from purchasing items	A list of numbers as long as the number of unique attributes where the attributes on items bought is summed into its corresponding position in the list. Thereafter the list is normalized for each attribute over time frames.

the included parameters were *Batch Size*, *size of LSTM layer*, *L2 regularization*, *Learning Rate* and *size of dense layer*.

Following the initial coarse search, we refined the top-performing model by altering minor values in the hyper-parameters, and added further regularization where tendencies of overfitting were indicated.

Following our search for the optimal hyper-parameter settings for our models, our results clearly indicated better performance in smaller LSTM models, and they were inherently used in favor of larger models. The optimal hyper-parameter and model size settings are presented in *Table 3*.

We decided not to utilize dropout as a regularization method in this application as L2 regularization proved more useful on the numerous experiments conducted. This was the case regardless of the size of the model and the settings of our hyper-parameters. This was quite unexpected, as dropout has proven to be a successful regularization method in other research [25].

Table 3. Optimal network architecture hyper-parameters

ID	Data set	Batch Size	LSTM Layer Size	Dense Layer Size	L2	Learning Rate
0	Expert data	256	191	1160	0.0436	0.000425
1	Generic data	512	69	567	0.0813	0.001349
2	Generic data	256	596	NA	0.0198	0.000447
3	Static data	100	NA	500	0.0164	0.00137

4.2 Network performance

Given the optimized models from earlier experiments, these were used for the final classification on the test set, and the results are presented in *Table 4*. In order to refine our results, we decided to also include the performance of the network where the outcome probability is $> 60\%$ - as it would be a sensible limitation as any probability below 60% essentially is a random guess. With this limitation, some matches would be omitted from the prediction, but never more than 25%. These numbers are represented as the confidence metrics in *Table 4*.

Table 4. Model performance

ID	Test accuracy	AUC	Confidence test accuracy	Confidence AUC
0	65.02%	0.7081	72.64%	0.7629
1	65.72%	0.7119	72.35%	0.7631
2	63.95%	0.6958	69.73%	0.7341
3	64.4%	0.7028	70.94%	0.7522

5 Results

The results from our experiments shows promising performance for both the generic and the hand-crafted features data representation. As seen in *Table 4*, the top-performing model is the one using the generic data representation, with 65.02% accuracy on the entire test set, and 72.64% accuracy on the test set with a probability $> 60\%$.

The training progress of model ID 0 is illustrated in *Figure 3*. It can be seen that after about 50 epochs the model starts to overfit on the train set, where the validation and train loss diverge. This may indicate that there is not enough regularization in the model. Although higher values of regularization was tested during the random search for hyper-parameters, the training during the random searches did not exceed 20 epochs. For the final test, the network was trained for 100 epochs with the parameters that did perform best during the random search, and the model weights that got the best validation accuracy was picked to evaluate the test set. Therefore, the overfitting that can be seen in *Figure 3* is not a problem.

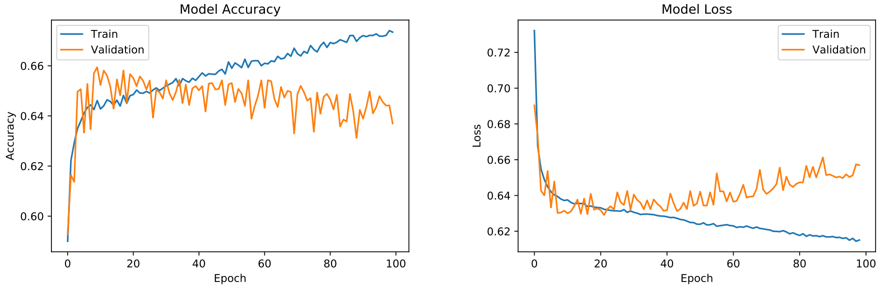


Fig. 3. Performance of expert model ID 0.

For model ID 1 the same type of overfitting was not present during the training. The progress of accuracy and loss can be seen in *Figure 4*. The train accuracy seems to stop increasing and this is the result of a much higher regularization term. It was found in the random search for hyper-parameters that model ID 1 with generic input data needed a higher regularization term to perform better. However, the random search did not train for more than 20 epochs where this problem was still not visible. The accuracy and loss of validation in *Figure 4* also seemed to be very unstable, whilst the train accuracy and loss was quite stable. This indicate that it was harder for the network to generalize on the generic data where the input space was a lot larger. However, we still achieved slightly better results with the generic data. Picking the model weights from the epoch with the best validation accuracy still helped to generalize to the test set.

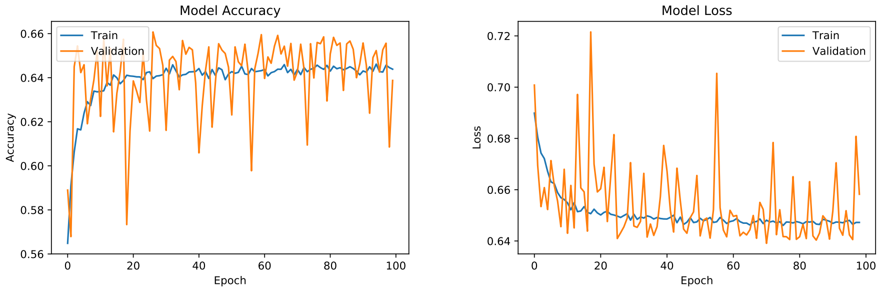


Fig. 4. Performance of generic model ID 1.

There are many possible reasons as to why the generic data representation performs better even though the dimension is much higher, the most probable one being loss of information as it is condensed to the more compact hand-crafted representation, which could alter the possible abstractions available for our model. Furthermore, the hand-crafted features could simply not be good enough, as producing these in a good manner is a tedious task which requires specific expertise skills. Evaluating hand-crafted features could further benefit from methods such as PCA.

Given the limited data in comparison to the official data of the Dota 2 competition [12], our model performs well. Given that confidence AUC score of 0.7629, we would place fairly high on the official leaderboard. Though, this comparison is not entirely fair, as the confidence AUC includes some filtering - but serves as an indicator of success.

The LSTM implementation outperforms the performance of the MLP, where the MLP only analyzes the static game state from the fifth minute of the game without any temporal data, as seen in *Table 4*. The difference in performance between the MLP and the LSTM model is quite modest with the generic data set, but as earlier mentioned the temporality in the data set is quite limited as well. Given a data set in a more temporal fashion, the difference in performance would most likely increase.

6 Conclusions

While both the generic data representation, and the hand-crafted data representation produce comparable results, it can be concluded from our results that the effort of producing hand-crafted features did not improve the performance of our models - on the contrary, the generic data representation performed better.

Our research indicates that a generic data representation is preferred over hand-crafted features when predicting the outcome of Dota 2 matches using an LSTM with the data from the five first minutes, which further reinforces claims in earlier research on the subject. Furthermore, the resulting models from our experiments proved themselves to be good predictors of the outcome of the matches based on data from the first five minutes.

References

[1] Grigory Antipov et al. “Learned vs. Hand-Crafted Features for Pedestrian Gender Recognition.” In: *ACM Multimedia*. Ed. by Xiaofang Zhou et al. ACM, 2015, pp. 1263–1266. ISBN: 978-1-4503-3459-4. URL: `http://dblp.uni-trier.de/db/conf/mm/mm2015.html#AntipovBRD15`.

[2] Ken Chatfield et al. “Return of the Devil in the Details: Delving Deep into Convolutional Nets”. In: *CoRR* abs/1405.3531 (2014). URL: `http://arxiv.org/abs/1405.3531`.

[3] Valve. *Dota 2*. 2017. URL: `http://blog.dota2.com/` (visited on 05/11/2017).

[4] John Cristian Borges Gamboa. “Deep Learning for Time-Series Analysis”. In: *CoRR* abs/1701.01887 (2017). URL: `http://arxiv.org/abs/1701.01887`.

[5] Ali Sharif Razavian et al. “CNN Features off-the-shelf: an Astounding Baseline for Recognition”. In: *CoRR* abs/1403.6382 (2014). URL: `http://arxiv.org/abs/1403.6382`.

[6] Moez Baccouche et al. “Sequential Deep Learning for Human Action Recognition”. In: *Human Behavior Understanding: Second International Workshop, HBU 2011, Amsterdam, The Netherlands, November 16, 2011. Proceedings*. Ed. by Albert Ali Salah and Bruno Lepri. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 29–39. ISBN: 978-3-642-25446-8. DOI: `10.1007/978-3-642-25446-8_4`. URL: `http://dx.doi.org/10.1007/978-3-642-25446-8_4`.

[7] Linmei Hu et al. *What Happens Next? Future Subevent Prediction Using Contextual Hierarchical LSTM*. 2017. URL: `https://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14324`.

[8] Adam Summerville, Michael Cook, and Ben Steenhuisen. *Draft-Analysis of the Ancients: Predicting Draft Picks in DotA 2 using Machine Learning*. 2016. URL: `https://www.aaai.org/ocs/index.php/AIIDE/AIIDE16/paper/view/14075/13619`.

[9] Kaushik Kalyanaraman. “To win or not to win? A prediction model to determine the outcome of a DotA2 match”. In: (2014). URL: `https://cseweb.ucsd.edu/~jmcauley/cse255/reports/wi15/Kaushik_Kalyanaraman.pdf`.

[10] Kevin Conley and Daniel Perry. *How Does He Saw Me? A Recommendation Engine for Picking Heroes in Dota 2*. 2013. URL: `http://cs229.stanford.edu/proj2013/PerryConley-HowDoesHeSawMeARecommendationEngin` `pdf` (visited on 05/15/2017).

[11] Nicholas Kinkade and Kyung Yul Kevin Lim. *DOTA 2 Win Prediction*. 2015. URL: `https://cseweb.ucsd.edu/~jmcauley/cse255/reports/fa15/018.pdf` (visited on 05/15/2017).

[12] Kaggle. *Dota 2: Win Probability Prediction*. 2016. URL: `https://inclass.kaggle.com/c/dota-2-win-probability-prediction` (visited on 05/11/2017).

[13] Devin Anzelamo. *Dota 2 Matches*. 2016. URL: `https://www.kaggle.com/devinanzelmo/dota-2-matches` (visited on 05/11/2017).

[14] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162%2Fneco.1997.9.8.1735.

[15] BiObserver. *Long Short-Term Memory*. 2017. URL: https://en.wikipedia.org/wiki/Long_short-term_memory#/media/File:Long_Short_Term_Memory.png (visited on 05/11/2017).

[16] Gamepedia. *Items*. 2017. URL: http://dota2.gamepedia.com/Items (visited on 05/15/2017).

[17] Gamepedia. *Table of hero attributes*. 2017. URL: http://dota2.gamepedia.com/Table_of_hero_attributes (visited on 05/15/2017).

[18] Gamepedia. *Table of attributes*. 2017. URL: http://dota2.gamepedia.com/Attributes (visited on 05/15/2017).

[19] Kaggle. *Public Leaderboard - Dota 2: Win Probability Prediction*. 2016. URL: https://inclass.kaggle.com/c/dota-2-win-probability-prediction/leaderboard (visited on 05/15/2017).

[20] François Chollet. *Keras*. 2017. URL: https://keras.io/ (visited on 05/11/2017).

[21] Nvidia. *CUDA*. 2017. URL: http://www.nvidia.com/object/cuda_home_new.html (visited on 05/11/2017).

[22] Google. *Tensorflow*. 2017. URL: https://www.tensorflow.org/ (visited on 05/11/2017).

[23] Mohamed A. Shahin Holger R. Maier Mark B. Jaksa. "Data Division for Developing Neural Networks Applied to Geotechnical Engineering". In: *Journal of Computing in Civil Engineering* 18.2 (Issue: object: doi:10.1061/jccee5.2004.18.issue-2, revision: rev:1479257530531-16244:doi:10.1061/jccee5.2004.18.issue-2,), pp. 105–114. DOI: 10.1061/(ASCE)0887-3801(2004)18:2(105). URL: http://ascelibrary.org/doi/abs/10.1061/%28ASCE%290887-3801%282004%2918%3A2%28105%29.

[24] James Bergstra and Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". In: *Journal of Machine Learning Research* 13.13 (Feb. 2012), pp. 1735–1780. URL: http://www.jmlr.org/papers/volume13/bergstra12a/bergstra12a.pdf.

[25] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *J. Mach. Learn. Res.* 15.1 (Jan. 2014), pp. 1929–1958. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=2627435.2670313.

Appendix A

Hero pick	Teams choose heroes in turn. Hero pick is when one team is to choose their next hero.
Hero selection	Teams choose heroes in turn. Hero selection is the choice of the next hero.
Ban order	Teams ban heroes in turn. Ben order is the order of which teams ban heroes.
Hero kill	When one player kills another players hero.
First blood	The first hero kill of them game. Rewards extra gold.
Ladder	Ranked matches, where the rating of the players are altered depending on the game outcome.
Group fight	When at least three players are in a fight with each other.
Objective	Goal or subgoal of the game.