Louisiana Tech University
Louisiana Tech Digital Commons

Mathematics Senior Capstone Papers
College of Engineering and Science

Spring 2019

# Predicting Win Rates in Competitive OverwatchTM

Andrea Sibley

# Predicting Win Rates in Competitive Overwatch™

Andrea Sibley

May 8, 2019

Project Advisor: Samira Fazel Anvaryazdi [1]

**Abstract**

Overwatch™ is a video game published by Blizzard Entertainment® where two teams comprised of six people each compete against one another to accomplish a specific goal. The goal of each game is dependent on which map is being played. The maps are divided into four categories: Assault, Escort, Control, and Hybrid. A data set comprised of 3000 games of competitive Overwatch™ is used to determine how likely a team is to win their match. The factors used to determine the likelihood of winning are the map type and the skill ranking for each team. The data set is pre-processed by standardizing and encoding the data through Python. After the data is encoded, 80% of the data is divided into a training set and 20% of the data is divided into a testing set. Classification algorithms are tested against the data to determine which classifying method returns the highest accuracy. After using the training set, the Bagging Classifier shows the highest accuracy when compared to the testing set.

*Keywords:* Overwatch™, Machine Learning, Bagging Classifier, Multiple Linear Regression, Win Rates, Support Vector Classification, K-Nearest Neighbors, Naive Bayes

## 1 Introduction

Overwatch™ is a video game created by Blizzard Entertainment®. This game is a first person shooter, or FPS, where the player sees through the eyes of one of the characters. It is comprised of two teams of six players, each team competing to get the highest number of points before the game is over. The goal is to work with team members to eliminate or repel opponents while attacking, defending, or competing for an objective.

[1] A thank you to Shashank Reddy Vadyala for supplying knowledge in machine learning.

In Overwatch™, there are four different categories of games: escort, assault, control, and hybrid. These are the different map types that will be considered in the data. Each team is also assigned a Skill Ranking, or SR. The SR is a number determined in game and will increase or decrease depending on the performance of the team. In the data, SR Delta corresponds to the difference between each team's SR. The data is from 3,000 different competitive matches of Overwatch™. Blizzard Entertainment® has created an Overwatch League™ where teams compete on the professional level. While there is currently not enough data on the professional matches to perform analysis on the Overwatch League™, it does indicate the game's popularity.

Overwatch™, and e-sports (electronic sports), are growing in popularity and demand. The Overwatch League™ Season One Grand Finale had record breaking viewership. The inaugural season of the Overwatch League™ had more viewers in the 18-34 age range than any other major sporting event, such as the Super Bowl and the NBA Finals. [3]

Even though Overwatch™ is a game with a large fan-base and a professional presence, the game has not been given the same amount of research and attention as other games, such as Dota 2, have been given. Many sports, both traditional and electronic, have some sort of bracket or betting system for the fans of that sport. A popular example of this is the March Madness bracket for the National Collegiate Athletic Association, or NCAA.

While Overwatch™ is becoming increasingly popular, there is a lack of research that corresponds to the game. Different video games, such as Dota 2, have had similar research performed on them. When looking for literature that focuses specifically on Overwatch™, there are only a couple of articles on this game. Specifically, the most prevalent article focuses on why lag may happen in a game of Overwatch™ [2]. Other articles that focus on similar video games have looked into more machine learning or classifying aspects. [1] looks into predicting an opponents strategy in a game called StarCraft®, another game produced by Blizzard Entertainment®. Similarly, [4] looks into which machine learning algorithm is the most effective in predicting the outcome of a game of Dota 2. Since there has been similar research done for games other than Overwatch™, the contribution of this paper is to bridge the gap between the machine learning approach used to predict win rates in other video games and the lack of research done on Overwatch™. This paper will take a data science approach on Overwatch™ and use machine learning to determine if it is possible to accurately predict the win rate of a competitive Overwatch™ game using the map type and the difference between the teams' skill ranking, denoted SR Delta.

We review the three main steps for a data scientist to consider when approaching a new problem. The first step is to decide the goal of the prediction. The second step is where the bulk of the work for a data scientist lies. Here, the features that affect the prediction are determined. The features will be reviewed for how strong their effect on the prediction is, and based on that effect the features will be selected. Next, the third step is to decide on an appropriate model for prediction. This step is mostly trial through different models until a specific model is chosen. Then, the model will be chosen based on the highest accuracy for the expected prediction.

A data set of 3,000 competitive Overwatch™ games with a map type column and a SR delta column is used to determine if it is possible to predict the winner of a competitive match when given the SR delta and the map type. Once the data is readable to Python, different classifying methods are used to determine which classifier has the highest accuracy when predicting the winner.

The goal of the research is to predict the winner of an Overwatch™ match, using both statistical and collective knowledge taken from the data set on competitive Overwatch™ matches. First, the standard data steps of pre-processing for the competitive Overwatch™ data is done. Next, a method for accurately determining the win rate of matches through the use of machine learning model is determined. A probabilistic machine learning method learns about the relationship between relevant match factors, Map Type and SR Delta, from 3,000 matches. The known factors of the Overwatch™ matches are then entered into the trained machine learning to determine a prediction for each match. Thus, an efficient prediction model through the use of Bagging Classifier method is produced. This paper is organized as follows: Section 2 reviews the relevant literature. Section 3 provides the problem description. Section 4 discusses models, multiple linear regression, and the different machine learning approaches. Section 5 provides data and results. Finally, the paper concludes in section 6.

## 2 Related work

Overwatch™ was released in 2016, and thus the game is relatively new and there has been little research done on Overwatch™. Dota 2 is a popular game that was released in 2013, so there have been several research papers written on Dota 2 utilizing methods that have yet to be applied to other video games.

For instance, [4] takes into account different types of machine learning algorithms and concludes which one can accurately predict the outcome of a single game. The authors look into the Naive Bayes classifier, Logistic Regression, Gradient Boosted Decision Trees, and Factorization Machines methods. Through looking into a publicly available data set and looking at different skill rankings of players, the authors found the Factorization Machines method with high accuracy, although the accuracy of each methodology changes with the skill ranking of the player.

Similarly, [5] uses a data set published by UCI Machine Learning Repository to find information of different matches of Dota 2. In this article the authors use the Naive Bayes Classifier method to analyze the data and to predict the outcome of the game. By analyzing the data with this classifier method, the accuracy rate is reported around 58.99%. While this accuracy rate is not high, the authors deem the Naive Bayes classifying method to be successful due to the lack of data and complexity of variables in a human played video game.

However, Dota 2 is not the only video game that has been analyzed in this method, although it is the most popular. Another Blizzard Entertainment game, StarCraft®, is reviewed in

[1]. StarCraft® is a real time strategy, or RTS. Although the authors use a model that is general enough to be applied to any RTS that has a similar structure to StarCraft®, they choose to analyze StarCraft® due to its popularity. The authors use the Bayesian Model to predict the opening move of the opponent in a match of StarCraft®.This paper uses this strategy to compare games played against both real opponents and bots.

Article [6] aims to predict the outcome of cricket games, namely of games in the Indian Premier League. They use a data set that shows several features relevant to a game of cricket, such as season, home team, away team, toss winner, man of the match, venue, umpires, referee, home team score, away team score. However, they do not use all of these features in their predictions. Scikit Learn machine learning library in Python is applied in order to apply feature selection. Their three steps for feature selection are to remove low variance features, univariate feature selection, and recursive feature elimination. Through these steps, the authors found the most important features to be the home team, away team, toss winner, toss decision, and stadium. Next, the data is organized and discussion about lack of data is presented. The authors then calculate a players points by using multiple linear regression. Then, a teams weightage is calculated and dummy variables are used in order to convert categorical variables, such as the home team. Moreover, Random Decision Forests is applied to reduce the uncertainty in the data. Finally, the modified data set is considered into a classifier to predict the win rates using Scikit Learn.

Articles [7] [8] use the Support Vector Classification method to classify the data. These papers allow a look into how Support Vector Classification is used and how the data should be structured in order for the classification to be successful. Similarly, [11] [12] use the Bagging Classifier method to classify the data. These papers use the Bagging Classifier, demonstrating what type of data this method needs and how successful the method is.

# 3 Problem description

As betting has arisen, there has been a need for both people and machines to predict the outcome of those games or matches. Machine learning is used to fine-tune prediction of win rates throughout sports, allowing betting and brackets to become more precise and to allow the fan-base to have a more exact knowledge of what to look for when viewing the games or matches. Despite the popularity of Overwatch™, there has not been any research or machine learning used to look into match predictions. This is the problem that this paper targets. The first step to machine learning, however, is to clean up the data. This is known as pre-processing the data.

Data pre-processing is the act of getting the data ready to be used. This is seen commonly when the data is not machine readable, and has to be converted from a string to a number. There are several ways to do this, but one example is by using Scikit Learn in Python. Depending on what the data is, Python can do Label Encoder or One Hot Encoder. Label Encoder takes the data and assigns each distinct string a value, starting at zero. The data is kept in the same format it is found in, all in one column. The problem with this is that when applied to machine learning, the machine may think that encoded values are in a certain pattern. One Hot Encoder gets rid of this issue as it takes each distinct value and makes a row just for that value. By doing this, the values do not appear to be in any pattern, and the data is machine readable.

After the data is encoded, it may need to be standardized to a smaller scale to get rid of interference in the data. This can also be done through Python. After the data has been encoded and standardized, it is ready to be processed through different methods such as linear regression. This approach is shown in Figure 1.

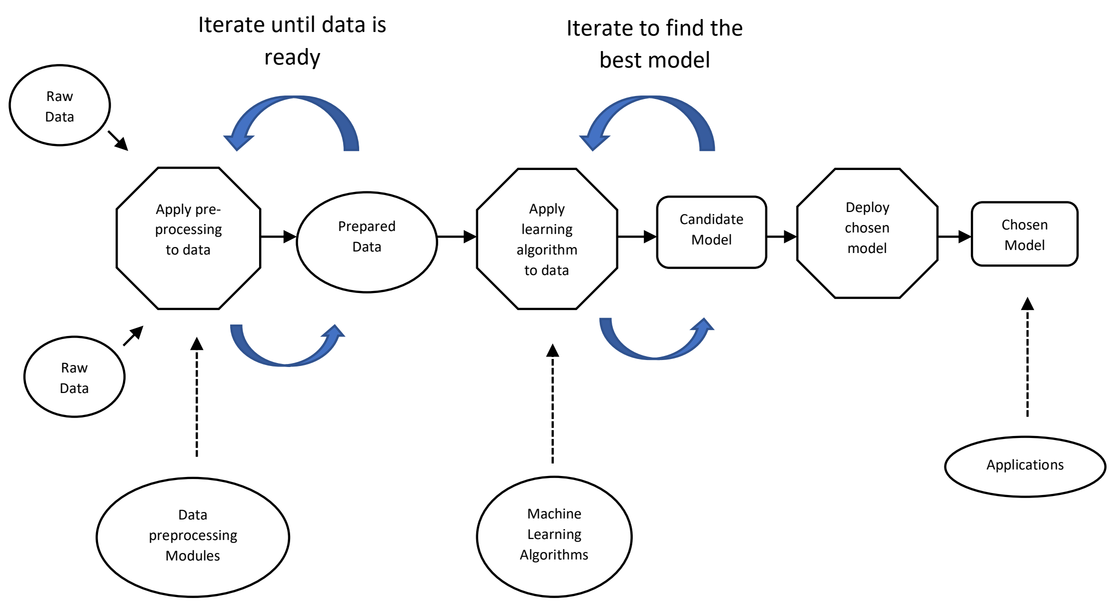


Figure 1: Overall Research Approach, Adapted from [10]

To standardize the data with normally distributed data, the Z-Score is calculated as shown in Equation 1.

$$Z = \frac{X - \mu}{\sigma} \tag{1}$$

In this formula, Z represents the Z score, X is the number that is being standardized, $\mu$ is the mean of the X values, and $\sigma$ is the standard deviation of the X values. In the data set used, the SR delta was standardized through this method.

The goal of the research presented in this paper is to predict the winner of Overwatch$^{TM}$, using both statistical and collective knowledge. The collective knowledge is found through a publicly available data set on competitive Overwatch$^{TM}$. After standard data pre-processing, the next step is to begin machine learning model building. The classification machine learning methods learn about the relationship between relevant match factors, maps and skill ranking, from 3000 matches data. The known factors of the Overwatch™ matches are then entered into the trained machine learning to determine a prediction for each match.

## 3.1 Approach

The data is first cleaned in a manner that makes it readable for Python. In order to do this, the Scikit Learn library in Python is used to apply Label Encoder to the Win Result column of the data set. By applying Label Encoder to this column, the distinct Win Result values are each assigned a distinct number.

After the Win Result column is encoded, the Map Type column is also encoded. There are two different methods used when encoding the data: Label Encoder and One Hot Encoder. Label Encoder assigns each unique value in the target column a dummy number, which is useful as it can convert a string into a machine-readable number. However, there are more categories to the map, so Label Encoder is not the best way to encode the data. Instead, One Hot Encoder is used. By using One Hot Encoder, a pivot table is created. One Hot Encoder takes every unique value and pivots it into a unique column.

Once every string variable is converted into a number, Python is able to read and use the data. However, it is not yet in its best form. While the SR Delta column is already in the integer datatype, the range of the numbers is large and arbitrary. In order to give more meaning to the SR Delta column, the column is standardized to fit a smaller range using the Z score.

Now, the data is ready to be analyzed with Python. The first approach to applying the machine learning methodology to the data is to ensure that the input columns for the data, the map and SR Delta, do have an affect on the output column, or the Win Result. This is done by performing a multiple linear regression test on the input columns.

Once the input columns are determined to be significant, the data is divided into two different sets, a testing set and a training set. The first 80% of the data is designated to test the machine learning algorithms, and the remaining 20% of the data is used to test the accuracy of each algorithm. Due to the nature of the data and how the structure is predetermined, the machine learning algorithms that are used are different types of classification in the supervised learning spectrum.

Using the Scikit Learn library in Python, four different types of classification algorithms are used in order to determine which algorithm provides the highest accuracy when determining the win result of a competitive Overwatch™ match. The different algorithms used are the Naive Bayes, K-Neighborhood, SVC, and Bagging Classifier methods.

# 4 Models

The two different approaches taken to analyze the competitive Overwatch™ matches are multiple linear regression and machine learning. For machine learning, K-Nearest Neighbors, Bagging Classifier, Support Vector Classification, and Naive Bayes Classification approaches have been used.

## 4.1 Multiple Linear Regression

Multiple linear regression determines the linearity of the relationship between multiple independent variables and one dependent variable. A relationship is determined to be insignificant if the $r^2$ is close to 0 and significant if the $r^2$ value is close to 1. The $r^2$ value is called the coefficient of determination as it is a metric used to determine the variation in the independent variables. The formula for multiple linear regression is shown in Equation 2.

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_n x_{in} + \epsilon, \tag{2}$$

where $y_i$:= output, $x_i$:= inputs, $\beta_0$:= y-intercept, $\beta_n$:= slope coefficient, and $\epsilon$:= error term

## 4.2 Machine Learning Approaches

Machine learning is a branch of artificial intelligence that relies on the thought that computers can learn from data and patterns, and then make decisions based on those patterns. Ultimately, machine learning is the automation of processing and predicting data. There are several different types of machine learning. For the purpose of this data, supervised learning with structured data is used. Because the data is structured in the way it is, different classification methods are used to see which algorithm can most accurately predict the win result of the testing data.

The first classification algorithm used on the data is the K-Nearest Neighbors algorithm. This algorithm does not make any assumptions on the structure of the data, so it is non-parametric. This algorithm uses a voting system, where each object is classified by the objects around it and how similar the features are to the training set. Because of this voting system, the K-Nearest Neighbors algorithm is susceptible to noise and interference. This algorithm looks at each object and compares it to the different classifications, seeing how many features it has in common with each classification. The classification with the most feature similarity to the object is the classification in which the object gets placed. Thus, the algorithm classifies an object based on how its neighbors are classified, leading to the name of the algorithm.

The Bagging Classifier methodology takes several sets of the training data and samples the input data. Each training set is randomly generated out of the data, leading to slightly different training sets. One run of the Bagging Classifier methodology takes multiple training sets and averages out its predictions found through classification. This allows it to correct its predictions while also running itself against multiple different sets of data. By taking multiple instances of the data, variance is minimized.

Support Vector Classification takes the data and arranges the $x$ values and the $y$ values into an array. This classification method finds a best fit hyper-plane that is dependent on the grouping of the objects. This algorithm divides each object into a specific classification depending on where the object falls in the hyper-plane. An example of this is seen in Figure 2, where there are two classification groups. This algorithm decides in which classification the object belongs depending on where it falls in the graph.

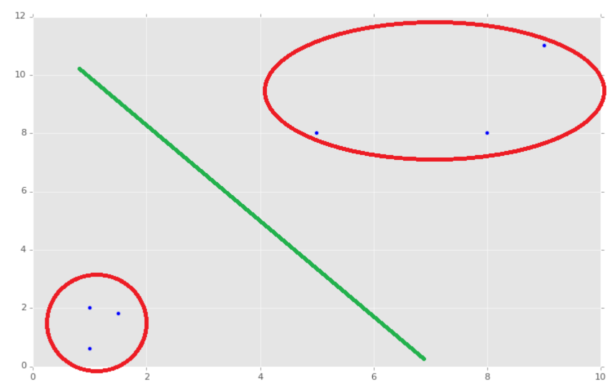


Figure 2: Support Vector Classification Visualized [9]

The Naive Bayes Classification method assumes each feature is independent of all the other features. This algorithm is called Naive Bayes as it builds off of Bayes Theorem of Probability. This theorem is mathematically described by Equation 3.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{3}$$

where: $P(A|B)$:= likelihood of event A occurring given that event B is true, $P(B|A)$:= likelihood of event B occurring given that event A is true, $P(A)$:= probability of event A, and $P(B)$:= probability of event B.

This algorithm calculates the $P(A|B)$ for each classification and then determines to which classification the object belongs by the highest probability. Due to the efficiency of Naive Bayes, it is often considered a good algorithm for large sets of data.

There are three result categories, win, loss, and draw that the machine learning predictions can make. For each category the prediction data is divided into precision, recall, f1-score, and support. Precision is the percentage of the accurate results to the total results. This can be written as shown in Equation 4.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}, \tag{4}$$

where true positives are results that were both predicted to be true and were actually true. False positives are results that were predicted to be true but were actually false.

Similiarly, Recall is the percentage of the accurate results to the predicted results. This can be written as shown in Equation 5.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}, \tag{5}$$

where false negatives are results that were both predicted to be false and were actually false.

The F1-Score is the harmonic mean of the precision and recall. This can be written as shown in Equation 6.

$$\text{F1-Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} \tag{6}$$

# 5 Data and Results

In the first stream of this section the data is reviewed in detail. Next, in the result section we review multiple linear regression and machine learning approaches. Finally, the most accurate methodology is decided.

## 5.1 Data

The data set is a publically available date set found at [13]. Each row of this data corresponds to a match played by two random teams. After cleaning the data to only include the information needed, the columns left are "Result," "Delta Standardized," "map_Assault," "map_Control," "map_Escort," and "map_Hybrid." For each row, there is only one "map_" column that is populated with a true value, 1, and all the rest are populated with false, 0. Similarly, the "Result" column is either populated with "0," "1," or "2," which are draw, loss, or win respectively. An example of the data set is seen in Figure 3.

| result | delta_standardized | map_Assault | map_Control | map_Escort | map_Hybrid |
|---|---|---|---|---|---|
| 2 | -3.896279236 | 0 | 0 | 1 | 0 |
| 2 | -3.896279236 | 0 | 0 | 1 | 0 |
| 2 | -3.309156543 | 1 | 0 | 0 | 0 |
| 2 | -3.309156543 | 1 | 0 | 0 | 0 |

Figure 3: Data Set Example

## 5.2 Results

The calculations were carried out on a Dell, 64-bit operating system, and 16 GB RAM. The solution scheme is implemented in Python 2.7.12.

### 5.2.1 Multiple Linear Regression

To determine how game results are affected by the type of maps, multiple linear regression has been applied. In the model "Delta Standardized," "map_Assault," "map_Control," "map_Escort," and "map_Hybrid" are selected as independent variables against the game results. The model demonstrated accuracy with $r^2$ values of 0.024. In other words, 2.4% of change in the game result would be explained by model these characteristics which demonstrates that multiple linear regression is not a precise way to classify the data. Figure 4 shows the coefficients obtained from the multiple linear regression study. The result justifies all factors have a significant impact on game results.

```
==============================================================================
Dep. Variable:                 result   R-squared:                       0.024
Model:                            OLS   Adj. R-squared:                  0.022
Method:                 Least Squares   F-statistic:                     14.36
Date:                Tue, 12 Feb 2019   Prob (F-statistic):           6.44e-14
Time:                        20:55:22   Log-Likelihood:                -2745.8
No. Observations:                2929   AIC:                             5504.
Df Residuals:                    2923   BIC:                             5539.
Df Model:                           5
Covariance Type:            nonrobust
======================================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------
const                 -0.9989      0.437     -2.284      0.022      -1.856      -0.141
map_Assault            2.2959      0.438      5.243      0.000       1.437       3.154
map_Control            2.5107      0.439      5.725      0.000       1.651       3.371
map_Escort             2.4197      0.438      5.526      0.000       1.561       3.278
map_Hybrid             2.3465      0.438      5.360      0.000       1.488       3.205
delta_standardized    -0.0265      0.011     -2.316      0.021      -0.049      -0.004
```

Figure 4: Coefficients from multiple linear regression model

#### 5.2.2 Machine Learning Approach

In this section, Naive Bayes, Support Vector Classification, K-Nearest Neighbors, and Bagging Classifier have been applied to the data and the results are shown in Tables 1, 2, 3, and 4 respectively. The support is the number of samples found for each of the categories in the generated training set. When run against the data in Python, these algorithms returned an accuracy of approximately 30.5%, 48.3%, 54.2%, and 55.8% for Naive Bayes, Support Vector Classification, K-Nearest Neighbors, and Bagging Classifier in order. The result indicates that Bagging Classifier has the highest accuracy out of the tested algorithms while Naive Bayes has the lowest accuracy out of the tested algorithms. Moreover, the results reporting precision, recall, and f1-score for win, loss, and draw are reported in Tables 1, 2, 3, and 4. In this data, win, loss, and draw are the three result categories.

Table 1: Naive Bayes, Accuracy = 30.55%

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Win | 0.15 | 1.00 | 0.26 | 89 |
| Loss | 0.55 | 0.33 | 0.41 | 480 |
| Draw | 0.57 | 0.14 | 0.22 | 457 |

Table 2: Support Vector Classification, Accuracy = 48.29%

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Win | 0.00 | 0.00 | 0.00 | 89 |
| Loss | 0.51 | 0.45 | 0.48 | 480 |
| Draw | 0.47 | 0.62 | 0.53 | 457 |

Table 3: K-Nearest Neighbors, Accuracy = 54.19%

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Win | 0.35 | 0.24 | 0.28 | 89 |
| Loss | 0.55 | 0.60 | 0.57 | 480 |
| Draw | 0.56 | 0.54 | 0.55 | 457 |

Table 4: Bagging Classifier, Accuracy = 55.80%

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| Win | 0.63 | 0.22 | 0.33 | 89 |
| Loss | 0.53 | 0.70 | 0.60 | 480 |
| Draw | 0.60 | 0.48 | 0.53 | 457 |

# 6 Conclusion

Out of the tested algorithms, the Bagging Classifier algorithm is found to most accurately predict the win result of a competitive Overwatch™ match. However, this accuracy is only 55.8%.

While there are many factors that may go into a decrease of accuracy, the most prevalent factor is the lack of data. Overwatch™ is still a relatively new game and there has been little data released by the publisher. Due to this fact, most of the data retained for the game is manually recorded.

While the data set used has 3,000 matches recorded, there are only two unique features, map type and delta standardized. Because this feature count is low, the machine learning algorithms do not have enough features from which to train. Once the game has released more data on the competitive matches, the machine learning models can be modified to include additional features. By doing this, the accuracy should increase for all of the observed algorithms which is the area for future research.

# References

[1] G. Synnaeve and P. Bessiere, *A Bayesian model for opening prediction in RTS games with application to StarCraft*, 2011 IEEE Conference on Computational Intelligence and Games (CIG'11) Computational Intelligence and Games (CIG) 2011.

[2] F. Metzger and R. Heger, *Exploring the Transmission Behaviour of Overwatch: The Source of Lag*, 2018 30th International Teletraffic Congress (ITC 30), 2018

[3] A. Ellingson, *Overwatch League Grand Finals averaged 860,000 viewers per minute*, L.A. Biz [Online], 2018

[4] A. Semenov, P. Romov, S. Korolev, D. Yashkov, and K. Neklyudov, *Performance of Machine Learning Algorithms in Predicting Game Outcome from Drafts in Dota 2*, Communications in Computer and Information Science Analysis of Images, Social Networks and Texts, 2017

[5] K. Wang and W. Shang, *Outcome prediction of DOTA2 based on Nave Bayes classifier*, 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), 2017

[6] R. Lamsal and A. Choudhary, *Predicting Outcome of Indian Premier League (IPL) Matches Using Machine Learning*, Transactions on Machine Learning and Artificial Intelligence, 2015

[7] P. Pai, L. Changliao, and K. Lin, *Analyzing basketball games by a support vector machines with decision tree model*, Neural Computing and Applications, vol. 28, no. 12, pp. 41594167, 2016

[8] K. Sriporn and C. Tsai, *Predicting Tourists' Behavior of Virtual Museum Using Support Vector Machine with Feature Selection Technique*, 2018 International Conference on Machine Learning and Cybernetics (ICMLC), 2018

[9] *Linear SVC Machine learning SVM example with Python*, Python Programming Tutorials. [Online]

[10] R. Folter, *Soccer and Machine Learning: 2 hot topics for 2018*, Data Science Central. [Online]

[11] M. Kang, Y. Kim, and Y. Kim, *Collecting large training dataset of actual facial images from facebook for developing a weighted bagging gender classifier*, Cluster Computing, vol. 20, no. 3, pp. 21572165, 2017

[12] J. A. Alzubi, *Diversity Based Improved Bagging Algorithm*, Proceedings of the The International Conference on Engineering  MIS 2015 - ICEMIS 15, 2015

[13] M. O'Neill, *Overwatch Game Records*, https://www.kaggle.com/mylesoneill/overwatch-game-records/downloads/overwatch-game-records.zip/1, 2018