Win prediction of League of Legends after ten minutes of play

Wesley Earl Stander
COMS4030
University of the Witwatersrand
Johannesburg, South Africa
1056114@students.wits.ac.za

***Abstract*—Win-prediction in sports helps to determine important concepts to focus on for victory. The same can be said about e-sports, although e-sports have data intrinsic to them. The paper aims to determine if predicting the winning team of a *League of Legends* match at ten minutes is feasible and determine the most important mechanics to focus on before that time point to ensure victory. Dimensionality reduction is attempted and fails due to irreconcilable overlap. Logistic regression and a random forest are used on the feature set to achieve a maximum accuracy of 73% with a random forest of maximum depth of 6 and 6 trees. The most important features are found to be gold attained by both teams.**

***Index Terms*—Win-Prediction, League of Legends, MOBA, E-sports**

I. Introduction

League of Legends is a multi-player online battle arena (*MOBA*) game created by Riot Games. The most played game-mode consists of game-play whereby two teams of five each controlling a character compete in a map with 3 lanes and a jungle. The outcome of a game is determined by the resources of each team alongside the usage of those resources. The problem that this paper aims to solve is win prediction at the ten minute mark. Predicting the results of sports matches allows for better sport analysis and with the rise of e-sports which have data intrinsic to them, the prediction of e-sports matches is even easier.*League of Legends* has a competitive scene that is ever-growing which makes match analysis even more important. The analysis of matches helps players to determine what is more important to winning and what should be the focus of the first ten minutes of the match. The first ten minutes of the match may be impactful enough to produce accurate results through prediction and the choice of the problem was made to provide insight on the most important mechanics to focus on to result in a victory.

This paper aims to determine if the first ten minutes of play can determine the outcome of the game accurately as well as analyse the feature set to determine the most impactful features. The problem is solved by using a logistic regression model as well as a random forest model. Principal component analysis is also attempted on the data to reduce the dimensionality of the data. The input is a file containing 9879 matches and their respective values at ten minutes of game time. The first output is the results of the principal component analysis to determine whether reduced dimensionality is feasible for this data-set. The second output is the regression model which predicts the outcome of the match. The learned theta values provide weighting as to whether values are important in determining the win prediction. The third outcome is the prediction of the random forest as well as the pivoting features which provide insight on the most important features to focus on when playing the game.

II. Related Work

The paper *Real-time eSports Match Result Prediction* [1] utilize logistic regression and a neural network to predict based on prior information before the game begins. They utilize three prior features including hero feature, player feature and her-player combined feature. Their prediction model utilizing prior information produced an accuracy of 71.46% for logistic regression and it was their highest accuracy achieved. The paper goes further to predict the outcome during the course of the game at 5 minute intervals. The real-time logistic regression model performs slightly better than the neural network model. utilizing the combination of prior and real-time models the performance is improved above the prior model slightly with the accuracy converging at the accurate prediction as the time of the game increases.

The paper *Continuous Outcome Prediction of League of Legends Competitive Matches Using Recurrent Neural Networks* [2] utilize a recurrent neural network, a long short term memory and a gated recurrent unit. A simple recurrent neural network was found to be the best choice with an accuracy that ranges from 63% to 83% depending on the time of the game that the model is used. The system achieves between 68.69% and 75.34% around the ten minute mark. The paper develops insight that gold and experience are required at certain times and fighting at others. The model can predict which is most suitable at that time which provides relevance to the most important features at the ten minute mark.

The paper *Predicting the winning side of DotA2* [3] utilizes a logistic regression model to predict the winner of *Dota 2*. The game has similarities to *League of Legends* and hence

the insight is transient. K-fold cross validation is used to perform feature selection and the paper attains an accuracy of as high as 78% with the optimal amount of features. The paper utilizes hero combination as well as player-proficiency to predict the winner. The implications as this is the highest accuracy achieved is that, the performance of individual players alongside the combination of the team has a slightly higher impact on the win than the resource accumulation.

The paper *DOTA 2 Win Prediction* [4] utilize logistic regression and a random forest on the gold, experience and kills of each team. The highest accuracy achieved is 73% utilizing the logistic regression model on the small feature set. The paper determines these features as the highest impacting features by viewing their respective distributions. This paper predicts after the match has ended which should be the highest prediction accuracy and makes the statement that the game is fundamentally a resource collection game. This insight is transient to *League of Legends*.

III. Data-set and features

The data-set is obtained from Kaggle and is a set of 9879 matches played in the diamond rank of League of Legends. The diamond rank contains the top 2.5% of players and hence will provide accurate results that are devoid of anomalies as all players have a high level of understanding of the game. The target feature is whether the blue team wins or not. The remaining nine-teen features that occur for both teams are:

- Wards placed
- Wards destroyed
- First blood
- Kills
- Deaths
- Assists
- Elite monsters killed
- Dragons killed
- Heralds killed
- Towers destroyed
- Total gold
- Average level
- Total experience
- Total minions killed
- Total jungle minions killed
- Gold difference
- Experience difference
- CS per minute
- Gold per minute

The feature set has some redundant features such as gold difference, gold per minute and experience difference. These have been removed for the random forest to reduce the dimensions of the data to thirty-two. The other features must remain except one teams first blood as the other can be inferred from the value because only one team can have a first blood making the dimensions thirty-one. Deaths are not inferred from the other teams kills as champions can die to jungle minions. Average level also does not infer from total experience as the experience distribution for levels is quadratic. The training and testing data split is 80% for training data to 20% for testing data. The data is normalised to a Gaussian distribution for the logistic regression with zero mean and unit variance.

IV. Methods

A. Principal Component Analysis

The data-set has a significantly high dimensionality so one approach attempted in order to discover a simpler method to predict the winner was principal component analysis. This method would produce abstract eigenvalues that would not provide insight but has the potential to increase the speed at which a prediction of the winning team can be made. The method to compute the principal components involves computing the mean vector μ for each feature x and then computing the co-variance matrix Σ. N is the number of features.

$$\mu = \frac{1}{N}\sum_{n=1}^{N} x_n \tag{1}$$

$$\Sigma = \frac{1}{N-1}\sum_{n=1}^{N}(x_n - \mu)(x_n - \mu)^T \tag{2}$$

The eigenvectors (principal components) P_D and eigenvalues Λ_D are found from the co-variance matrix Σ. The d principal components are then selected from P_D to retain a total variation of δ.

$$d = \underset{k\in\{1,2,\ldots,D\}}{\operatorname{argmin}} \; |\delta - \frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{D}\lambda_i}| \tag{3}$$

B. Logistic Regression

The outcome of the prediction is a binary classification which can be optimally found in the shortest time utilizing logistic regression. The algorithm trains to find a set of theta values that can transform the feature vector x into a prediction $h_\theta(x)$ that is either above 0.5 for class 1 or below 0.5 for class 0.

$$h_\theta(x) = \theta^T x + b \tag{4}$$

$$E(\theta, b) = \frac{1}{N}\sum_{n=1}^{N} J(y_i, h_\theta(x_i)) + \beta R(\theta) \tag{5}$$

The learning method aims to minimize the regularized training error $E(\theta, b)$ where cost function $J(y_i, h_\theta(x_i))$ is the log loss described in equation 6, b is the intercept, β is the amount of regularization with $R(\theta)$ is the regularization term described in equation 7.

$$J(y_i, h_\theta(x_i)) = \log(1 + \exp(-y_i h_\theta(x_i))) \tag{6}$$

$$R(\theta) = \frac{1}{2}\sum_{d=1}^{D}\theta_d^2 \tag{7}$$

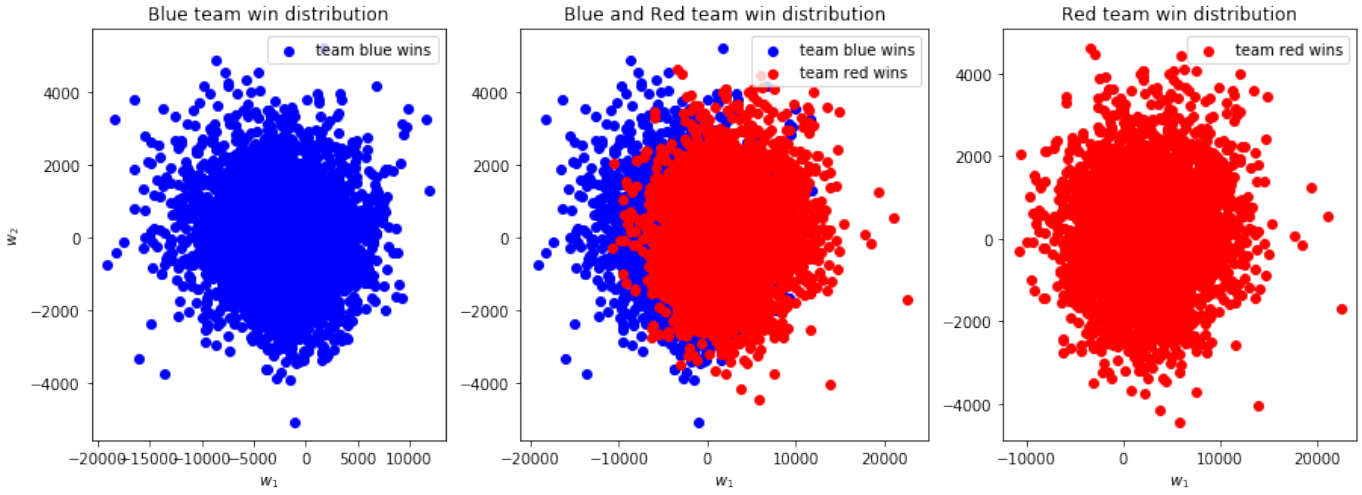

Fig. 1: Dimensionality-reduction to 2 dimensions for blue-wins and red-wins distributions

Each theta is updated for each data point with convergence rate α until the system converges. The update function for θ is in equation 8.

$$\theta \leftarrow \theta - \alpha[\beta \frac{\delta R(\theta)}{\delta \theta} + \frac{\delta J(y_i, \theta^T x_i + b)}{\delta \theta}] \tag{8}$$

α is not constant and is determined by the equation 9. This provides the optimal learning rate with respect to the initial α_0.

$$\alpha^{(t)} = \frac{1}{\alpha_0(t_0 + t)} \tag{9}$$

C. Random Forest

A decision tree is generated by splitting an initial node then recursively splitting on each split. The recursive algorithm splits the root node into 2 leaf nodes that recursively split on each sub set. The split feature and split value is determined by going through all possible splits to determine the best possible split. Every split S is tested on every feature f of every data point d in all data-points D. The split divides all values of d in feature f that are smaller than the split into the smaller sub-set α and all values of d in feature f that are bigger than the split into larger sub-set β. The gini index is determined using the 2 sub-sets α and β. The gini index G for the two groups is determined by equation 10 where N is the number of features, n the number of output targets(2) and p_i is the probability of a data point being classified into a particular target class. The gini index is used as it allows for continuous feature values.

$$G = \sum_{i=1}^{N}(1 - \sum_{i=1}^{n}(p_i)^2) \tag{10}$$

The lowest gini index G from all splits on the root node determines the actual split. The split S is then implemented on the feature and value which produced the lowest gini index G. Whence the root node has been split the algorithm determines if one of the nodes is less than a minimum size to prevent over-fitting. The algorithm also checks if the depth of the tree is larger than maximum depth and ends the recursive splitting as another measure to prevent over-fitting. If neither over-fitting criteria fails then the tree splits on both leaf nodes, treating them like root nodes as described in the opening steps of the algorithm.

The random forest is generated by randomly sampling the data into m distinct sub-sets. A decision tree h_m is generated from each sub-set. The set of trees H of size m are a random forest. Each tree predicts the prediction by comparing the values of the respective features to the respective decisions in the generated tree. Each tree predicts an outcome and the summation of the outcomes produce $H(x_i)$ which is tested against a sigmoid function to determine the prediction. The prediction $p(y_i = 1|x_i)$ is described in equation 12.

$$H(x_i) = \sum_{m} h_m(x_i) \tag{11}$$

$$p(y_i = 1|x_i) = \sigma(H(x_i)) = \frac{1}{1 + e^{-H(x_i)}} \tag{12}$$

V. Experimental Results and Discussion

A. Principal Component Analysis

The data is of a high dimensionality so principal component analysis is attempted as a form of preprocessing to reduce the dimensionality. The dimensionality-reduced data is represented in figure 1. It can be seen that the data has a large amount of overlap in the lower dimensions which prevents a high accuracy. The data however seems to be reflected about $x = 0$ from the red team win distribution to the blue team win distribution. This provides insight that the vertical eigenvector w_2 has little influence on the win probability of either team whereas the horizontal eigenvector w_1 has great influence on the win-rates. The first eigenvalue contains 93% of the information determining the win prediction with 2

dimensions of Principal Components. At higher dimensions of 10 dimensions the first eigenvector contains 86% of the accuracy and with the first 2 containing 92% of the accuracy there is a irreconcilable amount of overlap in the first 2 dimensions as seen in figure 1. This overlap will not produce accurate predictions and hence principal component analysis is not used for the remainder of the research.

The feature set can not have it's dimensions reduced so the feature set is kept as large as possible excluding redundant data features as described in section III. The insight gathered is that each feature has it's own independent affect on the which team will win. To gather further insight on each feature's influence a logistic regression model is fitted and a random forest.

B. Logistic Regression

The logistic regression converged after 150 iterations with an initial $\alpha_0 = 0.0001$. The classifier scored an accuracy of 71.50% on the training data and 71.45% accuracy on the test data indicating a very good fit due to the closeness of accuracy between training and testing data. The average term of the logistic regression is false indicating that it is easier for the classifier to predict red wins than blue wins. The mean of the coefficients $\theta_{average} = -66.16$ and the intercept $b = -4.27$ which shows that the model tends toward false indicating with the data set that it is easier to predict red wins. This is consistent with the training data where the blue team has a win-rate of 49.7% across all training data points. [2] acquire a similar accuracy utilizing a Recurrent Neural Network (RNN) with 68.69% in the 5-10 minute mark and 75.23% in the 10-15 minute mark. [4] achieve 68% accuracy utilizing a logistic regression model before the game starts which is similar to the RNN. [5] achieve 58.99% accuracy on a much larger data set using Naïve Bayes prediction before the game starts. It can be assumed that prediction accuracy is reduced the earlier the time step is from these 3 papers. [1] achieve 71.49% predicting prior to the start of the game utilizing an attribute sequence model and data that wasn't available to other papers representing the ability of the individual players.

Utilizing table I a qualitative analysis can be performed on the feature weightings to determine high impact features. Looking at the values that have magnitude bigger than half of the largest magnitude of coefficients, a set of 7 values is determined. The features corresponding to these values are blue wards placed, blue total gold, blue total minions killed, blue total jungle minions killed, red wards placed, red towers destroyed, red total experience. The largest coefficient being red total experience. [4] determine that gold, experience and kills are the highest accuracy predictors of post game performance. The logistic regression model makes similar assumptions with respect to gold and experience although kills have a much smaller weighting. This could be attributed to the difference of game although the similarity between the resource gathering is noticeable. The logistic model also finds the wards placed as a large indicator for predicting wins. It

TABLE I: Feature Weightings of Logistic Regression Model

Feature Index	Feature	Feature Weighting
1	blue Wards Placed	-1485.1791027
2	blue Wards Destroyed	-5.1068185
3	blue First Blood	45.0158463
4	blue Kills	70.3224071
5	blue Deaths	-117.087718
6	blue Assists	76.7539646
7	blue Elite Monsters	226.3786893
8	blue Dragons	221.5095149
9	blue Heralds	4.8691744
10	blue Towers Destroyed	-10.9923003
11	blue Total Gold	2152.1599311
12	blue Average Level	0.5652894
13	blue Total Experience	690.9120876
14	blue Total Minions Killed	-2230.2082975
15	blue Total Jungle Minions Killed	1454.9477487
16	blue CS Per Min	-223.0208298
17	red Wards Placed	-1350.2326889
18	red Wards Destroyed	-123.9743385
19	red First Blood	-117.087718
20	red Kills	70.3224071
21	red Deaths	-129.1923946
22	red Assists	-236.4709761
23	red Elite Monsters	-210.1379953
24	red Dragons	-26.3329807
25	red Heralds	13.1766029
26	red Towers Destroyed	-2063.8827598
27	red Total Gold	-4.734678
28	red Average Level	-881.6846234
29	red Total Experience	2430.6382815
30	red Total Minions Killed	-536.4384138
31	red Total Jungle Minions Killed	243.0638282

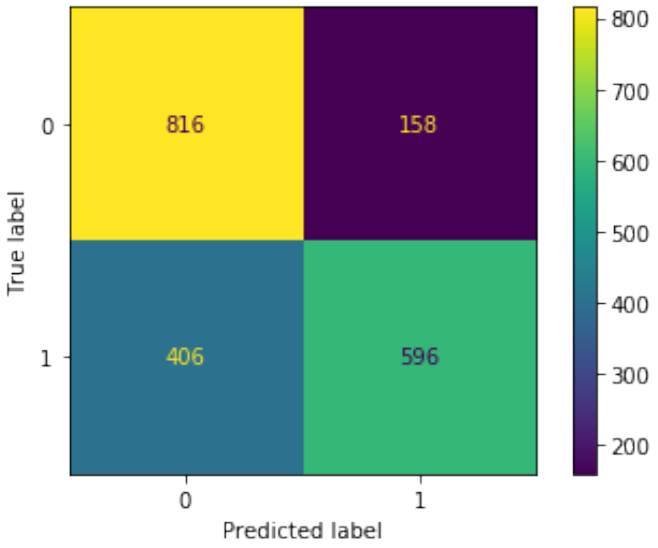

Fig. 2: Confusion Matrix for Logistic Regression with features in table I

can also be seen that blue wards destroyed, blue first blood, blue heralds, blue towers destroyed, blue average level, red dragons, red heralds and red total gold are irrelevant as they all have a magnitude smaller than the average coefficient magnitude. A further fit was attempted by removing non-impactful data points. After the removal of the non-impactful

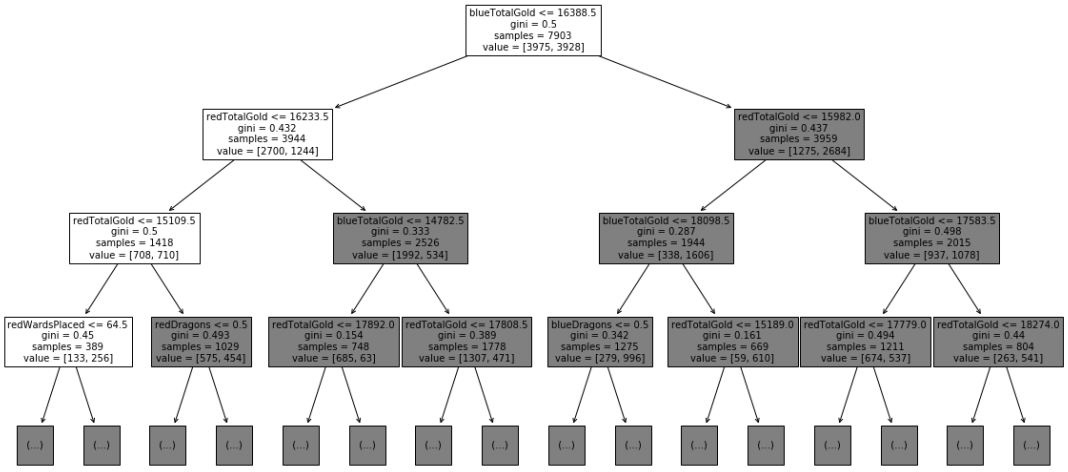

Fig. 3: Decision tree with max-depth of 6. Depth of 3 view-able.

features, the feature set is decreased to a size of 23 features. The fit of only relevant data points achieved an accuracy of 68.81% on the training data and 68.78% on the testing data which is a slight decrease indicating that all features are necessary to determine the optimal win-rate.

Upon further inspection of the model's prediction capabilities using the confusion matrix in figure 2, a further trend can be established. The similarity between true and false positives shows that The model struggles to predict wins and losses of the blue team and has a much higher accuracy on red wins and losses as the difference between true and false negatives is much larger. This is apparent as the weightings indicate lower magnitudes for corresponding weightings as compared to the red team except for total gold, minions killed and wards placed. The confusion matrix shows a higher accuracy for predicting red wins than blue wins. This can also be attributed to the larger amount of relevant features according to the weightings in table I and their higher variance to the feature mean. The larger variances in feature weightings indicate more impactful features per unit size. These can be further checked utilizing the decision tree classifier to determine the most impactful features and the random forest to find the best accuracy.

C. Random Forest

Viewing the first 10 max-depths of the decision tree it can be noted that the data begins to over-fit at a max-depth of 6. This is the point whereby the test accuracy begins to diminish while the training accuracy increases. The accuracy on the training data-set of a single decision tree at max-depth 6 is 73.34% and the accuracy on the testing data-set is 71.86%. This shows slightly better accuracy's than the logistic model. The decision tree in figure 3 reveals that the most important feature is total gold of the blue team and the second most important feature is the total gold of the red team. This coincides with the feature understanding that gold is one of the most important features [4]. Gold is further used as an important feature in the layers following as the game is a resource collecting game at it's essence [4].

Experience seems to be less impactful in *League of Legends* than in *Dota 2*. Through continued analysis of the feature set it can be seen that gold makes up the first two layers as well as the root node and the majority of the third layer. This implies that gold is the most important feature that decides if a team will win in *League of Legends*. This coincides with the logistic regression model as the feature weights with the largest magnitudes are the blue total gold and red total gold. The maximum depth is relatively shallow with respect to the lack of diversity in the first three layers. This lack of diversity can be rectified with a random tree classifier which will consider different sections of the data and reduce the overall presumption to only involve gold. The large amount of overlap in the principal component analysis reveals that there is indeed a large amount of overlap with respect to who wins and it involves the gold attained by each team.

The trend that a maximum depth of 6 provides the optimal fit continues into the random forest generation of all sizes. A maximum depth of 6 was then used to generate all random trees. The random forests tend to an accuracy of 72.7%

on the testing data-set and an accuracy of 75% on the training data-set. The largest accuracy for the testing data was a random forest with 6 decision trees. The accuracy is 73.23% and the accuracy on the training data was 75%. This provides the best accuracy on the data-set. This gives a slight improvement of the single decision tree classifier and accounts for the fluctuations in the data. It is assumed that any performance gains from this point would be negligible as the most important feature is gold and causes a large amount of accuracy to pivot on the gold of each team. The accuracy can only be improved with the increase of data-points. Similar results were achieved to the random forest usage in paper [4]. They achieved 73% accuracy on their logistic regression model whereas only 69% on their random forest model although their model utilized post-game information. The random forest and logistic model in this paper performed significantly better as the prediction of victory has comparable accuracy at ten minutes from the other paper which is post-game prediction.

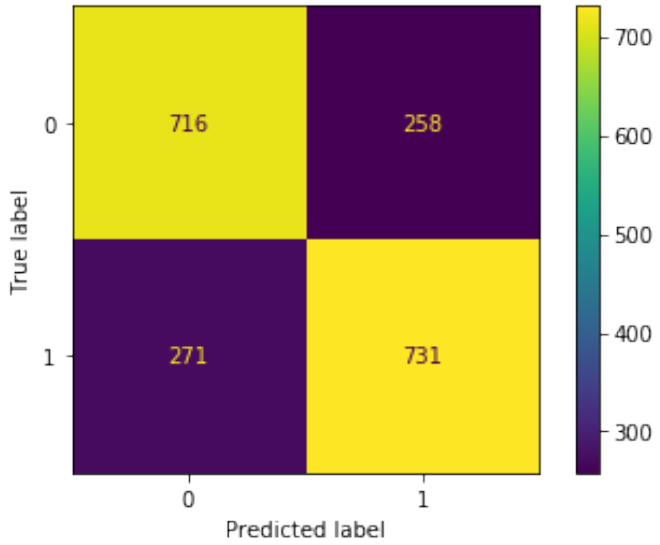

Fig. 4: Confusion Matrix for Random Forest with 6 trees and max-depth of 6

The confusion matrix in figure 4 provides insight that the random forest has better prediction accuracy for both red winning and blue winning as compared to the logistic regression. The false positives and false negatives are very similar in magnitude as well as the true positives and the true negatives are similar in magnitude. This indicates a more accurate quality of prediction for the random forest as compared to the lower quality of prediction with the logistic regression. The random forest tends to fit the data better due to the reduction of over-fitting and is evident in the comparison of confusion matrices in figure 4 and figure 2.

VI. Conclusion

The prediction of which team will win a *League of Legends* match at the ten minute mark is very viable. *League of Legends* produces a much higher accuracy at the ten minute mark as compared to similar games such as *Dota 2*. This paper comes to the conclusion that victory in *League of Legends* is highly deterministic on the first ten minutes of the game. The paper achieved a maximum prediction accuracy of 73% utilizing a random forest with 6 splits on the data-set and a maximum depth of 6. This alongside the feature analysis provides insight on the most important mechanics that must be focused on for a match to result in victory. The gold attained by both teams is the most important deciding factor as to who will win the game. This feature is dependant on many of the other features as dieing loses gold and killing enemies, minions, dragons and heralds gains gold. The optimization of attaining gold will provide the optimal route to winning a match of *League of Legends*.

The conclusion as to whether a game can be predicted at the ten minute mark is true. The prediction accuracy of 73% on the testing data-set shows that there is a significantly higher chance of the predicted team winning than losing which is 73% to 27%. This difference in chance indicates that predicting the winner is viable although much like any game, poor play can result in abnormal training values as well as the team combination and each individual player's ability with their champion can sway the win chance. These factors influence the win-rate significantly [4] [3].

VII. Future work

The future work that is proposed is research that considers both the accumulated resources at the ten minute marks as well as the individual champion choices alongside the respective player's ability with the champion and their performance in that team composition. This creates a significantly larger search space but will provide the most accurate prediction results.

Acknowledgment

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

References

[1] Y. Yang, T. Qin, and Y.-H. Lei, "Real-time esports match result prediction," *arXiv preprint arXiv:1701.03162*, 2016.

[2] A. L. C. Silva, G. L. Pappa, and L. Chaimowicz, "Continuous outcome prediction of league of legends competitive matches using recurrent neural networks," in *SBC-Proceedings of SBCGames*, 2018, pp. 2179–2259.

[3] K. Song, T. Zhang, and C. Ma, "Predicting the winning side of dota2," *Sl: sn*, 2015.

[4] N. Kinkade, L. Jolla, and K. Lim, "Dota 2 win prediction," *Univ. California, San Diego, CA, USA, Tech. Rep. FA15-018*, 2015.

[5] K. Wang and W. Shang, "Outcome prediction of dota2 based on naïve bayes classifier," in *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*. IEEE, 2017, pp. 591–593.