AITA
ARTIFICIAL INTELLIGENCE
THEORY AND APPLICATIONS

■ RESEARCH ARTICLE

A Machine Learning Based Predictive Analysis Use Case for eSports Games

Atakan Tuzcu [a], **Emel Gizem Ay** [a†], **Umay Uçar** [a], **Deniz Kılınç** [a]

[a] Department of Computer Engineering, University of Bakırçay, İzmir, Turkey
[†] aemelgizem@gmail.com, corresponding author

RECEIVED MARCH 5, 2023
ACCEPTED XX

CITATION Tuzcu, A., Ay, E.G., Uçar, U, & Kılınç. D. (2023). A Machine Learning Based Predictive Analysis Use Case for eSports Games. *Artificial Intelligence Theory and Applications, XX*(X), XX-XX.

Abstract

Multiplayer online games named of e-Sports generate data, just like all other technology and application-based online activities performed. It's clear that by leveraging sports analytics to analyze online game data, teams can gain significant advantages over their opponents. Through this approach, teams can gain valuable insights into their own performance and that of their opponents, allowing them to make data-driven decisions that can help them improve their game strategies and ultimately increase their chances of winning. In this study, it is aimed to predict the outcome of the game using machine learning classification algorithms on historical game data obtained through the official API provided by Riot Games for the game League of Legends (LoL). Two separate data sets were created for testing the models, one based on teams and the other based on players. The data set was divided into training and testing data, with a test data rate of 20%. For training the team-based models, 1045 data were used. For training the player-based models, 5232 data were used. Experimental studies showed that models trained on the team data set gave more successful results. The most successful model trained on the team data set was the AdaBoost algorithm, with an accuracy of approximately 0.98, while the most successful model trained on the player data set was the LightGBM algorithm, with an accuracy of approximately 0.95. In this study, a feature selection method based on Gini score was also applied to assess its effect. The 10 most significant features affecting the match prediction were identified, and a new team data set was recreated by using only these features. Machine learning algorithms previously used were retrained on this new data set. It was observed that the success rate of 71% of the trained models increased with this method. As a result of analysing the model performances, it was seen that the accuracy value was highest with Logistic Regression and Gradient Boosting algorithms, reaching 98%.

Keywords: *league of legends, riot games, machine learning, random forest, gradient boosting*

1. Introduction

Sports analytics is one of the most popular topics in data analysis today, with MOBA (Multiplayer Online Battleground Arena) style games being among the most interesting games in this field. The primary objective in MOBA games is to destroy the opposing

Artificial Intelligence Theory and Applications, ISSN: 2757-9778. ISBN : 978-605-69730-2-4 © 2022 University of Bakırçay

team's main base. In this study, data sets were created by pulling data from League of Legends (LoL) game through online platforms. LoL is a MOBA game and, like other MOBA games, has a 5v5 game style. The teams in the game consist of 5 players in roles such as top lane, mid lane, jungle, marksman, and support. The tasks of each player based on these roles vary according to different strategies. The items taken during the game can cause different reactions from the characters, leading to many possible game strategies through combinations of limited parameters in the game.

The main objective in the game is to destroy the opponent team's towers and subsequently their main center. The data of all five players in the teams should be taken into consideration. As it is a team game, the poor performance of some players can be compensated to a certain extent, and the possibility of the team winning still exists. In this study, two data sets, team-based and player-based, were created. The most used machine learning algorithms in the literature were experimentally tested on these data sets. After preprocessing the datasets, models were trained. It was observed that the most successful algorithm among the models trained on the team dataset was the AdaBoost algorithm with an approximate accuracy value of 0,9847, while the most successful algorithm among those used for the player dataset was the LightGBM algorithm with an approximate success value of 0,9541.

In the next step, Gini score-based feature selection was applied to analyze the top 10 attributes that have the most impact on match prediction. A new dataset was created with only these attributes, and the previously used machine learning algorithms were re-trained using this new dataset. After this step, it was observed that the accuracy rate increased in 71% of the models. When examining the performance of the models obtained, it was observed that the accuracy value was 0.98 for Logistic Regression and Gradient Boosting algorithms.

The remainder of the paper is structured as follows: Section two provides a comprehensive literature review on game analytics. Section three briefly describes the materials and methods used in the study, including the dataset collection process, preprocessing techniques, machine learning algorithms, and model performance evaluation criteria. In section four, we present the results of our experimental study, discuss the findings, and analyze the impact of feature selection on model performance. Finally, the conclusion summarizes the entire study.

2. Related Works

Even before the advent of computers and digitalization, data was generated from sports competitions, much like in all activities today. Analysis based on this data allowed for inferences to be made about game strategies that would give teams an advantageous position over others in these competitions. In Michael Lewis's book [2], the story of extracting decision-making information to elevate a mediocre team to the top ranks during baseball matches through data analysis is told.

The study conducted by Y. Yang et al. [3] stands apart from previous studies by incorporating data obtained during the game, in addition to pre-game data. This approach resulted in changes in the expected winning team, based on the in-game data. The researchers chose a logistic regression model as their prediction model and conducted their study on Dota 2. They used their trained model with real-time data and presented their results graphically. Their findings revealed that the team expected to win until the 7th minute of the game was different from the team that eventually won the game. This study illustrates how the use of in-game data can influence the accuracy of the output. However, by solely relying on logistic regression in their trained model, the researchers overlooked other models that could potentially have resulted in higher accuracy.

In their study, A. Silva and colleagues [4] aimed to compare RNN [5] models by leveraging the inherent characteristics of the data. They tested simple RNN, LSTM [6], and GRU [7] models and found that the simple RNN model had the highest accuracy rate. The researchers utilized a dataset where each row represented a minute of the game, with the goal of capturing changes in the data as the game progressed. Their results showed that the simple RNN model achieved a consistent accuracy rate of 76.29%. However, the researchers acknowledged that the model's performance may be affected by game updates and may not work as consistently.

3. Materials and Methods

3.1. Dataset

The dataset used in this study was created by obtaining game data from an online platform through the Riot API, which is provided by the game's developer. The Riot API is a tool used by developers to integrate Riot Games into their applications. Although Riot Games offers numerous APIs to researchers, only two were utilized in this project. Figure 1 illustrates the data extraction steps for the API used in this study.

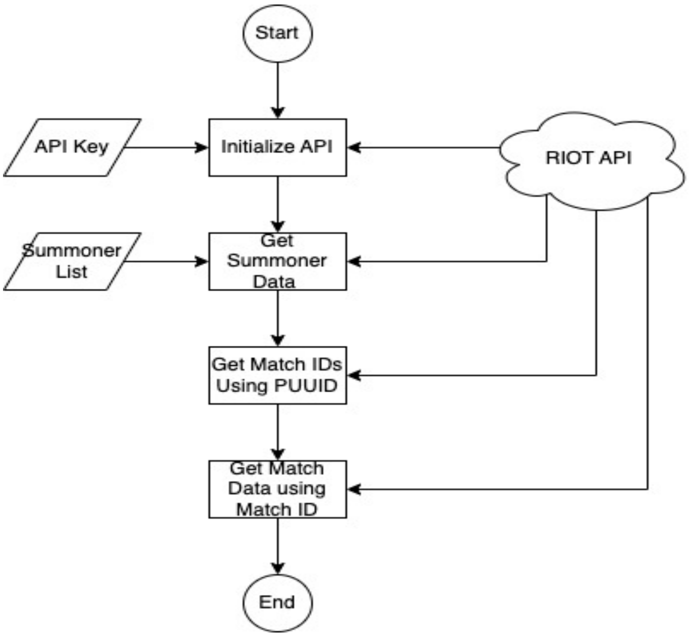

Figure 1. Data Collection with RIOT API

Summoner-v4: The API used in this study offers 6 different methods for obtaining summoner information. The method used in this research retrieves summoner data using the summoner name and stores the response value for retrieving the PUUID, a unique value for each summoner. This method is a GET method that requires the summoner name and region as input and returns a summoner object as response.

Match-v5: This API offers three methods that developers can use to retrieve information on match games. In our study, we utilized two of these methods to obtain game IDs and subsequently access each game's data. These methods are both GET methods, with one taking the PUUID as a parameter and responding with game IDs, while the other takes game IDs as a parameter and responds with the corresponding game data.The datasets are labeled with a binary label, where 0 indicates losing team and 1 indicates winning team. The numeric features of the datasets are presented in Table 1.

Table 1. Numerical Information of The Classes in The Datasets

Dataset	Data Groups (Labels)	Data Counts	Feature Count	Total Instance Count
DS1. Team-Dataset	Loser Winner	587 591	47	1,045
DS2. Player-Dataset	Loser Winner	2,594 2,638	47	5,232

Some important attributes in the dataset and their definitions are shown in Table 2.

Table 2. Description Of Some Features

Feature Name	Description
turretsLost	The number of towers lost
turretKills	The number of destroyed towers.
inhibitorKills	The number of destroyed inhibitors.
inhibitorTakedowns	The number of inhibitors destroyed by the player.
largestKillingSpree	The highest killing spree count.
deaths	The number of deaths of the player.
damageDealtToObjectives	Damage dealt to objectives.
totalTimeSpentDead	The time spent dead in the game..
kills	The number of kills.

The dataset was split into training and test data with a test data ratio of 20%. The data used in the test set was not used in any way in the training set.

3.2. Pre-processing

In this stage of the study, the game data collected with the RIOT API was pre-processed to ensure that the classification models to be used would yield accurate results. Firstly, the attributes in the dataset were examined separately, and missing values were detected in some of the attributes; if these missing values exceeded 80%, they were deleted. Unique identity information defining IDs for each team was recreated, and filtering was performed based on these IDs for both team and player-specific data. As a result, DS1 and DS2 datasets were obtained for model training, one based on teams and the other based on players.

3.3. Machine Learning Algorithms

As mentioned in the literature, the use of machine learning algorithms in game analytics has become increasingly widespread in recent years. In the conducted study, after pre-processing steps were completed on the dataset, different categories of machine learning classification algorithms were used on DS1 and DS2 datasets. The classification algorithms used were Random Forest [8], Decision Tree [9], Logistic Regression [10], LightGBM [11], Naive Bayes Classifier [12], Gradient Boosting [13], and AdaBoost [14]. Feature selection, one of the characteristics of the Random Forest algorithm, measures the importance of each feature on the prediction. By examining the importance of each feature, the features that have the most impact on the prediction were identified, reducing the prediction time and lowering the probability of overfitting and its opposite. The system's workflow is shown in Figure 2.

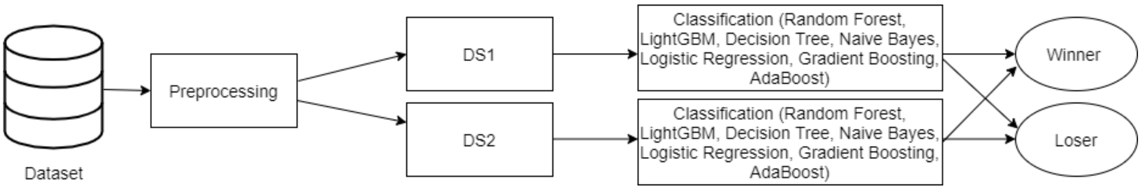
Dataset
Preprocessing
DS1
DS2
Classification (Random Forest, LightGBM, Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, AdaBoost)
Classification (Random Forest, LightGBM, Decision Tree, Naive Bayes, Logistic Regression, Gradient Boosting, AdaBoost)
Winner
Loser

Figure 2. Operating schema of the system

3.4. Evaluation Criteria

To evaluate the accuracy of the system that performs classification using machine learning algorithms, a confusion matrix was used. The confusion matrix determines the accuracy of the model, which represents the ratio of correct predictions to all data. Table 3 shows the structure of a two-class (positive, negative) confusion matrix [15].

Table 3. Confusion Matrix

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

Accuracy is a commonly used metric to measure the performance of a model. The accuracy value is calculated by the ratio of the total number of correctly predicted classes in the model to the entire dataset. True Positive and True Negative refer to the areas where the model correctly predicted, while False Positive and False Negative refer to the areas where the model incorrectly predicted. The equation for the accuracy metric used to evaluate the model's performance is shown in Equations 1.

$$Accuracy = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \qquad \text{(Eq.1)}$$

4. Experimental Study and Results

Initially, two separate datasets were obtained from the existing dataset, one based on team and the other based on player. The highest accuracy rate was achieved with the AdaBoost algorithm with 0.9847 when examined based on the team, and with the Gradient Boosting and LightGbm algorithms with 0.95 when examined based on the player. The accuracy rates of the machine learning models trained are given in Table 4.

Table 4. Performance Comparison of Models

Algorithm Name	DS1	DS2
Random Forest	0.9732	**0.9533**
Decision Tree	0.9503	0.9388
Naive Bayes	0.8015	0.7265
Logistic Regression	0.8969	0.7624
Gradient Boosting	0.9656	0.9541
LightGBM	**0.9770**	**0.9541**
AdaBoost	**0.9847**	0.9526

The confusion matrices of the top 2 models with the highest accuracy rates for both DS1 and DS2 datasets have been shown. Figures 3 and 4 represent the confusion matrices for the player dataset, while Figures 5 and 6 represent the confusion matrices for the team dataset.

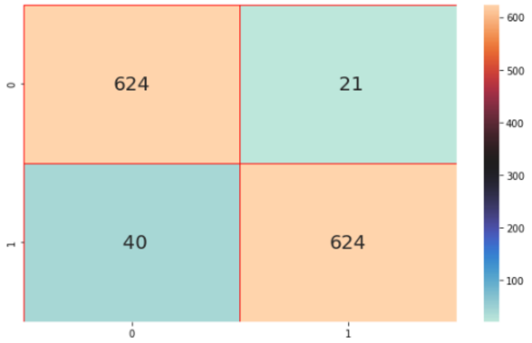

Figure 3. Random Forest

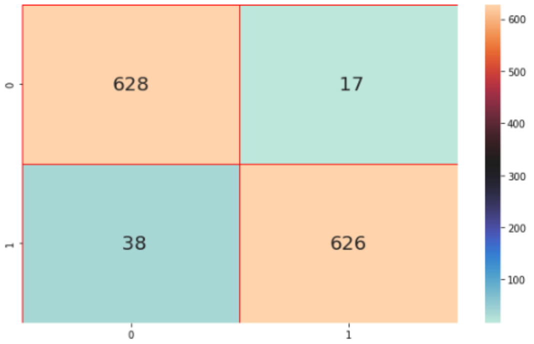

Figure 4. Gradient Boosting

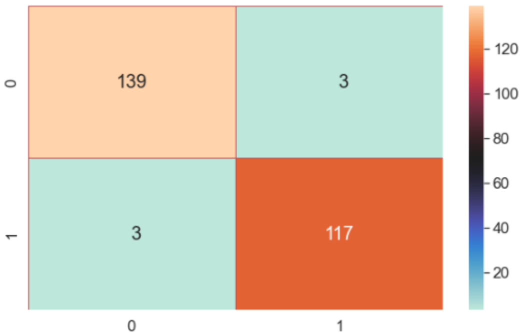

Figure 5. LightGBM

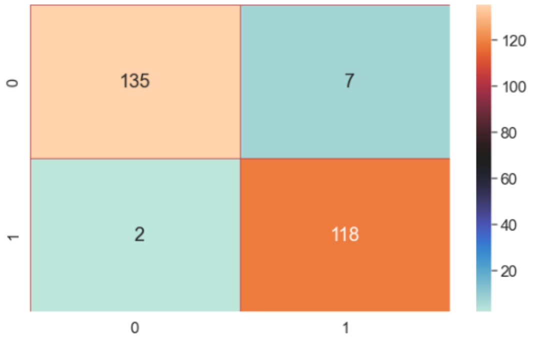

Figure 6. Gradient Boosting

4.1. Feature Selection

Feature selection is the process of reducing the number of input variables when developing a prediction-based model. It is desirable to reduce the number of input variables to decrease the computational cost of modeling and, in some cases, improve the model's performance [16]. The decision tree algorithms used in the study prune the branches of the tree based on the importance of the input variable.

In this study, a Gini score-based algorithm was used for feature selection, and the top 10 features that have the most impact on classification (Figure 7) were selected to train models and calculate their accuracy values.

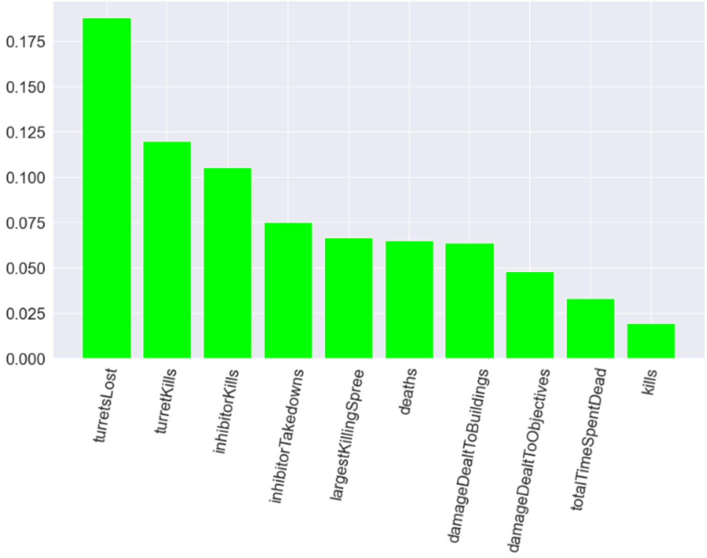

Figure 7. The Results of Gini Score-Based Feature Selection

4.2. The Effect of Feature Selection

In this study, feature selection was performed on a data set with 47 attributes to use fewer features. Out of the 7 models trained, performance improvement was observed in 5 models. The most significant performance increase was observed in the Naïve Bayes and Logistic Regression algorithms. According to the accuracy values obtained after the feature selection process, the most successful models were Logistic Regression and Gradient Boosting, as seen in Table 5.

Table 5. Performance Comparison of Algorithms After Feature Selection

Algorithm Name	**Accuracy Value Before Feature Selection**	**Accuracy Value After Feature Selection**
Random Forest	0.9732	0.9809
Decision Tree	0.9503	0.9618
Naive Bayes	**0.8015**	**0.9656**
Logistic Regression	**0.8969**	**0.9847**
Gradient Boosting	**0.9656**	**0.9847**
LightGBM	0.9770	0.9770
AdaBoost	0.9847	0.9809

4.2.1. Comparison of Confusion Matrices for Naïve Bayes

After the feature selection process, it was observed that the accuracy value of the Naïve Bayes model increased by 0.16%. The confusion matrices of the algorithm before and after feature selection are shown in Figure 7 and Figure 8, respectively.

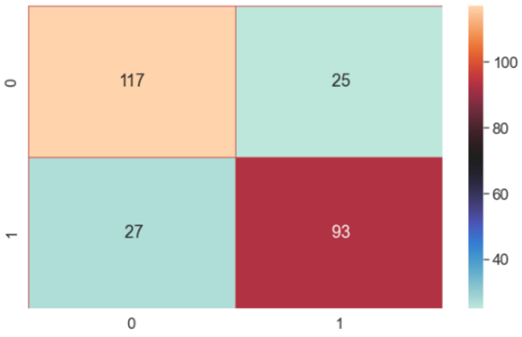

Figure 7. Before Feature Selection

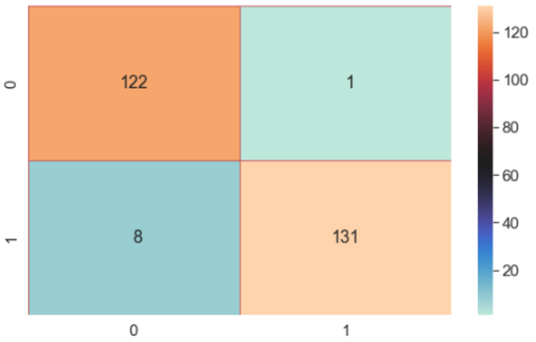

Figure 8. After Feature Selection

4.2.2. Comparison of Confusion Matrices for Logistic Regression

After the feature selection process, it was observed that the accuracy value of Logistic Regression model increased by 0.16%. The confusion matrices of the algorithm before and after feature selection are shown in Figure 7 and Figure 8, respectively.

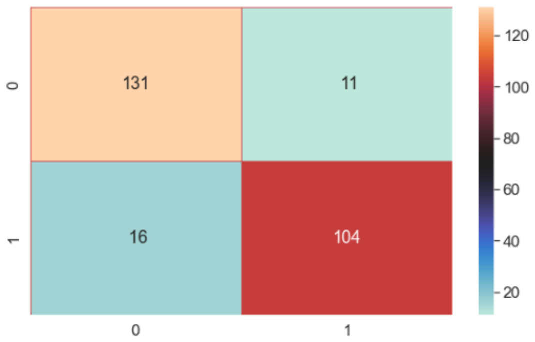

Figure 9. Before Feature Selection

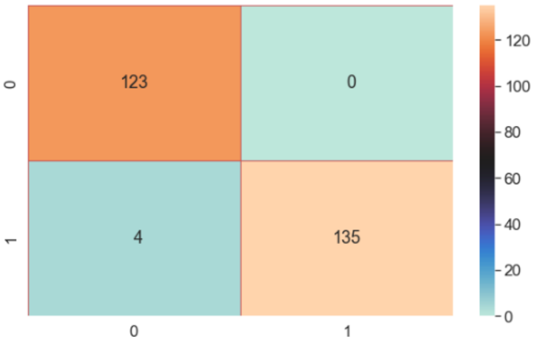

Figure 10. After Feature Selection

4.2.3. Comparasion of Confusion Matrices for Gradient Boosting

It has been observed that the accuracy value of the Gradient Boosting algorithm, which is one of the most successful models, increased by 0.01% after the feature selection process. The confusion matrices before and after the feature selection are shown in Figure 11 and Figure 12, respectively.

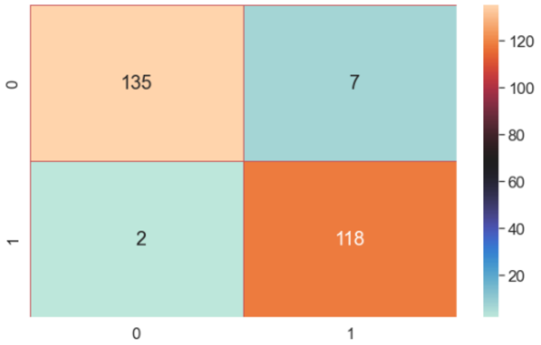

Figure 11. Before Feature Selection

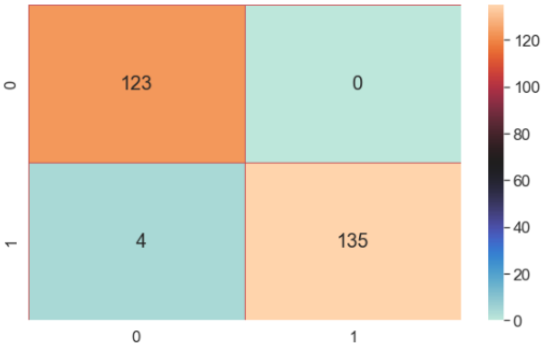

Figure 12. After Feature Selection

5. Conclusion and Future Works

The aim of this study was to predict the outcome of League of Legends games using historical game data obtained through the official API provided by Riot Games. The game data presents a classification problem, and machine learning models including Random Forest, Decision Tree, Logistic Regression, Light GBM, Naive Bayes Classifier, Gradient Boosting, and AdaBoost algorithms were used for classification. The highest accuracy rate obtained from the models for the team data set was 98.41% with the AdaBoost algorithm. It was observed that the selection of important features and the training of models with these features can result in high performance and using only 21% of the features in the data set reduced the workload of the model. After the feature selection process, the most successful algorithms were found to be Logistic Regression and Gradient Boosting with a success rate of 98.41%. It was also observed that this success rate was achieved with the AdaBoost algorithm even without the feature selection process. In the future, deep learning models can be created and optimized to achieve higher success rates for classification. In addition, more comprehensive and complex models can be trained with real-time data flow obtained during ongoing games to improve the accuracy of game outcome predictions.

References

[1] Mora-Cantallops, M., & Sicilia, M. Á. (2018). MOBA games: A literature review. Entertainment computing, 26, 128-138.

[2] Lewis, M. (2004). Moneyball: The art of winning an unfair game. WW Norton & Company.

[3] Yang, Y., Qin, T., & Lei, Y. H. (2016). Real-time e-sports match result prediction. arXiv preprint arXiv:1701.03162.

[4] Silva, A. L. C., Pappa, G. L., & Chaimowicz, L. (2018). Continuous outcome prediction of league of legends competitive matches using recurrent neural networks. In SBC-Proceedings of SBCGames (pp. 2179-2259).

[5] Medsker, L. R., & Jain, L. C. (2001). Recurrent neural networks. *Design and Applications*, *5*, 64-67.

[6] Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM--a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586.

[7] Dey, R., & Salem, F. M. (2017, August). Gate-variants of gated recurrent unit (GRU) neural networks. In 2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS) (pp. 1597-1600). IEEE.

[8] Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

[9] Freund, Y., & Mason, L. (1999, June). The alternating decision tree learning algorithm. In icml (Vol. 99, pp. 124-133).

[10] LaValley, M. P. (2008). Logistic regression. Circulation, 117(18), 2395-2399.

[11] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

[12] Murphy, K. P. (2006). Naive bayes classifiers. University of British Columbia, 18(60), 1-8.

[13] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

[14] Schapire, R. E. (2013). Explaining adaboost. In Empirical inference (pp. 37-52). Springer, Berlin, Heidelberg.

[15] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC genomics, 21(1), 1-13.

[16] Sima, C., & Dougherty, E. R. (2006). What should be expected from feature selection in small-sample settings. Bioinformatics, 22(19), 2430-2436.