Mobile Legends Win Rate Prediction and Team Recommendation Using Switched Hero Roles

Pir Hamid Ali Qureshi[1], Areej Fatemah Meghji[1], Rabeea Jaffari[1]
[1]Department of Software Engineering, Mehran University of Engineering and Technology, Jamshoro, Pakistan.
*** Correspondence:** Areej Fatemah Meghji, areej.fatemah@faculty.muet.edu.pk

Citation | Qureshi. P. H. A, Meghji. A. F, Jaffari. R, "Mobile Legends Win Rate Prediction and Team Recommendation Using Switched Hero Roles", IJIST, Vol. 7 Issue. 1 pp 623-636, March 2025
Received| March 4, 2025 **Revised|** March 24, 2025 **Accepted|** March 26, 2025 **Published|** March 29, 2025.

Introduction: Mobile Legends Bang Bang (MLBB) falls under the category of a multi-line battle arena game which requires players to have strong skills and strategic gameplay; team composition is an important factor influencing the chances of winning the game.
Novelty Statement: Although there is data currently available for MLBB, two aspects of this game that remain unexplored include: i) win rate prediction using nontraditional roles in heroes, and ii) team composition with switched hero roles.
Material and Method: This research aims to address this issue by predicting the win rate of heroes with switched roles. This unpredictability will lead to the formation of a team that can have a significant advantage over the enemy team thus leading to victory. The dataset for this study was formulated focusing on 67 heroes in the game. The win rates were generated with real-time simulations where the ally team members remained unchanged to avoid biased results.
Result and Discussion: The research utilized two model-building approaches and win rate predictions were made using 12 regression algorithms under 5 feature selection settings. The results show that LightGBM with AdaBoost as the base estimator provides better results and was used to formulate 5 teams. A recommendation system was designed to optimize team composition from the win rate prediction analysis. To validate the results, we simulated 50 matches with each team resulting in a 94% win rate.
Concluding Remarks: The research explores switched hero roles and provides promising results to help team formation with an increased chance of victory when using non-traditional hero roles.

Keywords: Machine learning; Recommendation; Feature Selection; Regression; Mobile Gaming

Introduction:

There are 4.88 billion smartphone users in the world today [1]. This number is expected to increase to 5.7 billion by the end of 2028 [2]. With the rapid increase of smartphone users, the number of people playing mobile games has also increased. The number of mobile gamers is expected to climb up to 2.3 billion by the end of 2027 [3]. Over the years, many types of game genres have been introduced, with Multi Online Battle Arena (MOBA) being the most successful genre in the history of gaming on smartphones. Due to its popularity, MOBA has become one of the main subjects of artificial intelligence (AI) research in gaming [4]. MOBA games belong to the category of Real-Time Strategy (RTS) [5], referring to time-based games which have two popular categories. The first type is where users control multiple units at the same time, build a base while collecting the resources, and defend their base or attack the opponent's base as done in Clash of Clans. Another example is where the user controls a main unit; the goal is to terminate the opponent's base through the collection of resources without making their base as it is already pre-built as in League of Legends (LoL). Similar to LoL, there is also a game called Mobile Legends Bang Bang (MLBB).

Five-on-five games are played in MLBB such that two teams of five heroes compete against each other. The game's basic gameplay pits two teams of five heroes against one another in real-time. Players engage in combat over three lanes to capture the enemy's tower and protect their own, all the while securing objectives to put pressure on the other team. This game features classic battle arena action with three lanes: the mid lane, the exp lane at the bottom, and the gold lane at the top [6]. Similar to other MOBAs, there is no pay-to-play element or hero training to level up; instead, skill, ability, and strategy determine who wins and loses. Each hero's responsibilities within their squad are determined by their role which may include either assassin, tank, mage, marksman, support, or fighter.

Each team aims to destroy the opponent base by collecting the gold from farming, either by killing minions, eating buffs, or getting the gold by slaying the opponent's heroes. There are matches of different types including; Classic, Ranked, Brawl, Magic Chess, and Arcade. As MLBB has grown in popularity, different regional and international tournaments are held. Each hero consists of different properties and skills that are included for overall team power. Heroes contain different strengths and weaknesses; therefore, drafting a team can be a difficult task. Limited research is available on MLBB, especially focusing on the aspect of the game where a hero is assigned a nontraditional role and the effect on the win rate when a team is composed where the hero roles are switched.

As MLBB is highly dependent on strategy, the element of surprise has great value in this environment. This surprise factor can be achieved when enemy teams are unaware of what to expect when hero roles are changed from their original ones. This particular kind of move can turn losses into wins when the enemy chooses counter heroes, completely unaware that the ally team will be using heroes with switched roles. In the current research, supervised machine learning based on regression tasks has been used to predict the win rate of heroes for team recommendations using switched hero roles. Supervised learning is a category of machine learning that uses labeled datasets to train algorithms [7] and then uses the trained models to predict future outcomes. There are two types of supervised learning: Classification, which is used to predict categorical outcomes for given inputs, and Regression, which is used for predicting outcomes in continuous or real numerical values [8]. The dataset inputs for this research include pre and post-characteristics of heroes with different roles. For example, Pre Health-Points and Post Health-Points, Pre Mana and Post Mana, Pre Physical Attacking Power, and Post Physical Attacking Power among others. The target variable was the win rate which was simulated by playing matches using each hero with every role. The goal of this research is to develop a predictive model that accurately estimates the win rate of heroes based on every role and to recommend team compositions that can ensure high chances of victory.

Our study offers a new perspective for coaches, players, and data analysts by predicting the outcome of matches when the roles of heroes are switched. It also offers guidance in building teams where the chances of being victorious are high when some hero roles are converted from their initial role.

Figure. 1. MLBB Roles map [6]

The outline of this paper is as follows: the second section consists of the related work, while the third section outlines the framework and methodology that provides details about the dataset, approaches, algorithms, methods, and the evaluation parameters used for the analysis. The fourth section describes the experimental results, discussion, and practical implications of the research. Lastly, the conclusion section sums up the core observations of the study and offers guidance for future recommendations.

Related Work:

Arik proposed three algorithms for winner prediction in LoL using player key performance metrics. The research found that the Light Gradient Boosting Machine (LightGBM), Logistic Regression (LoR), Support Vector Machine (SVM), and Gradient Boosting Classifier (GBC) are higher performers in terms of accuracy, with the LightGBM achieving an accuracy of 97% [9].

Tiffany et al. proposed a prediction approach in ranked matches for LoL. They applied multiple algorithms including Random Forest Tree (RFT), Gradient Boosting (GB), and Deep Neural Network (DNN). The parameters used in this dataset were the season total number of games played on the champion, the number of recent games played on the champion, and the player's champion win rate. They applied Pearson's correlation test and, based on it, decided to keep champion mastery points and player champion win rate and discarded the other parameters. The output by GB outperformed achieving an accuracy of 75.4%. A limitation of this research was the limited analysis of rank matches from only North American servers when the rank matches might differ from those of other servers such as Oceania (OCE) [10].

Chan, Fachrizal, and Lubis demonstrated that Naïve Bayes (NB) can effectively predict the result of matches based on selected hero roles in MLBB. The performance algorithm was based on four roles. The results for each category were obtained as: fighter 40% win, and 3% loss; mage 11% win and 30% loss; tank and marksman having a 40% and 28% win respectively with a 7% and 19% loss. The results obtained for fighters and tanks were similar to one another; however, since tanks had a higher loss rate, the conclusion was made that the fighters had higher chances of winning [11].

Ani et al. predicted winners in LoL using various ML models focusing on feature extraction and the use of assembling to achieve high accuracy. Among other algorithms, Extreme Gradient Boosting (XGB) was used with the Recursive Feature Elimination (RFE) technique. The parameters they set for data included pre-match, within-match, and combined [12]. Mahendra discussed the roles of champions in different categories and suggested accordingly as to what champion should a person pick for victory. The author explained graph theory and tree structures mainly by focusing on Decision Trees (DT) and applied discrete mathematics to categorize and select champions. Champions were classified into different roles including Top, Mid, Jungler, AD Carry, and Support. Sub-categorization of each role was based on stats and playstyles. The author concluded that DT can help players decide on champions by considering factors like team composition and personal playstyle preferences [13].

Pengmatchaya and Natwichai proposed an effective machine-learning pipeline to evaluate the player's skill. They collected data through API, and for feature engineering, they took different attributes that include end-game, or tactical decision-related statistics, harassment tactics, or spatiotemporal features. For machine learning, they applied LoR and RFT along with other algorithms. They concluded that the most effective model could achieve up to 0.7091 precision, 0.5850 recall, and 0.6411 F_1-score [14].

The Monte Carlo Tree Search algorithm was used in [15] to propose a recommendation system for a team based on the win rate. The researchers explored four approaches including gradient-boosted Decision Tree (GBDT) and Neural Network (NN). Since the NN performed well, they continued NN as a function in their simulation. They recommended that MCTS leads to stronger lineups of heroes in predicting the win rate. The limitation of this research work is that they did not consider certain player information. For example, players' skill levels with their chosen hero were not considered. Secondly, only well-versed players were utilized during the match simulations, so other players with other skill levels could not be determined [15].

Hong et al. presented a champion recommendation system using the pick-and-ban system in LoL. The authors used professional matches to create a dataset defining the pick and ban system and then used two different models for training namely RFT and NN. They concluded that neural networks perform better than RFT and also explained why the performance of both models was different by analyzing the sequences separately [16].

Carlos et al. aimed to apply classification algorithms to predict the winning team in MOBA games, specifically DOTA 2. They took two classification methods; first based on the composition of heroes in each team and second considering the duration of the match. Their research effectively demonstrated the use of machine learning in the analysis of datasets focusing on MOBA games [17].

Research has also emphasized the importance of choosing the right heroes for a team in Dota 2. Various machine learning models including GBC, RFC, Linear Regression (LiR), and SVM were used for predicting the outcome of a match based on the selection of heroes. The research concluded that out of all algorithms, LiR, Linear Support Vector Classifier, and NN with activation functions Softplus and sigmoid performed efficiently, but LiR was found to be the fastest therefore, making it best for practical implementation [18].

Junior and Campelo presented a study on the prediction of outcomes in matches in LoL. They used various models for predicting the outcome and concluded that in different stages of the match, the model performance varies, in the early stages of the match LoR and GB models were effective whereas, for the intermediate stage of the game LightGBM showed the best performance achieving an accuracy of 81.62% [19].

Tangniyom and Boonma explore the complexity of the hero selection in Realm of Valor (RoV). They compared four machine learning algorithms KNN, LoR, DT, and ensemble learning with their optimized parameters. The authors concluded that the winning prediction accuracy can be improved by considering three features that are only available in RoV, namely Hero Selection, Synergy, and Counter. They also mentioned that ensemble learning can outperform individual algorithms [20].

Kim, Prabhakar, and Dutta proposed a recommendation system for selecting champions in LoL. The authors composed a recommendation system to maximize the win rate by computing the unbiased synergy and counter relationships normalized by popularity. They utilized Riot API to collect data and the FP-Growth algorithm for association rule mining. To evaluate the system, they proposed novel metrics of Composite Win Rate and upper hand. They concluded that their recommendation system achieved 80.79% average Upper hand [21]. A recommendation system for selecting heroes in MOBA games was proposed in [22]. They implemented NN to predict the winning team and got an accuracy of 88.63%. To demonstrate a recommendation system in real gameplay scenarios, they simulated matches and got a 74.9% success rate. For future work, they intend to perform more experiments and analysis on their system by evaluating more metrics and other mechanisms for improved results.

Researchers also investigated how teammates affect players' performance in both the short and long term in online games, specifically DOTA 2. The authors proposed a computational framework to recommend teammates improve the performance of players using a modified deep neural autoencoder and demonstrated that DNN can be used to predict the skill transfer between players. They concluded that their proposed model significantly outperformed baseline models in predicting skill gain and also in recommending influential teammates. For future work, they proposed to extend their framework by using multiple other aspects of the game that can influence individual performance. Secondly, they intend to determine whether their framework can be applied to a broader range of scenarios that are beyond online games. Thirdly they intend to investigate the use of tensor-based factorization techniques for better performance. Lastly, they plan to conduct randomized control trials to test the recommendation in real-world settings and explore incentive-based strategies to motivate players [23].

From the reviewed literature, most research primarily focuses on DOTA 2 and LoL with very few discussions available on MLBB. Most of the win rate predictions are based on past tournaments or experience of players' skills. No point has been raised if the roles of heroes are changed, which can be very significant for wins. MLBB depends on strategy, so introducing and exploring the element of surprise is very crucial in MLBB-based research. Furthermore, the concept of switched hero roles remains unexplored. The prediction of the win rate and team recommendation based on the switched roles of heroes makes this research unique. This distinct element of surprise will help coaches, players, and data analysts to form a team that will have switched roles of heroes in it.

Research Methodology:

The research framework for this study has been presented in Fig. 2. The first section consists of synthesizing the dataset, the second section outlines the data preprocessing steps, the third section includes machine learning, the fourth section describes the model-building approaches, and the fifth section describes the evaluation parameters based on which the best model building approach was selected along with the top 3 performing algorithms. Finally, in the last section, the teams were formulated based on the best model-building approach. The formulated teams were validated through the simulation of matches.

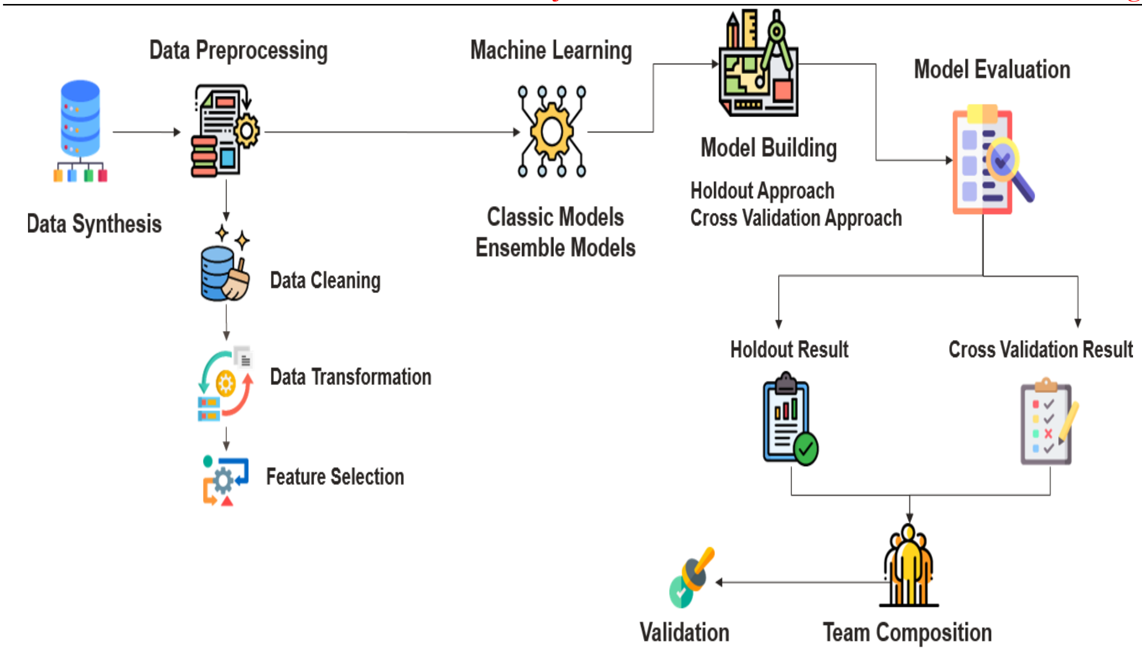

Figure. 2. Research Methodology

Dataset:

Due to the introduction of switched roles, the exact dataset needed for this research was not available. To synthesize the needed data for this research, we have used 67 heroes out of 125 heroes from January 2024 – June 2024. There are 6 categories of roles: tank, fighter, assassin, mage, marksman, and support. We took heroes from each category and utilized a total of 10 tanks, 13 fighters, 13 assassins, 13 mages, 13 marksmen, and 5 support. Each hero was assigned to all six roles, not just one. The pre and post-stats were determined by the characteristics of each hero in every role. As our research is focused on supervised machine learning, 20 simulation matches were played for each hero in every role the determination of the win rate. To avoid biased results, we kept the team constant throughout every match played. Our dataset consists of 403 entries with 46 feature variables.

Data Pre-Processing:

After the synthesis of the dataset, the data cleaning step was applied to check for any missing values. For accurate prediction of the win rate and to improve the performance of the models, a series of steps for the transformation of data were applied. First, the dataset was composed in Excel format and then loaded in the code using the Pandas library. After loading the dataset, it was separated into features such that X = the characteristics of heroes in every role and the target variable and Y = the win rate. After data transformation, 5 configurations were put forward for the experiments:

- Without Feature Selection (WFS): we did not use any feature selection techniques and all dataset features were used in the training step for the generation of a machine learning model.
- Filter methods: Relevant features were selected before the model training. SelectKBest (SB) method was taken amongst filter methods through which 10 key parameters were identified.
- Wrapper methods: Wrapper methods were used to select features based on feature subsets. [24]. The Recursive feature elimination (RFE) method was explored through which 10 key parameters were identified.
- Ensemble methods: This approach combined multiple models to improve classification accuracy. AdaBoost as a base estimator was taken amongst ensemble methods.

- Filter Ensemble method: A hybrid approach that leverages the strengths of both filter methods and ensemble methods to enhance feature selection and model performance. We explored the SB method from the filter methods in which it took 10 key parameters first and then AdaBoost as a base estimator was applied.

Machine Learning:

We experimented using 12 algorithms of which 4 were ensemble algorithms and 8 were classic machine learning algorithms. The working mechanism of the algorithms explored in this research is as follows:

- Classification And Regression Tree (CART) is used for both classification as well as regression problems. Regression tree structure contains a root, leaves, nodes, and branches and is produced by binary recursive partitioning. In recursive partitioning the data is split into small segments each is selected based on the minimum mean sum of squares among all segments [25]. The main advantage of using RT is the ease of readability. They not only predict target values but also help explain what attributes are used and how they contribute to the prediction.
- Linear Regression (LiR) is a statistical approach for the prediction of a dependent variable based on a single predictor variable [26]. This algorithm finds the relationship between two variables.
- Bayesian Ridge Regression (BRR) falls under the category of LR and uses probability to estimate predictions. It consists of a regularization technique to prevent the model from overfitting.
- Ridge Regression (RR), also known as L2 regularization, is a type of regularization for LR. It is a statistical regularization technique that helps prevent overfitting and can also be applied to Logistic Regression. RR was introduced to address the problem of multicollinearity in the analysis of regression. It adds a penalty on the residual sum of squares that shrinks the regression coefficients to zero thus transforming least squares estimation. This results in the variance of the coefficients being reduced and leads to more stable and reliable estimations [27].
- Least Absolute Shrinkage and Selection Operator (LaR) commonly known as Lasso regression is a L1 regularization technique, one of the regularization types for LR. It is a statistical regularization technique that helps to prevent overfitting by applying a penalty to ordinary least squares regression that shrinks the coefficient of less important variables to zero and therefore enhances the accuracy of statistical models. The advantages of LASSO regression are reduced overfitting, handling multicollinearity, and improvement in model generalization [28].
- Support Vector Machine with Poly Kernel (SVM Poly) is a non-linear kernel that deals with non-linearly separable datasets [29]. It utilizes a polynomial function to transform the data into higher-dimensional space.
- Support Vector Machine with Radial Basis Function (SVM RBF) deals with a non-linearly separable dataset as SVM with poly kernel the exception is that it uses Gaussian function [29]. It is also known as the Gaussian kernel and is used for both task classification and regression.
- Support Vector Machine using Sigmoid Kernel (SVM sigmoid) is related to the NN activation function.
- Extreme Gradient Boosting (XGB) is based on GBDT. XGB creates trees parallel instead of sequentially. XGB is based on a level-wise strategy. It utilizes L1 (Lasso) and L2 (Ridge) regularization techniques to control overfitting. The key advantages of XGB are: It can handle big amounts of data (Scalability); Due to its parallel and distributed computing capabilities, its performance and speed are enhanced [30].

- In Gradient Boosting (GB), to provide the estimation accurate new models are fit consecutively [31]. In GB several weak models, typically decision trees, are combined to build stronger models. To improve the accuracy of the model, training is done on the newer model by correcting the errors from the previous model. GB algorithms can be used to enhance the accuracy and prevent the model from overfitting by regularization techniques. They may also be used for classification and regression tasks.
- Random Forest Tree (RFT) is an Ensemble algorithm that is used for both classification as well as regression tasks. Random forest tree creates multiple decision trees based on class variable numerical values [32] and then merges them to get a high score in accuracy as well as prediction. In RFT each tree is trained with the selection of various randomly generated subsets of data that lead to the creation of uncorrelated trees and then those trees are combined into a single result, reducing the risk of overfitting and providing precise results.
- Light Gradient Boosting Machine (LightGBM) is a GB framework that uses tree-based learning algorithms, its implementation also proposes new features that are Gradient Based One Side Sampling (GOSS) and Exclusive Feature Binding (EFB) [33]. An advantage when using LightGBM is that the training is done faster and with more precision. LightGBM comes with several other advantages which include but are not limited to: increased accuracy and lower storage use.

Model Building:

The experiments for this research focused on two model-building approaches: i) Hold out Approach - where we split data into 80% training and 20% testing, and ii) the K-fold Cross Validation approach - where we took 5 folds to determine the results of cross-validation. In the holdout approach, the dataset was divided into a set used to train the classification model, and a set used to evaluate how well the model can perform on unseen data. In the K-Fold approach, the data was randomly split into 5 folds. One fold was used as a testing set and the remaining K-1 folds were used as a training set. The process was repeated until each fold was used as a testing set.

Model Evaluation:

For performance evaluation, the following parameters were considered:

- Mean Absolute Error (MAE) is calculated when the absolute differences between the predicted and actual values are averaged. The model performs better if the MAE value is less. The calculation of MAE is given in (1):

$$MAE = \frac{\sum_{i=1}^{n} |yi - \hat{y}i|}{n} \quad (1)$$

- Mean Squared Error (MSE) is the result of the differences between predicted and actual values squared. It highlights the importance of reducing large errors by penalizing these errors. Similar to MAE, the model performs better if the MSE value is less. MSE can be calculated using (2).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2 \quad (2)$$

- R^2 shows how well a model can predict the outcome of a dependent variable. The greater the R^2 value the more accurate the model. The formula for calculating R^2 has been provided in (3):

$$R^2 = 1 - \frac{\sum(yi - \hat{y}i)^2}{\sum(yi - \bar{y})^2} \quad (3)$$

Team Recommendation:

After the evaluation of the generated models, the win rate predictions of the best-performing algorithm were taken into consideration to compose 5 teams.

Validation:

To validate these 5 teams composed by the best algorithm, we simulated 50 matches and to avoid biased results we kept the team constant throughout the simulations of every match we played.

Experimental Results and Discussion

We compared results based on MAE, MSE, and R^2 values given by both approaches as can be seen in Table 1, Table 2, and Table 3.

Table 1. Experimental Results MAE

Algorithm	Hold-out Result					Cross Validation Result				
	WFS	**SB**	**RFE**	**Ada**	**Ada. SB**	**WFS**	**SB**	**RFE**	**Ada.**	**Ada. SB**
RT	0.156	0.194	0.156	0.165	0.174	0.170	0.1875	0.1712	0.1625	0.1703
LIR	0.168	0.175	0.180	0.187	0.179	0.365	0.1727	0.2471	0.2652	0.1816
BRR	0.173	0.178	0.178	0.178	0.171	0.180	0.1798	0.1833	0.2120	0.1798
RR	0.163	0.175	0.179	0.167	0.177	0.209	0.1727	0.1842	0.1865	0.1754
LAR	0. 176	0.180	0.177	0.179	0.180	0.185	0.1765	0.1813	0.2018	0.1772
SVM POLY	0.173	0.181	0.169	0.173	0.181	0.201	0.182	0.244	0.212	0.180
SVM RBF	0.169	0.178	0.168	0.172	0.178	0.175	0.173	0.174	0.183	0.183
SVM SIGMOID	0.371	1.848	1.063	1.131	0.682	0.461	1.303	1.03	0.91	0.910
XGB	0.144	0.157	0.167	0.141	0.161	0.165	0.1585	0.1445	0.1640	0.1635
GB	0.146	0.160	0.152	0.138	0.159	0.138	0.1600	0.1409	0.1438	0.1628
RFT	0.137	0.152	0.142	0.138	0.158	0.135	0.1586	0.1387	0.1604	0.1616
LIGHTGBM	**0.134**	0.158	0.141	**0.131**	0.157	**0.134**	0.165	0.151	0.146	0.1646

Table 2. Experimental Results MSE

Algorithm	Hold-out Result					Cross Validation Result				
	WFS	**SB**	**RFE**	**Ada.**	**Ada. SB**	**WFS**	**SB**	**RFE**	**ADA.**	**Ada. SB**
RT	0.045	0.063	0.044	0.038	0.0403	0.0513	0.0636	0.0491	0.0372	0.0393
LIR	0.044	0.045	0.046	0.049	0.0470	3.5065	0.0432	0.6539	1.2581	0.0557
BRR	0.043	0.046	0.046	0.046	0.042	0.0550	0.0541	0.0537	0.2813	0.0541
RR	0.042	0.045	0.044	0.041	0.044	0.1113	0.0432	0.0513	0.0503	0.0429
LAR	0.044	0.045	0.044	0.043	0.045	0.0565	0.0436	0.0470	0.1424	0.0436
SVM POLY	0.046	0.047	0.042	0.046	0.050	0.166	0.060	1.294	0.154	0.052
SVM RBF	0.043	0.045	0.047	0.040	0.046	0.046	0.044	0.046	0.047	0.047
SVM SIGMOID	0.230	7.630	1.929	1.777	1.011	0.619	4.620	2.90	1.18	1.180
XGB	0.034	0.038	0.043	0.030	0.038	0.0421	0.0390	0.0325	0.0396	0.0402
GB	0.033	0.040	0.037	0.030	0.037	0.0306	0.0403	0.0319	0.0310	0.0404
RFT	0.030	0.036	0.032	0.028	0.038	**0.0286**	0.0384	0.0309	0.0377	0.0402
LIGHTGBM	0.028	0.039	0.030	**0.027**	0.038	0.0288	0.042	0.037	0.032	0.0410

Table 3. Experimental Results R^2

Algorithm	Hold-out Result					Cross Validation Result				
	WFS	**SB**	**RFE**	**Ada**	**Ada. SB**	**WFS**	**SB**	**RFE**	**ADA**	**Ada. SB**
RT	-0.014	-0.413	0.007	0.157	0.0970	-0.1752	-0.4107	-0.1197	0.1543	0.1060

LIR	0.0155	-0.0123	-0.029	-0.092	-0.0528	-64.8426	0.01224	-13.7909	-27.1950	-0.277
BRR	0.030	-0.021	-0.021	-0.029	0.063	-0.2464	-0.2393	-0.2348	-5.3401	-0.239
RR	0.058	-0.012	0.010	0.090	0.019	-1.2965	0.0123	-0.1640	-0.1482	0.0221
LAR	0.019	-0.017	0.019	0.030	-0.005	-0.2598	0.0086	-0.0727	-2.4941	0.0061
SVM POLY	-0.032	-0.046	0.049	-0.041	-0.126	-2.317	-0.421	-28.956	-2.115	-0.200
SVM RBF	0.039	-0.045	0.003	0.095	-0.036	-0.0483	-0.001	-0.067	-0.072	-0.072
SVM SIGM	-4.157	-170.0	-42.512	-38.81	-22.362	-13.65	-108.55	-72.92	-25.88	-25.88
XGB	0.245	0.140	0.039	0.320	0.140	0.0542	0.1121	0.2600	0.1060	0.0830
GB	0.255	0.103	0.161	0.334	0.160	0.3031	0.0820	0.2718	0.29	0.0784
RFT	0.326	0.183	0.277	0.371	0.155	**0.3510**	0.1368	0.2969	0.1508	0.0819
LIGHTGBM	0.369	0.130	0.324	**0.392**	0.158	0.345	0.046	0.160	0.262	0.071

Analyzing the values for MAE (see Table 1), the best value of MAE (0.131) was generated by LightGBM with AdaBoost as the base estimator, indicating the lowest number of errors in the prediction. The second-best value was observed in the model generated using LightGBM WFS (0.134), and the third-best value was observed in the model generated using RFT WFS (0.137). Analyzing MSE (see Table 2), the best value of MSE was generated by LightGBM with AdaBoost as a base estimator (0.027) indicating the fewest amount of errors in the prediction. The second and third best values were generated by LightGBM WFS (0.028) and RFT with AdaBoost as a base estimator (0.028).

The analysis of R^2 uncovered the best value of 0.392 by LightGBM with AdaBoost as a base estimator, indicating good accuracy in the prediction. The second-best value was given by RFT with AdaBoost as a base estimator (0.371) with the third-best value generated using LightGBM WFS (0.369). From the analysis of evaluation parameters, LightGBM with AdaBoost as a base estimator outperforms. Determination of the second-best algorithm was done based on consistency and trade-off. LightGBM WFS has performed consistently very well. In terms of R^2, LightGBM WFS is close to RFT with AdaBoost as the base estimator. In terms of MAE, LightGBM WFS gave 0.134 whereas, RFT exhibited a score of 0.138 very close to RFT WFS which gave a value of 0.137. Therefore, the value difference in terms of R^2 being very small at 0.001 can be neglected. In terms of MAE difference between LightGBM WFS and RFT with AdaBoost as the base estimator was recorded at 0.004. We can, thus, conclude that LightGBM WFS is the second best-performing algorithm followed by RFT with AdaBoost as a base estimator.

Discussion:

Comparing the results of Hold out with cross-validation, the only similarity observed from both approaches is that ensemble models are performing very well from classic regression algorithms, but there are two major differences in both approaches:

- BEST_VALUES: We can observe that the evaluation parameter values in the Holdout approach are better than the Cross Validation approach.

MAE: Holdout: 0.131, Cross Validation: 0.134
MSE: Holdout: 0.027, Cross Validation: 0.0286
R^2: Holdout: 0.392, Cross Validation: 0.351

- TRADE-OFF: In the cross-validation approach we cannot determine the results, according to evaluation parameters, the best MAE value is given by LightGBM WFS i.e. 0.134, whereas the best MSE and R^2 values are given by RFT WFS. For a model to be considered good, it should give results without any trade-off but here, the trade-off becomes evident across MAE, MSE, and R^2, making it hard to establish the final result. In the Holdout approach, LightGBM with AdaBoost as base estimator has performed consistently and gave values of MAE, MSE, and R^2 better.

From these differences, it is quite clear that the Holdout Approach has achieved consistently better results specifically while working with LightGBM with AdaBoost as a base estimator. Therefore, the cross-validation results were discarded, and holdout approach results were considered for further analysis.

In Table 4, we created 5 teams according to the win rate predicted by LightGBM with AdaBoost as a base estimator algorithm and played 50 matches with each team recommended by the algorithm. Angela's role is usually support but in TC-1, we have changed her role from support to mage. TC-1 gave 47 wins and 3 losses out of 50 total matches giving a win rate of 94%. Baxia's role is usually tank but in TC-2, we have changed his role from tank to assassin. TC-2 gave 45 wins and 5 losses out of 50 total matches giving a win rate of 90%. Karrie's role is usually marksman and Akai's role is usually tank but in TC-3, we have changed Karrie's role from marksman to tank and Akai's role from tank to assassin. TC-3 gave 44 wins and 6 losses out of 50 total matches giving a win rate of 88%. Brody's role is usually marksman but in TC-4, we have changed his role from marksman to fighter. TC-4 gave 40 wins and 10 losses out of 50 total matches giving a win rate of 80%. Marti's role is usually fighter but in TC-5, we have changed his role from fighter to assassin. TC-5 gave 40 wins and 10 losses out of 50 total matches giving a win rate of 80%.

Table 4. Team Recommendations

Role	Team Combination				
	TC-1	**TC-2**	**TC-3**	**TC-4**	**TC-5**
Match Win	47	45	44	40	40
Match Lost	3	5	6	10	10
Total	50	50	50	50	50
Win %	94%	90%	88%	80%	80%

After the simulation of 50 matches with each team based on the highest wins gained, we can conclude that the first team predicted by the algorithm that is TC-1 is the best team that gave the highest chances of victory with an overall win rate of 94%. The second team predicted by algorithm gave a result of 90%. Likewise, the third team combination showed an accuracy of 88%. The fourth and fifth teams gave the same results of a win rate of 80%.
There are several practical implications for exploring this line of research:

There is a wide possibility of improving team performance by using non-traditional roles. The use of switched roles has the potential to give rise to unconventional strategies that can greatly improve gameplay and establish stronger synergies. Exploring switched hero roles can also lead to meta diversification by pushing meta shifts; it can also lead to added competitive edge and greater variety in competitive play. Exploring switched roles can expand the skill set of a player and allow them to outmaneuver their opponents, especially ones slow to adapt. Understanding switched roles can also lead to a more enriched experience with gaming strategies being tailored more effectively during real-world competitions. This will also significantly affect the decision-making process during the real-world drafting phase of the competition in ranked matches in tournaments and will help create dynamic user engagement by removing predictability. eSports teams can leverage this approach to create unusual and novel team combinations providing their teams a strategic edge. For instance, players can use data-driven insights to select heroes with higher win rates in untraditional or switched roles to form better teams. eSports coaches could also use empirical data to fine-tune their training and gaming strategies. The exploration of win rate prediction using switched hero roles in MLBB will allow not only the players and coaches to gain critical insights regarding team composition and planning counter-strikes but also help the developers with strategic innovation and adaptive gameplay.

Conclusion:

As online games have piqued the interest of users, many different tournaments of online games are held. This research is based on a mobile platform online game, called Mobile Legends Bang Bang or MLBB. This research offers insights for players, coaches, and team data analysts. The goal of the research was to help pick heroes with switched roles, adding an element of surprise for the enemy. The research can also help with team formation with increased chances of victories even if some hero roles are converted from original to different ones. Out of 125 heroes available in this game, we used 67 to synthesize our dataset. We simulated matches with these heroes in which they were used in every role instead of just their one traditional role. This helped determine what role works best if the roles are to be changed and allowed us to recommend teams that ensure high chances of victory. We played 20 matches to determine the win rate with each hero according to the roles they were switched to. We used 2 different approaches with 12 different regression algorithms. For each algorithm, we applied 5 different methods, producing 5 variants of each algorithm. After the analysis, we concluded that the Holdout approach was better than the cross-validation approach with LightGBM with AdaBoost as a base estimator outperforming other algorithms. Based on the win rate prediction of LightGBM we formed the top 5 best performing teams. To solidify the estimation of the win rate, we played 50 matches and got the results that TC-1 gave 47 wins and 3 losses resulting in a win rate of 94%. In the future, the prediction of the win rate will be done using all the heroes from MLBB. Each hero will have their roles switched again and more algorithms with further feature selection methods will be utilized.

Author's Contribution: Pir Hamid Ali Qureshi: Writing – original draft, methodology, visualization, literature review, data extraction, result reporting. Areej Fatemah Meghji: Supervision, methodology, writing – editing and review, validation, result reporting. Rabeea Jaffari: Supervision, methodology, writing – editing and review, validation, result reporting

Conflict of interest: The authors declare they have no conflict of interest in publishing this manuscript in IJIST.

Project details: This research was not part of any project.

References:

[1] A. Turner, "How Many Smartphones Are in the World?," BankMyCell, Aug. 2023. [Online]. Available: https://www.bankmycell.com/blog/how-many-phones-are-in-the-world. [Accessed: Oct. 1, 2023].

[2] J. Degenhard, "Number of smartphone users worldwide 2013-2028," Statista, Aug. 14, 2023. [Online]. Available: https://www.statista.com/forecasts/1143723/smartphone-users-in-the-world. [Accessed: Oct. 11, 2023].

[3] Statista, "Mobile Games - Worldwide | Statista Market Forecast," Statista, 2023. [Online]. Available: https://www.statista.com/outlook/dmo/digital-media/video-games/mobile-games/worldwide. [Accessed: Oct. 12, 2024].

[4] S. K. Lee, S. J. Hong, and S. I. Yang, "Predicting game outcome in Multiplayer Online Battle Arena Games," in 2020 International Conference on Information and Communication Technology Convergence (ICTC), 2020, doi: 10.1109/ICTC49870.2020.9289254.

[5] M. Lobanova, "What Makes Moba Games So Successful?," Hackernoon, Nov. 11, 2022. [Online]. Available: https://hackernoon.com/what-makes-moba-games-so-successful. [Accessed: Oct. 1, 2024].

[6] UniPin Blog, "Explanation of the Lane in Mobile Legends," https://blog.unipin.com/en/explanation-of-the-lane-in-mobile-legends/ [Accessed: Jan. 27, 2025]

[7] D. A. Team, "Unveiling the Mysteries of AI Supervised Learning: A comprehensive overview," Deep AI — Leading Generative AI-powered Solutions for Business, Jul. 14, 2024. Accessed: Oct 04, 2024 [Online]. Available: https://deepai.tn/glossary/supervised-learning-ai-explained/
[8] A. F. Meghji, N. A. Mahoto, Y. Asiri, H. Alshahrani, A. Sulaiman, and A. Shaikh, "Early detection of student degree-level academic performance using educational data mining," PeerJ Computer Science, vol. 9, pp. e1294, 2023, doi: 10.7717/peerj-cs.1294.
[9] K. Arık, "Machine learning models on MOBA Gaming: League of Legends Winner Prediction," Acta Infologica, vol. 7, no. 1, pp. 139–151, 2023, doi: 10.26650/acin.1180583.
[10] T. D. Do, S. I. Wang, D. S. Yu, M. G. McMillian, and R. P. McMahan, "Using machine learning to predict game outcomes based on player-champion experience in League of Legends," in Proc. 16th Int. Conf. Foundations of Digital Games (FDG), 2021, doi: 10.1145/3472538.3472579.
[11] A. S. Chan, F. Fachrizal, and A. R. Lubis, "Outcome prediction using naïve Bayes algorithm in the selection of role hero mobile legend," J. Phys.: Conf. Ser., vol. 1566, no. 1, p. 012041, 2020, doi: 10.1088/1742-6596/1566/1/012041.
[12] R. Ani, V. Harikumar, A. K. Devan, and O. S. Deepa, "Victory prediction in league of legends using feature selection and ensemble methods," in Proc. 2019 Int. Conf. Intell. Comput. Control Syst. (ICCS), 2019, doi: 10.1109/iccs45141.2019.9065758.
[13] D. Mahendra, "Decision Tree Application for Choosing Champions in League of Legends," no. 13518003, Dec. 2019. Accessed: Oct. 01, 2023. [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2019-2020/Makalah2019/13518003.pdf.
[14] M. Pengmatchaya and J. Natwichai, "Identifying player skill of dota 2 using machine learning pipeline," Discover Artificial Intelligence, vol. 4, no. 1, May 2024, doi: 10.1007/s44163-024-00139-y.
[15] Chen, Z., Nguyen, T. H. D., Xu, Y., Amato, C., Cooper, S., Sun, Y., & El-Nasr, M. S. (2018, September). "The art of drafting: a team-oriented hero recommendation system for multiplayer online battle arena games", In Proceedings of the 12th ACM Conference on Recommender Systems (pp. 200-208), doi: 10.1145/3240323.3240345.
[16] S.-J. Hong, S.-K. Lee, and S.-I. Yang, "Champion recommendation system of League of Legends," 2020 International Conference on Information and Communication Technology Convergence (ICTC), vol. 1215, pp. 1252–1254, Oct. 2020. doi:10.1109/ictc49870.2020.9289546
[17] C. E. M. Almeida, R. C. M. Correia, D. M. Eler, C. Olivete-Jr, R. E. Garcia, L. C. Scabora, and G. Spadon, "Prediction of Winners in MOBA Games," in Proceedings of the Department of Mathematics and Computing, Universidade Estadual Paulista (UNESP), São Paulo, Brazil, 2024.
[18] I. Porokhnenko, P. Polezhaev, and A. Shukhman, "Machine Learning Approaches to Choose Heroes in Dota 2," in Proc. 24th Conf. FRUCT Assoc., 2024, pp. 346-350.
[19] J. B. S. Junior and C. E. C. Campelo, "League of Legends: Real-Time Result Prediction," in Proc. of the IEEE, 2024, pp. 1-10.
[20] N. Tangniyom and P. Boonma, "A Study on Using Machine Learning to Predict Winner in Multiplayer Online Battle Arena (MOBA) Game," ASEAN Journal of Scientific and Technological Reports, vol. 27, no. 5, pp. e252289, 2024, doi: 10.55164/ajstr.v27i5.252289.
[21] Y. S. Kim, P. Prabhakar, and A. Dutta, "Champion Recommender System For League of Legends," University of Washington, Preprint.

[22] L. Hanke and L. Chaimowicz, “A Recommender System for Hero Line-Ups in MOBA Games,” in Proc. Thirteenth AAAI Conf. Artificial Intelligence and Interactive Digital Entertainment (AIIDE-17), 2017, pp. 43-49.
[23] A. Sapienza, P. Goyal, and E. Ferrara, “Deep Neural Networks for Optimal Team Composition,” Frontiers in Big Data, vol. 2, no. 14, pp. 1-13, Jun. 2019, doi: 10.3389/fdata.2019.00014.
[24] Elavarasan, D., Vincent PM, D. R., Srinivasan, K., & Chang, C. Y. (2020). A hybrid CFS filter and RF-RFE wrapper-based feature extraction for enhanced agricultural crop yield prediction modeling. Agriculture, 10(9), 400.
[25] Jadhav, P., Patil, D. V., & Gore, D. S. (2020, May). A comparative study of linear regression and regression tree. In 2nd International Conference on Communication & Information Processing (ICCIP).
[26] James, Gareth, et al. "Linear regression." An introduction to statistical learning: With applications in python. Cham: Springer International Publishing, 2023. 69-134.
[27] R. W. Hoerl, “Ridge Regression: A Historical Context,” Technometrics, vol. 62, no. 4, pp. 420-425, Oct. 2020, doi: 10.1080/00401706.2020.1742207.
[28] L. Zhang, X. Wei, J. Lu, and J. Pan, “Lasso Regression: From Explanation to Prediction,” Advances in Psychological Science, vol. 28, no. 10, pp. 1777-1788, 2020. doi: 10.3724/SP.J.1042.2020.01777.
[29] Jayanth, S. B., Anthony, A., Abhilasha, G., Shaik, N., & Srinivasa, G. (2018). A team recommendation system and outcome prediction for the game of cricket. Journal of Sports Analytics, 4(4), 263-273.
[30] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 2016, pp. 785-794, doi: 10.1145/2939672.2939785.
[31] A. Natekin and A. Knoll, "Gradient boosting machines, a tutorial," *Frontiers in Neurorobotics*, vol. 7, art. no. 21, 2013. DOI: [10.3389/fneur.2013.00021](https://doi.org/10.3389/fneur.2013.00021).
[32] J. K. Jaiswal and R. Samikannu, "Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression," 2017 World Congress on Computing and Communication Technologies (WCCCT), Tiruchirappalli, India, 2017, pp. 65-68, doi: 10.1109/WCCCT.2016.25.
[33] Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54, 1937-1967.