JATIKOM: Jurnal Aplikasi dan Teori Ilmu Komputer
ISSN: 2615-577X (Online)

Predicting Team Wins from Time-Snapshot Features in Professional League of Legends Matches: Machine Learning Models Against a Single-Indicator Baseline

Muhamad Nursalman[1], Mochamad Rizky Farhan Auliya[2], Herbert Siregar[3]

Program Studi Ilmu Komputer FPMIPA Universitas Pendidikan Indonesia
Jl. Dr. Setiabudi No. 229 Bandung, West Java, Indonesia

[1]mnursalman@upi.edu
[2]farhan16@upi.edu
[3]herbert@upi.edu

Abstract— **Studies on esports win prediction typically compare algorithms without a simple reference rule, and leakage audits are mostly limited to aggregate end-of-match features, leaving the temporal validity of objective indicators without timestamps untested. This study evaluates whether machine learning adds value over single-indicator baselines and audits the temporal validity of the features used. Using 555 matches from the 2025 League of Legends Champions Korea season, numeric difference features were taken as snapshots at minutes 10, 15, 20, and 25. Then, Logistic Regression, Random Forest, and XGBoost were evaluated with a chronological split. Accuracy rose from 64–68% at minute 10 to about 82% at minute 25, and AUC-ROC from 0.68–0.71 to 0.91, first reaching 0.85 at minute 25. No model exceeded the gold difference sign baseline beyond its confidence interval, and none differed significantly in AUC from a single-feature Logistic Regression on gold-difference after Bonferroni correction. Adding first Baron, which cannot be known at minute 25 because the objective spawns at that moment, raised AUC by 0.02–0.03, whereas end-of-match features reached 97–99% accuracy. Future work should extend the snapshots beyond minute 25 and replicate the analysis on other leagues and seasons.**

Keywords— Data leakage, Esports analytics, Machine learning, Temporal validity, Win prediction

Abstrak— ***Penelitian prediksi kemenangan esports umumnya membandingkan berbagai algoritma tanpa aturan pembanding sederhana dan audit kebocoran data, sebagian besar terbatas pada fitur agregat akhir pertandingan, sehingga validitas temporal indikator objektif tanpa penanda waktu belum teruji. Penelitian ini mengevaluasi apakah machine learning memberi nilai tambah dibandingkan baseline satu indikator sekaligus mengaudit validitas temporal fitur yang digunakan. Data berasal dari 555 pertandingan League of Legends Champions Korea musim 2025, dengan fitur selisih numerik disusun sebagai snapshot pada menit 10, 15, 20, dan 25. Lalu Logistic Regression, Random Forest, dan XGBoost dievaluasi dengan pembagian kronologis. Akurasi meningkat dari 64–68% pada menit 10 menjadi sekitar 82% pada menit 25, dan AUC-ROC dari 0,68–0,71 menjadi 0,91 dengan ambang 0,85 pertama kali tercapai pada menit 25. Tidak ada model yang melampaui baseline tanda selisih gold di luar selang kepercayaannya, dan tidak ada yang berbeda signifikan dalam AUC dari Logistic Regression satu fitur (selisih gold) setelah koreksi Bonferroni. Penambahan first Baron, yang tidak dapat diketahui pada menit 25 karena objektif tersebut baru muncul pada saat itu, menaikkan AUC sebesar 0,02–0,03, sedangkan fitur akhir pertandingan mencapai akurasi 97–99%. Penelitian lanjutan disarankan memperluas snapshot melewati menit 25 dan mereplikasi analisis pada liga dan musim lain.***

***Kata kunci*— Analitik esports, Kebocoran data, Machine learning, Prediksi kemenangan, Validitas temporal**

I. INTRODUCTION

League of Legends (LoL) is one of the most widely played titles in the Multiplayer Online Battle Arena (MOBA) genre, in which two teams of five players compete to destroy the opposing team's Nexus. Teams accumulate gold and experience during the match and secure objectives such as dragons and the Baron, which convert into combat advantages. The popularity of the game has made it a frequent subject of qualitative and quantitative research [1].

Professional matches generate detailed statistics, including gold differentials, experience leads, objective control, and vision scores, which are recorded through official and third-party data sources. These data make professional MOBA matches well suited to machine learning, particularly for real-time outcome prediction at the international level [2]. Research on Dota 2 shows that predictive accuracy increases as a match progresses, ranging from 58.7 to 71.5 percent when only pre-match data are used and reaching 93.7 percent when real-time data up to minute 40 are included [3].

A systematic review of thirty-five machine learning studies in MOBA esports identifies Dota 2 and LoL as the most frequently studied titles [4]. These studies apply

Random Forests, Decision Trees, Logistic Regression, and Neural Networks to game mechanics and player profiles, and their accuracy depends strongly on the information supplied to the model. Models that rely only on pre-match data tend to achieve lower accuracy: a neural network trained on player-champion experience reaches about 75 percent [5], and models based on past match performance fall in a similar range [6]. Combining pre-match and in-game data raises accuracy to 76.8 percent in a comparison of linear and ensemble models [7]. Models using late-game data report the highest accuracy [3], but this raises a methodological concern [7]: features recorded shortly before a match ends may reflect an outcome that is already nearly determined rather than support prediction, so high accuracy may indicate data leakage rather than model quality.

In the Indonesian context, research attention has centred on Mobile Legends: Bang Bang (MLBB). Prior studies applied regression-based machine learning to the Malaysian professional league [8] and Naive Bayes to the Indonesian league within the CRISP-DM framework [9], [10]. The closest study to the present work examined the 2025 Mobile Legends Southeast Asia Cup using Random Forest and XGBoost [11]. It contrasted the draft phase with in-game snapshots of gold, kills, turrets, and objectives, and found that draft-based models reached only 57 percent accuracy, whereas in-game snapshot models reached 88 percent (Random Forest) and 84 percent (XGBoost) with an AUC of 0.94.

Despite this progress, two limitations remain across the literature, including the studies cited above. First, model performance is usually compared across algorithms rather than against a simple rule based on a single indicator. None of the accuracy figures quoted above is accompanied by such a reference point, so it is unknown how much of each figure could have been obtained from a single resource difference, and therefore whether model complexity adds value beyond the heuristics that practitioners already use. A similar pattern appears in a classification study conducted by Hambali et al. [12], in which the proposed network is compared with library implementations of the same model family on a single random 80/20 split, without a majority-class reference or a significance test. Second, data leakage is normally associated with aggregate end-of-match features [8], whereas binary objective indicators in public datasets carry no timestamp. A value such as first Baron states whether a team secured that objective at any point in the match, not the state of the match at a given minute, so including it in an earlier snapshot risks injecting information that was not available at prediction time. In addition, most existing studies use a single point in time, such as the pre-match draft, a single in-game snapshot, or end-of-game summaries [8], [9], [11], and few trace how predictive performance develops within a single match.

This study addresses these gaps by benchmarking three models against single-indicator references, identifying the most informative indicators, tracing how predictive performance develops across four snapshots, and auditing the temporal validity of the features used. Three research questions are posed. First, do machine learning models add value over single-indicator references, and which statistical indicators are the most dominant? Second, at which minute does predictive discrimination first reach a pre-specified threshold (AUC-ROC of at least 0.85), and has performance levelled off by then? Third, are features whose acquisition time cannot be verified, whether aggregate end-of-match features or objective indicators without timestamps, genuinely predictive, or do they merely reflect an outcome that has already occurred? The scope is limited to the 2025 season of the League of Legends Champions Korea (LCK), obtained from oracleselixir.com, covering the LCK Cup, Rounds 1 to 2, and Rounds 3 to 5, with timestamped numeric difference features taken as snapshots at minutes 10, 15, 20, and 25. The study is intended for esports analysts, coaches, casters, and fans who require real-time analysis, and contributes to the esports analytics literature.

II. METHODS

A. Research Stage

The study proceeds through six stages, summarised in Figure 1. The first stage acquires professional match data for the 2025 LCK season from Oracle's Elixir. The second stage prepares the data by restricting it to team-level rows, retaining a single row per match, and verifying completeness. The third stage constructs snapshot feature sets at minutes 10, 15, 20, and 25 from timestamped numeric difference features, with every objective indicator screened against its acquisition window and excluded from the main models wherever its timing cannot be verified. The fourth stage partitions the data chronologically and trains Logistic Regression, Random Forest, and XGBoost separately for each snapshot. The fifth stage evaluates the resulting twelve model and time combinations against a majority-class baseline, a single-indicator sign rule, and a single-feature Logistic Regression, using accuracy, AUC-ROC, and the Brier score. The sixth stage interprets the models through SHAP values and permutation importance, and audits temporal validity by comparing the snapshot models against models built on aggregate end-of-match features and by adding first Baron and the Atakhan difference to the minute-25 snapshot. The subsections that follow describe each stage in detail.

B. Data and Preprocessing

This study uses data from Oracle's Elixir for the 2025 season, restricted to LCK matches, yielding 555 unique matches between 15 January and 28 September 2025. This range spans three competitive phases, namely the LCK Cup, Rounds 1 to 2, and Rounds 3 to 5, and thirteen game patches from 15.01 through 15.17.

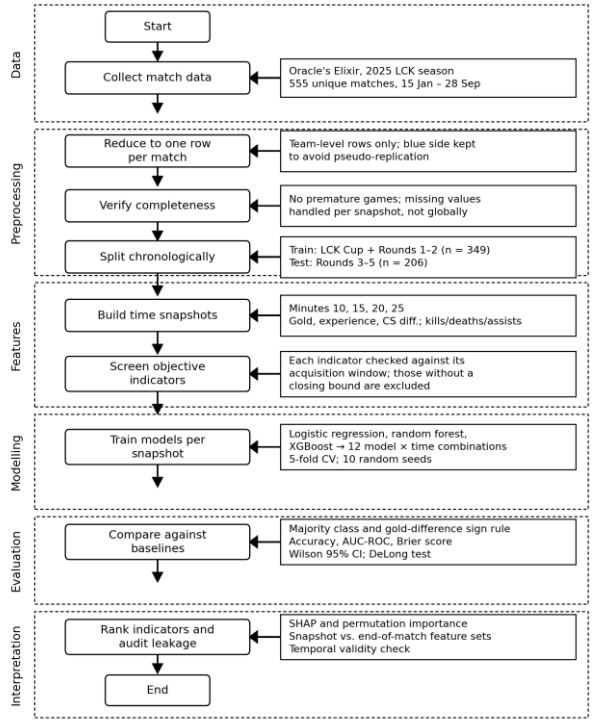

Figure 1. Research Workflow.

Each row in the source dataset contains either per-player or per-team statistics for a match. Only team-level rows are used. The gold, experience, and CS differences provided in the dataset are taken directly, and kill and assist differences are computed as the team's count minus the opponent's. Each match contributes two team rows, blue side and red side, that mirror one another. Treating both as independent samples would introduce pseudo-replication and risk leakage between the training and test partitions. Preprocessing therefore retains a single row per match, namely the blue side. All difference features are oriented from the same perspective, and the target variable result takes the value 1 when the blue side wins and 0 when it loses. The blue-side win rate of 50.99 percent indicates a relatively balanced class distribution and serves as the majority-class baseline.

Further preprocessing began with a completeness check for prematurely ended games, typically caused by a remake. None were found: all rows were marked complete and the shortest match lasted 21.9 minutes, so no match was discarded. Missing features were handled per snapshot rather than globally, so that matches were not dropped from snapshots for which they were complete. A match that ended before a snapshot minute is excluded from that snapshot, because the value recorded for that minute would describe the end state of the match rather than its state at that minute. The binary indicators used in the sensitivity analysis were already stored as 0 and 1 and required no encoding. Numeric features were standardised for Logistic Regression only, with parameters computed from the training partition alone so that no statistical information leaked into the training process.

C. Feature Construction and Temporal Validity

Each snapshot model uses only information recorded with a timestamp at or before the corresponding minute. The main feature set therefore consists of five numeric difference features, namely gold, experience, CS, kill, and assist differences, all oriented from the blue-side perspective and recorded at minutes 10, 15, 20, and 25 (Table I). Deaths are not entered separately, since the deaths of one team equal the kills of its opponent. Because the feature set is identical at every snapshot, differences in performance across snapshots can be attributed to the sampling time alone. Objective indicators are treated differently. The dataset records them only as whether a team secured the objective at any point in the match, with no timestamp attached. Whether such a flag reflects the state of the match at a given minute depends on spawn and closing times that the dataset does not store and that changed within the season: patch 15.09 (announced by Riot as patch 25.09) moved the first void grub spawn from 6:00 to 8:00, removed the second grub group, and moved the Rift Herald spawn from 16:00 to 15:00 [13], consistent with the fall in the mean number of void grubs per team from 3.00 to 1.50 at that patch. The patch notes do not document when these objectives close; a community wiki lists a despawn at 14:45 for the grub camp [14], but this is not a primary source and the value may have changed within the season. First blood and first dragon likewise cannot be assigned to a minute. No binary objective indicator is therefore used in the main models. First blood, first dragon, the void grub difference, and first herald are instead added in a sensitivity analysis, at every snapshot after the objective has spawned (Table I), to test whether the conclusions depend on their exclusion. First tower, Atakhan, and first Baron have acquisition windows that remain open past every snapshot and are excluded from both analyses. First Baron and Atakhan appear only in the leakage audit, which adds them to the minute-25 snapshot to quantify the effect of an unverifiable feature. Figure 2 shows the acquisition window of each objective against the snapshot points.

TABLE I
FEATURES AND TIME SNAPSHOTS USED

Feature	Acquisition window	Main Model	Sensitivity analysis
Gold, experience, CS difference	Recorded at time of occurrence	10, 15, 20, 25	10, 15, 20, 25
Kill, assist difference	Recorded at time of occurrence	10, 15, 20, 25	10, 15, 20, 25
First blood	Relative to time of occurrence	not used	10, 15, 20, 25 (assumed)
First dragon	Spawns at 5:00, no closing boundary	not used	10, 15, 20, 25

Feature	Acquisition window	Main Model	Sensitivity analysis
Void grub difference	Spawns 6:00 (8:00 from patch 15.09); closing time not in patch notes	not used	10, 15, 20, 25
First herald	Spawns 16:00 (15:00 from patch 15.09); closing time not in patch notes	not used	20, 25
First tower	Until the first tower falls	not used	not used
Atakhan	20:00 to end of match	not used	not used
First Baron	25:00 to end of match	not used	not used

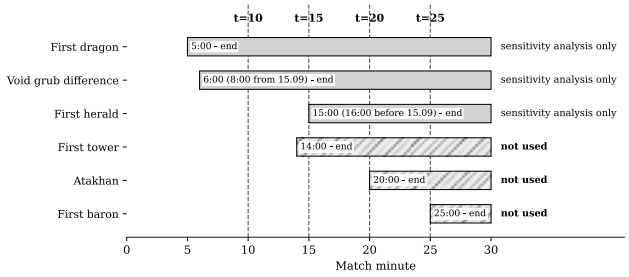

Figure 2. Objective acquisition windows against snapshot points.

Sample sizes are not uniform, since fifteen matches ended in under 25 minutes, leaving 540 samples for the minute-25 snapshot (201 in the test partition). Such matches typically show a highly lopsided statistical profile between the winning and losing team, producing right censoring. Cross-snapshot comparisons are therefore also reported on the common subset as a sensitivity analysis.

Features that only materialise at the end of a match, namely damage per minute, total gold, vision score, match length, and the differences in towers, dragons, and barons, are not used as primary indicators but as a comparison set for testing data leakage [15]. A feature set is regarded as leaking if its values can only be known after the match has ended, so that they are unavailable at prediction time, and if models built on it reach a level of performance unattainable by any snapshot model. A recent survey documents leakage across seventeen scientific fields and organises it into a taxonomy of eight types [16], which places the audit reported here within a broader methodological concern rather than a peculiarity of esports data. The target variable of this study is result, where 1 denotes a win and 0 a loss.

D. Models

Three models representing a range of complexity from simple to complex are used. All three receive the same snapshot feature vector and return a win probability, so that any difference in performance is attributable to how each model represents the relationship between features and outcome rather than to differences in the information available to it.

Logistic regression is a linear classification model that expresses the probability of a class, in this context a team win, as a function of a linear combination of the input features. It serves as a baseline because its coefficients can be interpreted directly as the direction and magnitude of each feature's influence on the probability of winning, as summarised in Figure 3.

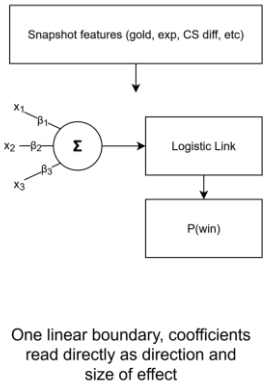

Figure 3. Logistic Regression Schematic.

Random Forest is an ensemble model that builds a large number of decision trees independently on randomised subsets of samples and features, then aggregates the predictions of all trees through majority voting [17]. This approach is relatively robust to collinearity among features and provides a built-in feature importance measure in the form of Gini importance, as illustrated in Figure 4.

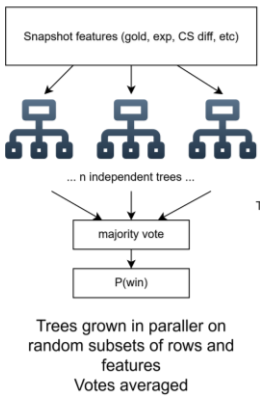

Figure 4. Random Forest Schematic

XGBoost (Extreme Gradient Boosting) is based on the gradient boosting framework [18] and builds decision trees sequentially, with each new tree trained to correct the errors of its predecessors. It adds regularisation mechanisms to limit overfitting [19]. It serves as the reference contemporary model in comparable studies [7] and is

compatible with SHAP value analysis [20] for deeper feature interpretation, as illustrated in Figure 5.

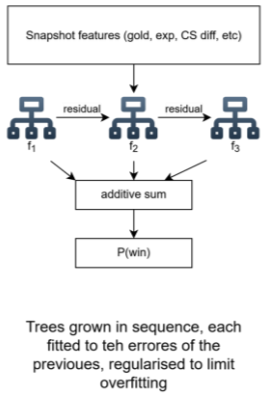

Figure 5. XGBoost Schematic.

Models are trained separately for each snapshot, namely minutes 10, 15, 20, and 25, producing a total of twelve model and time combinations whose performance can be compared, together with the numeric-only feature variants described above.

E. Validation Scheme

The training and test partitions follow the chronological order of the competitive season rather than a random split. The LCK timeline runs from the LCK Cup, held from 15 January to 23 February 2025, through Rounds 1 to 2, held from 2 April to 15 June 2025, to Rounds 3 to 5, held from 23 July to 28 September 2025. Training uses the LCK Cup together with Rounds 1 to 2 (n = 349; 339 at minute 25), while testing uses Rounds 3 to 5 (n = 206; 201 at minute 25). This choice reflects the fact that the LCK Cup falls at the start of the season, immediately after the off-season roster changes, when team performance is not yet stable and is therefore less representative of the settled rosters seen once the season is under way. Stratified five-fold cross-validation with shuffling is applied within the training partition for hyperparameter tuning, with AUC-ROC as the selection metric.

A second analysis examines whether the observed patterns hold in the play-in and playoff stages, which use a best-of-five format rather than the best-of-three used in the regular rounds. No separate training scheme is used. The models fitted on the chronological training partition are evaluated separately on the regular-season matches (n = 145, or 141 at minute 25) and the playoff matches (n = 61, or 60 at minute 25) of the test partition, identified through the playoffs column. This analysis relates to the problem of dataset shift [21] and asks whether the statistical signature of winning and losing teams holds under the elimination pressure of the playoffs. It is exploratory, since the playoff subgroup is small.

F. Baselines and Evaluation

So that the added value of the models can be assessed fairly, performance is compared against three references: the majority-class baseline; a naive rule based on a single feature, which predicts a win for whichever team leads in gold at the given minute; and a Logistic Regression fitted on gold difference alone, which serves as a probabilistic reference because the sign rule yields only a binary decision. Table II gives the sample size and the accuracy of the sign rule over the full sample at each snapshot, and Table IV gives all references on the test partition.

TABLE II
SAMPLE SIZE AND NAÏVE BASELINE PERFORMANCE FOR EACH SNAPSHOT

Snapshot	**Sample size**	**Gold difference sign baseline**
Minute 10	555	67.2%
Minute 15	555	71.5%
Minute 20	555	77.7%
Minute 25	540	82.6%

Evaluation uses accuracy, precision, recall, F1-score, and AUC-ROC, together with the Brier score [22] as a measure of the quality of the predicted probabilities, since the output intended for analysts, coaches, and casters is a real-time win probability. Class labels are assigned at a probability threshold of 0.5. Given the limited size of the test partition, each accuracy figure is reported with a 95 percent Wilson confidence interval, and AUC comparisons are tested with the DeLong test as implemented by [23], because two families of twelve comparisons are made, namely between models and between each model and the gold-only regression, a Bonferroni correction is applied within each family (threshold 0.05/12 = 0.0042). Hyperparameters are selected once for each model and snapshot, and the selected configuration is refitted with ten random seeds. Reported metrics are means over seeds, and the DeLong test and the subgroup analyses use the probabilities averaged over seeds. A model is said to exceed the sign rule only if the lower bound of its accuracy confidence interval lies above the accuracy of the rule.

Two separate questions are asked of the AUC-ROC trajectory. The first is the earliest snapshot at which the test AUC-ROC reaches a threshold of 0.85 fixed in advance. The second is whether performance has levelled off by that snapshot, defined as an AUC increase of less than 0.02 to the following snapshot, which cannot be assessed at the final snapshot. Neither result is described as reliable prediction, and a plateau is claimed only if the second condition is met.

Indicator analysis uses SHAP values [20] and permutation importance on the test partition. SHAP values are compared as rankings within each model, not as magnitudes across models, because Logistic Regression and XGBoost report them in log odds units and Random Forest in probability units. Permutation importance, expressed as the drop in AUC-ROC, is comparable across models, but permute-and-predict procedures can produce misleading diagnostics when features are strongly dependent [24]. Which is the case here because the gold, experience, and CS differences derive from the same

underlying events. Reporting both metrics is therefore a deliberate cross-check. Standardised Logistic Regression coefficients and the built-in importance of the tree models were also computed and are provided with the code; the Gini importance of tree models is known to be biased toward high-cardinality continuous features [25] and not used for inference.

All experiments were implemented in Python with the scikit-learn, XGboost, shap, pandas, and numpy libraries (versions are listed in the repository). Random seed values were fixed across all stochastic components. The hyperparameter search space and the values selected by five-fold cross-validation on the training partition are given in Table III, so that the reported configurations can be reproduced.

TABLE III
HYPERPARAMETER SEARCH SPACE AND SELECTED VALUES

Model	Hyperparameter	Search Space	Min 10	Min 15	Min 20	Min 25
Logistic Regression	C	0.01, 0.1, 1.0, 10.0	0.1	0.01	0.1	0.01
Random Forest	n_estimators	300, 600	600	600	600	300
Random Forest	max_depth	4, 8, none	4	4	none	4
Random Forest	min_samples_leaf	1, 5	5	5	5	5
XGBoost	n_estimators	200, 400	200	200	200	200
XGBoost	max_depth	2, 3, 5	2	5	2	2
XGBoost	learning_rate	0.03, 0.1	0.03	0.03	0.03	0.03
XGBoost	subsample	0.8, 1.0	1.0	1.0	0.8	0.8
XGBoost	reg_lambda	1.0, 5.0	5.0	5.0	5.0	5.0

The dataset is publicly available from Oracle's Elixir. The analysis code, including the snapshot construction routine, the objective screening rules summarised in Table I, and the scripts that reproduce every table and figure reported below, is available at https://github.com/FaN0016/LCK2025PredictionWinSnapshot-main.

III. RESULTS AND DISCUSSION

A. Base Rate Shift and Model Performance

One characteristic of the data shapes every interpretation that follows. Using blue-side rows, the win rate declines steadily over the season, standing at 61.5 percent in the LCK Cup, 50.8 percent in Rounds 1 to 2, and 45.6 percent in Rounds 3 to 5 (Figure 6). The training partition therefore carries a blue-side win rate of 54.2 percent against 45.6 percent in the test partition, a base rate shift of 8.5 percentage points. This is a concrete instance of dataset shift [21] that the chronological validation scheme cannot remove. Recent work on estimating model performance under covariate shift treats such shifts as a routine deployment condition and stresses that performance measured before a shift cannot be assumed to carry over [26]. It also means that the majority-class baseline on the test partition is 54.4 percent rather than the 51.0 percent computed over the full sample.

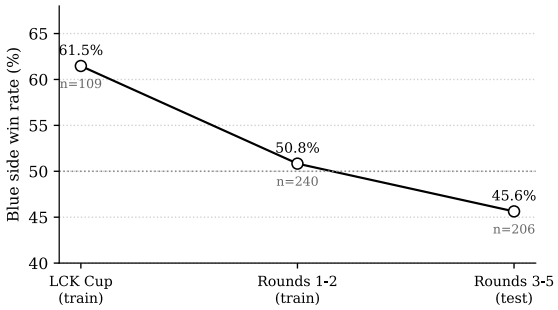

Figure 6. Blue side win rate across the three competitive phases of the 2025 season.

Table IV summarises the test performance of the three models alongside two single-indicator references: the gold difference sign rule, and a Logistic Regression fitted on gold difference alone. Figures 7 to 9 plot accuracy, AUC-ROC, and the Brier score respectively. All three metrics improve as the match progresses. Accuracy rises from 63.6-68.2 percent at minute 10 to 81.6-82.2 percent at minute 25, AUC-ROC from 0.683-0.714 to 0.905-0.906, and the Brier score falls from 0.220-0.229 to 0.128-0.134. The improvement is not strictly monotone for every model, since Random Forest accuracy falls from 68.2 to 66.5 percent and the XGBoost Brier score rises from 0.229 to 0.231 between minutes 10 and 15. The overall pattern agrees with findings in other esports that predictive accuracy increases as in-game information accumulates [3]. The test partition contains 206 matches at minutes 10 through 20 and 201 at minute 25, so each 95 percent Wilson confidence interval for accuracy spans 11 to 13 percentage points.

TABLE IV
TEST PERFORMANCE OF THE MODELS AND THE SINGLE-INDICATOR REFERENCES

Metric and Method	Min 10	Min 15	Min 20	Min 25
Accuracy (%)				
Logistic Regression	63.6	67.5	73.8	81.6
Random Forest	68.2	66.5	76.0	81.6
XGBoost	66.5	66.5	74.6	82.2
Gold Difference sign rule	68.4	69.9	74.3	82.1
Gold-only Logistic Regression	65.5	72.3	74.3	82.1
AUC-ROC				
Logistic Regression	0.714	0.764	0.842	0.906
Random Forest	0.702	0.753	0.848	0.905
XGBoost	0.683	0.725	0.844	0.905
Gold-only Logistic Regression	0.711	0.779	0.835	0.897
Brier Score				
Logistic Regression	0.220	0.202	0.165	0.134
Random Forest	0.228	0.213	0.167	0.128
XGBoost	0.229	0.231	0.171	0.131
Gold-only Logistic Regression	0.221	0.194	0.170	0.132

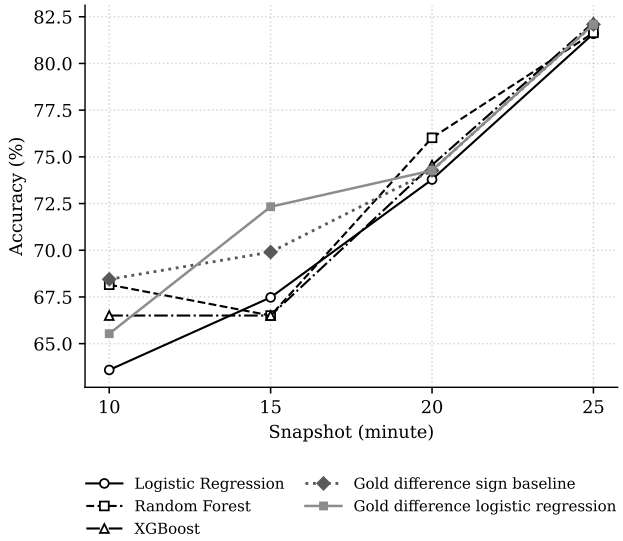

Figure 7. Test accuracy of the three models against the gold difference sign baseline and the gold-only Logistic Regression at each snapshot.

None of the twelve model and time combinations exceeds the gold difference sign rule beyond its confidence interval (Figure 7). At minute 10, the point estimates trail the rule by 0.3 to 4.9 percentage points, and at minute 15 by 2.4 to 3.4. At minutes 20 and 25, they lie between 0.5 below and 1.7 above it, the largest margin being Random Forest at minute 20, and the lower bound of every confidence interval remains below the accuracy of the rule.

Against the probabilistic reference, the picture is the same. AUC-ROC is higher than that of the gold-only Logistic Regression by 0.007 to 0.013 at minutes 20 and 25, but lower by 0.015 to 0.054 at minute 15. The Brier score is lower in only five of the twelve combinations, by at most 0.005, and higher in the other seven. In the DeLong test against the gold-only regression, the smallest p value among the six comparisons at minutes 20 and 25 is 0.18. The only significant difference in the whole set is XGBoost at minute 15 ($p = 0.012$), where the model is worse than the single-feature reference, and it does not survive the Bonferroni correction. The models therefore add no measurable value over either reference, in classification accuracy or in probability quality, within the resolution that a test partition of this size allows.

Comparison between models using the DeLong test shows that, of the twelve pairs tested, two are significant at the 5 percent level, namely Logistic Regression and Random Forest against XGBoost at minute 15, with p values of 0.045 and 0.038. That gap is the one visible separation in Figure 8, where the XGBoost curve sits below the other two at minutes 10 and 15 before the three converge. Neither difference survives the Bonferroni correction. At minutes 20 and 25, the AUC differences between models are at most 0.006. No statistically significant difference between the models is therefore detected, although, with 206 test matches, the test has limited power against small differences. Logistic regression, the simplest of the three, has the highest AUC at three of the four time points.

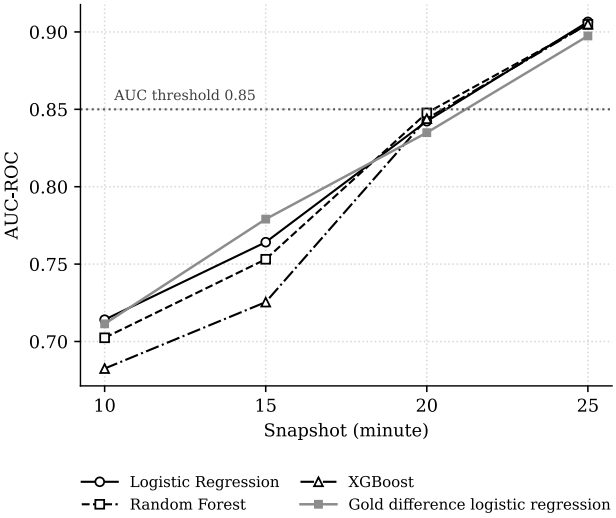

Figure 8. Test AUC-ROC of the three models and the gold-only Logistic Regression at each snapshot. The dotted line marks the 0.85 threshold fixed in advance (Section II-F).

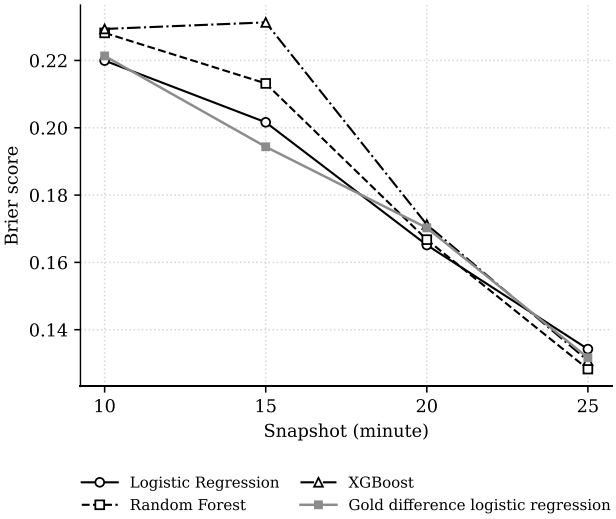

Figure 9. Test Brier score of the three models and the gold-only Logistic Regression at each snapshot; lower values indicate better probabilities.

B. AUC Threshold and Performance Plateau

The AUC-ROC of Logistic Regression is 0.714, 0.764, 0.842, and 0.906 at the four snapshots, and a similar pattern holds for the other two models. The earliest snapshot at which the pre-specified threshold of 0.85 is reached is minute 25 for all three models. Two caveats apply. First, the answer is sensitive to the threshold, since the AUC at minute 20 ranges from 0.842 to 0.848 and misses 0.85 by only 0.002 to 0.008. Had the threshold been 0.84, the answer would have been minute 20. Because the threshold was fixed before the experiments were run, its use remains legitimate, but its fragility should be visible to the reader. Second, the plateau condition cannot be met within the observation window. AUC still rises by 0.057 to 0.064 from minute 20 to minute 25, far above the 0.02 limit, and minute

25 is the final snapshot, so no later increase can be examined. This study therefore reports that the threshold is first reached at minute 25 and does not conclude that predictions are reliable or stable at that point. Locating the plateau requires snapshots at minute 30 or beyond.

C. Feature Importance Results

Indicator importance is assessed with mean absolute SHAP values and permutation importance, compared as rankings within each model. Figure 10 presents permutation importance at the minute-20 snapshot of the sensitivity variant, averaged across the three models, which is the variant in which numeric and binary indicators can be compared directly. In the main models, gold difference is the strongest indicator at minutes 10 and 20, and experience difference at minute 25, by both metrics and in all three models. At minute 15, the two metrics disagree. Permutation importance ranks gold difference first in every model, whereas SHAP ranks experience difference first for Random Forest and XGBoost and places it level with gold difference for Logistic Regression (0.194 against 0.199). For XGBoost, the permutation importance of experience difference at minute 15 in the main model is negative (-0.013), which suggests that the model relies on a feature that does not generalise to the test partition, and this is the model and snapshot at which XGBoost has significantly lower AUC than the other two models (uncorrected). No conclusion about a dominant indicator at minute 15 is therefore drawn.

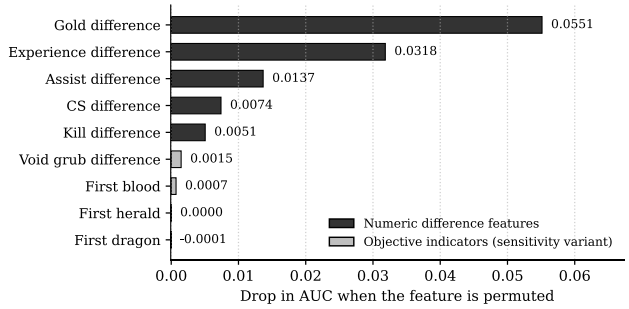

Figure 10. Permutation importance at the minute-20 snapshot of the sensitivity variant, averaged across the three models on the test partition. Numeric difference features and objective indicators are shown separately.

Gold difference reaches 0.0551 at minute 20 in Figure 10, followed by experience difference at 0.0318, assist difference at 0.0137, CS difference at 0.0074, and kill difference at 0.0051. The contribution of the objective indicators is close to zero at this minute: the void grub difference, first blood, first herald, and first dragon return 0.0015, 0.0007, -0.00002, and -0.00006, less than a third of the weakest numeric feature. At minute 25, they remain at or below 0.0005, with first herald at 0.0005, first dragon at 0.0002, the void grub difference at -0.00002, and first blood at -0.0002, so that permuting the negative ones slightly raises AUC, which is consistent with the documented behaviour of permute-and-predict measures under feature dependence [24]. The exception is minute 10. There, the void grub difference ranks third by permutation importance (0.0218), above first dragon (0.0077) and the experience (0.0055), kill (0.0044), and assist (0.0021) differences, and third in the XGBoost SHAP ranking, while first blood returns -0.0011.

The sensitivity analysis adds the objective indicators at every snapshot after the objective has spawned (Table I) and compares the result with the main models in Table V. The change in accuracy ranges from -3.2 to +1.0 percentage points and the change in AUC from -0.004 to +0.010, and the direction is not consistent. At minutes 20 and 25, the AUC changes lie between -0.004 and +0.001 and the accuracy changes between -2.0 and +1.0 points. At minutes 10 and 15, AUC rises in all three models by 0.004 to 0.010, while accuracy falls in all three models at minute 10, by 1.0 to 3.2 points. The AUC gain at the earlier minutes coincides with the void grub difference, which carries real weight only at minute 10. Because the dataset records only the final grub count and the closing time of the encounter is not given in the patch notes (a community wiki lists a despawn at 14:45 [14], which would make the count final from minute 15 onward), the minute-10 count may include grubs taken after the snapshot, so the gain is not read as evidence that the indicator is informative at prediction time. The conclusions about the main models do not depend on these indicators, which remain excluded on temporal grounds.

TABLE V
ACCURACY (%) OF THE MAIN MODELS AND THE SENSITIVITY VARIANT WITH OBJECTIVE INDICATORS

Model and feature variant	**Min 10**	**Min 15**	**Min 20**	**Min 25**
LR, main	63.6	67.5	73.8	81.6
LR, with objective indicators	62.6	66.5	74.3	82.6
RF, main	68.2	66.5	76.0	81.6
RF, with objective indicators	65.5	66.8	74.0	81.7
XGB, main	66.5	66.5	74.6	82.2
XGB, with objective indicators	63.3	67.1	74.5	82.1

D. Why Complex Models Do Not Exceed the Gold Baseline

Three observations bear on why the models fail to improve on a single resource difference. First, gold, experience, and CS differences all derive from the same underlying events (Section II-C), so that once gold difference is known, the remaining features carry little independent signal. At minutes 20 and 25, the gold-only Logistic Regression reproduces the accuracy of the sign rule exactly (74.3 and 82.1 percent), and the three models exceed its AUC by at most 0.013.

Second, with 349 training matches, added flexibility is more likely to fit noise than structure. At minute 15, the gold-only regression (72.3 percent, AUC 0.779) outperforms all three multi-feature models (66.5–67.5 percent, AUC 0.725–0.764). The base rate shift cannot

explain this gap, since the single-feature regression is trained on the same partition and carries the same intercept shift. The pattern is consistent with variance introduced by additional features rather than with a missing signal, although these data cannot separate the two explanations.

Third, the base rate shift may account for part of the shortfall at minute 10, where the gold-only regression (65.5 percent) also trails the sign rule (68.4 percent). Models trained when the blue side won 54.2 percent of matches tend to over-predict blue-side victories at a threshold of 0.5, whereas the sign rule carries no such prior and is immune to the shift. These findings do not show that the models are equivalent to the references, since with 206 test matches, gains of a few points in accuracy or of 0.01 to 0.02 in AUC cannot be excluded. They show that no gain was detected, and that any gain would be small.

E. Temporal Validity and Data Leakage

The third research question is answered through two tests. The first compares the minute-25 snapshot model against a model built on aggregate end-of-match features, as shown in Table VI. The end-of-match model reaches an accuracy of 97.1 to 98.5 percent and an AUC of 0.996 to 0.999, far beyond any snapshot model. Both leakage conditions are satisfied, in that the values of these features are known only after the match has ended and are therefore unavailable at prediction time, and their performance is unattainable by any snapshot model. This confirms the methodological concern raised in earlier literature [7] and underlines that near-perfect accuracy in comparable studies needs to be read critically.

TABLE VI
COMPARISON OF THE MINUTE-25 SNAPSHOT FEATURE SET AND END-OF-MATCH FEATURES

Feature set	LR	RF	XGB
Accuracy (%)			
Minute-25 snapshot	81.6	81.6	82.2
End-of-match features	98.5	97.6	97.1
AUC-ROC			
Minute-25 snapshot	0.906	0.905	0.905
End-of-match features	0.999	0.998	0.996

The second test emerged from the research process itself and yields a subtler finding. In an earlier configuration, the minute-25 snapshot included first Baron and the Atakhan difference, on the assumption that both objectives were available from minute 20:00 and could therefore be secured before minute 25. With both features added, permutation importance ranked first Baron first for Logistic Regression and Random Forest and second only to gold difference for XGBoost, ahead of the experience difference in all three models. SHAP gave the same picture, ranking first Baron first for Logistic Regression and XGBoost and second only to experience difference for Random Forest. A binary flag recording who secured a single objective should not outrank the accumulated resource position of an entire team, and this anomaly prompted a re-examination of the game patch notes.

That examination showed the initial assumption to be mistaken. Throughout the 2025 season, Baron no longer spawns at minute 20:00 but at minute 25:00, because the 20:00 slot is occupied by Atakhan, an objective introduced in that season. First Baron is therefore not merely difficult to verify but cannot hold a true value at the minute-25 snapshot, since the objective has not yet spawned at that point. The data confirm this: of the 555 matches none records a Baron at a duration below 26.28 minutes, and only 12 of the 29 matches (41.4 percent) lasting between 26 and 27 minutes contain a Baron. The value of the feature therefore derives entirely from events occurring after the snapshot was taken.

To isolate the effect, each feature was added separately to the main minute-25 snapshot (Table VII). Adding first Baron raises AUC-ROC from 0.905-0.906 to 0.927-0.934 in all three models, lowers the Brier score by 0.011 to 0.018, and raises accuracy by 0.7 to 3.4 percentage points, taking the models from the level of the sign rule to 0.8 to 3.0 points above it. The Atakhan difference alone changes AUC by at most 0.002 and lowers accuracy slightly in all three models, so the entire inflation is attributable to first Baron. The accuracy gain is small and inconsistent across models (+0.7 points for XGBoost) and remains within the confidence interval, so AUC is the more consistent signal. Adding both features gives accuracies of 83.9 to 85.6 percent, but differences between the one-feature and two-feature variants (up to 2.7 points for XGBoost, about five matches) are within noise at n = 201. The presence of the leaking feature also distorts the apparent importance of legitimate ones: the permutation importance of experience difference is 60 to 67 percent lower in all three models when first Baron is present. The apparent advantage over the baseline at minute 25 therefore rested on a single feature whose temporal validity could not be defended.

TABLE VII
AUDIT OF FIRST BARON AND ATAKHAN AT THE MINUTE-25 SNAPSHOT

Feature set	Accuracy (%) LR / RF / XGB	AUC-ROC LR / RF / XGB
Main (numeric feature)	81.6 / 81.6 / 82.2	0.906 / 0.905 / 0.905
First Baron	83.6 / 85.1 / 82.9	0.934 / 0.930 / 0.927
Atakhan difference	80.6 / 80.5 / 82.1	0.904 / 0.906 / 0.906
both	84.6 / 83.9 / 85.6	0.933 / 0.926 / 0.928

This finding broadens the understanding of data leakage in esports analytics. Leakage takes the form not only of aggregate end-of-match features that are plainly post hoc, but can also enter through binary objective indicators that appear reasonable yet carry no time marker. Research using public datasets without timestamps needs to verify that each indicator could genuinely have been known at the moment of prediction, rather than merely that it might have

occurred earlier. Verification of this kind should become a standard part of reporting rather than an optional step, in line with proposals to make leakage checks an explicit and documented component of reporting practice [16].

Two broader rules follow. First, a feature whose acquisition window has no closing boundary cannot be used at any snapshot as long as the dataset stores no timestamp, because moving the snapshot later does not prevent the objective from being secured afterwards. This disqualifies first tower and Atakhan as well as first Baron, and it is the reason first blood and first dragon are excluded from the main models. The void grub difference and first herald are excluded for a related reason, since their spawn times changed within the season and their closing times are not given in the patch notes. Second, feature importance can serve as a leakage detection tool. When an indicator that ought on substantive grounds to be marginal instead occupies the top rank, the anomaly should be treated as a signal to re-examine assumptions about its temporal availability rather than as a finding to be reported directly.

F. Exploratory Playoff Subgroup

Beyond the three research questions, a subgroup analysis of the test partition produced a result that was not anticipated and is therefore reported as exploratory. As shown in Table VIII, at the minute-25 snapshot all three models predict playoff matches far more accurately than regular-season matches, at 91.7 to 93.3 percent against 76.6 to 78.0 percent. A test of the difference in proportions gives p values between 0.005 and 0.016, and Fisher's exact test reaches the same conclusion. None of these values survives a Bonferroni correction across the twelve model and snapshot comparisons (threshold 0.0042), and because the three models are evaluated on the same matches, the three tests are not independent. At minute 20, the difference is not significant (p between 0.21 and 0.62) and cannot be claimed.

TABLE VIII
PERFORMANCE ON TEST PARTITION SUBGROUPS AT THE MINUTE-25 SNAPSHOT (ACCURACY, %; COMPUTED FROM PROBABILITIES AVERAGED OVER SEEDS)

Model	**Regular season (n=141)**	**Playoff (n=60)**	**Diff.**	**p (proportion test)**
LR	77.3	91.7	+14.4	0.016
RF	76.6	93.3	+16.7	0.005
XGB	78.0	93.3	+15.3	0.009

This result runs counter to the initial expectation that elimination pressure would make playoff matches harder to predict. Two explanations are equally plausible and cannot be separated by these data. The first is behavioural, in that teams leading at minute 25 in the playoffs waste that lead less often, tightening the relationship between match state and final outcome. Work on comeback victories in MOBA titles supports the plausibility of this reading, since it treats the squandering of a lead as a distinct phenomenon driven by identifiable in-game conditions rather than as random variation [27]. The second is a matter of selection, since only eight of the ten teams qualify for the playoffs, so that team composition and playing style differ from the regular season. Given a subgroup of only 60 matches, this finding requires verification in another season or league before it can serve as the basis for any conclusion.

G. Comparison with Previous Studies

Read by accuracy alone, this study ranks below the closest comparable work, since the minute-25 snapshot reaches 81.6 to 82.2 percent against 88 percent for the Random Forest of [11], and its AUC of 0.906 falls below the 0.94 reported there. The figures are not measured on comparable material, however. Yang et al. [3] report the highest value of the four studies reviewed in the introduction, 93.7 percent, but on Dota 2 and from real-time features accumulated to minute 40, far beyond the window examined here. Do et al. [5] predict before the match begins, from player-champion experience alone, and reach about 75 percent. Chowdhury et al. [7] combine pre-match and in-game features on League of Legends at an aggregate rather than a timed resolution, reporting 76.8 percent. Firmansyah et al. [11] contrast the draft phase with a single in-game snapshot on Mobile Legends. The present study samples resource differences at four fixed points within a match, so that the accuracy it reports is tied to a stated prediction time rather than to the match as a whole.

Two reporting practices separate this study from all four. None of them accompanies its accuracy with a single-indicator baseline, so it is not known how much of each reported figure could have been obtained from a resource difference alone. The present study reports that baseline, and a one-feature Logistic Regression, and finds that neither is exceeded by its own models, a result that would have remained invisible under the reporting conventions of the comparison studies. None of them audits the temporal validity of its features either, and only [7] raises the issue at all, in discussion rather than as a test. The two omissions compound each other, since a study that screens for neither can report a figure that no heuristic comparison and no timing check has been allowed to reduce.

The position of this study is therefore not one of higher accuracy but of accuracy that can be accounted for. It supplies the missing reference point, showing that twelve model and time combinations fail to exceed a rule based on the sign of the gold difference. It traces the evolution of predictive performance across four points within a single match rather than reporting one figure for one phase. And it audits the temporal validity of every feature, a step that removed up to 3.4 percentage points of apparent accuracy and 0.02 to 0.03 of AUC from its own results. The last point bears directly on the comparison studies, since in-game objective features of the kind used in [11] belong to the category this audit found problematic when acquisition times are not recorded.

H. Implications

For analysts, coaches, and casters, the results suggest that a live win estimate at minutes 20 and 25 gains little from a complex model. The sign of the gold difference is within two points of every model tested, and a one-feature Logistic Regression gives a Brier score of 0.170 and 0.132 at those minutes, comparable to the models. This is consistent with the observation that broadcast narratives often treat first blood or first dragon as markers of the direction of a match: the sensitivity analysis shows that these indicators add no consistent information once gold and experience differences are known, most plausibly because their value is converted directly into resources. A real-time system would then need to monitor resource differences without separately tracking objective ownership.

The ranking at minute 25 also shows that experience difference, not gold difference, carries the largest weight there, so a one-feature rule is adequate for prediction but not for explanation. These recommendations apply to the league, season, and snapshot window studied.

Theoretically, the study supports three reporting practices for win prediction research: a single-indicator baseline reported beside every accuracy figure, an explicit audit of the temporal availability of every feature when the dataset has no timestamps, and the treatment of an implausible importance ranking as a diagnostic for leakage [16]. A further implication is methodological. A study that reports only the accuracy of its models, without the references used here, would have concluded that gradient boosting adds value over simpler alternatives at every snapshot.

I. Robustness Check and Limitations

As a robustness check, all cross-snapshot comparisons were repeated on the subset of 540 matches available at all four time points, which gives 201 test matches at every snapshot. The pattern is the same with a small drop in level: for example, Random Forest accuracy at minute 20 falls from 76.0 to 73.7 percent and at minute 10 from 68.2 to 67.1 percent, and AUC falls by at most 0.014. No conclusion reverses direction, so that excluding short matches at the minute-25 snapshot does not alter the findings of this study.

Several limitations should be noted. This study covers one league in one season, so generalisation to other leagues cannot yet be assured, particularly because the game meta changes both across regions and within a single season. Within-season change is clearly visible in the data, since the mean number of void grubs per team falls from 3.00 to 1.50 at patch 15.09, so that the weight of a given objective cannot be treated as constant throughout the season. A test partition of 206 matches yields accuracy confidence intervals spanning 11 to 13 percentage points, so that only large differences are detectable. Binary objective indicators are excluded from the main models because their timing cannot be verified, and their inclusion in the sensitivity analysis rests on the assumption that the events occurred before the snapshot. Finally, the base rate shift between the training and test partitions is a genuine property of the competition observed, but it simultaneously limits the accuracy attainable by any model that uses a fixed threshold.

IV. CONCLUSION

This study evaluated team win prediction in the 2025 LCK season from snapshots at four time points, using three models of differing complexity and two single-indicator references. In response to the first research question, machine learning models added no measurable value over either reference. None exceeded the gold difference sign rule in accuracy beyond its confidence interval, and none differed significantly in AUC-ROC from a Logistic Regression on gold difference alone after correction for multiple comparisons. In the main models, gold difference was the most informative indicator at minutes 10 and 20 and experience difference at minute 25, while binary objective indicators contributed essentially nothing at minutes 20 and 25.

In response to the second question, predictions first reached the pre-specified AUC-ROC threshold of 0.85 at minute 25. Because AUC was still rising at that point and minute 25 is the last snapshot observed, the study does not conclude that performance has stabilised or that predictions are reliable at any particular minute.

In response to the third question, features whose acquisition time cannot be verified inserted information that was not available at prediction time. End-of-match features produced near-perfect scores, and first Baron, which cannot have occurred before the snapshot, raised AUC-ROC in all three models and accounted for the entire apparent advantage over the baseline. Leakage can therefore enter through binary objective indicators without timestamps, and not only through aggregate end-of-match features.

The findings are limited by the use of a single league and season, a test partition of 206 matches that detects only large differences, a base rate shift between training and test partitions, and the exclusion of objective indicators whose timing cannot be verified. The playoff result is exploratory. Future work should extend the snapshots beyond minute 25 to locate a performance plateau, replicate the analysis across additional leagues and seasons, and use data with event timestamps so that objective indicators can be evaluated under verified timing.

REFERENCES

[1] M. Mora-Cantallops and M.-Á. Sicilia, “MOBA games: A literature review,” *Entertainment Computing*, vol. 26, pp. 128–138, May 2018, doi: 10.1016/j.entcom.2018.02.005.

[2] V. J. Hodge, S. Devlin, N. Sephton, F. Block, P. I. Cowling, and A. Drachen, “Win prediction in multiplayer esports: Live professional match prediction,” *IEEE Transactions on Games*, vol. 13, no. 4, pp. 368–379, Dec. 2021, doi: 10.1109/TG.2019.2948469.

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

[4] Ahmad Alif Kamal, M. A. Mansor, L. Truna, N. Mohd. Shaipullah, and N. H. Habib Sultan, “Machine learning applications in multiplayer online battle arena esports—a systematic review,”

Pertanika Journal of Science and Technology, vol. 33, no. 2, Mar. 2025, doi: 10.47836/pjst.33.2.11.

[5] 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 *The 16th International Conference on the Foundations of Digital Games (FDG) 2021*, New York, NY, USA: ACM, Aug. 2021, pp. 1–5. doi: 10.1145/3472538.3472579.

[6] C. J. A. Cordova, C. V. A. Villaceran, and C. F. Peña, "Predicting league of legends match outcomes through machine learning models using past match player performance," in *2024 IEEE International Conference on Computing (ICOCO)*, IEEE, Dec. 2024, pp. 522–527. doi: 10.1109/ICOCO62848.2024.10928259.

[7] S. Chowdhury, M. Ahsan, and P. Barraclough, "Applications of linear and ensemble-based machine learning for predicting winning teams in league of legends," *Applied Sciences*, vol. 15, no. 10, p. 5241, May 2025, doi: 10.3390/app15105241.

[8] I. G. W. Sena and A. W. R. Emanuel, "Mobile legend game prediction using machine learning regression method," *JURTEKSI (Jurnal Teknologi dan Sistem Informasi)*, vol. 9, no. 2, pp. 221–230, Mar. 2023, doi: 10.33330/jurteksi.v9i2.1866.

[9] S. Bayulianto, I. Purnamasari, and M. Jajuli, "Prediksi tingkat kemenangan mobile legends profesional league Indonesia season 9 dengan menggunakan algoritma naïve bayes," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 8, no. 2, pp. 538–550, May 2023, doi: 10.29100/jipi.v8i2.3562.

[10] A. P. Putra and P. N. Andono, "Win probability of heroes in mobile legends MPL ID S12 competitions using naive bayes algorithm," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 8, no. 1, p. 203, Jan. 2024, doi: 10.30865/mib.v8i1.7185.

[11] D. F. Firmansyah, A. P. Kuncoro, and R. Riyanto, "Match outcome prediction in draft pick and in-game phases of MSC 2025 mobile legends using random forest and xgboost," *Journal of Applied Informatics and Computing*, vol. 9, no. 6, pp. 3892–3903, Dec. 2025, doi: 10.30871/jaic.v9i6.11658.

[12] Y. A. Hambali, R. Megasari, and R. R. Santoso, "Implementasi metode machine learning menggunakan algoritma evolving artificial neural network pada kasus prediksi diagnosis diabetes," *Jurnal Aplikasi dan Teori Ilmu Komputer*, vol. 6, no. 1, pp. 9–20, May 2023, doi: 10.17509/jatikom.v6i1.56536.

[13] Riot Games, "Patch 25.09 notes." [Online]. Available: https://www.leagueoflegends.com/en-us/news/game-updates/patch-25-09-notes/

[14] League of Legends Wiki, "Voidgrub camp." [Online]. Available: https://wiki.leagueoflegends.com/en-us/Voidgrub_camp

[15] S. Kaufman, S. Rosset, C. Perlich, and O. Stitelman, "Leakage in data mining: Formulation, detection, and avoidance," *ACM Transactions on Knowledge Discovery from Data*, vol. 6, no. 4, pp. 1–21, Dec. 2012, doi: 10.1145/2382577.2382579.

[16] S. Kapoor and A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," *Patterns*, vol. 4, no. 9, p. 100804, Sep. 2023, doi: 10.1016/j.patter.2023.100804.

[17] Y. Mansour and M. Schain, "Learning with maximum-entropy distributions," *Machine Learning*, vol. 45, no. 2, pp. 123–145, Nov. 2001, doi: 10.1023/A:1010950718922.

[18] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001, doi: 10.1214/aos/1013203451.

[19] 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*, New York, NY, USA: ACM, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[20] S. Lundberg, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[21] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognition*, vol. 45, no. 1, pp. 521–530, Jan. 2012, doi: 10.1016/j.patcog.2011.06.019.

[22] B. Van Calster, D. J. McLernon, M. van Smeden, L. Wynants, and E. W. Steyerberg, "Calibration: The Achilles heel of predictive analytics," *BMC Medicine*, vol. 17, no. 1, p. 230, Dec. 2019, doi: 10.1186/s12916-019-1466-7.

[23] X. Sun and W. Xu, "Fast implementation of delong's algorithm for comparing the areas under correlated receiver operating characteristic curves," *IEEE Signal Processing Letters*, vol. 21, no. 11, pp. 1389–1393, Nov. 2014, doi: 10.1109/LSP.2014.2337313.

[24] G. Hooker, L. Mentch, and S. Zhou, "Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance," *Statistics and Computing*, vol. 31, no. 6, p. 82, Nov. 2021, doi: 10.1007/s11222-021-10057-z.

[25] S. Nembrini, I. R. König, and M. N. Wright, "The revival of the gini importance?," *Bioinformatics*, vol. 34, no. 21, pp. 3711–3718, Nov. 2018, doi: 10.1093/bioinformatics/bty373.

[26] J. Białek, J. Kivimäki, W. Kuberski, and N. Perrakis, "Estimating model performance under covariate shift without labels," in *Advances in Neural Information Processing Systems 38*, San Diego, California, USA: Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025, pp. 178849–178880. doi: 10.52202/085713-5379.

[27] J. Lee and N. Kim, "Development of machine learning-based indicators for predicting comeback victories using the bounty mechanism in MOBA games," *Electronics*, vol. 14, no. 7, p. 1445, Apr. 2025, doi: 10.3390/electronics14071445.