Hero featured learning algorithm for winning rate prediction of Honor of Kings

1st Peng Tian
College of Computer Science
South-Central Minzu University
Wuhan, China
tianpeng_znmd@163.com

2nd Wenfei Lan
College of Computer Science
South-Central Minzu University
Wuhan, China
446487309@qq.com

3rd Xiao Zhang
College of Computer Science
South-Central Minzu University
Wuhan, China
xiao.zhang@my.cityu.edu.hk

***Abstract*—Recent years have witnessed much research effort on automatically predicting game results (win predictions), which has great potential in esports live streaming and game commentator AI systems. The exsiting works on win prediction largely overlook the interpretability and the predication accuracies are hardly satisfactory. To address this issue, we collected a large-scale dataset that contains real-time game records with rich input features of the popular game *Honor of Kings*. For interpretable and more accurate predictions, we proposed a Hero Featured Network (HFN) by learning from real *Honor of Kings* combat data and heros' mutual attributes and interactions. For each time stage of the game, the prediction accuracy can archive 84.7%. The results show that HFN model can not only provide accurate real-time win predictions but also attribute the ultimate prediction results to the contributions of heros' mutual attributes and interactions for interpretability.**

***Index Terms*—MOBA esports; Hero interactions; Interpretability; Winning rate prediction;**

I. Introduction

With the rapid development of the gaming industry, esports live streaming and game commentator AI systems are becoming increasingly popular. Win prediction, a core part of the game commentator AI systems, has become one of the most-watched parts of esports live streaming. Along with MOBA games flourishing, one of the world's most popular MOBA games, *Honor of Kings (HoK)*, has an enormous potential audience among the streaming media community.

Nowadays, much work has been done on automatic MOBA esports win predictions. These works can be divided into two categories: pre-game predictions that use pre-game features such as heroes to predict the results before the games begin, and in-game predictions that use in-game features such as gold to indicate the results during games. In a sense, in-game prediction is more critical than pre-game prediction. Although some works have been done to perform in-game prediction, existing works on win prediction largely overlook the interpretability, and the prediction accuracies are hardly satisfactory. For example, Yang et al. [1] predict four types of events, including win, Kill, be-kill, and Tyrant. Unfortunately, the winning task could only predict the winning side. Yang et al. [2] propose a Two-Stage Spatial-Temporal Network (TSSTN) that can not only provide win predictions but also attribute final predictions to different features; however, the prediction is inaccurate, and the interpretability is not strong enough.

We incorporate lineup relationships and heroes' interactions into the win prediction process to address the above issues. As the initiator of events in the game, the hero plays an essential role in the game's direction. It is vital to understand synergy and restraint relationships between heroes in different game stages. On the one hand, it improves the accuracy of in-game predictions. On the other hand, it can improve the interpretability of forecasts. As one can see in Fig. 1, the HFN with hero features can provide more interpretable predictions, including the contribution of in-game features and each hero.

We train a sequence modeling network, the Long Short-Term Memory network (LSTM), to get accurate win predictions. However, LSTM is known to be a black-box model, so the reasons for the result are unknown. To solve this challenge, we separated the effects of feature value and importance to decouple their contributions. We attribute win predictions to the two most essential parts of the game: combat and hero. This paper proposes a Hero Featured Network (HFN), which gives interpretable and more accurate predictions. HFN is attributed to the two parts: combat and hero, and it is the first time to add the contribution of heroes' interactions into the in-game predictions. An important conclusion from our research on *HoK* and other MOBA games is that the hero feature is not always crucial. The hero, for example, is the essential feature in the early stage of the game. It is the basis for attributing the in-game prediction to the two major parts.

To get more interpretable predictions, in the combat space of the *Spatial-stage*, we have chosen just three of the most critical combat features ($Gold$, $Kill$, and $Tower$) and projected onto three independent representation spaces. In hero space, we tossed ten heroes onto ten different representation spaces. We

Corresponding author: Wenfei Lan. This work was supported in part by the Fundamental Research Funds for the Central Universities, South-Central Minzu for Nationalities(Grant Number CPT22017 and CSQ21018).

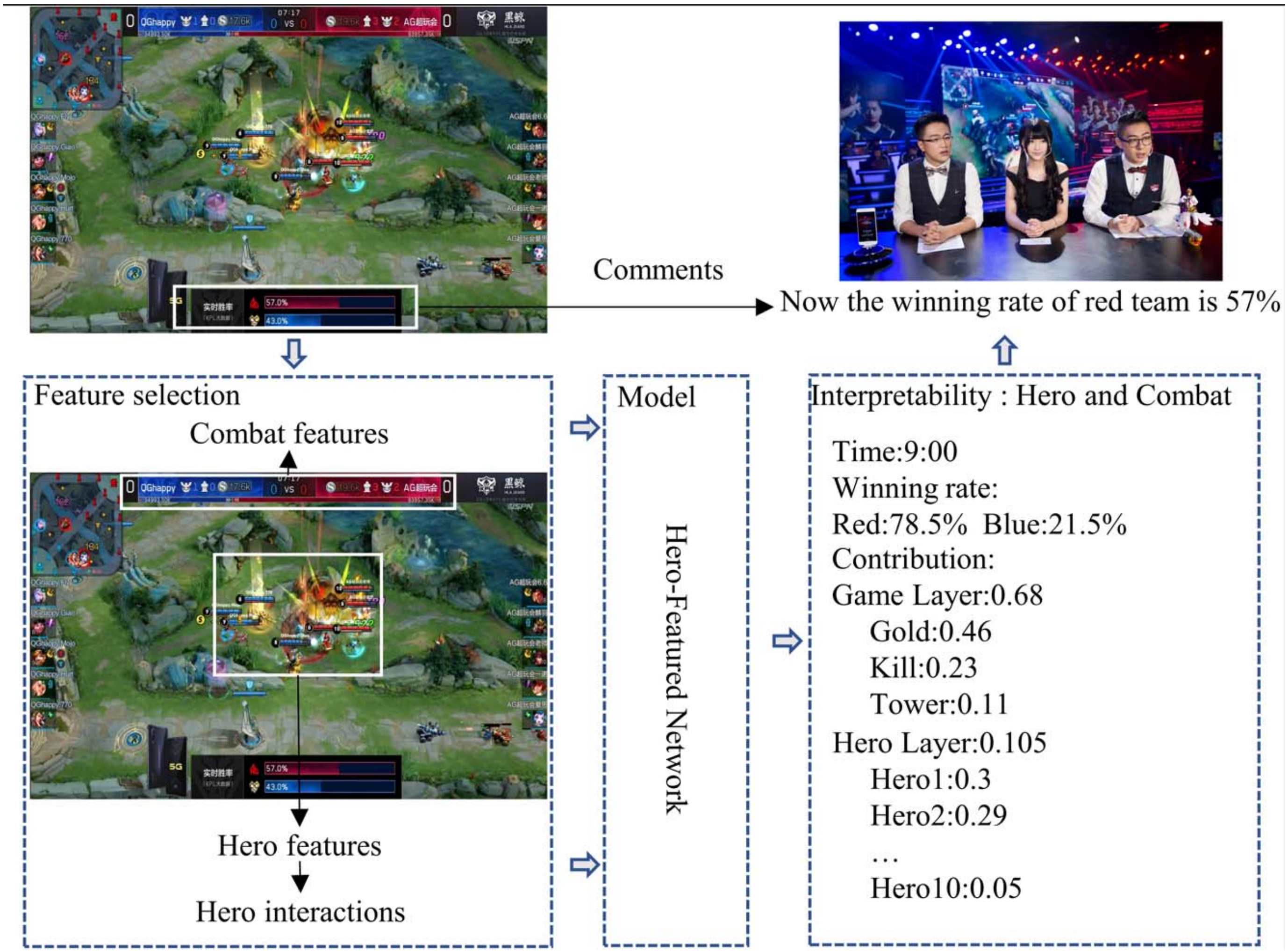

Fig. 1. The winning rate prediction process of Honor of Kings.

build thirteen *Spatial-models* that use only a single feature group or the relationship of a single hero as input to make win predictions. We discovered that specific characteristics are not consistently important throughout the games. To simulate this time characteristic more accurately, we apply three time-variant weight vectors to the *Spatial-stage* in the *Temporal-stage*. The outcome is generated by weighting the outputs of combat and hero spaces.

Furthermore, as explained in a subsequent section, it is worth noting that crucial features do not always imply a significant contribution. Experiments show that the Hero-Featured Network model provides the more accurate and interpretable in-game prediction for *HoK*. Our contribution is summarized as follows:

• We collected a large dataset of real-time game records from the popular game *HoK* and used this data to train reliable prediction models.

• We proposed a *Hero Featured Network (HFN)* by learning from real *HoK* combat data and heros' mutual attributes and interactions to make more reliable and interpretable results by attributing the forecasts to distinct features' contributions.

II. RELATED WORKS

With the increasing popularity of MOBA esports, much works have been done on MOBA games. This section introduces related work on MOBA games, including Game AI and MOBA esports win predictions. To better meet the needs of game-related industries, much research has been done on Game AI. For instance, Wu [3] proposed a macro strategy model to guide the destinations in the game. Ye et al. [4] implemented an RL system for 1V1 mode to control agents. Then, Ye et al. [5] proposed a MOBA AI learning paradigm for scalability. In terms of methodology, deep reinforcement learning can be used to play complete MOBA games. The large-scale performance test of MOBA AI was carried out for the first time, proving the superiority of AI.

MOBA esports win predictions as another central research direction has also made many achievements. Pre-game prediction and in-game prediction are the two most essential parts of win prediction research. Pre-game predictions mainly use pre-game features to make predictions. For example, Wang et al. [6] used heroes' historical records to train several machine learning algorithms and predict game outcomes. Chen et al. [7] studies the synergy and constraint relationship between heroes. Wang et al. [8] proposed an improved Naive Bayes classifier to predict the accuracy by analyzing the lineup, with an overall accuracy of 68%. It is inaccurate to predict the results through lineup analysis. These works lack precision and information, which limits their application. In-game predictions, with higher accuracy, use combat data to predict the results. Among these works, Yang et al. [9] began by predicting the winning rate using combat data. However, the model's accuracy is limited by low-quality dataset, and the lack of interpretability is another limitation. Hodge et al. [10] collected game data at different skill levels, which improves the quality of training data and improves the performance of the algorithm. Demediuk et al. [11] used unsupervised learning techniques to study how heroes perform in games. Andono et al. [12] recommended using data from the four AI-controlled heroes in the game to predict the winning rate, with a maximum accuracy of 80%. However, the AI can only acquire and control the data of four heroes, and the results are somewhat limited. Grutzik et al. [13] used SVM and neural networks to better predict game outcomes based on the historical performance of players and characters. Kinkade et al. [14] proposed two ways to predict the result of a match. The first method uses post-match data, while the second relies on lineups. These two methods are used to predict the result of the entire game, not the real-time winning rate. Ravari et al. [15] used the features of the lineup and the game to predict the winning rate by the logistic regression model. Its accuracy rate was 71%. Since the timeliness of the two parts is not considered, the effect is relatively general. The goal of [1] is to predict events in the game, such as "wins", and this work can give only the winning team. Yang et al. [2] trained a Two-Stage Spatial-Temporal Network (TSSTN) with high-quality training sets, and the most significant advantage of TSSTN is that it can attribute the predicted results. But this method is not very accurate and ignores the effect of lineups on win prediction.

III. DATA PREPROCESSING

A. DATASET

HoK is one of the most popular mobile games. In *HoK*, ten heroes will be divided into two teams, and their ultimate goal is to destroy the enemy's base. Working with the publisher of the *HoK* game, we collected data from the official game database for 27256 games that occurred on a random day in November 2019. Core data is the key to recreating game playback. To get a high-quality dataset, we only chose games between top human players. Considering the time of game data, we extract two data frames per minute, this meant that we would get 20 data frames in ten minutes, and we got 496342 data frames. Each data frame mainly includes combat data and hero data. A data frame is shown in Fig. 2. Apart from some helpful information such as each team's gold, kill, and tower, we also calculated valuable features such as hero interaction and composition relations. After data preprocessing, we selected three combat features ($gold$, $kill$, and $tower$) and six hero features (id, $type$, $kill$, $dead$, $gold$, and $assist$) from each data frame.

B. Synergy and restraint

To learn and acquire more affluent entity attributes from the role vector, we propose mapping game role features into a low-dimensional latent space. Given the heroes number N, denoted by $H = \{ H_1, H_1, \ldots, H_N \}$, given the heroes types M, denoted by $K = \{ K_1, K_1, \ldots, K_M \}$, we denote the sets of heroes and hero types for the red and blue teams in each game as $T_r^h = \{ H_i \}$, $T_b^h = \{ H_i \}$ and $T_r^k = \{ K_i \}$, $T_b^k = \{ K_i \}$. For hero H_i, its feature vector is denoted as $h_i \in R^V$, $H \in R^{N \cdot V}$ is the feature matrix of the hero. For hero type K_i, its feature vector is denoted as $k_i \in R^U$, $K \in R^{M \cdot U}$ is the feature matrix of the type. We choose to use a bilinear model to model synergy and restraint relations. First, we extract the relations between the two heroes. Synergy score function $S_C^H(i,j)$, calculates the synergy score of H_i and H_j, is introduced:

Time:8.0	
Win team:…	hero:
Lose team:	36:
gold:54832	type:tank
kill:41	kill:11
assist:66	dead:2
dead:31	assist:6
tower:4	gold:12212
hero:	46:…
Hero id:	27:…
36,46,27,66,31	66:…
	31:…
	…

Fig. 2. An example data frame at time 8.0 minutes .

$$S_C^H = H_i^T \cdot C^H \cdot H_j = \sum_{m=1}^{V} \sum_{n=1}^{V} H_{im} \cdot C_{mn}^H \cdot H_{jn} \quad (1)$$

$C^H \in R^{V \cdot V}$ is the synergy matrix of H_i,H_j. We also define the score function $S_R^H(i,j)$, which calculates the restraint score of H_i and H_j,

$$S_R^H = H_i^T \cdot R^H \cdot H_j = \sum_{m=1}^{V} \sum_{n=1}^{V} H_{im} \cdot R_{mn}^H \cdot H_{jn} \quad (2)$$

$R^H \in R^{V \cdot V}$ is the restraint matrix of H_i,H_j. Second, we extract the team composition relations (multi-type). Synergy score function $S_C^K(i,j)$, calculates the synergy score of K_i and K_j, is introduced:

$$S_C^K = K_i^T \cdot C^K \cdot K_j = \sum_{m=1}^{U} \sum_{n=1}^{U} K_{im} \cdot C_{mn}^K \cdot K_{jn} \quad (3)$$

$C^K \in R^{U \cdot U}$ is the synergy matrix of K_i,K_j. We extract the team composition relationship (multi-type). We also define the score function $S_K^R(i,j)$, which calculates the restraint score of K_i and K_j,

$$S_R^K = K_i^T \cdot R^K \cdot K_j = \sum_{m=1}^{U} \sum_{n=1}^{U} K_{im} \cdot R_{mn}^K \cdot K_{jn} \quad (4)$$

$R^K \in R^{U \cdot U}$ is the restraint matrix of K_i,K_j. Fig. 3 demonstrates the flowchart of the relational embedding process.

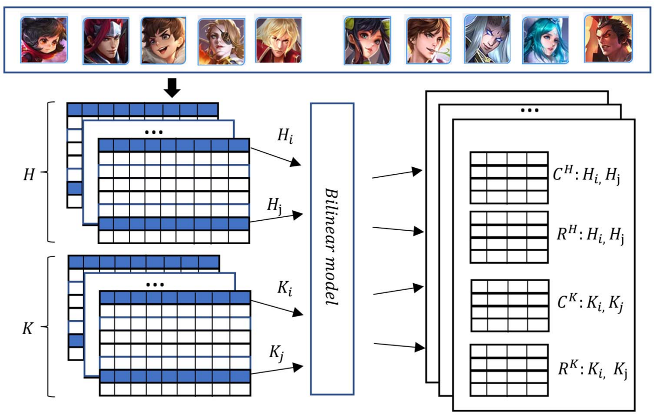

Fig. 3. Relational embedding process.

IV. HERO FEATURED NETWORK

This section introduces the Hero Featured Network (HFN) model for interpretable MOBA esports win predictions. We incorporated the effects of combat and hero relations into the HFN model. The *Spatial-stage* is divided into two spaces: combat space that takes combat features such as "gold" to make the respective win predictions, and hero space that takes relationship features of individual heroes to make the respective win predictions. In combat space, we just picked the three most essential features ($gold$, $kill$, and $tower$) and treated them separately. In hero space, ten heroes are built up relational feature groups, which contain all hero features. Then, thirteen feature groups were projected onto two different representation spaces, and we constructed several network models in each space. Each model would get a $score$ $S_i \in$ [0.0, 1.0], which represents the predicted score based on the feature group or a single hero. Moreover, feature groups in each space are not equally important to each other, and the importance of the same feature changes over time. For example, the role of *Marksman* becomes more and more important as the game progresses. To better simulate the changes in the importance of feature groups, we assigned a time-varying and learnable weight vector to each space in the *Temporal-stage*. In each space, the $win\text{-}score$ vector S = [S_1, S_2, …, S_n] and the important weight vector w^t = [w_1^t, w_2^t, …, w_n^t] are weighted together to calculate the layer score F_P = [F_C, F_H], and then we assign a important weight vector w_P^t = [w_C^t, w_H^t] to the layer score vector to get the final predicted score F_t. Fig. 4 demonstrates the flowchart of the proposed HFN model. The left part displays the selection procedures. The middle part is the Spatial-stage, including two spaces: combat and hero. Each model (Sequence modeling network LSTM) in both spaces is designed to give separate predictions. The right part shows the Temporal-stage. Multiple time-dependent importance weight vectors are combined with the resulting combat and hero space vectors to give the ultimate prediction results.

A. Input Feature

The features of the dataset are mainly divided into two categories: one is the classifiable feature and the other is the numerical feature. For these classification features, we chose to encode them as a vector to represent them better. At the same time, we normalize numerical features like gold to actual values in the range [0, 1]. The specific input is shown in Table I and Table II. After preprocessing, we concatenate all the vectors and variables into two multidimensional vectors. Due to the continuous nature of game data, we select game records within a specific time interval as training data to capture this sequential characteristic. We choose m data frames in l seconds time interval as input, which means representing data X = [x^{t-m}, …, x^t] at game time t.

TABLE I
THE INPUT FEATURE OF COMBAT PART.

Feature	Dimension	Value
Time	1	[0, 20]
Gold	4	[0, 1]
Kill	4	[0, 1]
Tower	4	[0, 1]

TABLE II
THE INPUT FEATURE OF HERO PART.

Feature	Dimension	Value
Time	1	[0, 20]
ID	1	[0, 96]
Type	1	[0, 5]
Win rate	1	[0, 1]
Record	8	[0, 1]
Synergy	19	[0, 1]
Restraint	36	[0, 1]

B. Spatial-stage

This section introduces the Spatial-stage of the HFN model. As mentioned earlier, the Spatial-stage is divided into combat space and hero space. The combat space consists of three LSTM sequential networks called Combat-model. The input of the Combat-model consists of a single feature group ($gold$, $kill$, and $tower$) and the current game time t. The hero space is composed of ten LSTM sequential networks, called Hero-model. The input of the Hero-model consists of the relevant features of a single hero and game time t. Time is an important feature that gives us a better understanding of how the game progresses. When time is added to the input features, the model's accuracy at different time nodes is improved, and the accuracy of the HFN model is also enhanced. The output of

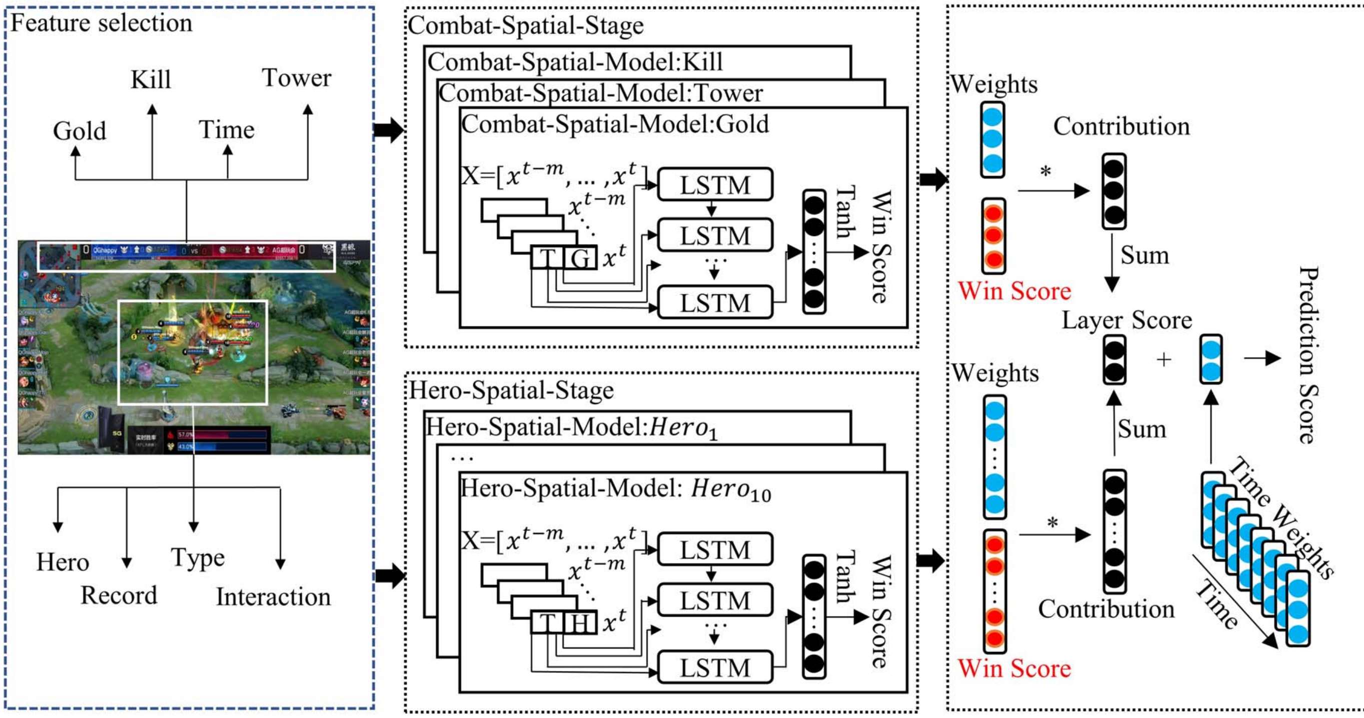

Fig. 4. The flowchart of the proposed HFN model.

Combat-model is $S_i \in [0, 1.0]$, which represents the prediction score based on the value difference of a single combat feature group. The output of the Hero-model is $H_i \in [0, 1.0]$, which represents the prediction score based on the difference between the eigenvalues of a single hero.

C. Temporal-Stage

This section introduces the Temporal-stage of the HFN model. The Temporal-stage is also divided into the combat part and the hero part. The combat part has three time-dependent weights that reflect the relative importance of the three combat features throughout the games. The hero part has ten time-dependent weights that reflect the relative importance of different heroes at different stages. Given the time node t, we linearly combine score vector $c(x, t)$ (where x represents feature group) provided by Combat-model in combat space with importance weight w^t to obtain C^t:

$$C^t = w^t \cdot c(x, t) = \sum_i^n w_i^t \cdot c_i(x_i, \mathrm{t}) \tag{5}$$

Similarly, we linearly combine score vector $h(x, t)$ (where x represents hero feature) provided by Hero-model in the hero space with importance weight w^t to obtain H^t:

$$H^t = w^t \cdot h(x, t) = \sum_i^n w_i^t \cdot h_i(x_i, \mathrm{t}) \tag{6}$$

n indicates the number of feature group or hero. Where C^t represents the effect score of combat data at time point t, and H^t represents the effect score of hero data at time point t. The importance weight of combat at time point t is w^t, we can reach the final P:

$$P^t = C^t \cdot w^t + H^t \cdot \left(1 - w^t\right) \tag{7}$$

In addition, the two weight vectors of Temporal-stage are obtained in different ways. The w^t of the combat part in Temporal-stage is obtained through baseline models using single-node data for predictive analysis. The w^t of hero part in the Temporal-stage is based on the importance of each hero at different stages. Each hero has its skills and attributes, which means that the hero is unique, and each hero has a distinct influence at different game stages. According to the process of the game and the attributes of heroes, the whole game is generally divided into early stage, middle stage, and late stage. Similarly, each hero can be divided into the weak period and the strong period. The same hero has different importance in different game stages, and different heroes have different significance in the same game stage. Fig. 5 shows the game time distribution for the three heroes in 1000 wins. We can see that hero 2 performs best before node 8, indicating that it may have an advantage over other heroes in the early stage. From 8 to 10.5, hero 1 is more vital than other heroes, and from 11 to 14, Hero 3 is stronger. In addition, we can see that the winning rate of the same hero in different time stages is also different. We can understand that the importance of heroes is different in different game stages. Therefore, we randomly select 5000 matches on each node and make statistics on the winning rate of a single hero on a single node. To make the results more accurate, we will perform the same operation ten times and average the results:

$$w_{\hbar}^{t} = \left(\sum_{i}^{10} \frac{h_w}{h_S} \right) /10 \quad (8)$$

$w_{\hbar}^{t}$ represents the hero's stage win rate, h_w represents the number of times the hero wins, and h_s represents the number of times the hero is selected.

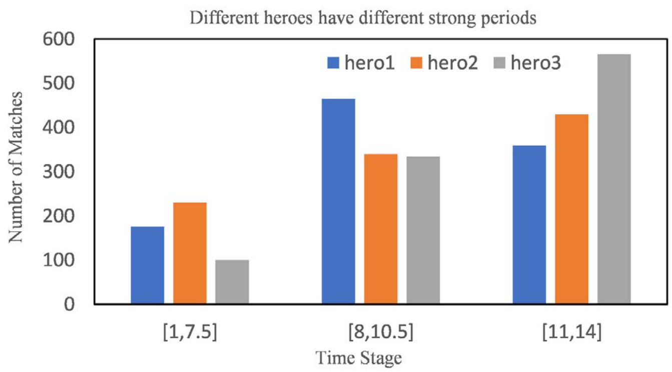

Fig. 5. Different heroes have different strong periods.

D. Contribution

HFN can obtain contribution by multiplying the two result vectors produced by the Spatial-stage and the two time weight vectors provided by the Temporal-Stage. The details are shown on Fig. 4. The contribution of combat feature is C_i^C

$$C_i^c = w_i^t \cdot c_i\left(x_i, \mathrm{t}\right) \quad (9)$$

and the contribution of hero is C_i^H:

$$C_i^H = w_i^t \cdot h_i\left(x_i, \mathrm{t}\right) \quad (10)$$

Contribution is used to represent the role played by feature groups in win predictions of a particular game at a certain time point. It is unequal to the importance of features, and features with significant contributions do not necessarily have high importance. For example, the tower has always been low in importance, but if towers differ significantly between teams, it can have a significant contribution. This also means that if the predicted score of the Spatial-stage is high enough, then relatively unimportant features can still make a non-negligible contribution. Importance is an average score that describes the importance of a feature across all games but may not apply to a particular game. In contrast, the contribution can explain the specific effect of a feature on a game. As shown in Fig. 6, we can see that gold is the most important at almost all time points, but as the game progresses, the gap between the two teams in gold decreases at the later stage, so its contribution also decreases. On the contrary, the number of towers becomes more important because the goal of a MOBA game is to push towers, and the more towers a team pushes, the more likely they are to win. In addition, we also used a single feature to make predictions at seven equidistant nodes in a game. The result is shown in Fig. 7. In the first five minutes of the game, the accuracy of predictions based on the lineup is the highest. When the game enters the middle stage, the accuracy of relying on gold is the highest; in the later stages of the game, the accuracy of predictions based on kills and towers is the highest. This is also consistent with the importance of the features described in Fig. 8 at different time nodes. It can be seen that at the beginning of the game, the only feature with non-zero importance is the hero, which plays a decisive role in the initial win predictions. As the game progresses, the importance of the hero has also declined, while the importance of the other three features has gradually increased.

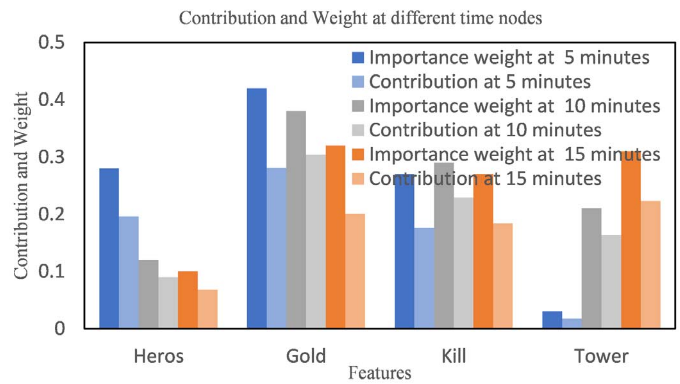

Fig. 6. Contribution and Weight at different time nodes.

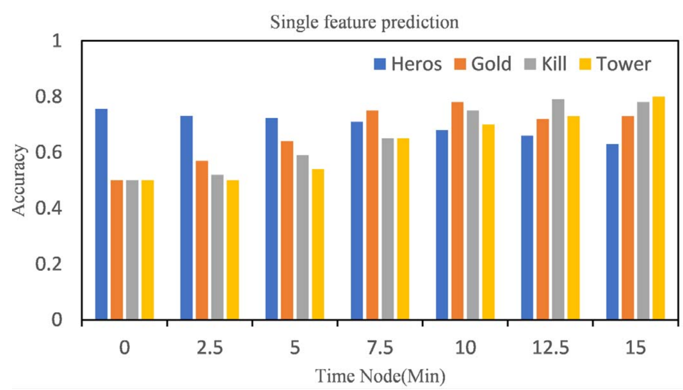

Fig. 7. Single feature prediction.

E. Interpretability: Combat and Hero

High interpretability is an essential advantage of the HFN model. In the Temporal-stage of HFN, two feature contribution vectors are generated for the combat and hero layers. The contribution vector can more intuitively explain the influence of features on the win predictions. More importantly, the sum over each contribution vector will give the layer score at that time point. For example, the score vector of the three feature groups in the combat space at 10 minutes is [0.82, 0.76, 0.75], the weight vector is [0.56, 0.3, 0.14], and the contribution of the three feature groups was [0.46, 0.23, 0.11], by ordering these contributions, we can conclude that the score for the combat layer is 0.8, and the gold is the essential feature in

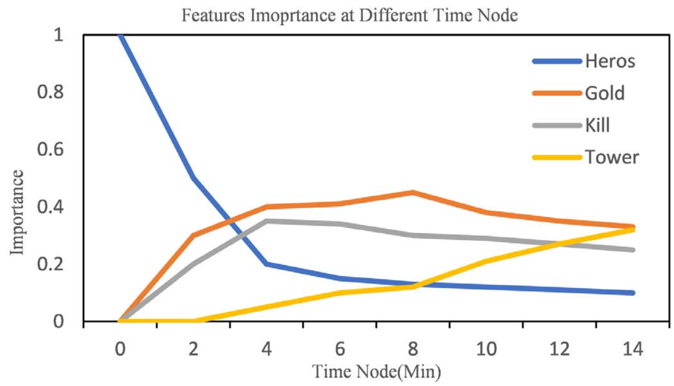

Fig. 8. The features importance at different time nodes.

the game layer. The predicted score of the ten heroes in the hero space is [0.72, 0.56, …, 0.60], and the contribution of heroes is [0.2, 0.19, …, 0.05]. Hero 1 is the most important hero, and the final score of the hero layer is 0.7. At this time, the importance weight of combat layer and hero layer is [0.85: 0.15], so the final win score is 0.785. Based on this information, the commentator can make insightful comments such as: "*At present, the winning rate of the red team is 78.5%, because the red team has a huge advantage in gold. For the red team, hero 1 has gained a huge advantage and tends to dominate the game. At the same time, the lineup is far less influential than the combat data, which makes it difficult for the blue side with the advantage of the lineup to overcome the opposition. But the hero's late performance can still influence the direction of the game.*" In addition, this information can be used to resume the game after the game and determine phased goals of the game, such as pushing towers.

V. EXPERIMENTS

A. Experimental Settings

Our experiment was conducted on the dataset mentioned above. We selected 10% of the 496342 data frames as test set, 10% as validation set, and the remaining 80% as training set. We set the sequential input interval l to 60 seconds. For the parameter settings of the HFN model, in the Spatial-stage, we use a bidirectional LSTM with a dropout probability of 0.2. The model first has two hidden layers of size 128, then a fully connected layer of 256-dimensions is added, and finally, the tanh function is used to calculate the score. It is worth noting that all models use all features as input.

B. Comparing Methods

- Logistic Regression(L R): The model's input is the same as previously mentioned.
- SVM: The model's input is the same as previously mentioned. Parameter C = 0.01, Parameter Gamma = 10000.
- LSTM: we use a bidirectional LSTM with a dropout probability of 0.2. The model first has two hidden layers of size 128, then a FC layer of 256-dimensions is added, and finally, the tanh function is used to calculate the score.
- Transformer: We added two hidden layers and eight attention heads, set the embedding dimension to 256 and dropout to 0.2, and then added 256-dimension FC layers and the tanh function calculation results.

C. Results

This paper mainly adopts Hero-Featured Network (HFN) model to predict the result of *HoK* and compares it with other models. A total of six models were used for comparative experiments, and the accuracy of the models is shown in Table III. The HFN model shows advantages in the early stages, while the other models have slightly lower accuracy at node 0. The reason is that the combat data of the two teams at node 0 is identical, and the only difference is the lineup, while HFN can make better use of role information to predict results, so its accuracy is higher. As the game progresses, combat data play an increasingly important role in in-game prediction, especially the gold of both teams. Due to the HFN model adding the contribution of heroes to the winning rate at different stages, the result of HFN is slightly higher than that of other models. In addition, As shown in Fig. 9, the accuracy decreases after node 7.5 for the following two reasons: 1) the game enters the later stage when the equipment of both teams reaches the maximum value, so the result of the game depends largely on the performance of the players. This makes the later game more difficult to predict; 2) Because most of the games ended before they entered the later stage, the data in the later stage of the game are far less than that in the middle stage. The data is more difficult to train and test, resulting in a significant decline in the prediction accuracy.

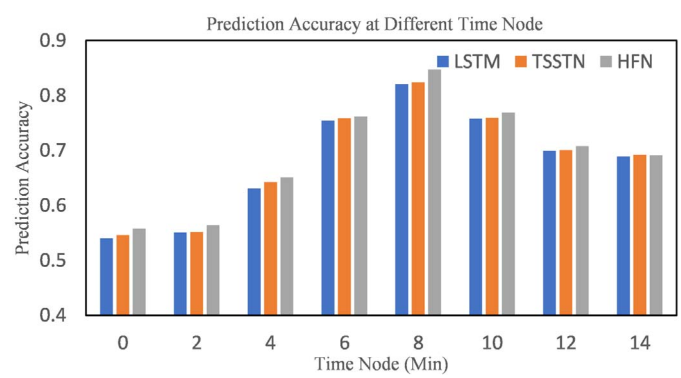

Fig. 9. The average prediction accuracy of three models on eight nodes.

VI. CONCLUSION

In this paper, we collected a large dataset of essential real-time game records from the popular game *HoK*. Then we proposed a Hero Featured Network (HFN) to provide interpretable intermediate results and reliable final win predictions based on this dataset. The core idea of the model structure is to introduce hero interaction into in-game prediction to improve accuracy. It also makes it easier to understand the role of each

TABLE III
THE ACCURACY OF THE SIX PREDICTION MODELS AT FIVE EQUIDISTANT TIME-POINTS.

Model	0.0	3.5	7.0	10.5	14+
SVM	51.80	61.8	73.8	68.7	66.3
LSTM	54	62.9	78.1	73.8	68.9
TSSTN[2]	54.6	63.2	75.6	74	69.2
L R	53.3	63	78.1	70.7	66.8
Transformer	54.3	63.1	78.5	74.3	69.5
HFN	55.8	64.9	81.8	74.6	69.1

hero and the composition of heroes in the game. At the same time, HFN separates the impact of the value difference between the combat and hero data and the relative importance of each feature included in both to decouple the contributions of the different features for a more comprehensive interpretation. Our explainable results can be used in various similar scenarios to enhance the growth of related industries.

REFERENCES

[1] Z. Yang, Y. Wang, P. Li, S. Lin, S. Shi, and S.-L. Huang, "Predicting events in moba games: Dataset, attribution, and evaluation," *arXiv preprint arXiv:2012.09424*, 2020.

[2] Z. Yang, Z. Pan, Y. Wang, D. Cai, S. Shi, S.-L. Huang, W. Bi, and X. Liu, "Interpretable real-time win prediction for honor of kings–a popular mobile moba esport," *IEEE Transactions on Games*, 2022.

[3] B. Wu, "Hierarchical macro strategy model for moba game ai," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1206–1213.

[4] D. Ye, Z. Liu, M. Sun, B. Shi, P. Zhao, H. Wu, H. Yu, S. Yang, X. Wu, Q. Guo *et al.*, "Mastering complex control in moba games with deep reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6672–6679.

[5] D. Ye, G. Chen, W. Zhang, S. Chen, B. Yuan, B. Liu, J. Chen, Z. Liu, F. Qiu, H. Yu *et al.*, "Towards playing full moba games with deep reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 621–632, 2020.

[6] N. Wang, L. Li, L. Xiao, G. Yang, and Y. Zhou, "Outcome prediction of dota2 using machine learning methods," in *Proceedings of 2018 International Conference on Mathematics and Artificial Intelligence*, 2018, pp. 61–67.

[7] Z. Chen, Y. Xu, T.-H. D. Nguyen, Y. Sun, and M. S. El-Nasr, "Modeling game avatar synergy and opposition through embedding in multiplayer online battle arena games," *arXiv preprint arXiv:1803.10402*, 2018.

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

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

[10] V. Hodge, S. Devlin, N. Sephton, F. Block, A. Drachen, and P. Cowling, "Win prediction in esports: Mixed-rank match prediction in multi-player online battle arena games," *arXiv preprint arXiv:1711.06498*, 2017.

[11] S. Demediuk, P. York, A. Drachen, J. A. Walker, and F. Block, "Role identification for accurate analysis in dota 2," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 15, no. 1, 2019, pp. 130–138.

[12] P. N. Andono, N. B. Kurniawan, and C. Supriyanto, "Dota 2 bots win prediction using naive bayes based on adaboost algorithm," in *Proceedings of the 3rd International Conference on Communication and Information Processing*, 2017, pp. 180–184.

[13] P. Grutzik, J. Higgins, and L. Tran, "Predicting outcomes of professional dota 2 matches," Technical Report. Stanford University, Tech. Rep., 2017.

[14] N. Kinkade, L. Jolla, and K. Lim, "Dota 2 win prediction," *Univ Calif*, vol. 1, pp. 1–13, 2015.

[15] Y. N. Ravari, S. Bakkes, and P. Spronck, "Starcraft winner prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 12, no. 1, 2016.