Predicting pick-ban sequence in League of Legends games

Tim Inzitari
University of Notre Dame
Notre Dame, Indiana, USA

Benjamin Lyons
University of Notre Dame
Notre Dame, Indiana, USA

Md Nafee Al Islam
University of Notre Dame
Notre Dame, Indiana, USA

ABSTRACT

With the advancement of gaming technologies and advent of various MMOGs (Massively Multiplayer Online Games), online gaming has extended its span towards global competitions with high stakes. Fields such as understanding gaming psychology, predicting opponents move, predicting the winning probability etc has caught the sight of the research community since these can have significant impact on how the teams perform. League of Legends (LoL) is one of the most played MOBA (Multiplayer online battle arena) games. The picking and banning of champions at the beginning of the game has a huge impact on the game results. This paper aims at predicting the pick and ban sequence of the LoL games. For this task, we utilize a dataset of 4078 professional LoL games to train, validate and test a Recurrent Neural Network (RNN), a Long Short Term Memory (LSTM) Network, a bi-directional Long Short Term Memory network (BiLSTM) and a Convolutional LSTM. We found that a Feature Augmented system fed into an LSTM network performed best with an accuracy of 27.78% which is much better than weighted random guess.

ACM Reference Format:

Tim Inzitari, Benjamin Lyons, and Md Nafee Al Islam. 2022. Predicting pick-ban sequence in League of Legends games. In . ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Esports have spiked in popular over the past decade. There are numerous eSports [15] tournaments going on throughout the globe every year with professional gamers competing against each other for huge amount of rewards. These tournaments include games of various genre such as Role-playing games (RPG) [22], First-person shooter (FPS) [13], Real-time strategy [7], Turn-based strategy [8] etc. Multiplayer online battle arena (MOBA) games are a subgenre of real-time strategy games where there are two teams competing against each other on a predefined battlefield [3]. The difference with the strategy games is that each player controls a single character rather than a unit or group of characters. Also each character is equipped with distinctive abilities or armor. Usually the goal of each team is to destroy the base of their opponent at the other end of the battlefield or to defeat every player of the opponent. The typical ultimate objective is for each team to destroy their opponents' main structure, located at the opposite corner of the battlefield.

League of Legends (LoL), commonly known to as *League*, is one of the most popular MOBA games developed by Riot Games back in 2009 [14]. LoL is referred as the world's biggest esport by many [1]. Its professional scene is divided into multiple domestic leagues which are played regionally. The best teams from the domestic leagues play in the Mid-Season Invitational (MSI) which is a very prestigious international event and the best teams of the world finally go on to play the League of Legends World Championship. In 2019 the LoL World Finals boasted around 100 million viewers. The Super Bowl that same year had 100.7 million.

In LoL, two teams each consisting of five players fight in player vs player combat. Each team has their own half in the map and they try to defend it from the opponent. Each of the ten players from two teams controls a character referred as a *"champion"*. Also based on the battle strategy, the champions are placed in different categories most of which have 2-3 subcategories. Some of the important categories are - Supports(Enchanter, Catcher), Fighter (Juggernaut, Diver), Mage (Burst, Battlemage, Artillery), Marksman, Slayer (Assassin, Skirmisher), and Tanks (Vanguard, Warden). Each individual champion has unique abilities and have different styles of gameplay. The skill set of the human player combined with the champion they pick has a huge impact on the outcome of the game. That is why the pick-ban phase before each match becomes a critical factor in deciding the outcome of the match. In the pick-ban phase, both the teams get opportunity to ban certain champions from the game and pick certain champions for their team. In the professional LoL matches, at first, both teams ban 3 champions each, then pick 3 each followed by another ban of 2 and pick of 2 for each team. This way, both teams ban 5 champions each and pick 5 champions each. This leads to a total of 20 pick-ban events in each of the games. Figure 1 shows the sequence of bans and picks for two team denoted by colors purple and blue.

While making these pick-ban choices, there are many things to consider for both the teams. During the pick-ban phase, both the teams try to build a winning strategy by predicting the moves of the opponent while playing to the strength of their own team composition. Huge efforts go into scouting the opponent teams using various sources of information. These information involves understanding the mindset of the opponent, the nuances of the meta and many other factors. Using machine learning and AI based techniques to predict various events of physical sports and esports is a thriving research topic. Recently it has grown a huge demand in the video games industry because of all the hypes and huge influx of monetary rewards associated with the game.

In this paper, we present our approach to predict the pick and ban sequence of the LoL games. We tested the performance of

https://doi.org/10.1145/1122445.1122456

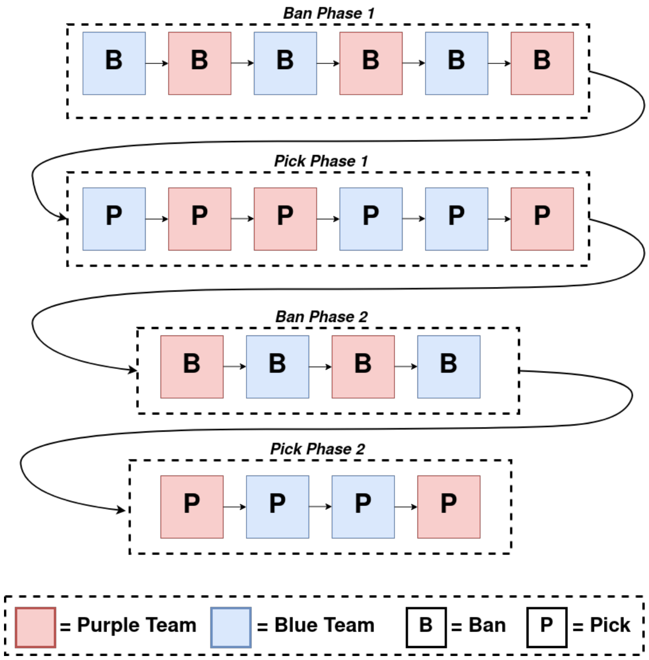


Figure 1: Pick-ban sequence for two teams in LoL games

four sophisticated auto-regressive deep neural network models to predict the pick and ban sequences.

We used RNN [28], LSTM [32] (a variant of RNN), BiLSTM [21] and Convolutional LSTM [29] networks for this experiment which are specially suitable for time series predictions. At first, we devised a baseline solution with one hot encoded champion IDs as feature vectors for the models. In that solution, we found that the BiLSTM works best among the four models tested. Further, we provide our final solution with a feature augmentation method applied. We then show that the feature augmentation significantly improved the performance of the BiLSTM model from the baseline solution.

The remainder of this paper is structured as follows. Section 2 shows an overview of our literature survey. In Section 3, we define our problem mathematically. Section 4 shows our solution idea followed by Section 5 which lays out the data and experimental setting. We present our results and relevant discussion in Section 6. Finally, Section 7 and Section 8 shows some of the limitations of our work along with concluding remarks.

2 RELATED WORKS

For this work, we reviewed two types of literature. At first, we focused on the prior works on application of machine learning and AI in predicting events in games. Later we focused on our specific problem which involves time series forecasting on sequential data.

2.1 Machine learning and AI in games

Adopting machine learning and AI techniques to predict moves and events in games, guessing the opponents' mindsets, enhance the performance of virtual agents has been vastly explored by the researchers. Significant efforts have been made to predict moves of physical strategy board games such as game of Go [16, 33] and Chess [17, 26]. There are numerous works on predicting sports outcome, events and actions too [2, 6, 23, 24]. However, in this work, we mostly focused on ML and AI for predictions in computer games. Semenov et al., evaluated and compared the performance of various machine learning algorithms in predicting the outcome of a MOBA game - Dota 2 [27]. They tried out four ML algorithms - Naive Bayes classifier, Logistic Regression and Gradient Boosted Decision Trees and Factorization Machines and found the best results from Factorization Machines. Conley et al., proposed a recommendation system for picking champions in Dota 2 using simple simple techniques like Logistic Regression and K-nearest neighbors [10]. Yang et al., presented a data-driven method for identifying patterns in combat which lead to successful game outcomes [36]. They used sequence of graph representation; each of which describes how a player involved in combat with another. Further, they built a decision tree from the graphs to determine successful combat rules. Yang et al., released a large dataset of in-game features of the MOBA game - Honor of Kings and proposed to predict four types of important events in interpretable way [37]. They also proposed an evaluation metric for these sort of works - fidelity-based evaluation. Do et al., used ML based technique to predict the outcome of League of Legends games based on the human controller's skill level combined with the selected champion [12]. Costa et al., used the ban-pick data to train ML based classifiers to predict the outcome of the LoL games [11]. However, they do not predict any events during the ban-pick phase like ours. Summerville et al., used Bayesian networks and LSTM to predict the ban-picks in DOTA games [34]. Hong et al., used LoL ban-pick data to predict the bans and picks of both teams and proposed a recommendation system based on that [20]. However, they made separate models for each team to predict 10 events each. We are predicting all 20 events using a single model. Also, they used two classifiers - random forest and fully connected neural networks which are not specialized for predicting on sequential data.

There have been many other works on predicting events and outcomes of LoL games such as [4, 18, 30, 31]. However, we did not find any prior work which works on predicting the bans and picks sequence of League of Legends games using sophisticated algorithms like LSTM or RNN.

2.2 Deep learning methods on sequential data

Various statistical methods have been applied before for time series prediction. One of them is Autoregressive Moving Average (ARMA) by Brockwell et al., which is able to capture temporality in the data [5]. Some of its variants - Seasonal Autoregressive Integrated Moving Average (SARIMA) and Autoregressive Integrated Moving Average (ARIMA) was implemented by Valipour for long-term runoff forecasting in the United States [35]. Chen et al., tried out hybrid methods and combined SARIMA with SVM to improve the predictions [9].

Deep learning based unsupervised algorithms are prominently used in time series anomaly detection tasks because of their ability to capture the non-linearity of data. Recurrent Neural Network (RNN) or Long Short-term Memory (LSTM) structures have been especially popular in time series anomaly detection because of their ability to capture the patterns of time series. Sherstinsky described fundamentals of RNN and LSTM and how they capture the temporal properties of time series data [28]. Both of the structures are capable of remembering the previous states while processing variable length sequences of inputs. However, Pascanu et al., pointed out the limitation of training RNN on long sequences such as the vanishing and exploding gradient problems [25]. Hochreiter shows how these problems are solved in LSTM as it involves no exponentially fast decaying or growing factors in its back propagation process [19].

3 PROBLEM DEFINITION

As discussed before, we tackle the problem of predicting the pick and ban sequence of the LoL matches where component in the sequence is a unique character. So, this problem perfectly fits the description of a *categorical time-series prediction problem.* In this section we describe our problem mathematically.

Let P_i be a vector of i unique sequential elements representing the pick/ban events (v) of a LoL game.

$$P_i = v_1, v_2, v_3, v_4, \ldots., v_i \qquad \text{where } 1 \leq i \leq 19$$

Also, $P_i \subset C$, where C represents the set of all champions. Currently, the number of elements in C is 157. At any point,

$$v_i \subset C - \{P_{i-1}\} \qquad \text{for all } i \geq 2$$

Now, given the sequence of i events as shown above – where i has a range of 1 to 19, We want to estimate a function $G(v)$ which can predict the $(i+1)^{th}$ event.

$$G(v) = v_{i+1}$$

4 OUR SOLUTION

We propose a solution that treats the problem as a categorical time series prediction. Since picks and bans happen in the same order every time, we can organize the data into a sequence of 20 events in time. We used this data to train, validate and test four neural network based systems: a RNN system, a LSTM system, a Bidirectional LSTM system, and a Convolutional-LSTM system. These experiments helped determine that a Bidirectional LSTM model performed the best. We provide our solution to this problems in two folds:

- **Baseline solution – using just the one hot encoded champion IDs as feature vector:** We do this by encoding the series of champions by their in game ID for their hero name. This is then encoded in a one hot fashion to generate our feature vectors for each pick ban event, a set of sequences that are each 20 in length with feature vectors encoded for the hero chosen at that event. We then split these sequences into windows as described later and we have our labels and inputs into a Bidirectional LSTM system to generate the remaining sequences.
  Each instance of our LSTM system only predicts the exact next value of the pregame event. This is then repeated as input for any additional predictions needed to complete the sequence of 20.

Most Picked	Most Banned	Highest Presence
Kai'Sa 37%	Thresh 48%	Renekton 72 %
Leona 32%	Renekton 44%	Thresh 71%
Renekton 28%	Lucian 29%	Kai'Sa 53%

Table 1: The three most picked, most banned, and highest presence champions in the dataset. This is measured according to rate across all games

- **Augmenting the previous feature vectors with additional information:** Further, to optimize the results, we applied an additional method to augment our feature vectors to gain more knowledge of the current hero selection pool. This augmentation included information such as a hero's global pick and ban rates, as well as how often they play each role in the game. After the input of previous selected heroes we apply Equation 1 as a transformation onto the input in the next cycle for the feature vector. To do this we define a custom model structure that allows for this augmentation layer to be implemented before an LSTM system.

$$v_c = \begin{cases} 0 \text{ if Champion not available} \\ BanRate_c \text{ if choice is a ban} \\ PickRate_c \text{ if teams first pick} \\ elsePickRate_c * \sum PosRate_c * \left[1 - max\left(PosRate_{pc}\right)\right] \end{cases} \tag{1}$$

where c is the champion id and
pc is the list of previous picked champions on a team

Figure 2 shows a description of how we implement this new system. The new system still has outputs and inputs of the one-hot-encoding for a champion but it applies a transformation layer to augmented information to the system before applying it to the rest of the system.

5 DATA AND EXPERIMENT SETTINGS

5.1 Dataset acquisition and description

The dataset for this application consists of pick/ban sequences from 4078 professional games in the 2021 season and was downloaded from the League of Legends Esports Wiki . Each data point consists of a sequence of 10 champion picks and 10 champion bans. There are 158 champions total, which means that about 12.5% of all champions are present in each sequence.

Figure 3 displays the pick and ban rate distributions for every champion, along with the win rate distribution. Its evident that the vast majority of champions are neither picked nor banned at a substantial rate. However, there are a few that are selected around 30% of the time. Table 1 shows the top three most picked, most banned, and highest presence champions in the game. Note that presence rate is equal to the sum of pick rate and ban rate. Unsurprisingly, there is significant overlap between the top 3 in each of these three categories. This makes sense because typically the most picked and most banned champions are the strongest. Thus, in most draft sequences, these champions are either picked or banned because they are so powerful.

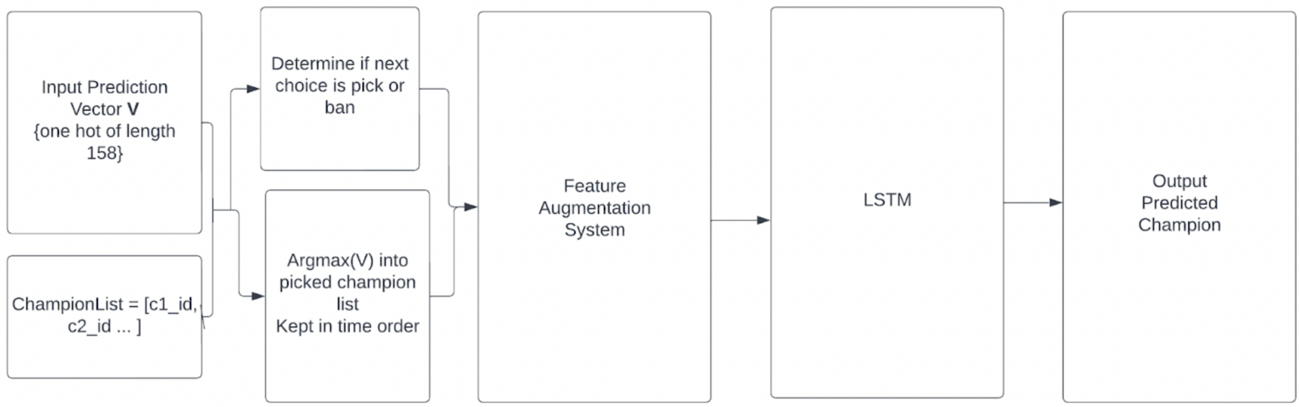


Figure 2: Augmented Model Description

It is also worth noting that the winrates for most champions hover around 50%, as shown in Figure 3. There are a couple outliers, which is simply because some champions are played no more than a handful of times. These few champions have such extreme winrates since they are unpopular, so the sample size if far to small. However, given enough games the win rate of a champion is a good measure of strength in professional games.

In professional drafts, the champions selected for picks and bans are strategic. Some champions are considered quite strong and are highly contested, whereas others are considered to be weak. Additionally, there are some combinations of champions that work well together or counter each other. Figure 4 shows the frequency with which pairs of champions are selected in the drafts. Note that no pairings with a frequency of less than 50 occurrences are shown, since the vast majority of possibilities do not occur. However, there are a few pairings that stand out as being selected far more often than others.

First, we consider the synergistic pairings. These are pairs of two champions that are picked by the same team, and theoretically they should reflect strategic synergies. The first column in Table 2 displays the top five most common synergies. Amongst these synergies, 4 of them reflect botlaner and support champion combinations. Without going too much into the details of the game, the botlaner and support spend the most time working together in game, so picking synergistic champions is particularly important in these roles. In fact, the most frequent pairing of Aphelios and Thresh reflects arguably the most powerful botlane and support pairing in the game.

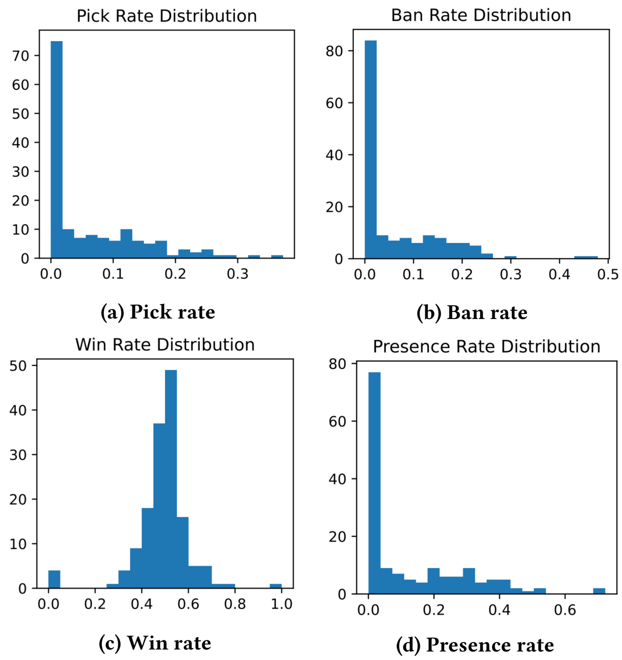
There are also counter pairings, which are pairs of two champions that are frequently found on opposing teams. Part of draft strategy is selecting champions that effectively counter those in the same role on the enemy team. Thus, we would expect most counter pairings to reflect these strategic ideas as well. Indeed, this is evident again in Table 2. Four of the top five counter pairings reflect champions selected in the same role for different teams: Kai'Sa + Xayah, Alistar + Rell, Kai'Sa + Tristana, and Lee Sin + Viego. The top picked counter of Kai'Sa + Xayah is very well known in the community, so its unsurprising to see it listed as the most common one. The remaining counter of Kai'Sa + Rell reflects a botlane and support counter pick. This is not surprising since opposing botlanes and supports spend a significant amount of the game in the same area of the map, which means that they can effectively counter each other. Thus overall, the champion synergies and counter frequencies are reflective of common strategic considerations in the game. This implies that our model should be able to learn these pairings and use them to make strategic predictions about drafting.

Synergy Frequencies (#)	Counter Frequencies (#)
Aphelios + Thresh 960	Kai'Sa + Xayah 326
Alistar + Kai'Sa 610	Alistar + Rell 308
Kai'Sa + Rell 536	Kai'Sa + Tristana 286
Gnar + Kai'Sa 528	Lee Sin + Viego 285
Kai'Sa + Leona 476	Kai'Sa + Rell 260

Table 2: Top five most common synergistic pairings and counter pairings

5.2 Experimental Setup

5.2.1 Data Preprocessing. Regarding data preprocessing, we had to convert the raw draft data to a form which is usable by a neural network model. With the input data being a simple list of hero names in the order their events occurred, the first step was to enumerate the text data. This was done by enumerating each hero possible to be chosen, as well as an <UNK> character to handle missing data into 158 possible inputs for our system. Resulting in each game sequence to be a series of 20 numbers representing various heroes chosen and banned. We then feed these numbered heroes and one-hot encode them, turning out sequence of 20 integers, into a sequence of 20 one-hot-158 length vectors. This system allows for the extension of this feature vectors should we desire additional inputs about the sequence state. We then split our set of sequence matrices into train, test and validation sets. We do this on a split of 80% train, 10% test, and 10% validation.



(a) Pick rate **(b) Ban rate**

(c) Win rate **(d) Presence rate**

Figure 3: Champion pick/ban distributions across all champions

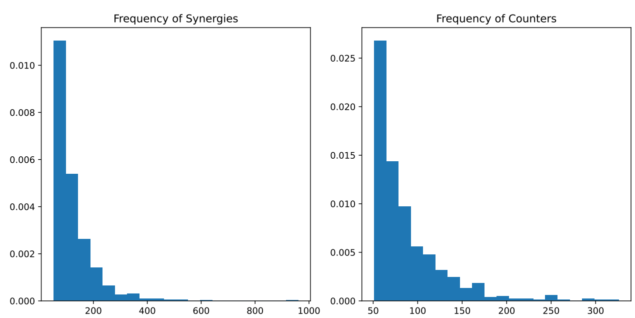


Figure 4: Champion Pairs Pick/Ban Distributions (a) The distribution of frequencies of synergistic pairings. (b) The distribution of frequencies of counter pairings.

We then send these sequence matrices through our Window Generator to convert them into inputs and outputs. Each window is two sided and sum to a total sequence length of 20. Order is always maintained from the original sequences since there are time elements involved. Figure 4 shows how a window would be generated for a simplified example. The previous elements of the system are used to predict the next value.

Our system needs to be able to peer further than one step in the future and we do this as shown in Figure 5. We treat multi-step sequences as consecutive single step solutions feeding into each other, allowing us system to build one system for any variation of prediction. After the Window Generator system is completed, we have our dataset ready for input into our systems.

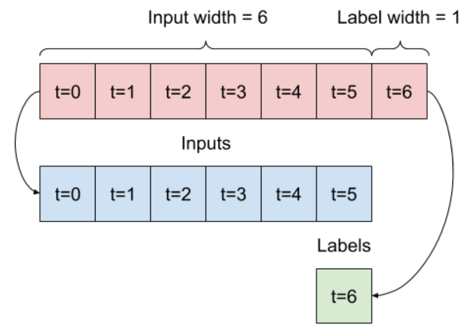


Figure 5: Example Window Generation

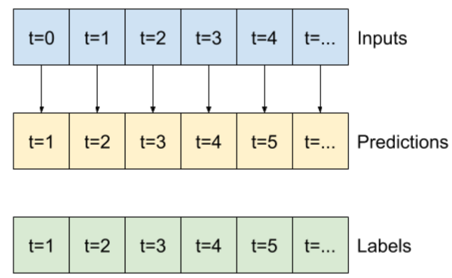


Figure 6: Consecutive Single Step Label Windows

5.2.2 Training. For our initial experiment we set up the system to predict only the final pick of the draft. We implemented several models and compared the results. The architectures we used for testing were a convolutional neural network set to accept sequence data, an RNN system designed to accept sequence data and an LSTM system applied to the sequence, as well as a Bidirectional LSTM system. All systems utilize categorical cross entropyloss functions and Adam optimizers.

Our pre-processed data is fed into these models to generate our output, and then repeated as necessary to predict any multi-step predictions. The output for each model is a set of probabilities for each champion, the maximum of these probabilities is our predicted hero. As of now our feature vectors only include this one hot encoded list of heroes, this will be changed in the future to include additional information about the team making the pick.

For the feature augmented system, we applied it only to the LSTM model, as the systems augmentation as the augmentations make little sense when replied in reverse, as the previous picked heroes will always be evaluated to zero causing a nonsensical result to be produced.

5.3 Random Guess and Weighted Random Model

We also wanted to establish an absolute lower bound for performance, based on a randomized selection model. The initial randomized model we established selects a random champion for every pick. Since there are 158 total champions, this yeilds to an accuracy of 0.6%.

18 Input 2 Output: Accuracy			
	Train	Validation	Test
RNN	22.9%	5.26%	7.21%
LSTM	24.1%	6.74%	8.23%
BiLSTM	28.9%	8.46%	10.5%
CNN	15.8%	6.88%	6.84%

Table 3: Accuracy of the baseline solution for sequence setup of 18 input, 2 step prediction

Split	Total Dataset Accuracy
15:5	0.0%
17:3	0.0%
18:2	0.075%
19:1	1.72%

Table 4: Accuracy for sequence splits with weighted random strategy

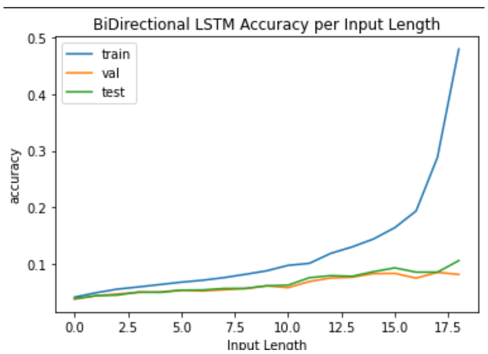


Figure 7: Bidirectional LSTM Accuracy over various input lengths

However our data analysis shows that not all champions are picked with equal probability. We took into account the pick/ban distribution across all the champions as well as the distribution of synergistic pairings and counter pairings, and we came up with the weighted random model. The weighted random model provides slightly more educated guesses to generate a prediction. We then tested this educated random model on the entire dataset.

Table 4 shows the prediction results of the weighted random model for various input:output split. We expect that, our solution for this problem performs significantly better than the weighted random model.

6 RESULTS AND DISCUSSION

In this section, we at first discuss the prediction performance of our baseline solution. Then we discuss how the final solution with augmented feature vectors improved the accuracies of the model.

6.1 Baseline Solution

At first, we evaluated the performance of the four models with our baseline solution. For this experiment, we evaluated the models

Feature Augmentation Accuracy			
Split	Train	Validation	Test
15:5	29.25%	12.89%	11.83%
17:3	38.56%	21.42%	21.63%
18:2	44.59%	26.48%	27.78%
19:1	53.98%	38.32%	37.84%

Table 5: Accuracy for the feature augmented system defined by Figure 2

Prediction Step	Accuracy
15 -> 16	84.4%
19 -> 20	37.84%

Table 6: Shows different accuracies at sepearate steps of the augmented system

with the sequence of (18,2) for our Timeseries window on the pick ban phase. This means that given 18 events, we are evaluating the models on predicting the next two events. The results of the baseline solution from our four models is shown in Table 3. The accuracy display are for each individual sequence cycle, counting each output in the sequence as its own event.

While all models performed significantly better than the weighted random model (see Table 4), it can be observed that the Bidirectional LSTM model yielded the best results. Note that this system also took the longest to train. It still has significant room for improvement as more experiments occur, tuning hyper parameters and hidden layer numbers should improve the model. The baseline was conducted on a simple network that only had one hidden layer outside of the BiDirectional LSTM, and we suspect that further tuning should lead to improved results.

The worst performing model was the CNN, which makes sense as it is not as fundamentally designed for Sequential Data as recurrent networks are. The more specialized networks performing better is not a surprise.

It is also not a surprise that within these sequential designed networks the test accuracy increased as their complexity increased. This can be seen by how the BiLSTM had the best performance, followed by the LSTM and then the simpler RNN on the test data.

Also interesting is the consistency of the Bidirectional LSTM model, which is shown in Figure 7. One of the main reasons we decided to ultimately focus on the Bidirectional LSTM system is how consistent the predictions were at all steps in the sequence. It does show an increase as you get near the end of sequences, but even with little information the number of correct predictions is high above random chance.

6.2 Final solution with augmented feature vectors

We found significant improvement in the prediction performance of the models after adding the feature augmentation. Table 5 shows how the different splits improve accuracy compared to previous experiments.

Its interesting to note that the accuracy rates are not static per pick, as Table 6 shows, there is a large gap between a 15 input sequence when predicting its 16th step and a 19 input sequence predicting the 20th step. This is due in large to the diversity of champion choices showed in Figure 3. As you reach the twentieth choice the number of commonly picked heroes is diminished, so teams end up picking from a wider range of second or third tier power champions for their role, which encompasses a much higher percent of choices compared to higher power champions.

7 LIMITATIONS AND FUTURE WORK

For future work we will test other time series configurations on the system as a whole. We will also attempt to make the system more dynamic in the series explanations, allowing for more flexible inputs to be issued as currently it would require seperate training for any new window periods applied.

We also will explore different additions to the feature vectors such as pairings between champions and more complex systems of role calcuation. Other additions that will be tested are physical aspects of each champion such as base statistics and abilities.

8 CONCLUSION

MOBA games have become a massive attraction for many; specially because of professional eSports tournaments. ML and AI techniques are frequently employed to predicting outcomes, events and player strategy/mindset of these games. This paper uses LSTM networks to predict the pick-ban phase events of one of the most played MOBA games which is League of Legends. This models can be very useful for teams ton determine opponents mindset and set strategies accordingly.

REFERENCES

[1] Agha, B. *League of Legends: Players and esports.* PhD thesis, 2015.
[2] Al Islam, M. N., Hassan, T. B., and Khan, S. K. A cnn-based approach to classify cricket bowlers based on their bowling actions. In *2019 IEEE International Conference on Signal Processing, Information, Communication Systems (SPICSCON)* (2019), pp. 130–134.
[3] Alejandro, C., and Ramirez, E. Towards procedural map and character generation for the moba game genre. *Ingeniería y Ciencia 11* (08 2015), 95–119.
[4] Ani, R., Harikumar, V., Devan, A. K., and Deepa, O. Victory prediction in league of legends using feature selection and ensemble methods. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (2019), pp. 74–77.
[5] Brockwell, P. J., and Davis, R. A. *Time series: theory and methods.* Springer Science & Business Media, 2009.
[6] Bunker, R., and Susnjak, T. The application of machine learning techniques for predicting results in team sport: a review. *arXiv preprint arXiv:1912.11762* (2019).
[7] Buro, M. Real-time strategy games: A new ai research challenge. In *IJCAI* (2003), vol. 2003, pp. 1534–1535.
[8] Caldwell, N. Theoretical frameworks for analysing turn-based computer strategy games. *Media International Australia 110*, 1 (2004), 42–51.
[9] Chen, K.-Y., and Wang, C.-H. A hybrid sarima and support vector machines in forecasting the production values of the machinery industry in taiwan. *Expert Systems with Applications 32*, 1 (2007), 254–264.
[10] Conley, K., and Perry, D. How does he saw me? a recommendation engine for picking heroes in dota 2. *Np, nd Web 7* (2013).
[11] Costa, L. M., Mantovani, R. G., Monteiro Souza, F. C., and Xexéo, G. Feature analysis to league of legends victory prediction on the picks and bans phase. In *2021 IEEE Conference on Games (CoG)* (2021), pp. 01–05.
[12] Do, T. D., Wang, S. I., Yu, D. S., McMillian, M. G., and McMahan, R. P. 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, 2021), FDG'21, Association for Computing Machinery.
[13] Frostling-Henningsson, M. First-person shooter games as a way of connecting to people:"brothers in blood". *Cyberpsychology & behavior 12*, 5 (2009), 557–562.
[14] Games, R. *League of legends.* GALERA, 2020.
[15] Hamari, J., and Sjöblom, M. What is esports and why do people watch it? *Internet research* (2017).
[16] Harrison, B. A. *Move prediction in the game of go.* PhD thesis, Citeseer, 2010.
[17] Hassabis, D. Artificial intelligence: chess match of the century. *Nature 544*, 7651 (2017), 413–414.
[18] Hitar-Garcia, J.-A., Moran-Fernandez, L., and Bolon-Canedo, V. Machine learning methods for predicting league of legends game outcome. *IEEE Transactions on Games* (2022).
[19] Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6*, 02 (1998), 107–116.
[20] Hong, S.-J., Lee, S.-K., and Yang, S.-I. Champion recommendation system of league of legends. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)* (2020), pp. 1252–1254.
[21] Huang, Z., Xu, W., and Yu, K. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).
[22] Mackay, D. *The fantasy role-playing game: A new performing art.* McFarland, 2017.
[23] McCabe, A., and Trevathan, J. Artificial intelligence in sports prediction. In *Fifth International Conference on Information Technology: New Generations (itng 2008)* (2008), IEEE, pp. 1194–1197.
[24] Min, B., Kim, J., Choe, C., Eom, H., and McKay, R. B. A compound framework for sports results prediction: A football case study. *Knowledge-Based Systems 21*, 7 (2008), 551–562.
[25] Pascanu, R., Mikolov, T., and Bengio, Y. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning* (Atlanta, Georgia, USA, 17–19 Jun 2013), S. Dasgupta and D. McAllester, Eds., vol. 28 of *Proceedings of Machine Learning Research*, PMLR, pp. 1310–1318.
[26] Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature 588*, 7839 (2020), 604–609.
[27] Semenov, A., Romov, P., Korolev, S., Yashkov, D., and Neklyudov, K. Performance of machine learning algorithms in predicting game outcome from drafts in dota 2. In *Analysis of Images, Social Networks and Texts* (Cham, 2017), D. I. Ignatov, M. Y. Khachay, V. G. Labunets, N. Loukachevitch, S. I. Nikolenko, A. Panchenko, A. V. Savchenko, and K. Vorontsov, Eds., Springer International Publishing, pp. 26–37.
[28] Sherstinsky, A. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena 404* (2020), 132306.
[29] SHI, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-k., and WOO, W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems* (2015), C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28, Curran Associates, Inc.
[30] Silva, A. L. C., Pappa, G. L., and Chaimowicz, L. Continuous outcome prediction of league of legends competitive matches using recurrent neural networks. In *SBC-Proceedings of SBCGames* (2018), pp. 2179–2259.
[31] Smit, R. A machine learning approach for recommending items in league of legends.
[32] Staudemeyer, R. C., and Morris, E. R. Understanding lstm - a tutorial into long short-term memory recurrent neural networks. *ArXiv abs/1909.09586* (2019).
[33] Stern, D., Herbrich, R., and Graepel, T. Bayesian pattern ranking for move prediction in the game of go. In *Proceedings of the 23rd International Conference on Machine Learning* (New York, NY, USA, 2006), ICML '06, Association for Computing Machinery, p. 873–880.
[34] Summerville, A., Cook, M., and Steenhuisen, B. Draft-analysis of the ancients: predicting draft picks in dota 2 using machine learning. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference* (2016).
[35] Valipour, M. Long-term runoff study using sarima and arima models in the united states. *Meteorological Applications 22*, 3 (2015), 592–598.
[36] Yang, P., Harrison, B. E., and Roberts, D. L. Identifying patterns in combat that are predictive of success in moba games. In *FDG* (2014).
[37] Yang, Z., Wang, Y., Li, P., Lin, S., Shi, S., and Huang, S.-L. Predicting events in moba games: Dataset, attribution, and evaluation. *arXiv preprint arXiv:2012.09424* (2020).