2022 4th International Conference on Electrical, Computer Telecommunication Engineering (ICECTE)
29-31 December 2022, Rajshahi-6204, Bangladesh

# DOTA2 Winner Team Prediction based on Stacked Bidirectional LSTM Network

Jalal Uddin
*Dept. of MTE, RUET*
Rajshahi 6204, Bangladesh
com.jalal.uddin@gmail.com

Iffat Fahmida
*Dept. of MTE, RUET*
Rajshahi 6204, Bangladesh
iffatfahmida23@gmail.com

Sumaya I. Moyeen
*Dept. of MTE, RUET*
Rajshahi 6204, Bangladesh
sumaya@mte.ruet.ac.bd

Md. Mehedi Hasan
*Dept. of MTE, RUET*
Rajshahi 6204, Bangladesh
mmehedihasann@gmail.com

***Abstract*—With the swift development of e-sports technology, games are now using techniques of machine learning in several scopes of application. One of the most significant applications of gaming is gaining insights into match details and various types of predictions. Competitive games provide enormous amounts of data from a match. In the context of data and analytics, the word ”insight” refers to the analysis of those game data to find out the pattern or relationship among the variables. In this work, the match stats data as kills-deaths-assists along with the match duration and the heroes has been used to predict the winner of a particular match based on a stacked Bi-LSTM neural network model. The match data which is used in training the model, is collected from the OpenDota API. The results demonstrate the performance of the stacked Bi-LSTM model in predicting the winning team of a game compared to other machine learning classifiers.**

***Index Terms*—Dota 2, E-sports, Bi-LSTM, Machine Learning**

## I. Introduction

DOTA, which stands for ”Defense of the Ancients,” is one of the most phenomenally popular multiplayer online battle arena (MOBA) games, which is also known to be one of the most complex games of this era [1]. Ten people play the game primarily in two teams, the Radiant team and the Dire team, consisting of five players each. Each player chooses a hero according to his playing role in the game, and generally, each game has a duration of about 30 to 40 minutes on average [2]. The complexity of the game can be addressed by the playing strategies. Each player plays with one unique hero out of 137 heroes (by 2022), where each hero has multiple unique abilities and spells. Also each player has to build items based on game timing, situation and playing role which adds special abilities to each hero. Along with the heroes, the mechanism of the game consists of farming creeps to earn gold, taking down enemy towers and buildings and finally the enemy Ancient to win the game [3]. All these decisions made by a player composes enormous data and as a result of the accessibility of high-volume and multidimensional data produced throughout every match, DOTA2 has become a part of research in the machine learning field. Recently, five bots created by Elon Musk-funded research lab OpenAI and trained by reinforcement learning beat the professional champion human team in a best-of-three competitive match [4].

OpenDota is a data analysis platform, one of whose sponsors is OpenAI. Detailed information about any particular match, can be gathered from their API. The high volume of data that is generated from a match can be used to carry out research on the game's strategy. One of the most crucial stages of winning a match is hero selection. Every hero has a counter-hero, thus every spell can be countered with ”items,” and the items themselves can be countered by other ”items” as well. The number of kills, deaths, and assists conquered by a hero possesses a high impact on a match. Eggert et al. [5] proposed a strategic machine learning approach to player-role classification using game replays containing each player's kill-death stats. The match duration also heavily impacts the winning decision. In the game, there are early strong heroes and recent strong heroes. In some specific hero combinations, a team may decide to finish the game early if they have early strong heroes on the team. This circumstance affects the match duration impact, alongside the game stats of the heroes. Every single piece of information about a particular match can be gathered through the OpenDota API. Furthermore, an e-sports player can achieve strategic knowledge from post-game data analysis [6].

The research introduces a method of match winner prediction based on heroes' choices along with their game statistics of kills, deaths, and assists, combined with the impact of game duration. With building a classifier of winners from the statistical data, this work proposes a intuitive match winner prediction model. The experimental design of the work was to train a neural network model based on the features of game stats to classify the winner as Radiant or Dire. This work also compared the model with the machine learning classifiers KNN, SVM, and Random Forest and evaluated performance. The analytical design of the work was to provide a comprehensive analysis of the dataset and the model we prepared. Another intended section of our work was to build a web application to run live experiments with our model.

In the next section of the paper, we will discuss the related works on game prediction, especially DOTA2. In the following section, an in-depth analysis of the dataset will be provided. In Section IV, we then explore the methodology of this work. The result analysis and performance metrics will be discussed in the later section of the paper. The concluding remarks section has been provided in Section VI.

## II. Related Works

Research into e-sports is rapidly growing with the increase in computing capability and the increasing involvement of artificial intelligence in games. Such as OpenAI Five [4], a five team AI bot, played against themselves for 10,000 years to compete over human champion teams. However, in the academic field, the majority of the research has been focused on building machine learning models to predict some outcomes. Anshori et al. [7] proposed a method for predicting the outcome (win or lose) based on SVM classifier combined with Particle Swarm Optimization (PSO) technique. In their work, they used the UCI ML Repository dataset to train their model. PSO was used to determine the optimum parameters for the SVM classifier. While ranging the C-parameter of SVM from $10^{-6}$ to $10^{6}$ their work had achieved accuracy improvisation from 0.54 to 0.60 approximately. Hodge et al. [8] discussed a comparative analysis of real-time win prediction among Linear Regression, Random Forest and LighGBM models. They also had implemented a real-time system to predict winning team with the help of OpenDota API and Dota2 GSI (Game State Integration), using C# live parser. Authors have used almost 5.7k game replays in their work and used an open-source parser to extract information from the replay binaries. Katona et al. [3] proposed a deep learning method to predict the time to die of a hero. Authors also used game replays and parsed game data from the replays to make a dataset for their work. Almost 9.8k replays of professional and high rank matches were used in their work. Their best performing feed-forward neural network model has achieved a precision of 0.54 approximately. Zhang et al. [9] proposed a Bidirectional LSTM (Long Short-Term Memory) method to provide line-up recommendation. The best combination of heroes to win against another combination is the outcome of their work. As a result of their work, 1 hero recommendation accuracy was 78.32% and for team line-up, their Bi-LSTM model had achieved 67.74% accuracy. Akhmedov and Phan [10] showed a comparative analysis of the Random Forest, Linear Regression and LSTM model to predict outcome based on the features of game scores. With the help of DOTA2 GSI, they collected real-time data from matches. Their analysis resulted in better accuracy with the LSTM model than the other models in comparison. Looi et al. [11] presented an item suggestion system based on commonly used purchase strategies. They approached Rule based classifier using the Apriori algorithm [12] and Logistic Regression classifier and evaluated it based on how accurately it predicted the purchase by players. The item recommendation was improved by their clustering methodology and trained separately for each clusters. Similarly, sequential item recommendation method was presented by Dallmann et al. [13]. They evaluated a set-based approach using Linear Regression and some neural network models.

## III. Dataset Preparation

The primary step of our work was to prepare a dataset containing match data from recent games played by professional players. In order to prepare the dataset, the OpenDota API has been used to collect match information in JSON format. As selection criteria, only professional matches and top-tier ranked players' public matches have been collected in the range from April 2022 to August 2021. DOTA 2 changes patch updates very frequently, and major updates are released almost yearly span. Hence, very old match data weren't used in this work, since major changes often bring changes in team combination and game play. In total 27,000 matches data were scapped and after cleaning corrupted collections, 26,062 match data were used to perform our work. Since the winning team has been elected as the classification label for this work, there are 13,168 Radiant Team wins and 12,894 Dire Team wins, which almost satisfy the balance of the prepared dataset.

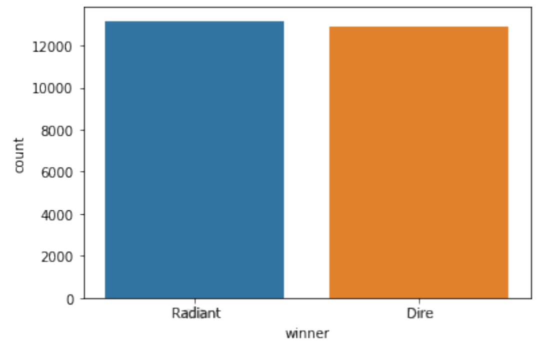

Fig. 1: The dataset class labels count

There are 137 heroes currently (by 2022) in DOTA2. The last two heroes added to the game were in October 2021, and hence we collected match data from August 2021 to keep the hero data balanced. The histogram of the heroes of this dataset has been shown in figure 2.

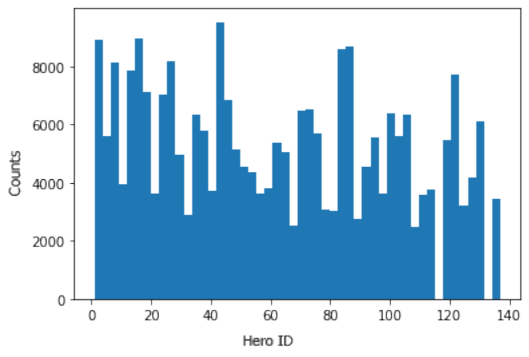

Fig. 2: The histogram of heroes

The histogram of the match durations in the dataset has been shown in figure 3

For analysis, this working dataset has 41 features, with 4 features for each player and 10 players in total in a match, in addition to the duration of the match. Each player has a chosen hero ID, kills, deaths, and assists statistical values as the feature. Labels of the outcome class has only two classes,

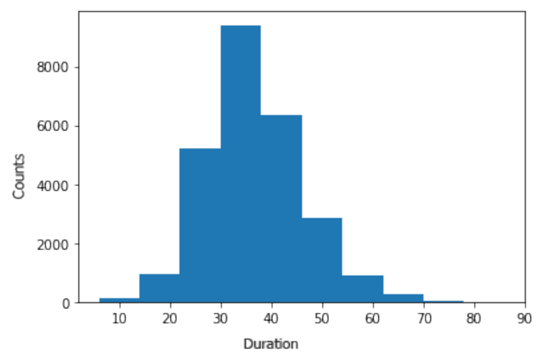

Fig. 3: The histogram of match duration

either Radiant or Dire team win and converted into numerical value as 0 and 1 accordingly. The dataset preparation script can be found at our Github repository https://github.com/dgvai/dota2-match-dataset-scrapper.

## IV. Methodology

### A. LSTM and Bi-LSTM

Recurrent Neural Networks or RNN [14] are often used in time-series applications with temporal dependencies. Basically, this network uses previous state data to process current data; hence, it has problems training long-term dependencies data. Long Short-Term Memory (LSTM [15]) is a variant of RNN that solves the short-term memory dependency limitation. It uses a hidden cell called memory cells, which are self-connected and store network temporal states.

There are three gates to control the network temporal states, input gate $i(t)$, output gate $o(t)$ and forget gate $f(t)$. Controlling the flow of memory cells to the remainder of the network is the function of the input gate and output gate. Additionally, forget gate passes the output information, including high weights, to the next neuron from the previous one. $C(t)$ is the current memory cell of the network. Figure 4 shows a single LSTM network.

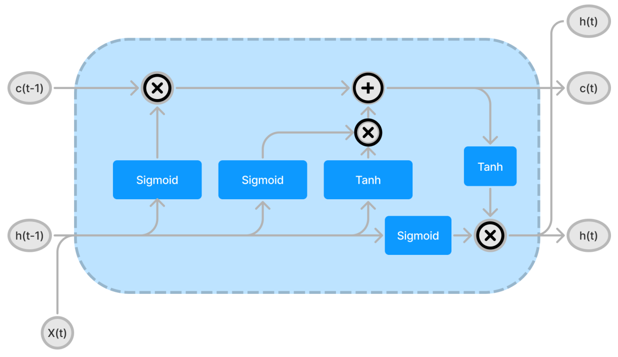

Fig. 4: An LSTM Cell

The expressions of the function gates as follows:

$$i(t) = sigmoid(W_{xi}X(t) + W_{hi}h(t-1) + b_i) \quad (1)$$

$$f(t) = sigmoid(W_{xf}X(t) + W_{hf}h(t-1) + b_f) \quad (2)$$

$$o(t) = sigmoid(W_{xo}X(t) + W_{ho}h(t-1) + b_o) \quad (3)$$

Where, $W(x)$ is the input weight matrix, $W_h$ at time $t-1$ defines the state weight matrix of the hidden layer. In addition, $b$ is the bias constant.

Bidirectional LSTM (Bi-LSTM) is able to work on previous states and can't use the future state, and thus it overcomes the limitation of the LSTM network [16]. Schuster and Paliwal [17] proposed the idea of Bidirectonal RNN by comprising two distinct LSTM hidden layers. $X = X_1, X_2, X_3, ...X_n$ as an input sequence in Bi-LSTM is propagated in forward direction as $\overrightarrow{h_t} = \overrightarrow{h_1}, \overrightarrow{h_2}, ...\overrightarrow{h_n}$ and backward direction as $\overleftarrow{h_t} = \overleftarrow{h_1}, \overleftarrow{h_2}, ...\overleftarrow{h_n}$. A Bi-LSTM network has been shown in figure 5 bellow.

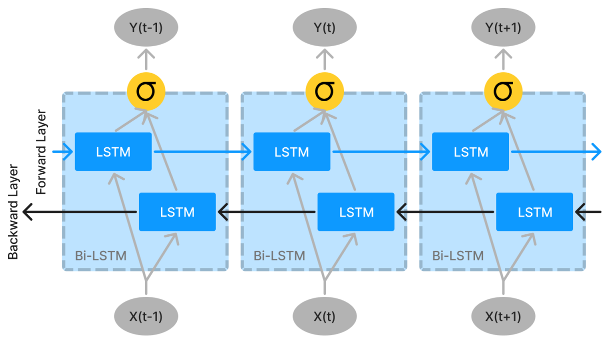

Fig. 5: A Bi-LSTM Network

### B. Stacked Bi-LSTM Network

In this work, we used stacked Bi-directional LSTM layers followed by an LSTM layer as the neural network of the model. Pascanu et al. [18] explored several ways of combining LSTM layers and discussed various difficulties among them. Similar approach was explored by Graves et al. [19] while investigating stacked LSTM for text generations.

There is a degradation problem when stacked LSTM layers are present in the model, which makes it hard to train the model. Wang et al. [20] described the problem's cause as the low convergence rate at training error. In this work, a residual connection between stacked LSTM layers has been made to reduce the low convergence problem. Intermediate ReLU activation layers were used as residual connections in between the stacked LSTM layers of our model. Since our strategy follows binary classification, a single neuron output was added to the model. Figure 6 shows the model architecture of our work.

### C. Model Training

The dataset we have prepared used to contain 43 columns. The match ID column is excluded at the very beginning of the process. From the rest of the 42 columns, the last column, "winner," was separated as the output labels. The input layer of

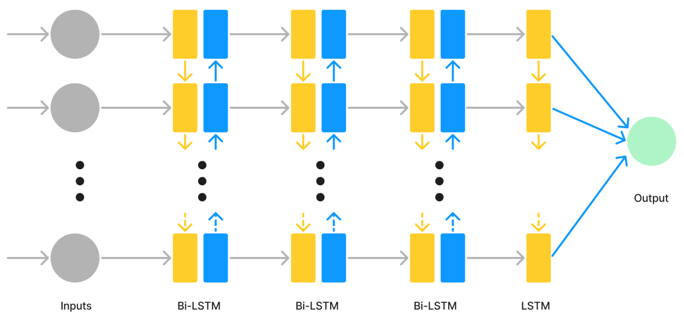

Fig. 6: Stacked Bi-LSTM Model Architecture

the network received the remaining 41 columns as the feature sequence.

The whole dataset were split by 0.2 training and testing samples ratio and the 20% of the training samples were used as validation samples while training. The model was implemented using the Tensorflow Keras framework with the Python programming language. The sequential model approach was used in this work. Adam was used as a gradient optimizer with a learning rate of $10^{-4}$. The loss criterion was calculated based on Binary crossentropy.

## V. Performance Analysis

To evaluate our model, the dataset was also trained with classic machine learning algorithms such as, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Random Forest (RF) classifiers. The optimum value of K was chosen based on square root of number of samples $N$. Using support vector, the optimum C parameter, $C = 1$ was chosen by trial and error. A similar approach was applied for the Random Forest classifier. With 1000 as the number of estimators in 5 maximum depths of the forest, the classifier was trained using the prepared data.

The Bi-LSTM network model was trained for 40 epochs as optimum parameter before the model gets over-fitted. The evaluated result on test set, resulted in approximately 92% accuracy with a loss of 0.2084. The comparison among the models has been presented in the following table I.

TABLE I: The comparative performance analysis of the models

| Model | Accuracy | AUC | F1 Score | Precision | Recall |
|---|---|---|---|---|---|
| KNN | 0.731 | 0.732 | 0.719 | 0.760 | 0.682 |
| RF | 0.911 | 0.911 | 0.911 | 0.912 | 0.911 |
| SVM | 0.915 | 0.915 | **0.915** | 0.920 | 0.911 |
| **Stacked Bi-LSTM** | **0.919** | **0.921** | 0.914 | **0.940** | **0.911** |

The training and validation accuracy and loss curve has been presented bellow:

However, SVM classifies almost similarly to the proposed Stacked Bi-LSTM network. As the LSTM network shows recurrency while learning, it is more suitable for real-time applications for classification time series data. In this experiment, the features of the data are variable with time. Hence, the stacked Bi-LSTM network has showed effective performance for real-time application. The table II shows a comparison of outcome prediction with previous research.

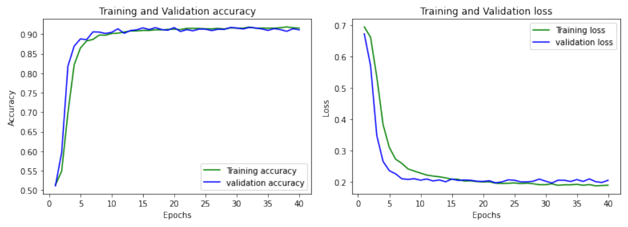

Fig. 7: Training & Validation Accuracy-Loss Curve

TABLE II: The comparison between previous researches on winner team prediction.

| Ref. | Data | Method | Accuracy |
|---|---|---|---|
| [7] | UCI, 92.3k | SVM+PSO | 60.1% |
| [21] | OpenAPI, 38.6k | SVM | 64.3% |
| [8] | OpenAPI, 5.7k | RF | 77.51% |
| **This work** | **OpenAPI, 26.1k** | **Stacked Bi-LSTM** | **91.9%** |

An application has been built using the saved model binary, using a Python web service and the Streamlit library. The main purpose of our application is to provide predictions based on given stats provided by the end-user. There are 10 select boxes to select 10 different heroes, and also three text boxes with each hero to input the kill-death stats. There's a slider bar to select the duration in minutes. The application is hosted on a streamlit server with the binary model, and the source codes are synchronized with the github repository. The application architecture is shown in figure 8.

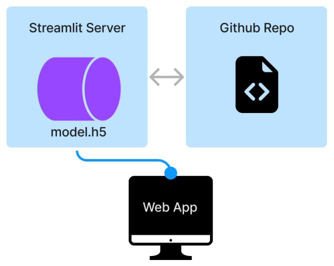

Fig. 8: Application architecture

The application is served at https://share.streamlit.io/dgvai/dota2-winner-predictor-web/main.py.

## VI. Conclusion

With the progress in the field of artificial intelligence, e-sports are now entering in the field of researchers. Games are now getting more intelligent than before with the involvement of AI. The tournaments are providing more statistical information, like match prediction, bracket prediction, win rate, etc., with the help of data science. DOTA2 has always been one of the most complex computer games of the world, and has been leading in the highest prize pool among the e-sports events. The enormous amount of data generated in each match is precious for the data science studies. In our work, a very small area of the data has been grabbed. It is observed that a Bidirectional LSTM network has better class prediction based on the sequence of matches' statistical data as feature other than classical machine learning classifiers. However, using more information from a match, or using other different data, a future scope of the work can be improvising the prediction strategy.

## References

[1] M. Schubert, A. Drachen, and T. Mahlmann, "Esports analytics through encounter detection," in *MIT Sloan Sports Analytics Conference*. MIT Sloan, 2016.

[2] F. Block, V. Hodge, S. Hobson, N. Sephton, S. Devlin, M. F. Ursu, A. Drachen, and P. I. Cowling, "Narrative bytes: Data-driven content production in esports," in *Proceedings of the 2018 ACM international conference on interactive experiences for TV and online video*, 2018, pp. 29–41.

[3] A. Katona, R. Spick, V. J. Hodge, S. Demediuk, F. Block, A. Drachen, and J. A. Walker, "Time to die: Death prediction in dota 2 using deep learning," in *2019 IEEE Conference on Games (CoG)*. IEEE, 2019, pp. 1–8.

[4] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[5] C. Eggert, M. Herrlich, J. Smeddinck, and R. Malaka, "Classification of player roles in the team-based multi-player game dota 2," in *International Conference on Entertainment Computing*. Springer, 2015, pp. 112–125.

[6] F. Block and A. Drachen, "Opinion: The case for data in esports," 2017.

[7] M. Anshori, F. Mar'i, M. W. Alauddin, and F. A. Bachtiar, "Prediction result of dota 2 games using improved svm classifier based on particle swarm optimization," in *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*. IEEE, 2018, pp. 121–126.

[8] V. Hodge, S. Devlin, N. Sephton, F. Block, P. Cowling, and A. Drachen, "Win prediction in multi-player esports: Live professional match prediction," *IEEE Transactions on Games*, 2019.

[9] L. Zhang, C. Xu, Y. Gao, Y. Han, X. Du, and Z. Tian, "Improved dota2 lineup recommendation model based on a bidirectional lstm," *Tsinghua Science and Technology*, vol. 25, no. 6, pp. 712–720, 2020.

[10] K. Akhmedov and A. H. Phan, "Machine learning models for dota 2 outcomes prediction," *arXiv preprint arXiv:2106.01782*, 2021.

[11] W. Looi, M. Dhaliwal, R. Alhajj, and J. Rokne, "Recommender system for items in dota 2," *IEEE Transactions on Games*, vol. 11, no. 4, pp. 396–404, 2018.

[12] B. Liu, W. Hsu, Y. Ma *et al.*, "Integrating classification and association rule mining." in *Kdd*, vol. 98, 1998, pp. 80–86.

[13] A. Dallmann, J. Kohlmann, D. Zoller, and A. Hotho, "Sequential item recommendation in the moba game dota 2," in *2021 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 2021, pp. 10–17.

[14] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for covid-19 with deep learning models of lstm, gru and bi-lstm," *Chaos, Solitons & Fractals*, vol. 140, p. 110212, 2020.

[17] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[18] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to construct deep recurrent neural networks," *arXiv preprint arXiv:1312.6026*, 2013.

[19] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 273–278.

[20] J. Wang, B. Peng, and X. Zhang, "Using a stacked residual lstm model for sentiment intensity prediction," *Neurocomputing*, vol. 322, pp. 93–101, 2018.

[21] 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.