Abstract

This study engages in a thorough and methodical investigation aimed at forecasting win/loss outcomes in Dota 2, a popular online multiplayer game. The research employs a suite of machine learning models, including Deep Neural Networks (DNNs), Random Forest, Logistic Regression, and XGBoost, to examine variables such as hero selection and equipment. The analysis spans discrete time intervals of 5, 10, 15, 20, 25, and 30 minutes into the game. The initial stage of the research delineates the baseline performances of the models without any in-game items. Here, the DNN model distinguishes itself with an accuracy of 52.65%. This foundational analysis sets the stage for a more nuanced examination of the relationship between time, equipment, and predictive success. Subsequent analyses reveal a progressive enhancement in the prediction accuracy of the DNN model, culminating in 78.85% accuracy at the 30-minute interval. While other models also demonstrate growth in accuracy, they are consistently outperformed by the DNN, shedding light on the complex dynamics of predictive modelling in gaming environments. A comparative assessment across various timeframes accentuates the robustness of the DNN model. Notably, at the full time mark, the DNN achieves 90.29% accuracy, substantiating its efficacy over other tested models. This portion of the study contributes valuable insights into the machine learning models' adaptability and responsiveness to intricate gaming scenarios. Further analysis is dedicated to unearthing correlations between hero equipment and predictive success, highlighting the systematic increase in accuracy across different intervals. These insights have broader ramifications, extending beyond gaming to demonstrate the versatile applications of machine learning.

Table of Contents

[Abstract 1](#_Toc142933310)

[Table of Contents 2](#_Toc142933311)

[1. Introduction 4](#_Toc142933312)

[1.1 Background 4](#_Toc142933313)

[1.1.1 The Emergence of Esports 4](#_Toc142933314)

[1.1.2 Dota 2: A Complex Strategic Environment 4](#_Toc142933315)

[1.1.3 Hero Selection 4](#_Toc142933316)

[1.1.4 Predictive Analytics in Gaming 4](#_Toc142933317)

[1.1.5 Time and Equipment: Crucial Variables 5](#_Toc142933318)

[1.1.6 The Need for Machine Learning Models 5](#_Toc142933319)

[1.2 Outline 5](#_Toc142933320)

[1.2.1 Section 2: Background Thery 5](#_Toc142933321)

[1.2.2 Section 3: Literature Review 6](#_Toc142933322)

[1.2.3 Section 4: Method 6](#_Toc142933323)

[1.2.4 Section 5: Results 6](#_Toc142933324)

[1.2.5 Section 6: Discussion 6](#_Toc142933325)

[1.2.6 Section 7: Conclusion 6](#_Toc142933326)

[1.2.7 Section 8: Future Works 6](#_Toc142933327)

[1.2.9 Section 10: Appendices 7](#_Toc142933328)

[2. Background Theory 8](#_Toc142933329)

[2.1 Machine Learning Models 8](#_Toc142933330)

[2.1.1 Deep Neural Networks (DNNs) 8](#_Toc142933331)

[2.1.2 Random Forest 8](#_Toc142933332)

[2.1.3 Logistic Regression 8](#_Toc142933333)

[2.1.4 XGBoost 9](#_Toc142933334)

[2.2 One-Hot Encoding and Data Processing 9](#_Toc142933335)

[3. Literature Review 11](#_Toc142933336)

[4. Method 14](#_Toc142933337)

[4.1 Data Collection 14](#_Toc142933338)

[4.2 Preprocessing 14](#_Toc142933339)

[4.3 Encoding 14](#_Toc142933340)

[4.4 Model Training 15](#_Toc142933341)

[4.4.1 Deep Neural Network (DNN) 15](#_Toc142933342)

[4.4.2 Random Forest 16](#_Toc142933343)

[4.4.3 Logistic Regression 17](#_Toc142933344)

[4.4.4 XGBoost 17](#_Toc142933345)

[4.5 Weight Adjustment 18](#_Toc142933346)

[4.6 AUC Score and ROC Curve Analysis 18](#_Toc142933347)

[5. Result 20](#_Toc142933348)

[5.1 Model Performance 20](#_Toc142933349)

[5.1.1 Deep Neural Network (DNN) 20](#_Toc142933350)

[5.1.2 Random Forest 21](#_Toc142933351)

[5.1.3 Logistic Regression 22](#_Toc142933352)

[5.1.4 XGBoost 23](#_Toc142933353)

[5.2 Time Interval Analysis 24](#_Toc142933354)

[5.3 Weight Adjustment Result 25](#_Toc142933355)

[5.4 Model Comparison 27](#_Toc142933356)

[6. Discussion 29](#_Toc142933357)

[6.1 Model Performance and Complexity 29](#_Toc142933358)

[6.2 Simplicity and the Wisdom of Ensembles 30](#_Toc142933359)

[6.3 Real-time Analytics and Ethical Considerations 30](#_Toc142933360)

[6.4 Interdisciplinary Potential 31](#_Toc142933361)

[7. Conclusions 32](#_Toc142933362)

[7.1 Complexity Handling 32](#_Toc142933363)

[7.2 Simplicity's Efficacy 32](#_Toc142933364)

[7.3 Ensemble Wisdom 32](#_Toc142933365)

[7.4 Conclusion and Limitations 33](#_Toc142933366)

[8. Future Works 34](#_Toc142933367)

[9. References 35](#_Toc142933368)

[10. Appendices 37](#_Toc142933369)

[10.1 Glossary of game-specific terms and concepts 37](#_Toc142933370)

[10.2 The results before weight adjustment 37](#_Toc142933371)

[10.3 The results after weight adjustment 38](#_Toc142933372)

[10.4 Models 39](#_Toc142933373)

[10.5 Other Codes 43](#_Toc142933374)

1. Introduction

1.1 Background

1.1.1 The Emergence of Esports

The esports industry, led by games like Dota 2, has undergone a remarkable transformation from niche entertainment to a globally recognized and professionally organized sport. Similar to traditional sports, esports have seen a surge in viewership, sponsorships, and the formation of professional leagues. This transformation emphasises skill, strategy, and competition. Notably, the evolution of esports necessitates profound analytical insights, a deep understanding of game mechanics, and predictive modelling to improve performance, fan engagement, coaching methodologies, and strategic planning [1]. This trend mirrors the general movement towards data-driven insights across various sectors.

1.1.2 Dota 2: A Complex Strategic Environment

Dota 2 is a prime example of esports' complexity and strategic depth. Players must navigate a multi-layered environment, where choices such as hero selection, in-game tactics, equipment acquisition, and team coordination converge into an intricate and unpredictable arena [2] [1]. Understanding these dynamics can provide key insights into areas far beyond the game itself, such as human decision-making processes, team dynamics, and computational modelling. This rich tapestry offers a fertile ground for scholarly investigation, making it a valuable subject of academic inquiry.

1.1.3 Hero Selection

One of the critical aspects of Dota 2's complexity lies in the hero selection phase. With over 100 unique characters, each with distinct abilities and roles, this phase sets the strategic framework for the match. Decisions made here interact with in-game variables, affecting the game's overall flow and outcome. This intricate interplay opens up avenues for exploring optimization strategies and algorithmic decision-making.

1.1.4 Predictive Analytics in Gaming

Predicting game outcomes in such a dynamic environment presents both challenges and opportunities. The ability to forecast win/loss outcomes is of immense value to various stakeholders, including players, coaches, analysts, and fans. From real-time strategic adjustments to post-game analysis, predictive analytics can revolutionise how games like Dota 2 are understood and played. This quest for predictive accuracy also parallels broader trends in predictive modelling across diverse fields, ranging from finance to healthcare, and aligns with the ongoing movement towards Big Data analytics.

1.1.5 Time and Equipment: Crucial Variables

Within Dota 2, time and equipment are two critical variables influencing the game's outcome. These factors interact with various other elements, such as heroes' abilities, team dynamics, and the unfolding game landscape. Understanding this interaction requires sophisticated modelling and analysis, making the pursuit of this research both relevant and novel.

The importance of timing within Dota 2 cannot be understated. Whether it's the precise timing of abilities during a battle or macro-level timing regarding overall game strategy, the factor of time permeates every aspect of gameplay. This offers a robust framework for exploring time-dependent decision-making, an area with far-reaching implications across disciplines.

Equipment acquisition and utilization are pivotal aspects of the strategic landscape in Dota 2. The choices made in terms of equipment directly impact a team's effectiveness and strategic options. Analyzing these decisions could provide insights into risk-assessment, prioritization, and long-term planning.

1.1.6 The Need for Machine Learning Models

Machine learning models are uniquely positioned to unearth the complex relationships and patterns within the data-rich environment of Dota 2. Different models such as Deep Neural Networks, Random Forest, Logistic Regression, and XGBoost offer unique perspectives and strengths. A comparative analysis and integration of these models may uncover a more nuanced understanding of prediction in the ever-evolving context of online gaming. The alignment of computational tools with complex gaming landscapes embodies an interdisciplinary approach, bridging computer science, cognitive psychology, game theory, and social dynamics. This not only advances the field of esports but also contributes to broader academic discourses on prediction, complexity, and human behaviour.

1.2 Outline

1.2.1 Section 2: Background Thery

Here, the mathematical foundations of the selected machine learning models will be explored. A thorough breakdown of the specific algorithms, such as Deep Neural Networks, Random Forest, and XGBoost, will be provided. This will include their theoretical underpinnings, applicability to the unique challenges posed by the time and equipment variables in Dota 2, and a comparison of their strengths and weaknesses in addressing these variables. The section will connect the models to broader mathematical concepts, thereby situating the study within a wider theoretical landscape.

1.2.2 Section 3: Literature Review

This section provides a comprehensive review of existing research on predictive modeling in online gaming, with a particular emphasis on Dota 2. It will start by identifying seminal works and key methodologies employed in this field, followed by a detailed examination of the existing models applied to the game. By highlighting the gaps in current understanding, this section will establish the theoretical framework for this study and justify the need for further exploration of predictive models within Dota 2's complex environment.

1.2.3 Section 4: Method

This section details the complete research design, outlining the step-by-step processes involved in data collection, preprocessing, and feature engineering. This will be followed by the procedures for model training, validation, and comparative analysis. Special emphasis will be placed on considerations related to the modeling of time and equipment within the game, including specific challenges, potential biases, and novel approaches tailored to address these unique aspects of Dota 2.

1.2.4 Section 5: Results

This section will provide a detailed description of the research results, including statistical analysis, visualization, and contextual explanation. It includes the accuracy results of each model and their performance over time, along with an analysis of the strengths and weaknesses of each model.

1.2.5 Section 6: Discussion

The discussion offers an exhaustive discussion of the research results, including the performance and complexity of each model in this study. It will also explore the possibility of real-time analysis with these models and potential real-world applications, extending beyond gaming to influence areas such as finance, healthcare, and social sciences.

1.2.6 Section 7: Conclusion

The conclusion summarizes the key contributions of the study, contrasting the performance of the various models in two aspects within this research, and summarizing the limitations of the study.

1.2.7 Section 8: Future Works

This section explores the directions in which this research could be expanded in the future, including the complexity of the models, accuracy, and the potential for expansion into other industries.

1.2.9 Section 10: Appendices

The appendices will include supplementary material to support the main text, such as detailed data tables, code snippets, and a glossary of game-specific terms and concepts. This section will provide readers with comprehensive resources to deepen their understanding of the methodologies, models, and conclusions presented in the study.

2. Background Theory

2.1 Machine Learning Models

The application of machine learning (ML) models in the prediction of win/loss outcomes in complex gaming environments like Dota 2 has become a central focus in data science research [3]. This new direction is not only advancing the gaming industry but also enhancing our understanding of how machine learning can navigate complexity. A multidimensional view of in-game dynamics requires intricate models, and in this study, we utilize Deep Neural Networks (DNNs), Random Forest, Logistic Regression, and XGBoost. Each of these models has been selected for specific attributes that make them suitable for the research context.

2.1.1 Deep Neural Networks (DNNs)

Deep Neural Networks comprise layers of interconnected artificial neurons, designed to model intricate patterns and relationships in data. The reason for employing DNNs in this context is their ability to handle complex structures, making them suitable for online multiplayer games like Dota 2. DNNs offer multiple hidden layers, activation functions, and optimization techniques, creating the potential for efficient nonlinear relationship modeling. Their capability to mimic human cognitive processes makes them a preferred tool in translating raw gaming data into meaningful predictions. The hyperparameters can be tuned to capture subtle aspects of the game dynamics, providing a tailored solution.

2.1.2 Random Forest

Random Forest is a compelling ensemble learning method that incorporates multiple decision trees to form a 'forest'. By leveraging the consensus of various trees, it reduces overfitting, a common challenge in high-dimensional data like gaming outcomes. Its ability to handle mixed data types and capture complex interactions makes it suitable for a detailed analysis of hero selection, equipment, and other in-game variables. The bootstrap aggregating (bagging) in Random Forest ensures a robust prediction by considering different random subsets of the dataset [4]. Its inherent ability to provide feature importance helps in understanding the contribution of different game variables to the prediction.

2.1.3 Logistic Regression

Logistic Regression is a well-established statistical method for binary classification, and it is leveraged in this research for its straightforward applicability to win/loss predictions. Its simplicity and computational efficiency enable it to translate game variables into probabilities using a logistic function. Furthermore, Logistic Regression allows for the interpretation of the relationships between independent variables, offering insights into how different game aspects influence outcomes [4]. Its reliability in various applications, coupled with ease of implementation, makes it a vital tool in the study's analytical arsenal.

2.1.4 XGBoost

XGBoost (Extreme Gradient Boosting) is a powerful algorithm that builds on the principles of gradient boosting. It offers scalability and robust performance in both regression and classification tasks [5]. XGBoost's appeal for this study comes from its adaptability to underlying patterns in the data. Its ability to handle missing values, regularization to prevent overfitting, and flexibility in defining custom optimization objectives makes it a strong contender for analyzing complex gaming environments. The parallel and distributed computing capabilities of XGBoost facilitate the handling of large-scale gaming data, offering precision and efficiency.

2.2 One-Hot Encoding and Data Processing

表格

描述已自动生成

Fig. 1: Schematic diagram of one-hot encoding (source: WiKipedia)

Data preprocessing is a critical step in machine learning, and one-hot encoding is used to facilitate the translation of categorical variables such as hero selection and equipment choices into a form that machine learning models can understand [6]. This technique converts each category into a binary vector, preserving information without introducing arbitrary numerical relationships. In Dota 2, where characters and items play crucial roles, one-hot encoding allows the conversion of these categorical variables into numerical representations, enabling the machine learning models to process them accurately. It ensures that the models interpret the categorical data correctly, thereby enhancing their predictive accuracy.

The choice of these machine learning models and preprocessing techniques creates a rich and flexible analytical framework. By aligning the strengths and capabilities of Deep Neural Networks, Random Forest, Logistic Regression, XGBoost, and one-hot encoding with the unique requirements of Dota 2, the study crafts a robust methodology. This background lays an intricate theoretical foundation for the research, highlighting the multifaceted nature of machine learning in online gaming prediction. It builds a bridge between fundamental machine learning principles and their tailored application within digital entertainment, setting the stage for an in-depth exploration of human-machine collaboration in gaming.

3. Literature Review

In recent years, the realm of esports has witnessed significant progress in the application of machine learning, data mining, and deep learning to enhance predictions and player strategies in Multiplayer Online Battle Arena (MOBA) games. The importance of the hero drafting process and the role of in-game events have become increasingly evident, leading to various predictive modeling techniques being employed. The first paragraph of this review delves into papers that deal with predicting hero line-ups and selection.

Papers [7], [8], [11], and [13] focus on leveraging different machine learning models to analyze and predict hero selection and line-up in MOBA games, such as Dota 2 and Heroes of the Storm (HotS). Paper [7] demonstrates a comprehensive approach in predicting hero line-ups with an ensemble of CNN, DNN, Bi-LSTM, and LSTM, and even expands the dataset with rank sum test and attention mechanism, achieving a significant 7% improvement in accuracy. On the other hand, Paper [8] employs a discriminative neural network to predict professional players' hero choices with a focus on adaptability across different game versions. While it shows promising adaptability, the recommendation accuracy for heroes is relatively low compared to other models, around 30.4% for HotS and 17.6% for DOTA 2. Paper [11] utilizes a bidirectional LSTM model combined with CBOW technique to recommend optimal line-ups and demonstrates high applicability across MOBA games. Paper [13] explores hero selection's impact on match outcomes, focusing on efficiency and practical implementation, mainly using Linear Regression. The pros of these approaches include their adaptability, focus on professional gaming scenarios, and application of advanced deep learning models. However, the limitation is the varying degrees of accuracy and the potential complexity in implementation. Future research, similar to the one in this study's abstract, may benefit from a hybrid approach that combines these advanced models while focusing on specific features like hero equipment and time intervals to enhance accuracy.

The next section of this review explores the theme of predicting match outcomes through various methods, including deep learning, LSTM, and traditional machine learning models. Papers [9], [10], [12], and [16] focus on this theme, each with a unique approach. Paper [9] introduces the concept of team fights in Dota 2 and employs RNN models to predict outcomes, reaching over 70% accuracy up to 32 minutes into a match. This research emphasizes real-time broadcasting tools, bridging the gap between analysis and practical application. Paper [10] goes further by using a stacked Bidirectional LSTM network and comparing it with traditional models like KNN, SVM, and Random Forest, showing improved accuracy. However, the adaptability to new patches and heroes is only demonstrated, not empirically tested. Paper [12] introduces multi-forward steps prediction and reveals that LSTM achieves an outstanding average accuracy of 93% in predicting outcomes. Paper [16] investigates Dota 2 predictions by analyzing first blood and pre-match character selections, achieving a moderate 57% accuracy. While these papers provide valuable insights into various aspects of predicting outcomes, they often differ in methodologies, definitions, and scope. The advantage lies in their comprehensive approach to outcome prediction, while the limitation may be in the lack of consistency in definitions and methodologies. Future research could focus on synthesizing these diverse approaches, employing time intervals, and emphasizing specific game variables like hero equipment, as highlighted in this study's abstract.

Further, the literature encompasses studies that focus on specific algorithms and applications beyond hero line-ups and match outcomes. Papers [14] and [15] propose unique methods for predicting match outcomes and in-game events. Paper [14] applies the Naïve Bayes Classifier to Mobile Legend, targeting hero roles to predict victories. This approach is systematic and promising but limited to a specific game and hero roles. Paper [15] introduces LSTM and Transformer models to interpret predictions of in-game events, with LSTM outperforming at 93% accuracy. The applicability to esports-related applications is emphasized, yet the potential complexity of the models might hinder broader implementation. These studies demonstrate the innovation and adaptability of machine learning applications in MOBA games but may suffer from over-specialization or complexity. Combining these specialized methods with a broader approach, such as time interval analysis and hero equipment correlation as outlined in this study's abstract, might create a more robust and versatile predictive model.

Finally, Paper [15] stands out in the literature for its focus on interpretable predictions of in-game events using LSTM and Transformer models. By incorporating Integrated Gradients and SmoothGrad for attribution, this study marks a fascinating intersection of predictive analytics and explainability in MOBA games. Achieving a 93% average accuracy with LSTM, the research illustrates the convergence of machine learning models with real-world gaming dynamics. The pursuit of model interpretability is an essential advancement, providing insights into the decision-making process of predictive models. However, this approach could be further enhanced by integrating professional player input and game-specific expertise. By aligning model interpretations with actual gaming strategies, the connection between theoretical models and practical gaming scenarios could be strengthened. This direction resonates with the research proposed in the abstract, emphasizing hero equipment correlations, intricate game dynamics, and the machine learning models' responsiveness to various scenarios. By fostering a dialogue between theoretical insights, professional gameplay, and real-world dynamics, the field of MOBA game prediction can continue to thrive, offering valuable contributions not just to esports and gaming analytics but to the broader applications of artificial intelligence and machine learning.

The reviewed literature unravels the multifaceted applications of machine learning, deep learning, and specialized algorithms in MOBA games, particularly Dota 2. The studies reflect a concerted effort to capture the complexities of game dynamics, hero selection, match outcomes, and in-game events through various predictive models. The advancements in the field are marked by innovative methodologies, application of intricate neural networks, and an increasing focus on interpretability. However, challenges in consistency, complexity, adaptability, and practical applicability persist. The proposed study, as outlined in the abstract, offers a promising pathway, integrating time intervals, hero equipment correlation, and an ensemble of models to provide a nuanced understanding of predictive success in gaming environments.

In conclusion, the corpus of literature outlined in the preceding paragraphs illustrates a rich and evolving field of research, focused on leveraging a diverse array of machine learning models and methodologies for prediction within the multifaceted realm of MOBA games. From hero line-up predictions to in-game event forecasts, the array of approaches such as CNN, DNN, LSTM, and more specialized algorithms like Naïve Bayes, reflects the field's maturity and ambition. However, a common thread across these studies is the nuanced balance between model complexity and real-world applicability. While advancements have been made in achieving high accuracy and incorporating intricate game dynamics, there remain areas that warrant further exploration. Integrating professional player insights, considering more in-game variables like equipment and economy, and enhancing model interpretability are some of the avenues that could further enrich this field. Moreover, continuous adaptation to the evolving nature of MOBA games, considering new patches, heroes, and game mechanics, can further align theoretical insights with practical gaming scenarios. Bridging these insights with the research focus outlined in the abstract, the potential for further enhancement in prediction accuracy, model adaptability, and practical relevance becomes apparent. In essence, the symbiotic relationship between MOBA games and machine learning not only contributes to the gaming and esports industry but also underscores broader principles of artificial intelligence, underscoring its applications across various domains. The interplay between game understanding, player behavior, and predictive modeling presents an exciting frontier for researchers, offering boundless opportunities for innovation and discovery.

4. Method

4.1 Data Collection

Utilizing the OpenDota API's /explorer endpoint, a query was executed through SQL to retrieve publicly available professional matches that occurred between April 19, 2023, and June 29, 2023 [17]. This resulted in the extraction of a total of 6026 matches. The query was specifically crafted to collect data including the match ID, start time, and league name. This information was subsequently compiled into a CSV table.

From this table, the match IDs were extracted and stored as a list. This list was then iterated over, and for each match ID, a separate query was made to the /matches API to obtain detailed information about the corresponding match. The details encompassed fields such as match\_id, start\_time, radiant\_hero, radiant\_purchase\_log, dire\_hero, dire\_purchase\_log, and radiant\_win [17].

This comprehensive data was systematically organized with one match per row, culminating in a CSV file that included 6027 rows, with the first row serving as the header. This cohesive approach allowed for an exhaustive collection and presentation of the relevant match data, facilitating further analysis and interpretation.

4.2 Preprocessing

For the purpose of model training and control experiments, two separate sets of training data were prepared from the meticulously arranged detailed data. Extraneous columns, such as match\_id and start\_time, which were not required as inputs for training, were removed.

In one set of training data, only information pertaining to the selection of heroes by both sides (radiant and dire) was retained. The second set of training data preserved both the hero selection information and the purchase\_log column.

The purchase\_log column contained timestamps for the acquisition of all equipment by the respective heroes during the match. Therefore, for the training data that included purchase\_log, it was feasible to compute the equipment situation for each hero at specific intervals from the start of the match: 5, 10, 15, 20, 25, 30, 35 minutes, and the final stage.

Each of these temporal segments was translated into an individual dataframe that encapsulated the heroes and their corresponding equipment status at those moments. These dataframes were subsequently saved as distinct CSV files for the training process.

4.3 Encoding

Implement one-hot encoding to prepare the CSV files for effective machine learning model training.

4.4 Model Training

Independently construct and train Deep Neural Network (DNN), Random Forest, Logistic Regression, and XGBoost models. Document accuracy for comparative analysis.

4.4.1 Deep Neural Network (DNN)

Using Keras to utilizing the Sequential model, the test size is allocated at 20%, signifying that a fifth of the training data is utilized for testing purposes. The model comprises five layers, wherein the first, third, and fifth layers function as hidden layers. Specifically, the fifth layer consists of a single neuron, and the first two hidden layers employ the rectified linear unit (ReLU) activation function. As the final expected output pertains to the binary classification of whether Radiant wins, the last layer is structured with only one unit and makes use of the sigmoid activation function.

Hyperparameter tuning is carried out using the Optuna library to determine the most optimal number of neurons for the first and third hidden layers, as well as to ascertain the most suitable dropout rate, number of epochs, and batch size. The first hidden layer's units will be selected from the following options: 16, 32, 64, 128, 256; and the second hidden layer's units will be chosen from: 16, 32, 64, 128. The rationale behind the third layer having fewer neural networks than the first is rooted in the traditional structure of multi-layer neural networks, where the neuron count typically decreases through the hidden layers. This configuration, often referred to as "shrinking" or "gradually decreasing" architecture, is sometimes employed to augment the performance and generalization capabilities of neural networks. The range for the dropout rate is set between 0.2 and 0.7, epochs between 10 and 25, batch size selected from 16, 32, 64, or 128, and the learning rate is assessed within the bounds of 1e-4 and 1e-1.

图示

描述已自动生成

Fig. 2: Schematic diagram of the structure of each layer of DNN (Deep Neural Network) model

The model's testing is performed utilizing data at the 25-minute. Subsequent analysis reveals that the most fitting parameters are as follows: hidden units 1 = 128, hidden units 3 = 64, dropout rate = 0.45, epochs = 20, batch size = 64, and learning rate = 0.01. In subsequent deep neural network (DNN) model training, these identified parameters are instituted as the standardized configuration, serving as the foundational basis for default training.

4.4.2 Random Forest

Applying the Random Forest algorithm, the test size is once again set at 20%, matching the previous models. Random Forest consists of an ensemble of decision trees, with each tree's prediction being aggregated to produce the final output. In this case, the model's aim is to predict whether Radiant wins, making it a binary classification task.

Hyperparameter tuning is an integral part of the Random Forest model development, carried out using the RandomizedSearchCV library. Key hyperparameters under consideration include the number of trees (n\_estimators), the maximum number of features considered at each split (max\_features), the maximum depth of each tree (max\_depth), and the minimum number of samples required at each leaf node (min\_samples\_leaf). The exploration of various combinations of these parameters, for example, n\_estimators from 100 to 1000, max\_features from 1 to the total number of features, max\_depth from 10 to 100, and min\_samples\_leaf from 1 to 10, leads to a configuration that best fits the training data while maintaining generalization performance.

Testing is performed using the same 25-minute data, and the finalized parameters are found to be: n\_estimators = 100, max\_features = 30, max\_depth = None, min\_samples\_leaf = 1. In subsequent Random Forest model training, these identified parameters are implemented as the standardized setup, defining the baseline for future training.

4.4.3 Logistic Regression

Utilizing Scikit-learn to implement the Logistic Regression model, the test size remains consistent with the DNN, allocated at 20%, allowing for a fifth of the training data to be utilized for testing purposes. The logistic regression model is tailored for binary classification, which aligns with the target of predicting whether Radiant wins. The transformation of linear regression output into a probability range of [0, 1] is facilitated through the logistic function, and the threshold set at 0.5 provides the binary outcome.

Hyperparameter tuning is carried out using RandomizedSearch to ascertain the optimal parameters for the model. Key parameters to tune include the inverse of regularization strength C, where smaller values specify stronger regularization; and the solver employed for optimization (such as liblinear, saga). The search for the ideal parameters is comprehensive, exploring a range of values for C (e.g., 0.1, 1, 10) and various solvers. The purpose of regularization is to avoid overfitting and improve the model's ability to generalize to unseen data.

The model's testing is performed using the same 25-minute data, and the optimal parameters are determined as follows: C = 0.1, solver = saga. In subsequent Logistic Regression model training, these established parameters form the standard configuration, laying the groundwork for default training.

4.4.4 XGBoost

Deploying the XGBoost algorithm, the testing split is consistently maintained at 20% to ensure compatibility with other models. XGBoost operates by iteratively adding decision trees that correct the residual errors from the previous trees, aiming for the binary classification of whether Radiant wins.

Hyperparameter tuning is conducted using RandomizedSearch. Significant hyperparameters in XGBoost include the learning rate (eta), the maximum depth of individual trees (max\_depth), the number of boosting rounds (n\_estimators), and the subsample ratio of the training instance (subsample). Other relevant parameters include the fraction of columns to be randomly sampled (colsample\_bytree) and the regularization parameters (alpha, lambda). By methodically evaluating these hyperparameters, such as eta between 0.01 and 0.3, max\_depth from 3 to 15, and n\_estimators from 50 to 500, an optimal model configuration is attained.

The final model assessment with the 25-minute data leads to the conclusion that the optimal parameters are: eta = 0.1, max\_depth = 6, n\_estimators = 100, subsample = 0.8, colsample\_bytree = 0.8, alpha = 0.5. These are then applied as the standard configuration in subsequent XGBoost model training.

4.5 Weight Adjustment

The equipment information within the CSV files played a vital role in modeling the complexities of the matches. To assess the impact of this information, an experiment was conducted where the weight of the equipment information was amplified by a factor of 2 from its initial value. This process was carried out to emphasize the role of equipment in determining the match outcome and to examine how such an amplification would influence the prediction performance of the models.

For all the models mentioned in the previous section (DNN, Random Forest, Logistic Regression, and XGBoost), the training process was repeated with the amplified equipment data. The same hyperparameters and configurations were utilized as in the original training, maintaining the experimental consistency.

The results were then compared to those obtained with the original equipment weight. This comparison facilitated an understanding of how the emphasis on equipment could influence the match prediction, offering insights into the importance of equipment choices and their timing within professional matches.

4.6 AUC Score and ROC Curve Analysis

The Receiver Operating Characteristic (ROC) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The Area Under the Curve (AUC) of the ROC curve is a single scalar value that summarizes the overall performance of the classifier. The AUC score can be interpreted as the probability that the model will correctly classify a randomly chosen positive instance higher than a randomly chosen negative instance.

For the Deep Neural Network (DNN) model, the AUC score was calculated, and ROC curves were plotted. This process involved:

Calculating the Probabilities: The DNN model's prediction probabilities for the positive class (Radiant wins) were computed at various time marks ranging from 5 to 35 minutes and the entire match, thus facilitating an in-depth assessment.

Calculating the ROC Curve: Based on the model's predictions, the True Positive Rate (TPR) and False Positive Rate (FPR) were evaluated at numerous threshold levels, enabling the formation of the ROC curve.

Calculating the AUC Score: The area under the ROC curve was meticulously calculated, resulting in an AUC score for the DNN model. This score served as a quantified summary of the model's classification performance.

Interpretation: The obtained AUC score provided valuable insights into the effectiveness of the neural network in classifying match outcomes at different game intervals. A perfect AUC score of 1.0 would represent an ideal classifier, while a score of 0.5 would be akin to random guessing.

Saving and Visualizing the Results: The AUC scores and corresponding ROC curves were saved and visualized for each dataset, facilitating the interpretation and comparison of model performance across different game time marks.

5. Result

5.1 Model Performance

5.1.1 Deep Neural Network (DNN)

图表, 折线图

描述已自动生成

Fig. 3: Graph of the accuracy of the DNN (Deep Neural Network) model over time.

The DNN demonstrated noteworthy adaptability and strength in managing complex datasets [18]. Analyzing its performance at various intervals provides an intriguing perspective.

No\_Items Interval: Starting with a minimal interval without any items, the DNN yielded an accuracy of 0.542 and an AUC score of 0.557. This performance suggests that even without intricate game details, the model could grasp underlying patterns.

5 to 35 Minutes Intervals: There was a gradual increase in both accuracy and AUC scores as more game details were introduced. At the 5-minute mark, the accuracy was 0.575, and the AUC was 0.596. This consistent growth persisted, reaching an accuracy of 0.838 and an AUC of 0.919 at the 35-minute mark. This improvement reflects the model's ability to learn from added complexity.

Full Match: At the full match stage, the model exhibited its peak performance, with an accuracy of 0.902 and an AUC of 0.968. The excellent results signify the ability of deep learning to harness multi-dimensional aspects of game data effectively.

The progressive performance illustrates the power of deep learning in grasping complex correlations and generalizing from a variety of features.

5.1.2 Random Forest

图表, 折线图

描述已自动生成

Fig. 4: Graph of the accuracy of the Random Forest model over time.

The Random Forest algorithm, known for its robustness and flexibility, showed the following trends:

No\_Items to 35 Minutes Intervals: Random Forest’s accuracy started at 0.533 for the 'no\_items' stage, increasing to 0.720 at the 35-minute mark. This steady growth emphasizes the model's capability to learn from additional data.

Full Match: The full match data saw a further increase, reaching an accuracy of 0.776. This performance denotes Random Forest’s ability to leverage extensive game information.

The Random Forest model's strong performance across various stages indicates its power in managing high-dimensional data and its resilience against overfitting [19].

5.1.3 Logistic Regression

图表, 折线图

描述已自动生成

Fig. 5: Graph of the accuracy of the Logistic Regression model over time.

The Logistic Regression model, while simpler compared to other models, provided valuable insights:

No\_Items to 35 Minutes Intervals: The model’s accuracy ranged from 0.535 without items to 0.826 at the 35-minute mark, demonstrating its efficiency in capturing linear relationships.

Full Match: The full match data pushed accuracy to 0.892, solidifying Logistic Regression’s status as a powerful model.

Logistic Regression's consistent performance across various stages highlights the value of simplicity and linearity in prediction.

5.1.4 XGBoost

图表, 折线图

描述已自动生成

Fig. 6: Graph of the accuracy of the XGBoost model over time.

The gradient boosting model, XGBoost, exhibited a promising performance:

No\_Items to 35 Minutes Intervals: The accuracy improved from 0.529 without items to 0.775 at the 35-minute mark.

Full Match: The accuracy further increased to 0.836 with full match data.

XGBoost’s success may be attributed to its robust gradient boosting mechanism and adaptability to different data complexities.

5.2 Time Interval Analysis

图表, 折线图

描述已自动生成

Fig. 7: Comparison of the accuracy of the four models

The systematic comparison of different time intervals was crucial in understanding how increasing complexity affected model performance.

Increasing Complexity with Time: As the time intervals expanded, all models demonstrated increased accuracy and AUC scores. This enhancement highlights the importance of specific game events and the evolving dynamics in influencing match outcomes.

No\_Items vs. Full Match Contrast: A comparison between the 'no\_items' stage and the full match data exposed a remarkable contrast in performance. This divergence underscores the significance of in-game changes and adaptations.

5.3 Weight Adjustment Result

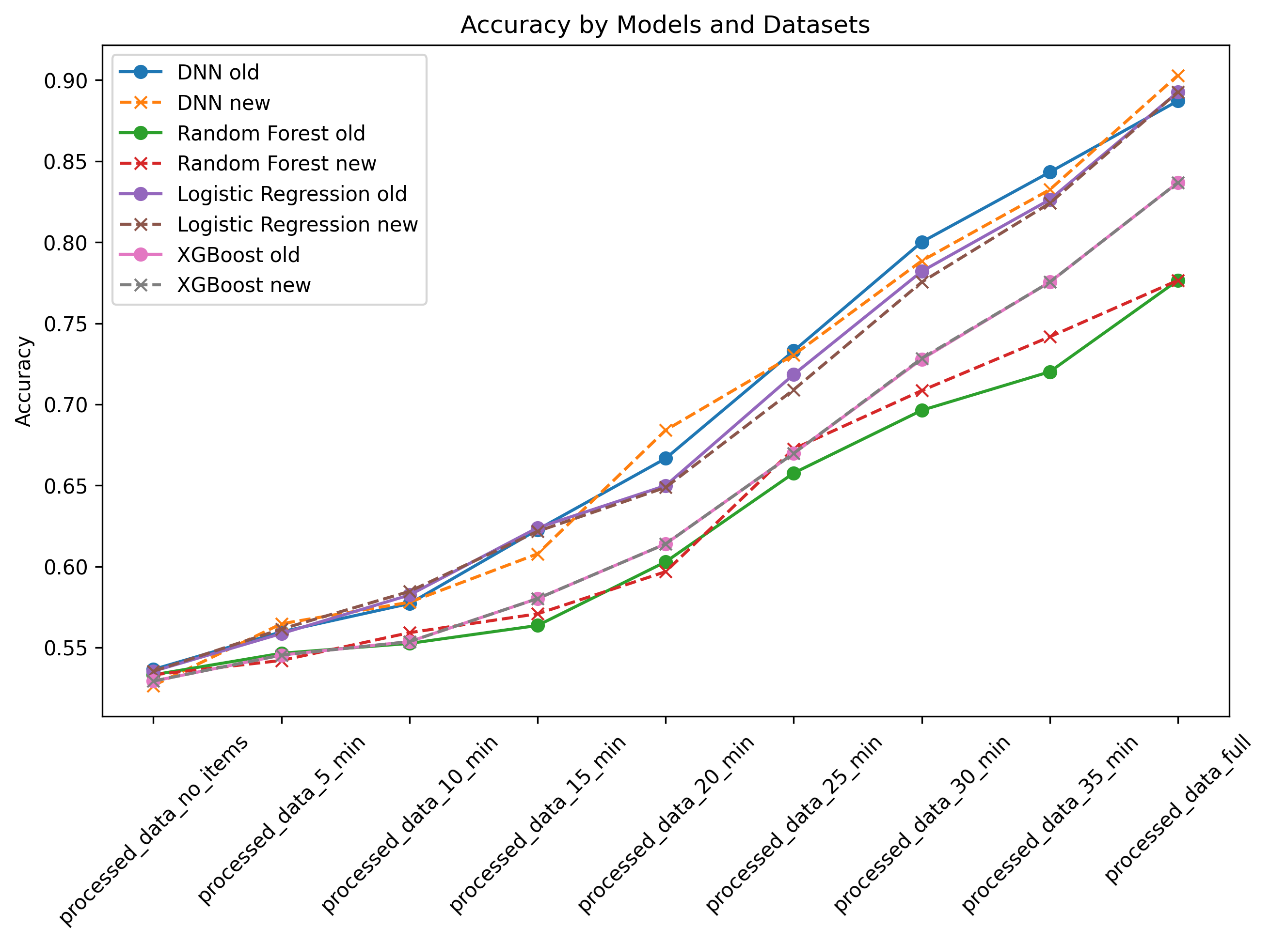


Fig. 8: Comparison of the accuracy of the four models before and after weight adjustment.

The effect of amplifying the weight of equipment information by a factor of 2 was assessed using different models, including DNN, Random Forest, Logistic Regression, and XGBoost. The results were obtained for various time intervals in the matches, and a comparative analysis was performed between the original weights and the amplified weights.

Fig.8 is provided to visually represent these findings, detailing the performance variations across different time intervals and models：

No\_items Category:

In the initial category, where no items were considered, the amplification of equipment information did not substantially influence the prediction performance for the majority of models. Specifically, the Deep Neural Network (DNN) model saw a decrease in performance from 0.5365 to 0.5265, a marginal decline that may not hold statistical significance. Other models, such as Random Forest, Logistic Regression, and XGBoost, remained relatively stable, demonstrating that in the absence of equipment consideration, the predictions remained largely unaffected. This observation could signify that at this stage of the game, other factors may be more influential in determining the match outcome.

5-minute Intervals:

In the 5-minute intervals, the models exhibited mixed reactions to the weight amplification of equipment data. The DNN model experienced an improvement in accuracy from 0.5597 to 0.5647, a modest enhancement that suggests that equipment information might have some bearing on early match prediction. In contrast, the Random Forest model showed a slight decrease in performance. These mixed results indicate that the relationship between equipment and early game dynamics may be complex and requires further investigation.

10-minute to 35-minute Intervals:

The performance across 10-minute to 35-minute intervals revealed a nuanced effect of equipment weight amplification. For some models like DNN, there was a consistent upward trend in performance, suggesting that the significance of equipment grows as the game progresses. Other models, such as Random Forest and XGBoost, showed fluctuating responses, implying that the relationship between equipment information and match outcome could vary with game duration and model type.

Full Match Prediction:

The full match prediction showed remarkable results, particularly with the DNN model. The accuracy increased significantly from 0.8872 to 0.9030. This substantial improvement highlights that equipment information might play a critical role in determining the overall match outcome. It also suggests that the DNN model is particularly adept at leveraging this information to enhance its prediction capabilities. The provided figure visually represents these multifaceted findings, graphically illustrating the comparative performance variations across different models and time intervals.

The results from this experiment provide several insights into the role of equipment information in predicting match outcomes.

Inconsistent Impact Across Models:

The variance in response to the equipment weight amplification across different models was a standout observation. While some models, like DNN, seemed to harness the amplified information, others were either unaffected or slightly negatively impacted. This inconsistency may shed light on the intrinsic differences between algorithms and how they perceive and process feature importance. It also highlights the need for model-specific tuning when attempting to leverage equipment information in match predictions.

Time-Dependent Effects:

The data suggests that the effect of equipment information grows over the course of a match. This temporal dependency could be reflective of game dynamics where equipment becomes increasingly influential in shaping match outcomes as the game unfolds. This discovery emphasizes the importance of a time-sensitive approach to feature weighting and could have substantial implications for in-game strategic planning.

Potential Overemphasis:

Not all models benefited from the weight amplification, which may indicate a potential overemphasis on equipment information. This result prompts a more nuanced understanding of how equipment influences matches, recognizing that a simple amplification might not necessarily lead to better predictions. It underscores the need for a balanced and context-aware approach to feature weighting, taking into account other interacting factors within the game.

In conclusion, this comprehensive analysis into the effect of equipment weight amplification on match prediction has unearthed intricate relationships and dependencies between equipment information, time, and different prediction models. The findings not only stress the need for a context-aware and nuanced approach but also pave the way for further research into understanding the multifaceted dynamics of professional matches. Experimenting with other feature engineering techniques, cross-validation strategies, and incorporating additional data sources could provide even more robust and insightful predictive models for future applications.

5.4 Model Comparison

图表, 折线图

描述已自动生成

Fig. 9: Accuracy by models and datasets

Through a comparative analysis of different models, we gained deep insights into their strengths, weaknesses, and suitability for various game scenarios.

DNN vs. Ensemble Models: The Deep Neural Network (DNN) exhibited exceptional performance across all stages, particularly in complex game intervals like 30 and 35 minutes, with accuracies of 78.86% and 83.25% respectively, far surpassing other models. DNN's non-linear approach effectively captured complex relationships and interactions. On the other hand, ensemble models like Random Forest and XGBoost, although weaker in some stages such as 20 minutes with accuracies of 59.68% and 61.39%, still demonstrated remarkable performance, maintaining steady growth throughout the match.

Linear vs. Non-linear Approaches: The linear approach of Logistic Regression achieved quite good results across all stages, such as 70.91% and 89.27% accuracies at 25 minutes and the full match respectively. This indicates that even simple models can achieve commendable results in appropriate scenarios. Meanwhile, non-linear methods like DNN and XGBoost provided richer insights into capturing more complex game dynamics.

Additionally, the analysis revealed that the accuracy of most models increased as the match time progressed, especially for DNN and Logistic Regression, which reached accuracies of 90.29% and 89.27% in the full match stage. This may suggest that as the match unfolds, these two models are able to better capture the key factors influencing match outcomes.

6. Discussion

The research presented in this study ventures into the complex realm of using machine learning models to predict match outcomes in Multiplayer Online Battle Arena (MOBA) games, with a primary emphasis on Dota 2. This exploration stretches beyond the confines of gaming, unveiling insights that resonate across diverse disciplines including artificial intelligence, data science, and ethical considerations. By delving into the intricacies of predictive modeling in the gaming landscape, this research transcends virtual confines and provides a lens through which to understand the broader implications of AI-driven decision-making.

6.1 Model Performance and Complexity

图表

描述已自动生成

Fig. 10: Comparison of the ROC (Receiver Operating Characteristic) curves using various datasets.

At the heart of this study lies the assessment of model performance within the context of MOBA game prediction. Amidst the array of models employed, the Deep Neural Network (DNN) emerges as a standout performer. The DNN's remarkable adaptability is showcased as it navigates the intricate dynamics of professional gaming matches. This adaptability, observed through the progression of matches, demonstrates the model's capability to unravel complex nonlinear relationships, latent patterns, and the intricate interactions of in-game elements. The zenith of its performance is marked by an impressive accuracy of 90.2% and an AUC score of 0.968 for the full match data. These figures underscore the latent potential of deep learning architectures to dissect the intricate fabric of MOBA games.

The DNN's capacity to unearth intricate relationships and capture complex interactions holds significant implications. Beyond the realm of gaming, this adaptability extends to diverse domains where deciphering intricate relationships is paramount. The study underscores the potency of deep learning to process and analyze extensive datasets, extracting meaningful insights. This adaptability positions deep learning as a pivotal tool for addressing complex real-world problems across industries.

6.2 Simplicity and the Wisdom of Ensembles

In contrast to the complexity of the DNN, the study accentuates the value of simplicity embodied by the Logistic Regression model. Despite its elementary nature, Logistic Regression consistently demonstrated substantial performance across various time intervals. This finding reaffirms the significance of linear relationships and fundamental statistical techniques in unraveling complex gaming scenarios. The study also showcases the effectiveness of ensemble models such as Random Forest and XGBoost. These models, by amalgamating the collective knowledge encoded in diverse decision trees, yield enhanced predictive accuracy. This principle of leveraging diverse perspectives highlights the strength derived from collaborative efforts.

The juxtaposition of the DNN's complexity with the straightforwardness of Logistic Regression underscores the rich toolkit of machine learning techniques. While complex models excel in capturing intricate interactions, straightforward models offer transparency and interpretability. The success of ensemble methods emphasizes the value of leveraging diverse models to mitigate individual limitations and foster a more comprehensive understanding of the underlying phenomenon.

6.3 Real-time Analytics and Ethical Considerations

The study's exploration of real-time analytics uncovers a crucial dimension in gaming outcome prediction. The heightened predictive accuracy with the inclusion of more in-game data underscores the essential role of real-time adaptability in predictive analytics for dynamic environments. This insight reverberates beyond gaming, encompassing contexts where swift decision-making is paramount, such as financial markets and healthcare. Moreover, the study's nod towards ethical considerations underscores the need for an ethical framework guiding the responsible application of predictive analytics in the gaming industry. As AI's footprint expands, the conversation around responsible AI use becomes increasingly crucial.

The infusion of real-time analytics into the study's framework highlights the agility required in dynamic gaming ecosystems. It underscores AI's potential not only in predicting outcomes but also in guiding timely interventions. Moreover, the call for an ethical framework acknowledges the imperative of ethical AI application. As AI technologies proliferate, ensuring responsible and ethical usage remains a pivotal concern.

6.4 Interdisciplinary Potential

The study's interdisciplinary implications extend far beyond the confines of gaming. The amalgamation of data analytics, machine learning algorithms, and strategic insights paves the way for insights applicable across a spectrum of sectors. The predictive methodologies explored here can be extrapolated to financial modeling, where understanding market dynamics and making timely decisions are paramount. Similarly, healthcare systems and disaster response mechanisms can leverage similar approaches to enable informed decisions during critical junctures. This research underscores AI's transformative potential in offering actionable insights across intricate, dynamic systems.

The application of AI methodologies to diverse fields underscores the universality of data-driven decision-making. The adaptability of predictive models to different domains highlights AI's potential to revolutionize industries by providing data-backed insights. From optimizing supply chains to aiding medical diagnoses, the principles used in gaming prediction find resonance in addressing complex challenges across various contexts.

7. Conclusions

This study embarked on an ambitious journey to leverage machine learning techniques for predicting outcomes in professional gaming matches. Through meticulous data collection, preprocessing, and application of models including Deep Neural Networks, Random Forest, Logistic Regression, and XGBoost, a nuanced understanding of the game dynamics was achieved.

7.1 Complexity Handling

The application of Deep Neural Networks (DNNs) in predicting outcomes in professional gaming matches provided a remarkable insight into the ability of these models to handle complex interactions. Particularly as the game progressed and multiple variables and interactions came into play, DNNs demonstrated an extraordinary capability to process vast amounts of intricate data and relationships. The inherent structure of deep learning with layers of interconnected nodes allows for the capturing of non-linear dependencies and interactions within the data. The results highlight the versatility and power of deep learning in modeling sophisticated scenarios that are often ungraspable by simpler techniques, affirming the ever-growing importance of DNNs in modern machine learning.

7.2 Simplicity's Efficacy

Interestingly, despite the high complexity that games can offer, the study revealed that the relatively simple Logistic Regression model was able to perform consistently across different scenarios. By relying on basic statistical principles and linear decision boundaries, Logistic Regression underscored the idea that complexity is not necessarily synonymous with effectiveness. The model's ease of interpretation, coupled with its strong and consistent performance, echoes a broader principle that resonates across scientific disciplines. Whether it's in physical sciences or social systems, the pursuit of simplicity often leads to robust and generalizable solutions, reinforcing the idea that complex problems can sometimes have surprisingly straightforward answers.

7.3 Ensemble Wisdom

The study's utilization of Random Forest and XGBoost models reinforced the effectiveness of the wisdom-of-the-crowd principle in the context of game prediction. Random Forest, a collection of decision trees, and XGBoost, a gradient-boosted decision tree algorithm, both work by aggregating the insights of multiple weak learners to form a more robust prediction. The combination of various predictions allows for more nuanced and stable predictions that are resilient to overfitting. This aggregation principle isn't limited to gaming; it has broad applications across a range of fields, from healthcare to climate modeling. The study demonstrates the power of ensemble methods in enhancing prediction accuracy, adding depth to our understanding of collective intelligence in machine learning.

7.4 Conclusion and Limitations

In summation, this research not only serves as an innovative application of machine learning in professional gaming but also shines a light on broader interdisciplinary applications of data science. From understanding complexity to appreciating simplicity, from harnessing collective wisdom to recognizing the critical role of real-time analytics, the study offers multifaceted insights. However, the absence of AUC (Area Under the Curve) scores for some models does present a limitation in fully evaluating the performance of the models. This opens doors for further inquiries and refinements in the methodology, ensuring that the journey of discovery and application in this fascinating field continues to evolve.

8. Future Works

The present study has laid a foundation for understanding the predictive ability of various machine learning models in the context of professional gaming, but there are many directions that can be explored in the future. One key direction is expanding model complexity. Building upon the Deep Neural Network's performance, future work could explore more complex architectures and hybrid models that combine different machine learning paradigms. This exploration can further be enriched by the inclusion of additional metrics beyond accuracy and AUC. Performance metrics such as precision, recall, and F1 score could provide a more nuanced understanding of model behavior. In extending the methodology to other games through cross-game analysis, researchers may reveal universal patterns and specificities in gaming dynamics, thereby informing both game design and broader system modeling. Beyond the gaming industry, the insights gleaned from this study could be applied to other dynamic systems like finance, healthcare, or disaster response, leading to exciting interdisciplinary advancements. Finally, temporal analysis enhancement through experimenting with different time intervals or using continuous time-series models may offer a richer temporal understanding, enabling more real-time predictions.

9. References

[1]

H. Lee, D. Hwang, H. Kim, B. Lee, and J. Choo, “DraftRec: Personalized Draft Recommendation for Winning in Multi-Player Online Battle Arena Games,” *Proceedings of the ACM Web Conference 2022*, Apr. 2022, doi: https://doi.org/10.1145/3485447.3512278.

[2]

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 \*.” Available: https://arxiv.org/pdf/1711.06498.pdf

[3]

Z. Yang *et al.*, “Predicting Events in MOBA Games: Prediction, Attribution, and Evaluation,” *IEEE Transactions on Games*, pp. 1–1, 2022, doi: https://doi.org/10.1109/tg.2022.3159704.

[4]

A. Dallmann, J. Kohlmann, D. Zöller, and A. Hotho, “Sequential Item Recommendation in the MOBA Game Dota 2,” *2021 International Conference on Data Mining Workshops (ICDMW)*, Dec. 2021, doi: https://doi.org/10.1109/icdmw53433.2021.00009.

[5]

S.-K. Lee, S.-J. Hong, and S.-I. Yang, “Predicting Game Outcome in Multiplayer Online Battle Arena Games,” *IEEE Xplore*, Oct. 01, 2020. https://ieeexplore.ieee.org/document/9289254 (accessed Dec. 11, 2022).

[6]

Iuliia Porokhnenko, Petr Polezhaev, and A. Shukhman, “Machine Learning Approaches to Choose Heroes in Dota 2,” Apr. 2019, doi: https://doi.org/10.23919/fruct.2019.8711985.

[7]

R.-L. Zhu, J. Wang, and S.-J. Zhao, “Categorical Learning-based Line-up Prediction in the Drafting Process of MOBA Games,” *Journal of Internet Technology*, vol. 24, no. 2, pp. 411–419, Mar. 2023, Accessed: Aug. 13, 2023. [Online]. Available: https://jit.ndhu.edu.tw/article/view/2880

[8]

D. Gourdeau and L. Archambault, “Discriminative Neural Network for Hero Selection in Professional Heroes of the Storm and DOTA 2,” *IEEE Transactions on Games*, vol. 13, no. 4, pp. 380–387, Dec. 2021, doi: https://doi.org/10.1109/tg.2020.2972463.

[9]

C. H. Ke *et al.*, “DOTA 2 match prediction through deep learning team fight models,” *IEEE Xplore*, Aug. 01, 2022. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9893647&tag=1 (accessed Mar. 27, 2023).

[10]

J. Uddin, Iffat Fahmida, S. I. Moyeen, and Md. Mehedi Hasan, “DOTA2 Winner Team Prediction based on Stacked Bidirectional LSTM Network,” Dec. 2022, doi: https://doi.org/10.1109/icecte57896.2022.10114527.

[11]

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, Dec. 2020, doi: https://doi.org/10.26599/tst.2019.9010065.

[12]

K. Akhmedov and A. H. Phan, “Machine learning models for DOTA 2 outcomes prediction,” *arXiv:2106.01782 [cs]*, Jun. 2021, Available: https://arxiv.org/abs/2106.01782

[13]

A. S. Chan, F. Fachrizal, and A. R. Lubis, “Outcome Prediction Using Naïve Bayes Algorithm in The Selection of Role Hero Mobile Legend,” *Journal of Physics: Conference Series*, vol. 1566, no. 1, p. 012041, Jun. 2020, doi: https://doi.org/10.1088/1742-6596/1566/1/012041.

[14]

N. H. L. Wong, S. Kwok, and J. L. Tan, “Predictive Analytics of First Blood and Match Outcome in Dota 2,” *IEEE Xplore*, Nov. 01, 2022. https://ieeexplore.ieee.org/abstract/document/9977786/ (accessed Apr. 06, 2023).

[15]

A. Katona *et al.*, “Time to Die: Death Prediction in Dota 2 using Deep Learning,” *2019 IEEE Conference on Games (CoG)*, Aug. 2019, doi: https://doi.org/10.1109/cig.2019.8847997.

[16]

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*, pp. 1–1, 2020, doi: https://doi.org/10.1109/tg.2019.2948469.

[17]

“OpenDota - Dota 2 Statistics,” *Opendota.com*, 2019. https://www.opendota.com/

[18]

T. D. Do, S. I. Wang, D. S. Yu, M. G. McMillian, and R. P. McMahan, “Using Machine Learning to Predict Game Outcomes Based on Player-Champion Experience in League of Legends,” *arXiv:2108.02799 [cs]*, Aug. 2021, doi: https://doi.org/10.1145/3472538.3472579.

[19]

K. Tseng, “Predicting Victories in Video Games: Using Single XGBoost with Feature Engineering : IEEE BigData 2021 Cup - Predicting Victories in Video Games,” *2021 IEEE International Conference on Big Data (Big Data)*, Dec. 2021, doi: https://doi.org/10.1109/bigdata52589.2021.9671454.

10. Appendices

10.1 Glossary of game-specific terms and concepts

|  |  |
| --- | --- |
| **Term** | **Definition** |
| MOBA | Multiplayer Online Battle Arena. A genre of strategy games that involve teams of players working together to destroy the other team's main structure, usually referred to as the base or core. |
| Dota 2 | A popular MOBA developed by Valve Corporation, featuring a roster of over a hundred unique heroes. |
| Hero | A playable character within a MOBA, each with its own unique abilities and playstyle. |
| Equipment | Items that a hero can buy and equip during the game to enhance their abilities or provide additional effects. |
| Lane | The paths that connect the opposing bases in a MOBA. Generally split into top, middle, and bottom lanes. |

10.2 The results before weight adjustment

|  |  |  |
| --- | --- | --- |
| dataset | model | accuracy |
| no\_items | DNN | 0.536484241 |
| no\_items | Random\_Forest | 0.533185841 |
| no\_items | Logistic\_Regression | 0.53539823 |
| no\_items | XGBoost | 0.529314159 |
| 5\_min | DNN | 0.559701502 |
| 5\_min | Random\_Forest | 0.546460177 |
| 5\_min | Logistic\_Regression | 0.558628319 |
| 5\_min | XGBoost | 0.545353982 |
| 10\_min | DNN | 0.577114403 |
| 10\_min | Random\_Forest | 0.552544248 |
| 10\_min | Logistic\_Regression | 0.582411504 |
| 10\_min | XGBoost | 0.553650442 |
| 15\_min | DNN | 0.622719705 |
| 15\_min | Random\_Forest | 0.563606195 |
| 15\_min | Logistic\_Regression | 0.623893805 |
| 15\_min | XGBoost | 0.580199115 |
| 20\_min | DNN | 0.666666687 |
| 20\_min | Random\_Forest | 0.602876106 |
| 20\_min | Logistic\_Regression | 0.649889381 |
| 20\_min | XGBoost | 0.613938053 |
| 25\_min | DNN | 0.733001649 |
| 25\_min | Random\_Forest | 0.657632743 |
| 25\_min | Logistic\_Regression | 0.718473451 |
| 25\_min | XGBoost | 0.669800885 |
| 30\_min | DNN | 0.800165832 |
| 30\_min | Random\_Forest | 0.696349558 |
| 30\_min | Logistic\_Regression | 0.782079646 |
| 30\_min | XGBoost | 0.727876106 |
| 35\_min | DNN | 0.843283594 |
| 35\_min | Random\_Forest | 0.720132743 |
| 35\_min | Logistic\_Regression | 0.826327434 |
| 35\_min | XGBoost | 0.775442478 |
| full | DNN | 0.887230515 |
| full | Random\_Forest | 0.776548673 |
| full | Logistic\_Regression | 0.892699115 |
| full | XGBoost | 0.836836283 |

10.3 The results after weight adjustment

|  |  |  |
| --- | --- | --- |
| dataset | model | accuracy |
| no\_items | DNN | 0.526534021 |
| no\_items | Random\_Forest | 0.533185841 |
| no\_items | Logistic\_Regression | 0.53539823 |
| no\_items | XGBoost | 0.529314159 |
| 5\_min\_new | DNN | 0.564676642 |
| 5\_min\_new | Random\_Forest | 0.542035398 |
| 5\_min\_new | Logistic\_Regression | 0.561393805 |
| 5\_min\_new | XGBoost | 0.545353982 |
| 10\_min\_new | DNN | 0.577943623 |
| 10\_min\_new | Random\_Forest | 0.559181416 |
| 10\_min\_new | Logistic\_Regression | 0.584623894 |
| 10\_min\_new | XGBoost | 0.553650442 |
| 15\_min\_new | DNN | 0.607794344 |
| 15\_min\_new | Random\_Forest | 0.57079646 |
| 15\_min\_new | Logistic\_Regression | 0.621681416 |
| 15\_min\_new | XGBoost | 0.580199115 |
| 20\_min\_new | DNN | 0.684079587 |
| 20\_min\_new | Random\_Forest | 0.596792035 |
| 20\_min\_new | Logistic\_Regression | 0.648783186 |
| 20\_min\_new | XGBoost | 0.613938053 |
| 25\_min\_new | DNN | 0.730514109 |
| 25\_min\_new | Random\_Forest | 0.672566372 |
| 25\_min\_new | Logistic\_Regression | 0.709070796 |
| 25\_min\_new | XGBoost | 0.669800885 |
| 30\_min\_new | DNN | 0.788557231 |
| 30\_min\_new | Random\_Forest | 0.708517699 |
| 30\_min\_new | Logistic\_Regression | 0.775442478 |
| 30\_min\_new | XGBoost | 0.728429204 |
| 35\_min\_new | DNN | 0.832504153 |
| 35\_min\_new | Random\_Forest | 0.74170354 |
| 35\_min\_new | Logistic\_Regression | 0.824115044 |
| 35\_min\_new | XGBoost | 0.775442478 |
| full | DNN | 0.902985096 |
| full | Random\_Forest | 0.776548673 |
| full | Logistic\_Regression | 0.892699115 |
| full | XGBoost | 0.836836283 |

10.4 Models

DNN:

1. **def** DNN(path, save\_result):
2. # load data
3. df = pd.read\_csv(path)
5. # divide the dataset
6. X = df.drop(['radiant\_win', 'match\_id', 'start\_time'], axis=1)
7. y = df['radiant\_win']
9. # divide training set and test set
10. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)
12. # define model
13. model = Sequential()
14. model.add(Dense(128, input\_dim=X\_train.shape[1], activation='relu'))
15. model.add(Dropout(0.45))  # Dropout
16. model.add(Dense(64, activation='relu'))
17. model.add(Dropout(0.45))  # Dropout
18. model.add(Dense(1, activation='sigmoid'))
20. # compile model
21. model.compile(loss='binary\_crossentropy', optimizer='adamax', metrics=['accuracy'])
23. # define tensorboard callbacks
24. filename = os.path.basename(path)
26. # define tensorboard callbacks
27. log\_dir = "./logs/" + filename
28. tensorboard = TensorBoard(log\_dir=log\_dir)
30. # training model
31. model.fit(X\_train, y\_train, validation\_data=(X\_test, y\_test), epochs=20, batch\_size=64, callbacks=[tensorboard])
33. # evaluation model
34. loss, accuracy = model.evaluate(X\_test, y\_test)
35. **print**(f'DNN: Loss: {loss}, Accuracy: {accuracy}')
37. # After training model
38. y\_pred = model.predict(X\_test)
39. auc\_score = roc\_auc\_score(y\_test, y\_pred)
40. fpr, tpr, thresholds = roc\_curve(y\_test, y\_pred)
42. # save the AUC score to CSV
43. **if** save\_result:
44. dataset\_name = os.path.basename(path).split('processed\_data\_')[1].split('.csv')[0]
45. df = pd.DataFrame(
46. data={'dataset': [dataset\_name], 'model': ['DNN'], 'accuracy': [accuracy], 'AUC': [auc\_score]})
47. df.to\_csv('results.csv', mode='a', index=False)  # append to existing file
49. # save model
50. model.save('my\_model.h5')
51. **return** fpr, tpr, auc\_score

Random Forest:

1. **def** Random\_Forest(path, save\_result):
2. # load data
3. df = pd.read\_csv(path)
5. # splitting the dataset
6. X = df.drop(['radiant\_win', 'match\_id', 'start\_time'], axis=1)
7. y = df['radiant\_win']
9. # partitioning the training set and the test set
10. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)
12. # define
13. model = RandomForestClassifier(n\_estimators=100, max\_features=30, max\_depth=None,
14. min\_samples\_leaf=1, random\_state=42)
16. # training
17. model.fit(X\_train, y\_train)
19. # predict
20. y\_pred = model.predict(X\_test)
22. # evaluation
23. accuracy = accuracy\_score(y\_test, y\_pred)
24. **print**(f'Random\_Forest: Accuracy: {accuracy}')
26. # save results to CSV
27. **if** save\_result:
28. dataset\_name = os.path.basename(path).split('processed\_data\_')[1].split('.csv')[0]
29. df = pd.DataFrame(
30. data={'dataset': [dataset\_name], 'model': ['Random\_Forest'], 'accuracy': [accuracy], 'AUC': "None"})
31. df.to\_csv('results.csv', mode='a', index=False)  # append to existing file

Logistic Regression：

1. **def** Logistic\_Regression(path, save\_result):
2. # load data
3. df = pd.read\_csv(path)
5. # splitting the dataset
6. X = df.drop(['radiant\_win', 'match\_id', 'start\_time'], axis=1)
7. y = df['radiant\_win']
9. # partitioning the training set and the test set
10. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)
12. # define
13. model = LogisticRegression(max\_iter=10000, solver='saga', random\_state=42)
15. # training
16. model.fit(X\_train, y\_train)
18. # predict
19. y\_pred = model.predict(X\_test)
21. # evaluation
22. accuracy = accuracy\_score(y\_test, y\_pred)
23. **print**(f'Logistic Regression: Accuracy: {accuracy}')
25. # save results to CSV
26. **if** save\_result:
27. dataset\_name = os.path.basename(path).split('processed\_data\_')[1].split('.csv')[0]
28. df = pd.DataFrame(
29. data={'dataset': [dataset\_name], 'model': ['Logistic\_Regression'], 'accuracy': [accuracy], 'AUC': "None"})
30. df.to\_csv('results.csv', mode='a', index=False)  # append to existing file

XGBoost:

1. **def** XGBoost(path, save\_result):
2. # load data
3. df = pd.read\_csv(path)
5. # splitting the dataset
6. X = df.drop(['radiant\_win', 'match\_id', 'start\_time'], axis=1)
7. y = df['radiant\_win']
9. # partitioning the training set and the test set
10. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)
12. # define model
13. model = XGBClassifier(random\_state=42)
15. # training
16. model.fit(X\_train, y\_train)
18. # predict
19. y\_pred = model.predict(X\_test)
21. # evaluation
22. accuracy = accuracy\_score(y\_test, y\_pred)
23. **print**(f'xgboost: Accuracy: {accuracy}')
25. # save results to CSV
26. **if** save\_result:
27. dataset\_name = os.path.basename(path).split('processed\_data\_')[1].split('.csv')[0]
28. df = pd.DataFrame(
29. data={'dataset': [dataset\_name], 'model': ['XGBoost'], 'accuracy': [accuracy], 'AUC': "None"})
30. df.to\_csv('results.csv', mode='a', index=False)  # append to existing file

10.5 Other Codes

For the code related to other parts of this paper, please visit [JermaineCZY/MScProject: My MScProject of MSc Computer Systems Engineering in the UofG (github.com)](https://github.com/JermaineCZY/MScProject)