Predicting Dota 2 Match Outcomes Through Chat Log Analysis: A Machine Learning Approach

Tshepiso Bapela
School of Computer Science
University of the Witwatersrand, Johannesburg
Email: 2346377@students.wits.ac.za

Branden Ingram and Pravesh Ranchod
Supervisors
School of Computer Science
University of the Witwatersrand, Johannesburg

Abstract—**This paper investigates the potential of Dota 2 in-game chat logs for match outcome prediction using supervised learning approaches. We analyze a dataset of 50,000 ranked matches, evaluating the predictive power of communication data alone and in combination with game events. Using LSTM and DistilBERT architectures, alongside a logistic regression baseline, we demonstrate that chat logs alone enable accurate prediction (up to 81.4% accuracy), while incorporating game events substantially improves performance (up to 98.4% accuracy). Our temporal analysis reveals that prediction reliability increases significantly during the mid-game phase (15-30 minutes), with models exhibiting different strengths - LSTM achieves higher accuracy while DistilBERT demonstrates greater prediction confidence. This study contributes to esports analytics by establishing chat logs as a viable predictive data source.**

I. Introduction

Esports popularity [1] has driven increased demand for sophisticated analytics tools that can empower audiences, teams and coaches with data-driven insights [2] [3]. In Dota 2[1], a highly competitive multiplayer game, match outcome prediction models are particularly valuable for deriving strategic insights [4]. However, current approaches that rely on in-game statistics face a significant challenge: frequent game updates [5] can quickly render these models obsolete as game mechanics and meta-strategies evolve.

We propose a novel approach focusing on in-game chat logs, which offer a more robust prediction basis as communication patterns remain relatively stable across game updates. These logs provide unique insights into team dynamics, strategic decision-making, and coordination patterns that may indicate match outcomes. Additionally, they capture real-time reactions to game events and team sentiments, offering a window into the psychological aspects of play.

Existing research has explored various avenues for match prediction, including hero draft analysis [6], player statistics [7], and live game state prediction [8]. While some studies have incorporated chat data, they primarily focus on toxicity analysis [9] or treat match prediction as a secondary objective [10]. Our work specifically investigates chat logs as the primary predictor of match outcomes, complemented by objective game events to enhance prediction accuracy.

This paper demonstrates that chat logs alone can effectively predict match outcomes, achieving accuracy comparable to traditional in-game statistics-based approaches while being more resilient to game updates. We further show how combining chat analysis with key game events can significantly enhance prediction accuracy[2].

II. Background

A. Dota 2

Dota 2 is a highly competitive esports title characterized by a blend of complex strategy, coordinated teamwork, and individual player skill [12]. Matches unfold on a geographically balanced map divided into two distinct territories (radiant[3] side and dire[4] side), each serving as the home base for a team of five heroes. The ultimate objective for each team is the complete destruction of the opposing team's Ancient[5], a central structure representing their final line of defense. This heavily fortified building is safeguarded by a series of defensive structures called turrets. Players assume the role of powerful heroes with unique skill sets, engaging in team fights, securing critical map objectives such as Roshan[6] and runes[7] that provide power-ups.

One of the unique features of Dota 2 is the in-game chat system, which allows players to communicate with their teammates and opponents during a match. This chat system serves as a viable source of data for match prediction, as it provides insights into player and team sentiments. Furthermore, Dota 2 records complimentary in-game event logs that detail key events like champion kills, turret destructions, and Roshan kills. This data can be leveraged to enrich the chat logs.

B. Models

1) Word2Vec and Continuous Bag of Words (CBOW): Word2Vec is a group of models used for generating word embeddings, which are dense vector representations of words in a continuous vector space. The CBOW architecture, one of the main models within the Word2Vec framework, predicts

[1] https://www.dota2.com/home

[2] Code available at: https://github.com/Crossofglory/Dota-2-Match-Outcomes

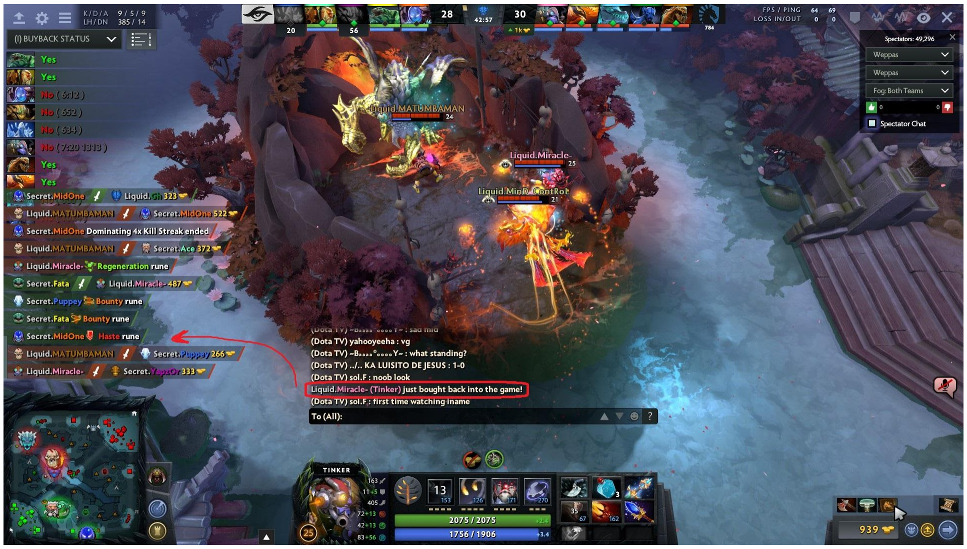
[3] https://dota2.fandom.com/wiki/Radiant

[4] https://dota2.fandom.com/wiki/Dire

[5] https://dota2.fandom.com/wiki/Ancient_(Building)

[6] https://liquipedia.net/dota2/Roshan

[7] https://liquipedia.net/dota2/Runes



Yes
Yes
No (6:12)
No (652)
No (634)
No (7:20 1313)
Yes
Yes

Secret.MidOne Liquid.Gh 323
Liquid.MATUMBAMAN Secret.MidOne 522
Secret.MidOne Dominating 4x Kill Streak ended
Liquid.MATUMBAMAN Secret.Ace 372
Liquid.Miracle- Regeneration rune
Secret.Fata Liquid.Miracle- 487
Secret.Puppey Bounty rune
Secret.Fata Bounty rune
Secret.MidOne Haste rune
Liquid.MATUMBAMAN Secret.Puppey 266
Liquid.Miracle- Secret.YapzOr 333

(Dota TV) ~Baaa®ooooY~ : sad mid
(Dota TV) yahooyeeha : vg
(Dota TV) ~Baaa®ooooY~ : what standing?
(Dota TV) ../.. KA LUISITO DE JESUS : 1-0
(Dota TV) sol.F : noob look
Liquid.Miracle- (Tinker) just bought back into the game!
(Dota TV) sol.F : first time watching iname

Fig. 1. Dota 2 match screenshot showing the Dire team taking down Roshan, a powerful neutral monster that grants significant rewards upon defeat. The in-game chat log (center screen) displays various messages from players. Additionally, the objective log on the left can be seen [11].

a target word based on its surrounding context words. By training on large text corpora, CBOW learns to represent words as dense vectors, with semantically similar words mapped to geometrically close vectors in the embedding space.

The input layer consists of one-hot encoded vectors representing the context words $\{x_1, x_2, ..., x_V\}$, where V is the vocabulary size. These inputs are projected into an N-dimensional hidden layer through a weight matrix $W(V \times N) = \{w_{ki}\}$, where w_{ki} is the embedding of the corresponding input word.

Given an input context word $x_k = 1$ and all other $x_{k'} = 0$ for $k' \neq k$, the hidden layer activations h are calculated as:

$$h = W^T x = W^T_{(k,:)} = v^T_{w_I}$$

For the output, a new weight matrix $W' = \{w'_{ij}\}$ of size $N \times V$ is used to compute a score u_j for each word:

$$u_j = {v'_{w_j}}^T h$$

These scores are then passed through a softmax function to obtain the posterior probability distribution over words:

$$p(w_j|w_I) = y_j = \frac{exp(u_j)}{\sum_{j'} exp(u_{j'})}$$

While effective for learning word embeddings that capture semantic similarities, CBOW's bag-of-words approach may struggle for long sequential text analysis like chat logs, where word order is crucial for understanding context and meaning.

2) LSTMs: LSTMs, proposed by Hochreiter & Schmidhuber [14], are a particular type of RNNs designed to fascilitate gradient flow. As shown in Figure 2, an LSTM has a gating mechanism and an internal cell state (c_t) that acts like a conveyor belt, transmitting relevant information through the entire sequence. At each time step t, the forget gate (f_t) determines what information from the previous cell state (c_{t-1}) should be retained or forgotten, using: $f_t = \sigma(W_f[a_{t-1}, x_t] + b_f)$. The input gate ($i_t$) and the tanh layer decide what new information from the current input (x_t) and previous hidden state (a_{t-1}) are added, using the equations: $i_t = \sigma(W_i[a_{t-1}, x_t] + b_i)$ and $c_{\tilde{t}} = \tanh(W_c[a_{t-1}, x_t] + b_c)$. The new state ($c_t$) is given by: $c_t = f_t \odot c_{t-1} + i_t \odot c_{\tilde{t}}$. Lastly, gate ($o_t$) controls what parts of the current cell state (c_t) should flow into the output hidden state (a_t), via the equations: $o_t = \sigma(W_o[a_{t-1}, x_t] + b_o)$ and $a_t = o_t \odot \tanh(c_t)$.

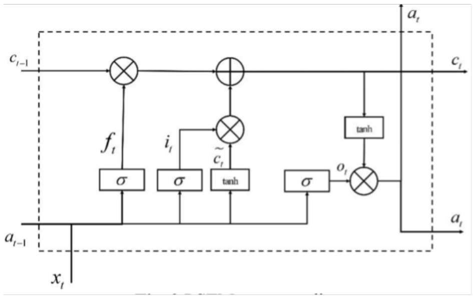


Fig. 2. Schematic diagram of the Long Short-Term Memory (LSTM) architecture, which consists of input, output, and forget gates that regulate the flow of information into and out of the cell state. With the cell state serving as the memory of the LSTM cell[13].

This gating mechanism enables LSTMs to selectively retain crucial details over long sequences, making them well-suited for analyzing contextual data like conversations in chat logs, where understanding the order and context of messages is essential.

3) Transformer-based models: The Transformer model [15] introduced a novel architecture that relies entirely on attention mechanisms to draw global dependencies between the input

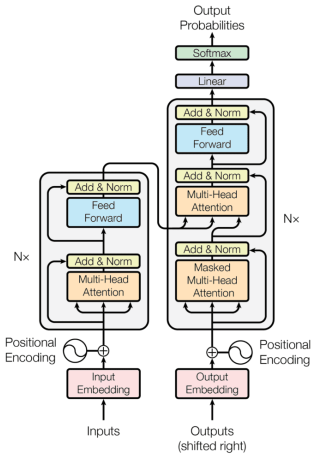


Fig. 3. The Transformer architecture consists of an encoder and decoder, each composed of multiple layers of multi-head attention and feed-forward neural networks. Positional encodings are added to the input and output embeddings to incorporate position information [15].

and output sequences. The core attention function is given by:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Where Q, V, and K are the queries, values, and keys derived from the input sequence. As seen in Figure 3, the model's major components are an encoder and a decoder, each containing a multi-head self-attention mechanism followed by a position-wise connected feed-forward network. The multi-head attention mechanism enables the model to jointly attend to information from different representational subspaces.

BERT (Bidirectional Encoder Representations from Transformers) [16] is a pre-trained language model that makes use of the transformer architecture and can be fine-tuned on specific tasks by adding a task-specific output layer and fine-tuning the model on labeled data.

For our analysis, DistilBERT [17], a distilled version of BERT, will be used. These models can effectively capture the context and meaning of chat messages, even when the relevant information is spread across different parts of the sequence, making them a promising choice for match outcome prediction based on chat logs.

III. Related Work

Previous research on predicting match outcomes in Dota 2 has employed various machine learning techniques and in-game data sources. Uddin et al. [18] used a Bidirectional LSTM neural network with match statistics to predict winning teams, achieving an accuracy of 91.9% and an F1 score of 0.914. Sándor and Wan [19] developed a hybrid deep learning and NLP model for match outcome prediction during the hero drafting phase, combining a CBOW model for hero embeddings with an improved LSTM architecture, achieving a prediction accuracy of 73%. Zhang et al. [20] proposed an improved Bi-LSTM model for recommending the 5th hero in a Dota 2 line-up, incorporating CBOW-generated hero embeddings and achieving an average line-up accuracy rate of 52%. Another paper [21] examined the impact of toxic behavior on match outcomes, finding that teams displaying toxic behavior had reduced chances of winning. Lastly, Bello et al. [22] proposed a framework combining the BERT language model with deep learning classifiers for sentiment analysis of short texts, achieving state-of-the-art performance with accuracy scores greater than 92% and F1-scores higher than 0.94.

IV. Research Design

Our research employs a systematic approach to investigate the predictive power of Dota 2 in-game communications. The methodology, illustrated in Figure 4, consists of three main phases. First, we process a dataset of 50,000 ranked matches, where chat logs undergo cleaning and standardization to remove noise and ensure consistency. Messages are chronologically ordered and attributed to teams, while non-English content and irrelevant information are filtered out.

In the second phase, these processed logs are transformed into tokens or embeddings. We then evaluate three distinct architectures: logistic regression as a baseline, LSTM network, and DistilBERT.

To assess the potential benefits of additional game state information, we concatenate our initial chat-based analysis with objective event data, including tower destructions, hero kills, and other key game events. This dual-dataset approach allows us to perform a systematic comparison of prediction capabilities with and without game state context, while maintaining consistent evaluation metrics and models configurations.

V. Dataset and Preprocessing

A. Data Preprocessing Pipeline

The dataset spans 50,000 Dota 2 ranked ladder matches initially obtained through the OpenDota API. Each match record contains several key attributes including match identifiers, player accounts, team-specific chat logs, and match outcomes (represented by the 'radiant_win' boolean flag). To address our research questions, we developed two distinct datasets: a chat-only dataset to evaluate the predictive power of communication alone, and a combined dataset incorporating both chat and objective events. Both datasets underwent preprocessing to ensure data quality and consistency.

1) Chat Log Processing: The chat log preprocessing pipeline, fundamental to both datasets, includes:

- **Team Attribution**: Messages were tagged with team affiliations (Radiant/Dire) based on player slots, with slots 0-4 corresponding to Radiant and 5-9 to Dire.
- **Message Standardization**: Chat entries were reformatted to a consistent structure: "Team: message content".
- **Data Cleaning**: Null entries and messages containing URLs were removed to ensure data quality. Non-textual content and special characters were standardized or removed as appropriate.

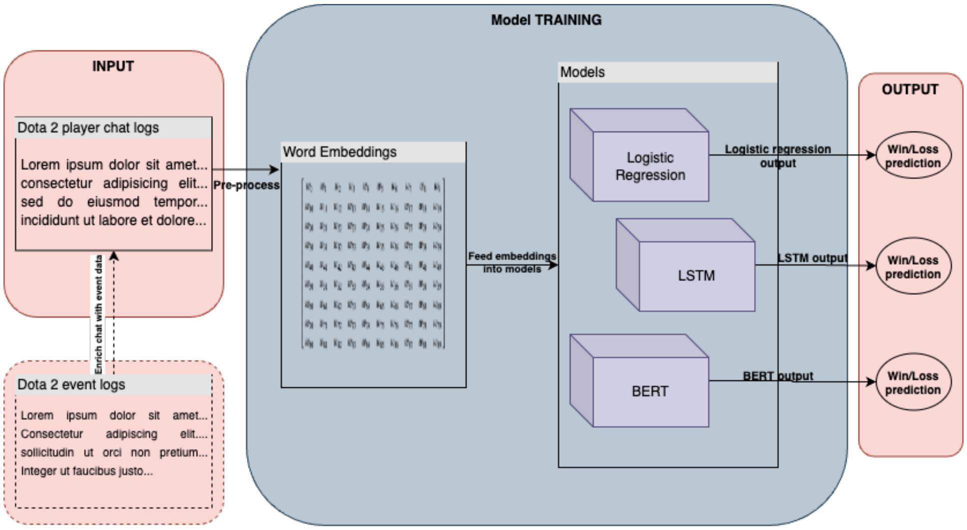

Model TRAINING
INPUT
Dota 2 player chat logs
Lorem ipsum dolor sit amet... consectetur adipisicing elit... sed do eiusmod tempor... incididunt ut labore et dolore...
Pre-process
Enrich chat with event data
Dota 2 event logs
Lorem ipsum dolor sit amet... Consectetur adipiscing elit.... sollicitudin ut orci non pretium... Integer ut faucibus justo...
Word Embeddings
Feed embeddings into models
Models
Logistic Regression
LSTM
BERT
Logistic regression output
LSTM output
BERT output
OUTPUT
Win/Loss prediction
Win/Loss prediction
Win/Loss prediction

Fig. 4. Methodology overview showing the complete pipeline: Input data consists of Dota 2 player chat logs and optional game event logs which undergo preprocessing to standardize format and clean text. The processed text is transformed into embeddings. These embeddings are then fed into three different model architectures - Logistic Regression, LSTM , and DistilBERT - each producing win/loss predictions. The dotted line from event logs indicates this data source is used only for the combined dataset experiments.

- **Temporal Ordering**: Messages were chronologically arranged within each match to preserve the sequential nature of in-game communication.

2) Objectives Data Processing: For our second dataset, we processed game objectives data to complement the chat logs:

- **Event Standardization**: Objective events were formatted with a consistent structure: "[EVENT] Team EventType [END_EVENT]"
- **Event Selection**: Key events including tower destructions, Roshan kills, Aegis pickups, and first blood were identified and extracted.
- **Temporal Alignment**: Events were temporally aligned with chat messages to maintain chronological consistency within each match.

B. Final Dataset Structures

Our preprocessing pipeline produced two distinct datasets for analysis:

1) Chat-Only Dataset: The first dataset contains only processed chat logs ,focusing exclusively on the relationship between team communication and match outcomes, including:

- Match identifier
- Chronologically ordered team-attributed chat messages
- Match outcome (radiant_win)

2) Combined Chat and Objectives Dataset: The second dataset integrates both chat logs and objective events, containing:

- Match identifier
- Formatted messages and events
- Timestamp
- Entry type (chat/event)
- Match outcome (radiant_win)

VI. Models and Training

This section details the three model architectures employed in our study: Logistic Regression as a baseline approach, Long Short-Term Memory (LSTM) networks for sequence modeling, and DistilBERT for contextual language understanding. Each model was trained and evaluated twice: first using only chat data to assess the predictive power of communication alone, and then using the combined chat-and-objectives dataset to evaluate potential improvements from incorporating game events.

A. Logistic Regression

Our baseline model employs logistic regression with Word2Vec embeddings for text representation. The Word2Vec model was pre-trained on the Google News dataset and subsequently fine-tuned on our Dota 2 dataset to better capture domain-specific language patterns. We process each match's chat log by converting messages into fixed-length vectors using these embeddings, then averaging them to create a single feature vector per match. The model uses L2 regularization (C=0.01) and is optimized using the LBFGS solver with a maximum of 250 iterations.

B. LSTM Architecture

To capture the sequential nature of in-game communication, we implemented an LSTM-based architecture. The model begins with an embedding layer of dimension 100, followed by an LSTM layer with 64 hidden units and recurrent dropout of 0.2. The LSTM's output passes through a global max pooling layer to create a fixed-size representation. This is followed by two dense layers (32 and 16 units respectively) with ReLU activation and L2 regularization, separated by dropout layers (0.5 and 0.3) to prevent overfitting. The final layer uses sigmoid activation for binary classification.

The network was trained using the Adam optimizer with a learning rate of 0.0005 and binary cross-entropy loss. We implemented early stopping with a patience of 30 epochs and learning rate reduction on plateau to prevent overfitting. Input sequences were padded or truncated to a maximum length of 512 tokens, with a vocabulary size of 10,000 most frequent words.

C. DistilBERT Model

We leveraged the DistilBERT architecture, a distilled version of BERT that maintains most of its performance while being lighter and faster. The model was initialized with pre-trained weights from 'DistilBERT-base-uncased' and fine-tuned for our binary classification task. We maintained the model's original architecture while adding a task-specific classification head.

The training process used a batch size of 16 and a learning rate of 2e-5 with the Adam optimizer. Weight decay was set to 0.01 to prevent overfitting. The maximum sequence length was set to 512 tokens, with longer sequences truncated and shorter ones padded. Training continued for 30 epochs with early stopping based on validation loss.

D. Evaluation Framework

Model performance was evaluated using a consistent set of metrics across all architectures. We employ accuracy as our primary metric, supplemented by F1-score to account for class imbalance and ROC-AUC to assess discrimination capability. We also track precision and recall to understand the models' trade-offs between false positives and false negatives.

The dataset was split into training (80%), validation (10%), and test (10%) sets, being careful to maintain chronological order to prevent data leakage.

VII. Results and Analysis

In this section, we present and analyze the performance of our three models, evaluating their effectiveness in predicting match outcomes using chat data alone versus the combined chat-and-objectives dataset. We explore both overall performance metrics and model-specific behaviors.

A. Overall Model Performance

Table I presents the comprehensive performance metrics for each model across both datasets. The most striking observation is the substantial improvement in prediction accuracy when combining chat logs with in-game events across all three models.

TABLE I
Model Performance Comparison

Model	Dataset	Accuracy	F1-Score	ROC-AUC
2*DistilBERT	Chat-only	0.814	0.815	0.899
	Combined	0.971	0.971	0.997
2*LSTM	Chat-only	0.799	0.804	0.881
	Combined	**0.984**	**0.984**	**0.999**
2*LogReg	Chat-only	0.564	0.606	0.599
	Combined	0.818	0.827	0.907

B. Impact of Objective Events

The incorporation of objective events significantly enhanced the predictive capabilities of all three models. The DistilBERT model showed a remarkable improvement of 15.7 percentage points in accuracy (from 81.4% to 97.1%), while the LSTM model demonstrated an even more substantial increase of 18.5 percentage points (from 79.9% to 98.4%). The logistic regression model showed the most improvement, with accuracy increasing by 25.4 percentage points (from 56.4% to 81.8%) when incorporating objective events.

C. Model-Specific Analysis

1) DistilBERT Performance: This light-weight DistilBERT model demonstrated strong performance across both datasets. On chat data alone, it achieved 81.4% accuracy with balanced precision (0.818) and recall (0.812). When incorporating objective events, its performance improved dramatically to 97.1% accuracy with near-perfect balance between precision (0.968) and recall (0.974). The high ROC-AUC score (0.997) on the combined dataset indicates excellent discrimination capability.

2) LSTM Performance: The LSTM model showed similar patterns to DistilBERT but achieved the highest overall performance on the combined dataset with 98.4% accuracy. Its balanced performance across precision (0.983) and recall (0.985) suggests consistent prediction capability across both positive and negative cases. The model maintained strong performance even on chat-only data (79.9% accuracy), though notably lower than with the combined dataset.

3) Logistic Regression Performance: The logistic regression model serving as a baseline showed the most pronounced benefit from incorporating objective events. While it struggled with chat-only data (56.4% accuracy), its performance improved substantially with the combined dataset (81.8% accuracy). This improvement suggests that the addition of structured game state information provides crucial features that can be effectively leveraged even by a linear model.

D. Error Analysis

Given that the models trained on a combination of the chat data and event data showed similar, relatively balanced errors, we exam the chat-only models which reveal more interesting variations in error patterns.

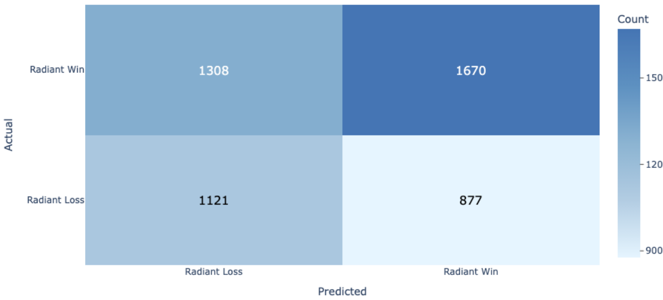


Fig. 5. Confusion matrix for Logistic Regression chat-only model showing poor discrimination ability and bias toward Radiant victories

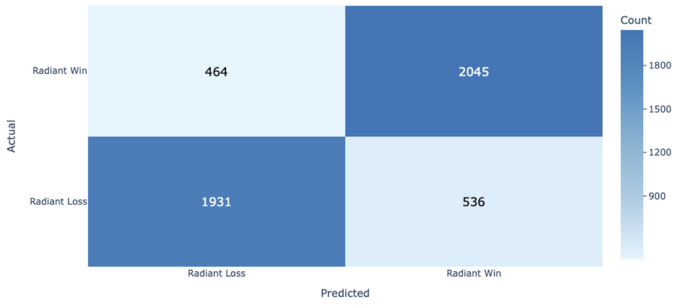


Fig. 6. Confusion matrix for LSTM chat-only model demonstrating balanced predictions and robust feature extraction

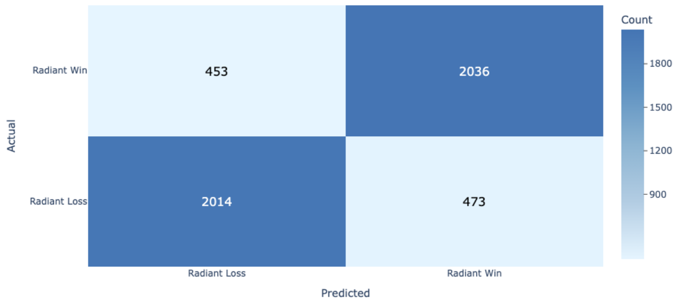


Fig. 7. Confusion matrix for DistilBERT chat-only model showing balanced predictions with accuracy comparable to LSTM

The logistic regression model (Figure 5) shows significant difficulty in learning from chat data alone, with 1,308 false positives and 877 false negatives out of 4,976 predictions, indicating poor discrimination ability and a slight bias toward predicting Radiant victories. The LSTM architecture (Figure 6) on the other hand demonstrates better balanced predictions with 464 false positives and 536 false negatives, suggesting more robust feature extraction from sequential chat data. The DistilBERT model (Figure 7) achieves slightly better performance with 453 false positives and 473 false negatives, indicating balanced and slightly more accurate predictions than its LSTM counterpart.

VIII. Temporal Analysis

An important real-world application of our models would involve looking at how prediction accuracy and confidence evolves throughout the duration of a match. Our analysis examines this temporal progression, focusing on the LSTM and DistilBERT architectures, as they outperform and exhibit similar performance compared to the logistic regression baseline. We will evaluate these architectures across both chat-only and combined (chat logs combined with in-game events) datasets. For clarity, we will refer to models using the chat-only dataset as "chat" and those using the combined chat logs and in-game objective events as "combined."

A. Accuracy Progression

Figure 8 illustrates the temporal evolution of prediction accuracy across all models. Both architectures begin with near-random performance (51-54%) during the pre-game phase, but their trajectories diverge upwards as matches progress with the LSTM models coming our ahead. Looking at the addition of objective events it is clear from Table II, that they substantially enhance prediction accuracy across all phases of the game. The LSTM architecture with combined data achieves the highest overall accuracy (74.81%), outperforming its DistilBERT counterpart (70.48%) consistently across all game phases.

The most striking improvements occur during the transition from early to mid-game (15-30 minutes). In this period, the

TABLE II
Comprehensive Model Performance Comparison

Metric	Chat-Only Models	Combined Models
Overall Accuracy (%)		
LSTM	65.46	74.81
DistilBERT	61.53	70.48
Phase Accuracy (%)		
Early Game (0-15min)		
LSTM	53.73	61.44
DistilBERT	51.77	57.94
Mid Game (15-30min)		
LSTM	63.88	81.47
DistilBERT	58.71	72.75
Late Game (30-60min)		
LSTM	**77.95**	**96.46**
DistilBERT	71.84	90.04
Confidence Metrics		
LSTM Avg Confidence (%)	51.19	52.70
DistilBERT Avg Confidence (%)	62.53	76.37

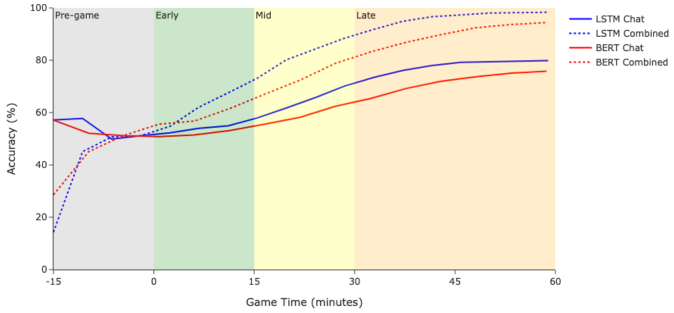


Fig. 8. Model accuracy progression across game phases. The graph shows the evolution of prediction accuracy over time for both LSTM and DistilBERT architectures, comparing chat-only and combined data approaches. Shaded regions indicate game phases: pre-game (gray), early game (green), mid game (yellow), and late game (orange).

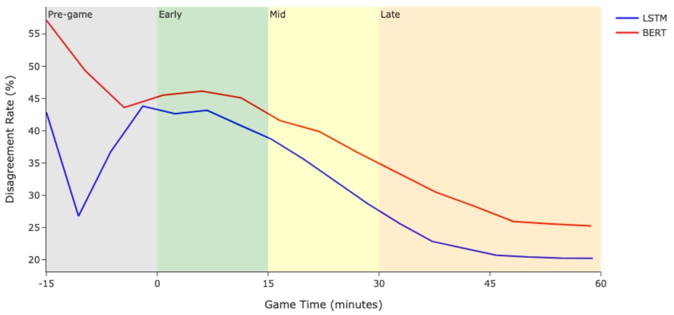


Fig. 9. Disagreement rates between chat-only and combined models over game time. Lower values indicate higher agreement between model predictions using different data sources.

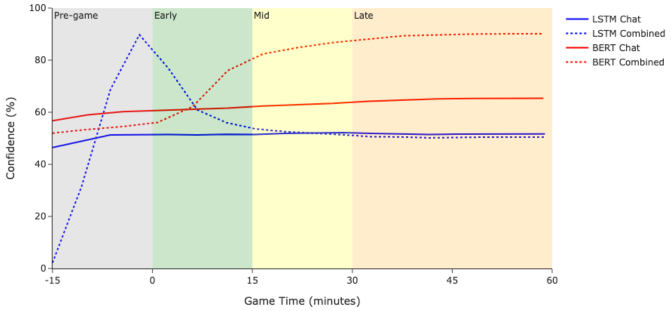


Fig. 10. Model confidence when making correct predictions across game phases. The plot shows how prediction confidence evolves over time for correct predictions, comparing both architectures and data approaches.

LSTM combined model demonstrates a remarkable jump from 61.44% to 81.47% accuracy, while the DistilBERT combined model shows a similar but less pronounced improvement from 57.94% to 72.75%. This pattern suggests that the accumulation of game events during this phase provides particularly valuable predictive information.

B. Model Agreement Analysis

The disagreement analysis (Figure 9) provides insights into prediction consistency between chat-only and combined models. Early game phases show the highest disagreement rates (40-55%), gradually decreasing as matches progress. Notably, LSTM models maintain lower average disagreement rates (31.33%) compared to DistilBERT models (38.26%), indicating more consistent predictions across data sources.

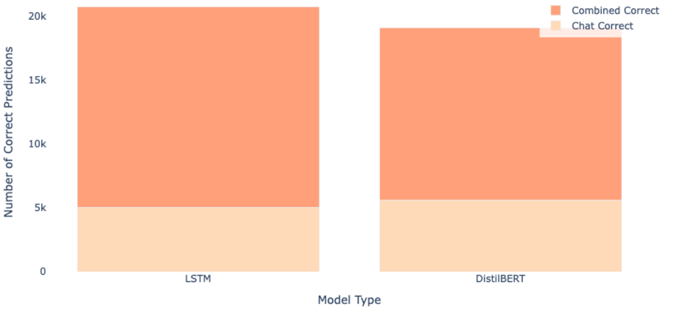


Fig. 11. Comparison of correct predictions when chat-only and combined models disagree.

When models disagree in their predictions, both architectures show substantially better performance with combined data, as illustrated in Figure 11. The LSTM combined model achieves 15,692 correct predictions compared to 5,044 for its chat-only version, a ratio of 3.11. Similarly, the DistilBERT combined model makes 13,478 correct predictions versus 5,609 for chat-only, a ratio of 2.40. This stark difference in performance highlights the value of incorporating game events for resolving prediction uncertainties.

C. Confidence Analysis

Model confidence represents the probability assigned to a predicted outcome, ranging from 0 to 1 (0% to 100%). For LSTM models, confidence is derived from the sigmoid function output, while DistilBERT models use softmax probabilities across prediction classes. The average confidence for correct predictions (C_{avg}) is calculated as:

$$C_{\text{avg}} = \frac{1}{N}\sum_{i=1}^{N} p_i \cdot (\hat{y}_i = y_i) \tag{1}$$

where p_i is the confidence score for prediction i, $\hat{y}_i$ and y_i are the predicted and actual labels respectively, and N is the total number of correct predictions. The indicator function $(\hat{y}_i = y_i)$ equals 1 for correct predictions and 0 otherwise.

The confidence patterns (Figure 10) reveal an interesting trade-off between accuracy and prediction confidence. While the LSTM model achieves higher accuracy, particularly with combined data (98.33% vs 94.40%), it demonstrates notably lower confidence in its correct predictions (52.70% vs 76.37%). This disparity between performance and confidence is particularly relevant for real-world applications, where how confident we are in a model's predictions matters. The DistilBERT model's higher confidence scores suggest it might be more suitable for applications where prediction certainty is prioritized, despite its slightly lower accuracy.

IX. Limitations and Future Work

Our study faces two primary limitations. First, the exclusion of professional matches due to their reliance on voice chat means our models may not capture the unique strategic dynamics of high-level competitive play. Second, filtering non-English messages introduces a significant language bias, potentially excluding valuable insights from Dota 2's global player base where English is not the primary language of communication.

Future work could focus on addressing these limitations by developing multilingual models, and exploring methods to incorporate voice chat analysis for professional matches. Additionally, future work could explore interpretable models to better understand the relationship between communication patterns and match outcomes.

X. Conclusion

Our study demonstrates the viability of using in-game chat logs for predicting Dota 2 match outcomes, with chat-only models achieving accuracy up to 81.4% and combined models reaching 98.4%. The temporal analysis reveals that prediction reliability increases significantly during the mid-game phase (15-30 minutes), with different architectures showing distinct strengths - LSTM achieving higher accuracy while DistilBERT demonstrates greater prediction confidence. Unlike traditional statistics-based approaches that can become obsolete with game updates, our communication-based method offers a more robust foundation for prediction, as patterns of team coordination and strategic discussion remain relatively stable across patches. This study thus establishes chat logs as a viable, update-resistant predictive data source for esports analytics.

References

[1] J Candela and K Jakee, “Can esports unseat the sports industry? some preliminary evidence from the united states,” *choregia*, vol. 14, 2018. DOI: 10.4127/ch.2018.0136.

[2] A. Gilles, “Statistical analysis and machine learning to improve league championship series teams,” 2023.

[3] A. Smerdov, A. Somov, E. Burnaev, and A. Stepanov, “Ai-enabled prediction of video game player performance using the data from heterogeneous sensors,” *Multimedia Tools and Applications*, vol. 82, no. 7, pp. 11 021–11 046, 2023.

[4] X. Du, X. Fuqian, J. Hu, Z. Wang, and D. Yang, “Uprising e-sports industry: Machine learning/ai improve in-game performance using deep reinforcement learning,” in *2021 International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, 2021, pp. 547–552. DOI: 10.1109/MLISE54096.2021.00112.

[5] X. Zhong and J. Xu, “Measuring the effect of game updates on player engagement: A cue from dota2,” *Entertainment Computing*, vol. 43, p. 100 506, 2022, ISSN: 1875-9521. DOI: 10.1016/j.entcom.2022.100506. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1875952122000301.

[6] A Summerville, M Cook, and B Steenhuisen, “Draft-analysis of the ancients: Predicting draft picks in dota 2 using machine learning,” *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertain.*, vol. 12, no. 2, p. 2, 2016. DOI: 10.1609/aiide.v12i2.12899.

[7] G. Aryanata, P. Rahadi, and Y. Sudarmojo, “Prediction of dota 2 match result by using analytical hierarchy process method,” *Int. J. Eng. Emerg. Technol.*, vol. 2, no. 1, pp. 22–25, 2017. DOI: 10.36948/ijfmr.2024.v06i02.14927.

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

[9] D Fesalbon, A De La Cruz, M Mallari, and N Rodelas, “Fine-tuning pre-trained language models to detect in-game trash talks,” *Int. J. Multidiscip. Res.*, vol. 6, no. 2, p. 14 927, 2024. DOI: 10.36948/ijfmr.2024.v06i02.14927.

[10] A Jumabayev, “The effect of sentiments in communication on a team performance,” Ph.D. dissertation, 2022.

[11] Reddit User, *Dear valve, can you put this here please?* Accessed: 2024-11-10, 2017. [Online]. Available: https://www.reddit.com/r/DotA2/comments/7hafzc/dear_valve_can_you_put_this_here_please/?rdt=58312.

[12] C. Berner, G. Brockman, B. Chan, *et al.*, “Dota 2 with large scale deep reinforcement learning,” *arXiv preprint arXiv:1912.06680*, 2019.

[13] H. Zhou, “Research of text classification based on tf-idf and cnn-lstm,” in *journal of physics: conference series*, IOP Publishing, vol. 2171, 2022, p. 012 021.

[14] S Hochreiter and J Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735.

[15] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[16] J Devlin, M.-W. Chang, K Lee, and K Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J Burstein, C Doran, and T Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[17] V Sanh, L Debut, J Chaumond, and T Wolf, “Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter,” 2020, arXiv:1910.01108. [Online]. Available: http://arxiv.org/abs/1910.01108.

[18] J Uddin, I Fahmida, S. Moyeen, and M. M. Hasan, “Dota2 winner team prediction based on stacked bidirectional lstm network,” in *2022 4th International Conference on Electrical, Computer Telecommunication Engineering (ICECTE)*, Dec, 2022, pp. 1–5. DOI: 10.1109/ICECTE57896.2022.10114527.

[19] B Sándor and Z Wan, “Predicting game outcome in dota 2 with nlp and machine learning algorithms,” 2023, Accessed: Mar. 23, 2024. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:hj:diva-62194.

[20] L Zhang, C Xu, Y Gao, Y Han, X Du, and Z Tian, “Improved dota2 lineup recommendation model based on a bidirectional lstm,” *Tsinghua Sci. Technol.*, vol. 25, no. 6, pp. 712–720, 2020. DOI: 10.26599/TST.2019.9010065.

[21] A Traas, “The impact of toxic behavior on match outcomes in dota,” 2017.

[22] A Bello, S.-C. Ng, and M.-F. Leung, “A bert framework to sentiment analysis of tweets,” *Sensors*, vol. 23, no. 1, p. 1, 2023. DOI: 10.3390/s23010506.