



MODELO DE PREDIÇÃO DE VITÓRIA PARA PARTIDAS DE MOBA

Eduardo Vieira Marques Pereira do Valle

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Geraldo Bonorino Xexéo

Rio de Janeiro
Agosto de 2025

MODELO DE PREDIÇÃO DE VITÓRIA PARA PARTIDAS DE MOBA

Eduardo Vieira Marques Pereira do Valle

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Geraldo Bonorino Xexéo

Aprovada por: Prof. Daniel Serrão Schneider

Prof. Carlos Eduardo Ribeiro de Mello

RIO DE JANEIRO, RJ – BRASIL

AGOSTO DE 2025

Vieira Marques Pereira do Valle, Eduardo

Modelo de predição de vitória para partidas de MOBA/Eduardo Vieira Marques Pereira do Valle. – Rio de Janeiro: UFRJ/COPPE, 2025.

XVII, 124 p.: il.; 29, 7cm.

Orientador: Geraldo Bonorino Xexéo

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2025.

Referências Bibliográficas: p. 70 – 83.

1. Ciência de Dados Aplicada a Jogos.
 2. Transformers.
 3. League of Legends.
- I. Bonorino Xexéo, Geraldo. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

*À minha família, amigos e
orientador, pelo apoio constante.*

Agradecimentos

Expresso minha sincera gratidão ao meu orientador, cujo suporte e discernimento analítico foram essenciais para a realização desta pesquisa. Ademais, agradeço à FAPERJ pelo apoio financeiro, que possibilitou a condução deste estudo de maneira abrangente e aprofundada. Agradeço também aos familiares e amigos que me apoiaram durante todo o processo de desenvolvimento desta dissertação.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MODELO DE PREDIÇÃO DE VITÓRIA PARA PARTIDAS DE MOBA

Eduardo Vieira Marques Pereira do Valle

Agosto/2025

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

O jogo do gênero Multiplayer Online Battle Arenas (MOBA), apresenta partidas caracterizadas por confrontos estratégicos em tempo real. A capacidade de prever a vitória em tempo real nesses contextos é de considerável importância prática para o desenvolvimento de ferramentas de suporte para jogadores e equipes profissionais, bem como para a criação de sistemas analíticos que auxiliem na compreensão da dinâmica do jogo. Este trabalho realizou a revisão da literatura acadêmica sobre a previsão de vitórias e eventos em jogos MOBA. Essa investigação identificou uma ausência de estudos analisando o modelo Transformer para previsão de vitória durante a partida.

Para investigar o potencial do modelo Transformer na predição de vitórias em jogos MOBA, foram concebidas duas arquiteturas distintas. A primeira aborda cada atributo da partida como um token, análogo a palavras em tarefas de linguagem natural, permitindo avaliar o impacto da troca da ordenação dos atributos na tarefa de predição de vitória. A segunda trata-se do FT-Transformer, o qual otimiza o tratamento de dados tabulares ao preservar a estrutura inerente da tabela e emprega embeddings específicos por coluna, facilitando a identificação de interações entre atributos sem depender de uma sequência pré-definida.

Os experimentos desse trabalho abordaram duas bases de dados públicas do jogo League of legends, uma oriunda de partidas profissionais de torneios e outra de partidas ranqueadas. Para ambas as versões do modelo Transformer realizou o estudo dos hiperparâmetros adequados para as bases de dados. O FT-Transformer apresentou desempenho superior em ambas as bases, destacando-se frente aos demais métodos avaliados em relação acurácia, revocação e F1-score. Enquanto a outra versão do Transformer, embora demonstre a influência da estrutura dos atributos, apresentou os piores resultados para as mesmas métricas.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

WIN PREDICTION MODEL FOR MOBA MATCHES

Eduardo Vieira Marques Pereira do Valle

August/2025

Advisor: Geraldo Bonorino Xexéo

Department: Systems Engineering and Computer Science

The Multiplayer Online Battle Arenas genre features matches characterized by strategic real-time confrontations. The ability to predict victory in real-time in these contexts is of considerable practical importance for the development of support tools for professional players and teams, as well as for the creation of analytical systems that aid in understanding game dynamics. This work conducted a literature review on victory and event prediction in MOBA games. This investigation identified a lack of studies analyzing the Transformer model for in-game victory prediction.

To investigate the potential of the Transformer model in predicting victories in MOBA games, two distinct architectures were conceived. The first treats each match attribute as a token, analogous to words in natural language tasks, allowing for the evaluation of the impact of attribute ordering on the victory prediction task. The second is the FT-Transformer, which optimizes the handling of tabular data by preserving the inherent structure of the table and employs column-specific embeddings, facilitating the identification of attribute interactions without relying on a predefined sequence.

Experiments in this work used two public datasets from the game League of Legends, one from professional tournament matches and another from ranked matches. For both versions of the Transformer model, a study of appropriate hyperparameters for the datasets was performed. The FT-Transformer presented superior performance on both datasets, outperforming other evaluated methods in terms of accuracy, recall, and F1-score. Meanwhile, the other Transformer version, while demonstrating the influence of attribute structure, yielded the worst results for the same metrics.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xiii
Lista de Símbolos	xv
Lista de Abreviaturas	xvi
1 Introdução	1
1.1 Contextualização	1
1.2 Complexidade e Potencial de Pesquisa	2
1.3 Revisão Sistemática da Literatura	3
1.4 Modelos Transformer e FT-Transformer	3
1.5 Problema de Pesquisa e Hipótese	4
1.6 Objetivos e Metodologia	4
1.7 Resultados e Contribuições	5
1.8 Estrutura do Trabalho	6
2 Revisão da literatura sobre jogos MOBA	7
2.1 História do gênero MOBA	7
2.2 Mecânicas fundamentais do gênero	9
2.3 Relevância do estudo do gênero	10
2.4 Justificativa para a revisão da literatura	11
2.5 Como foi realizado a revisão sistemática rápida	12
2.6 Análise dos estudos levantados na revisão sistemática rápida	13
2.6.1 Métodos na literatura	15
2.6.2 Quais eventos são analisados nos trabalhos da literatura	18
2.6.3 Momento de coleta de dados e predição	19
2.6.4 Bases de dados empregadas na literatura	20
2.6.5 Métricas abordadas na literatura	20

3	Técnicas de Aprendizado de Máquina	25
3.1	Modelos discriminativos	25
3.1.1	Regressão Logística	25
3.1.2	Máquina de Vetores de Suporte (SVM)	26
3.1.3	k-Nearest Neighbours (kNN)	26
3.1.4	Árvores de Decisão e Ensembles	26
3.1.5	Redes Neurais	27
3.1.6	Transformer	28
3.1.7	TSSTN	30
3.2	Modelos generativos	31
3.2.1	Naive Bayes	31
3.2.2	Factorization Machines	31
3.2.3	Modelos de Ranking	31
3.2.4	Outros modelos	32
4	Método Proposto	33
4.1	Visão Geral da Metodologia	34
4.1.1	Seleção dos Dados	34
4.1.2	Pré-processamento dos Dados	35
4.1.3	Definição dos Modelos	35
4.1.4	Ajuste de Hiperparâmetros	35
4.1.5	Avaliação e Comparação dos Resultados	36
5	Experimentos	37
5.1	Seleção e Pré-processamento dos dados	37
5.1.1	Partidas Rankeadas	37
5.1.2	Partidas Profissionais	39
5.2	Definição dos Modelos	42
5.3	Ajuste de Hiperparâmetros	42
5.3.1	Transformer	42
5.3.2	FT-Transformer	44
5.3.3	Regressão Logística	45
5.3.4	Classificador Naive Bayes	45
5.3.5	Random Forest	45
5.3.6	Extreme Gradient Boosting (XGBoost)	46
5.4	Avaliação e Comparação dos Resultados	46
6	Resultados	49
6.1	Resultados do Transformer	49
6.1.1	Impacto do número de cabeças de atenção	50

6.1.2	Impacto do <i>embedding</i> de posição	50
6.2	Resultados do FT-Transformer	52
6.2.1	Impacto dos hiperparâmetros investigados	53
6.2.2	Resultados na base de partidas ranqueadas	53
6.2.3	Resultados na base de partidas profissionais	55
6.2.4	Análise da validação cruzada interna	55
6.2.5	Síntese dos resultados	59
6.3	Discussão Comparativa dos Modelos	61
6.3.1	Desempenho nos dados de partidas ranqueadas	61
6.3.2	Desempenho nos dados de partidas profissionais	62
6.3.3	Análise comparativa e implicações práticas	62
7	Conclusão	67
7.1	Contribuições	68
7.2	Limitações	68
7.3	Trabalhos Futuros	68
	Referências Bibliográficas	70
	Apêndice A: Tabelas Complementares	84
A.1	Revisão de trabalhos sobre predição de vitória em jogos MOBA . . .	84

Lista de Figuras

1.1	Captura de uma partida de <i>Dota 2</i> na Steam. Desenvolvedor e distribuidor: Valve. Lançado em 9 de julho de 2013.	2
2.1	Tela do jogo Aeon of Strife. Fonte: MMOs.com ALTAY (2024).	8
2.2	Arena principal de League of Legends com destaque para as três rotas principais. Fonte: Fragster.com CUNHA (2021).	10
2.3	Arena lotada durante o The International 2017 de <i>Dota 2</i> na <i>KeyArena</i> (atualmente <i>Climate Pledge Arena</i>) em Seattle. Fonte: Fragster.com SHEVCHENKO (2021).	12
2.4	Fluxograma do processo da revisão sistemática rápida.	14
3.1	Arquitetura do Transformer. Fonte: VASWANI <i>et al.</i> (2017).	29
3.2	Arquitetura geral do FT-Transformer. Fonte: GORISHNIY <i>et al.</i> (2021).	30
4.1	Fluxograma da metodologia proposta neste trabalho.	34
6.1	A performance média do modelo Transformer, quantificada pela acurácia, foi avaliada em relação ao número de cabeças de atenção em ambas as bases de dados (partidas ranqueadas e profissionais).	50
6.2	Heatmap da acurácia média do modelo Transformer considerando diferentes estratégias de <i>embedding</i> de posição (Alternado, Agrupado, Aleatório e Sem posição) nas bases de partidas ranqueadas e profissionais.	52
6.3	Acurácia média obtida em 25 <i>folds</i> da validação cruzada aninhada para a base de partidas ranqueadas, utilizando o modelo FT-Transformer com diferentes combinações de hiperparâmetros. Os valores estão apresentados no formato <i>Blocos–Cabeças</i> e o eixo foi ajustado para destacar pequenas variações entre as configurações.	54

6.4	Acurácia média obtida em 25 <i>folds</i> da validação cruzada aninhada para a base de partidas profissionais, utilizando o modelo FT-Transformer com diferentes combinações de hiperparâmetros. Os valores estão apresentados no formato <i>Blocos-Cabeças</i> e o eixo foi ajustado para destacar pequenas variações entre as configurações. . .	55
6.5	Desempenho do modelo FT-Transformer nos diferentes <i>folds</i> externos da validação cruzada aninhada para a base de dados de partidas ranqueadas.	57
6.6	Desempenho do modelo FT-Transformer nos diferentes <i>folds</i> externos da validação cruzada aninhada para a base de dados de partidas profissionais.	58
6.7	Comparação do computacional e acurácia entre os modelos na base de partidas ranqueadas.	64
6.8	Comparação do F1-Score entre os modelos na base de partidas ranqueadas.	64
6.9	Comparação da revocação entre os modelos na base de partidas ranqueadas.	65
6.10	Comparação do custo computacional e acurácia entre os modelos na base de partidas profissionais.	65
6.11	Comparação do F1-Score entre os modelos na base de partidas profissionais.	66
6.12	Comparação da revocação entre os modelos na base de partidas profissionais.	66

Lista de Tabelas

2.1	Métodos empregados para a predição de vitória em jogos MOBA divididos por ano.	17
2.2	Síntese dos trabalhos que abordam a predição de eventos em jogos MOBA, especificando a metodologia adotada, o tipo de dados considerados e os eventos previstos.	21
5.1	Atributos selecionados para os experimentos do conjunto de dados de partidas ranqueadas de League of Legends.	40
5.2	Atributos selecionados para os experimentos do conjunto de dados de partidas profissionais de League of Legends.	48
6.1	Resultados médios (\pm desvio padrão) do modelo Transformer em termos de Acurácia(%), F1-Score(%), Revocação(%) e Tempo de execução(s) para a base de partidas ranqueadas, ao avariar a forma que realiza o <i>embedding</i> de posição.	52
6.2	Resultados médios (\pm desvio padrão) do modelo Transformer em termos de Acurácia(%), F1-Score(%), Revocação(%) e Tempo de execução(s) de execução para a base de partidas profissionais, ao avariar a forma que realiza o <i>embedding</i> de posição.	53
6.3	Resultados da acurácia média obtida em 25 <i>folds</i> para a base de partidas ranqueadas, empregando FT-Transformer e variando o número de blocos de atenção (<i>num_attn_blocks</i>) e o número de cabeças de atenção (<i>num_heads</i>).	54
6.4	Resultados da acurácia média obtida em 25 <i>folds</i> para a base de partidas profissionais, empregando FT-Transformer e variando o número de blocos de atenção (<i>num_attn_blocks</i>) e o número de cabeças de atenção (<i>num_heads</i>).	56
6.5	Acurácia média obtida para a melhor configuração de hiperparâmetros na validação cruzada interna, realizada em cada <i>fold</i> externo da validação cruzada aninhada, utilizando a base de partidas ranqueadas com o modelo FT-Transformer.	59

6.6	Acurácia média obtida para a melhor configuração de hiperparâmetros na validação cruzada interna, realizada em cada <i>fold</i> externo da validação cruzada aninhada, utilizando a base de partidas profissionais com o modelo FT-Transformer.	60
6.7	Resultados médios (\pm desvio padrão) dos principais modelos de classificação avaliados na base de dados de partidas ranqueadas, com base em Acurácia(%), F1-Score(%), Revocação(%) e Tempo de execução(s). 62	62
6.8	Resultados médios (\pm desvio padrão) dos principais modelos de classificação avaliados na base de dados de partidas profissionais, com base em Acurácia(%), F1-Score(%), Revocação(%) e Tempo de execução(s). 63	63
A.1	Síntese dos trabalhos que abordam a predição de vitória em jogos MOBA, destacando o momento da partida considerado, as métricas utilizadas, os métodos de predição aplicados e os principais resultados obtidos.	84

Lista de Símbolos

K	Número de vizinhos considerados na decisão	26
$P(x \mid y)$	Distribuição condicional dos atributos x dado a saída y	25
$P(x, y)$	Distribuição conjunta de atributos x e saída y	25
$P(y \mid x)$	Probabilidade condicional da variável de saída y dado os atributos x	25
$P(y)$	Distribuição marginal da variável de saída y	25
%	Porcentagem, utilizada em métricas de desempenho	38
\mathbf{w}	Vetor de parâmetros (pesos) do modelo	25
\pm	Intervalo “média \pm DP” reportado nas tabelas	51
b	Parâmetro de viés (bias) do modelo	25
k	Número de dobras (folds) na validação cruzada	46
x	Vetor de atributos (features) observados	25
x_i	Vetor de atributos (features) associado à i -ésima observação	25
y	Variável de saída ou rótulo associado ao vetor de atributos	25
y_i	Variável de saída binária da i -ésima observação, assume valores em $\{0, 1\}$	25
<code>bResult</code>	Rótulo binário de vitória do time azul (profissionais)	41
<code>blueWins</code>	Rótulo binário de vitória do time azul (ranqueadas)	38

Lista de Abreviaturas

<i>eSports</i>	Esportes Eletrônicos Competitivos.....1
AdaBoost	Adaptive Boosting 18
API	Application Programming Interface.....20
ARMA	Autoregressive Moving-Average.....20
AUC	Area Under the Curve.....20, 47
Colab	Google Colaboratory 46
CPU	Central Processing Unit (Unidade Central de Processamento) 46
CUDA	Compute Unified Device Architecture 46
F1	F1-Score 51
FT-Transformer	Feature Tokenizer Transformer 3
GBDT	Gradient Boosted Decision Trees.....18, 27
GPU	Graphics Processing Unit (Unidade de Processamento Gráfico) 46
GRU	Gated Recurrent Unit 18, 28
k-fold	Validação Cruzada em k dobras 46
KDA	Kills/Deaths/Assists (Eliminações/Mortes/Assistências)... 25
Keras	Keras 46
kNN	k-Nearest Neighbour 18
LoL	League of Legends.....9

LSTM	Long Short-Term Memory	18, 28
MLP	Multilayer Perceptron	18
MOBA	Multiplayer Online Battle Arena	1
NeuralAC	Neural Architecture for Competitive balance	18
NPC	Non-Player Character (Personagem Não Jogador)	18
OptMatch	Optimal Matching.....	18
PyTorch Tabular	Biblioteca PyTorch Tabular	46
RAM	Random Access Memory (Memória de Acesso Aleatório)...	46
ReLU	Rectified Linear Unit	27
ROC	Receiver Operating Characteristic	20
RPG	Role-Playing Game	2, 9
RTS	Real-Time Strategy	8
scikit-learn	Biblioteca scikit-learn	46
SVM	Support Vector Machine (Máquina de Vetores de Suporte).	18
tanh	Tangente hiperbólica (função de ativação)	27
TF	TensorFlow	46
TI7	The International de 2017 (torneio anual de Dota 2)	11
TS	TrueSkill (sistema de ranqueamento de jogadores)	18
TSSTN	Two-Stage Spatial-Temporal Network	18
XGBoost	Extreme Gradient Boosting	18

Capítulo 1

Introdução

1.1 Contextualização

Em tempos recentes, a percepção dos videogames evoluiu de meras ferramentas de entretenimento para se tornarem objetos de estudo significativos em diversas disciplinas acadêmicas, notadamente nas áreas de ciência de dados, inteligência artificial e aprendizado de máquina (PÉREZ *et al.*, 2023). O crescimento exponencial da indústria de *eSports* solidificou os jogos digitais como fenômenos culturais e econômicos, atraindo um público global massivo de jogadores, espectadores e investidores (BÁNYAI *et al.*, 2020). Dentro deste panorama, o gênero *Multiplayer Online Battle Arena* (MOBA) destaca-se por sua proeminência e popularidade, com partidas que se caracterizam por um elevado nível de competitividade, exigindo trabalho em equipe, planejamento estratégico e tomada de decisão em tempo real (YU *et al.*, 2019).

O gênero *Multiplayer Online Battle Arena* emergiu da comunidade de jogos e da imprensa especializada como um subgênero de jogos de estratégia em tempo real, inspirado por *Defense of the Ancients*. Esses jogos geralmente apresentam duas equipes, frequentemente compostas por cinco jogadores cada, competindo em um mapa simétrico. Cada participante controla um único personagem com habilidades distintas e um sistema de progressão in-game. O objetivo principal é a destruição da estrutura central da base adversária, ao mesmo tempo em que se defende a própria. Muitos MOBAs incluem ondas de unidades controladas por Inteligência Artificial que avançam por caminhos predeterminados, com estruturas defensivas posicionadas ao longo desses percursos para prolongar a partida e servir como objetivos secundários.

As origens do gênero *Multiplayer Online Battle Arena* podem ser rastreadas até modificações de jogos de estratégia em tempo real das décadas de 1990 e 2000, como *StarCraft* e *Warcraft III*, que permitiam a criação de mapas personalizados. Um

marco inicial proeminente foi o mapa *Aeon of Strife*, para *StarCraft*, que introduziu elementos fundamentais do gênero, incluindo rotas simétricas, torres defensivas e unidades controladas por inteligência artificial. A popularidade do formato aumentou significativamente com *Defense of the Ancients*, uma modificação de *Warcraft III*, que aprimorou esses conceitos ao incorporar uma diversidade de heróis, sistemas de itens e um cenário competitivo em desenvolvimento. O sucesso comunitário de *DotA* impulsionou a criação de títulos comerciais independentes, como *League of Legends*, que popularizou a nomenclatura MOBA, e *Dota 2* (Apresentado na Imagem 1.1), desenvolvido pela *Valve*. Esses jogos solidificaram o gênero como um fenômeno global, estabelecendo sua proeminência no cenário competitivo dos *eSports* (JORDAN-VALLVERDÚ *et al.*, 2024; YANG *et al.*, 2023).



Figura 1.1: Captura de uma partida de *Dota 2* na Steam. Desenvolvedor e distribuidor: Valve. Lançado em 9 de julho de 2013.

1.2 Complexidade e Potencial de Pesquisa

A complexidade inerente aos jogos do gênero MOBA advém da fusão de mecânicas de estratégia em tempo real com a progressão de personagens, emulando elementos de RPG, e é acentuada pela forte interdependência entre os jogadores. Essa sinergia resulta em partidas dinâmicas, influenciadas por múltiplos fatores, que, por sua vez, geram um vasto conjunto de dados. Tais dados são empregados em pesquisas para a predição de eventos e desfechos, de maneira análoga à análise de desempenho, treinamento de equipes, engajamento de audiências e apostas em esportes tradicionais, onde a previsão de resultados ou eventos durante a partida é de suma importância.

A tarefa de prever o desfecho de partidas no gênero MOBA transcende o interesse meramente acadêmico ou recreativo, apresentando diversas aplicações práticas que fundamentam a relevância do tema. Além da predição de resultados, a identificação automática de eventos específicos durante as partidas também possui aplicações práticas relevantes, como destacar momentos de maior impacto em transmissões ao vivo (LIE *et al.*, 2019), ou até mesmo detectar sinais comportamentais associados ao estresse e à possível evasão de jogadores (SMERDOV *et al.*, 2021).

1.3 Revisão Sistemática da Literatura

Uma revisão sistemática da literatura foi realizada para mapear e analisar trabalhos sobre predição de resultados e eventos em jogos MOBA. A revisão foi estruturada em quatro etapas: busca em bases de dados reconhecidas utilizando uma consulta de busca específica, aplicação de critérios de exclusão mediante a leitura de títulos e resumos, emprego da técnica de *backtracking snowballing* para identificação de referências adicionais e, por fim, filtragem de trabalhos indisponíveis ou em outros idiomas, totalizando 86 documentos selecionados.

A análise dos trabalhos identificados na revisão identificou as metodologias predominantes, dados comumente utilizados, tipos de eventos investigados, abordagens de estruturação de bases de dados e métricas de avaliação de modelos. A revisão foi guiada por questões de pesquisa que direcionaram a seleção, análise e categorização dos estudos, culminando na identificação do estado da arte na área. Observou-se que a maioria das investigações se concentra na predição de vitórias, enquanto um número reduzido aborda a predição de eventos durante a partida. Além disso, constata-se a escassez de estudos que explorem modelos baseados em Transformer neste contexto.

1.4 Modelos Transformer e FT-Transformer

Nos últimos anos, o modelo Transformer, conhecido por seu mecanismo de atenção que captura relações de longo alcance em sequências sem recursão, tornou-se uma ferramenta revolucionária no processamento de linguagem natural, sendo amplamente utilizado em tarefas como tradução automática, geração de texto e sumarização (VASWANI *et al.*, 2017). Devido ao seu sucesso em processamento de linguagem natural, pesquisadores passaram a investigar adaptações do Transformer para outros tipos de dados, incluindo visão computacional, séries temporais e dados tabulares.

Apesar do sucesso do Transformer em tarefas de texto e imagem, sua aplicabilidade direta em dados tabulares é limitada por fatores como a ausência de uma ordem inerente entre as colunas e a coexistência de atributos numéricos e categóricos. Para

superar essas limitações, GORISHNIY *et al.* (2021) propuseram o *Feature Tokenizer Transformer* (FT-Transformer), uma arquitetura que representa cada atributo como um *token* individual e emprega *embeddings* específicos para diferenciar variáveis. Essa adaptação permite que o modelo rivalize de forma consistente com métodos baseados em árvores de decisão em tarefas de classificação e regressão. No entanto, até o momento, não há registros de sua aplicação no contexto da predição de jogos MOBA, o que ressalta a relevância desta pesquisa.

1.5 Problema de Pesquisa e Hipótese

Embora a predição de resultados em jogos MOBA tenha se beneficiado de avanços recentes em aprendizado de máquina, a pesquisa existente tende a se concentrar em modelos estabelecidos, como regressão logística, árvores de decisão e métodos de *ensemble*. Há uma notável ausência de estudos que apliquem arquiteturas baseadas em Transformer a este domínio específico. Essa lacuna é particularmente significativa dada a natureza complexa dos dados de partidas, que englobam variáveis numéricas e categóricas e representam interações dinâmicas entre jogadores e equipes. Nesse contexto, o problema de pesquisa central deste trabalho é:

Em que medida os modelos Transformer e FT-Transformer podem superar ou complementar abordagens tradicionais da literatura na predição de vitórias em partidas de jogos MOBA durante a partida?

A hipótese que orienta este estudo é que:

Os modelos Transformer e FT-Transformer alcançarão desempenho superior aos principais modelos tradicionais da literatura, desde que adequadamente ajustados em relação a seus hiperparâmetros.

1.6 Objetivos e Metodologia

O presente estudo tem como principal objetivo a avaliação da aplicabilidade do modelo Transformer e de sua variação, o FT-Transformer, na predição de resultados de vitórias em partidas do jogo *League of Legends*. A análise será realizada com base em dados tabulares coletados de diversos contextos competitivos. Complementarmente, esta pesquisa buscará investigar a influência da representação e ordenação de atributos tabulares no desempenho do Transformer, comparar seus resultados com abordagens tradicionais como Random Forest e Regressão Logística, e discutir as vantagens, limitações e possíveis adaptações arquiteturais para futuras investigações na área.

Para alcançar os objetivos propostos, a metodologia adotada envolveu uma sequência de passos. Inicialmente, foram selecionadas bases de dados contendo partidas ranqueadas e profissionais do jogo *League of Legends*. Subsequentemente, realizou-se o pré-processamento dos atributos, que incluiu a seleção de atributos a serem considerados e determinar quais deles seriam categóricos para FT-Transformer. Na etapa seguinte, definiu-se o conjunto de modelos a serem avaliados, compreendendo Transformer, FT-Transformer, classificador Naive Bayes, Random Forest, regressão logística e XGBoost. Realizou a otimização de hiperparâmetros para determinar as configurações ideais para os modelos Transformer e FT-Transformer. O desempenho foi avaliado por meio de métricas estabelecidas, como acurácia, precisão, revocação e F1-score, permitindo uma comparação sistemática dos resultados e uma análise crítica do papel de cada abordagem no problema analisado.

1.7 Resultados e Contribuições

Os resultados experimentais indicaram que o FT-Transformer apresentou a mais elevada performance preditiva, superando os modelos concorrentes em todas as métricas avaliadas, embora com um custo computacional mais elevado. Em contrapartida, abordagens tradicionais como Random Forest e Regressão Logística demonstraram ser competitivas, equilibrando eficácia e eficiência temporal, enquanto o Naive Bayes se destacou pela sua rápida execução. O modelo Transformer, apesar de ter apresentado resultados inferiores, revelou que a ordenação sequencial dos atributos impacta diretamente seu desempenho, sugerindo potenciais adaptações arquiteturais futuras que preservem a interdependência entre equipes e otimizem sua aplicabilidade.

Além dos resultados experimentais, este trabalho apresenta quatro contribuições principais:

1. Uma revisão sistemática rápida da literatura foi realizada, abrangendo 86 estudos focados na predição de resultados e eventos em jogos MOBA. Esta análise identificou as metodologias recorrentes, as variáveis preditivas mais empregadas, as lacunas existentes no conhecimento e, notavelmente, a carência de investigações que apliquem modelos Transformer neste domínio específico.
2. A introdução da abordagem do modelo FT-Transformer para a previsão de vitórias em jogos MOBA, expandindo o conhecimento existente sobre a aplicação de modelos ba
3. A análise comparativa sistemática entre modelos tradicionais e arquiteturas Transformer, permitindo a identificação das vantagens e desvantagens inerentes a cada abordagem.

4. A investigação do impacto da representação e ordenação dos atributos em tabelas no desempenho de modelos de atenção, fornecendo evidências de que a estrutura dos dados influencia diretamente os resultados obtidos.

Essas contribuições reforçam a relevância científica do estudo, ao explorar metodologias de fronteira em aprendizado de máquina, e sua aplicabilidade prática, ao sugerir caminhos que podem beneficiar análises de desempenho em *eSports*, transmissões competitivas e desenvolvimento de novas ferramentas de suporte à decisão.

1.8 Estrutura do Trabalho

Este trabalho está estruturado em 7 capítulos descritos a seguir:

- Capítulo 1 – Introdução: a introdução contextualiza a problemática abordada, define a relevância do estudo e detalha os objetivos e a justificativa da pesquisa.
- Capítulo 2 – Revisão da literatura sobre jogos MOBA: descreve os conceitos inerentes aos jogos MOBA, detalhando as metodologias empregadas na revisão da literatura e examina os achados dos estudos pertinentes à temática.
- Capítulo 3 – Técnicas de Aprendizado de Máquina: revisa as técnicas de aprendizado de máquina predominantes na literatura para a previsão de vitória em jogos MOBA.
- Capítulo 4 – Metodologia: apresenta a estrutura metodológica adotada para a condução da pesquisa.
- Capítulo 5 – Experimentos: detalha as bases de dados empregadas, os procedimentos de pré-processamento de dados, os modelos analisados, os critérios de avaliação e a configuração dos experimentos.
- Capítulo 6 – Resultados: apresenta e analisa os resultados obtidos pelos experimentos.
- Capítulo 7 – Conclusão: sintetiza os principais achados, limitações e propostas para pesquisas futuras.

Capítulo 2

Revisão da literatura sobre jogos MOBA

Este capítulo revisa a literatura sobre jogos do gênero *Multiplayer Online Battle Arena*, para contextualizar o tema abordado nesse estudo, além de servir como base teórica. Inicialmente abrange a história do gênero, desde suas origens como modificações de jogos de estratégia em tempo real até a popularização do formato por títulos como *League of Legends* e *Dota 2*. Discute-se também as mecânicas que definem o gênero, além da importância dos MOBAs no meio acadêmico, econômico e competitivo. Por fim, justifica-se a revisão da literatura, detalha-se a metodologia e analisam-se os estudos levantados, cobrindo métodos, eventos, bases de dados e métricas.

2.1 História do gênero MOBA

O termo *Multiplayer Online Battle Arena* (MOBA) originou-se da comunidade de jogadores de jogos digitais e da imprensa especializada para definir um subgênero de jogos de estratégia em tempo real, inspirado por *Defense of the Ancients* (HILTS-CHER, 2021). Esses jogos tipicamente envolvem o confronto entre duas equipes, frequentemente de cinco jogadores cada, em um mapa simétrico. Cada jogador controla um único personagem com habilidades únicas e um sistema de progressão dentro da partida. O objetivo primordial é a destruição da estrutura principal da base adversária, enquanto se defende a própria. Muitos jogos desse gênero apresentam ondas de unidades controladas por inteligência artificial que avançam por caminhos predeterminados. Ao longo desses caminhos apresenta estruturas defensivas de ambas equipes no intuito de estender duração da partida por servirem como objetivos secundários.

As origens do gênero remontam a jogos de estratégia em tempo real das décadas

de 1990 e 2000, notadamente *StarCraft* e *Warcraft III*, que disponibilizavam aos jogadores editores de mapas avançados e acessíveis. Esses ambientes permitiram a criação de modos de jogo personalizados que alteraram a premissa usual dos RTS, onde o jogador controlava múltiplas unidades em uma perspectiva macroscópica, redirecionando o foco para o comando individual de um herói com atributos específicos em um campo de batalha compartilhado.



Figura 2.1: Tela do jogo Aeon of Strife. Fonte: MMOs.com ALTAY (2024).

Uma das primeiras experiências similar a jogos atuais de MOBA a alcançar notoriedade foi o mapa *Aeon of Strife* (Retratado na imagem 2.1), criado para *StarCraft* aproximadamente entre 2001 e 2002. *Aeon of Strife* consolidou conceitos centrais do gênero MOBA, como cada jogador controlar personagens distintos, rotas simétricas, torres defensivas, tropas controladas por IA e progressão dentro da partida, nesse caso através da aquisição de itens.

A popularização do gênero ocorreu com o lançamento do *Defense of the Ancients* (DotA), modificação de *Warcraft III: Reign of Chaos* e sua expansão *The Frozen Throne* em 2003. O DotA aperfeiçoou os elementos de *Aeon of Strife* com inovações em sistema de itens; a variedade de personagens com papéis específicos e habilidades que poderiam ser melhoradas durante a partida; e o cenário competitivo que surgiu de ligas amadoras online. A versão mais conhecida, *DotA: Allstars*, tornou-se um fenômeno global, com torneios internacionais ainda em meados dos anos 2000. Esse desenvolvimento estabeleceu o gênero como uma categoria distinta, mesmo antes de

uma designação oficial ser amplamente adotada.

A partir do sucesso comunitário de *DotA*, empresas perceberam o potencial de transformar o formato em produto comercial independente. Em 2009, a *Riot Games* lançou *League of Legends*, um título que se inspirou diretamente em *DotA*, além de difundir o termo MOBA, estabelecendo-se como a nomenclatura predominante para jogos análogos a *DotA* e *LoL*, mesmo com a existência de denominações alternativas como *Action Real-Time Strategy*. Tanto a expressão MOBA e *Action Real-Time Strategy* não retratam com precisão o que define jogos similares a *DotA* e *LoL*, mas devido a sua ampla aceitação e reconhecimento por parte da comunidade de jogadores e desenvolvedores, o termo MOBA se consolidou.

Em paralelo, outros títulos semelhantes foram lançados por diferentes desenvolvedoras, que visavam replicar a experiência do *DotA*, tendo destaque a desenvolvedora *Valve* que lançou em 2013 o *Dota 2*, sucessor de *DotA*. *Dota 2* e *LoL* solidificaram a posição do gênero no cenário competitivo de *eSports* globais (JORDAN-VALLVERDÚ *et al.*, 2024; YANG *et al.*, 2023).

2.2 Mecânicas fundamentais do gênero

O gênero MOBA é caracterizado por mecânicas centrais que definem sua identidade e o distinguem de outros gêneros de jogos digitais. Títulos como *Dota 2* e *League of Legends* exemplificam essas características, combinando elementos de jogos de estratégia em tempo real e RPG. O objetivo central em uma partida de MOBA reside na destruição da estrutura principal do oponente. Complementarmente, o mapa apresenta objetivos secundários de relevância estratégica, como os dragões elementares e o Barão Nashor em *League of Legends*, e o Roshan em *Dota 2*, cujas eliminações conferem vantagens significativas às equipes.

A progressão individual dos jogadores é um componente chave, com cada participante gerenciando um personagem com um conjunto único de habilidades. Essas habilidades podem ser aprimoradas ao longo da partida através do ganho de experiência, obtido pela eliminação de adversários e unidades controladas pela IA, o que permite o avanço de nível. Simultaneamente, a aquisição de ouro, acumulado por meio de abates, destruição de estruturas e coleta de recursos, possibilita a compra de itens que alteram significativamente as capacidades do personagem. Essa progressão dupla, baseada em experiência e itens, confere aos MOBAs uma dinâmica semelhante à dos jogos de RPG, mas integrada a um contexto competitivo em tempo real.

A representação usual da arena em jogos MOBA, como ilustrado na Figura 2.2, é definida por três rotas principais: superior, central e inferior. Estas rotas interligam as bases das equipes e funcionam como percursos para unidades controladas

por inteligência artificial. Em todas as rotas apresenta estruturas defensivas de ambas as equipes que criam barreiras ao avanço de tropas e jogadores. A selva, uma área adjacente às rotas, é povoada por criaturas neutras que, ao serem derrotadas, conferem aos jogadores recompensas em forma de ouro e experiência, além de vantagens estratégicas. A configuração espacial do mapa, caracterizada por rotas lineares interconectadas e a presença da selva oferecendo rotas alternativas, cultiva um ambiente dinâmico que incentiva tanto o confronto direto quanto a exploração de táticas de emboscada.



Figura 2.2: Arena principal de League of Legends com destaque para as três rotas principais. Fonte: Fragster.com CUNHA (2021).

Uma característica presente em MOBAs é o sistema de visão limitada do campo de batalha, comumente referido como "neblina de guerra", uma mecânica inerente aos jogos de estratégia em tempo real. A visibilidade é restrita às áreas onde os jogadores têm presença direta ou onde foram posicionados dispositivos de vigilância. O gerenciamento do mapa, que envolve a implantação e remoção desses dispositivos, representa um subsistema estratégico de considerável importância, uma vez que as informações adquiridas ou suprimidas podem ser determinantes para o sucesso em emboscadas, a segurança durante o avanço em rotas e a disputa por objetivos.

2.3 Relevância do estudo do gênero

A complexidade do gênero MOBA ocorre devido a fusão entre mecânicas de estratégia em tempo real, elementos de progressão de personagem similar aos de RPGs e um nível alto de interdependência tática entre os jogadores. Essa convergência de

características estabelece um cenário dinâmico e multifatorial, no qual as decisões tomadas individualmente e em conjunto se entrelaçam para determinar o resultado final da partida. A natureza virtual de jogos do gênero possibilita a coleta massiva de dados sobre o comportamento dos jogadores e o desenvolvimento de estratégias. Essa riqueza de dados oferece um terreno fértil para pesquisas sobre a predição de eventos e desfechos de partidas.

Dessa forma, a literatura atual sobre MOBA investiga essa abundância de dados, com o objetivo de entender como fatores temporais e contextuais impactam partidas, jogadores e eventos durante a partida. Semelhante a esportes tradicionais, prever o resultado ou eventos de uma partida em andamento tem implicações para o treinamento, análise de desempenho, mercado de apostas e engajamento de espectadores.

Um exemplo é o trabalho KATONA *et al.* (2019), que desenvolveu um modelo capaz de prever, com antecedência de cinco segundos, a morte de um jogador. Essa abordagem é particularmente útil para transmissões ao vivo, permitindo que comentaristas e o público antecipem momentos cruciais do jogo. Seguindo a mesma linha de suporte à análise em tempo real, YANG *et al.* (2022b) propuseram prever a vitória para o jogo *Honor of Kings* durante o decorrer da partida, além de fornecer análises sobre os fatores relevantes para o resultado previsto. Ao informar quais aspectos seriam importantes, auxilia o treinamento de times profissionais e avaliar a performance de jogadores e equipes.

A significância econômica dos principais títulos do gênero MOBA é notória, especialmente em torneios que estabelecem novos recordes no cenário de competições de *eSports*. Para exemplificar essa dinâmica, o título Dota 2 serve como um estudo de caso notório, tendo se consolidado não apenas em termos de audiência, mas também como um modelo de monetização dentro do setor de jogos eletrônicos. Um exemplo é o torneio *The International 2017* (TI7) de Dota 2, realizado na *KeyArena* (retratado na imagem 2.3), em Seattle, que distribuiu aproximadamente US\$ 24,7 milhões em premiações, estabelecendo à época um novo recorde para competições de e-sports (CAPITAL, 2018).

2.4 Justificativa para a revisão da literatura

Devido a complexidade dos dados e a natureza estratégica dos jogos MOBA, a comunidade acadêmica demonstra um interesse cada vez maior na análise e previsão de eventos e resultados. Essa crescente atenção motiva a condução de uma revisão sistemática rápida, norteadas pelas seguintes questões:

- Quais metodologias são empregadas para executar a predição de eventos e



Figura 2.3: Arena lotada durante o The International 2017 de *Dota 2* na *KeyArena* (atualmente *Climate Pledge Arena*) em Seattle. Fonte: Fragster.com SHEVCHENKO (2021).

desfechos de partidas de jogos MOBA?

- Quais informações dos jogos MOBA são utilizadas para treinar e efetuar a predição dos eventos ou resultados das partidas?
- Quais eventos são alvo de estudo dos trabalhos que realizam predição de eventos em jogos MOBA?
- Como as bases de dados utilizadas em pesquisas na literatura foram estruturadas?
- Quais métricas são utilizadas para avaliar as metodologias empregadas na predição de eventos e desfechos de partidas de jogos MOBA?

2.5 Como foi realizado a revisão sistemática rápida

A fim de responder as perguntas descritas anteriormente, a revisão da literatura foi estruturada com quatro etapas distintas. A primeira etapa foi realizada a procura de trabalhos nas bases de dados ACM Digital Library, IEEE Digital Library, Science Direct e Scopus através da seguinte string de busca: `("Multiplayer Online Battle Arena" OR dota OR "Defense of the Ancients" OR dota2 OR "League of Legends" OR MOBA) AND (predict* OR "micro-prediction")`. Essa etapa retornou

404 documentos, dos quais foram excluídos 338 documentos na segunda etapa, que consiste na identificação dos trabalhos a serem excluídos, por meio da leitura dos títulos e resumos de todos os documentos, pela verificação de quais trabalhos atendem os seguintes critérios de exclusão:

- Documentos que não abordam o estudo de predição de eventos ou vitória de jogos MOBA.
- Trabalhos que avaliam eventos externos a partidas de jogos MOBA, não estão relacionados diretamente a dinâmicas e mecânicas do jogo ou que ocorrem antes ou depois de uma partida.
- Pesquisas que estudam apenas a predição de comportamentos sociais atrelados a jogos MOBA.

A terceira etapa aplica *Backtracking snowballing* nos 66 trabalhos que restaram da etapa anterior, procedimento em que as referências dos artigos previamente selecionados são analisadas de forma retroativa, com o objetivo de identificar estudos adicionais relevantes. Por meio desse processo foram encontrados mais 34 documentos relacionados ao tópico abordado. E na última etapa são retirados os trabalhos redigidos em outras linguagens, além do português e inglês, ou sem a possibilidade de acesso, tendo esta revisão identificado 86 documentos. Todo processo retratado anteriormente é resumido na imagem 2.4.

2.6 Análise dos estudos levantados na revisão sistemática rápida

CONLEY e PERRY (2013) e YANG e ROBERTS (2013) foram os primeiros trabalhos a tratar da predição de vitória em jogos MOBA, este realizava a predição com dados finais da partida enquanto aquele utilizava a composição dos personagens em cada time para definir o vencedor da partida, que por meio dessa predição realizava a recomendação das melhores composições para garantir a vitória, que seria o objetivo principal de seu estudo. As próximas pesquisas mantiveram abordagens parecidas de predição de vitória no final da partida ou no início com a seleção de personagens, já que a coleta de dados e métodos de predição para abordar esses contextos são mais simples. Só em 2016, o trabalho YANG *et al.* (2016) tratou da predição de vitória no decorrer da partida ao analisar dados informados nos 10 minutos iniciais da partida. Já em 2017, CLEGHERN *et al.* (2017) foi a primeira pesquisa a abordar o estudo da predição de eventos de jogos MOBA, tentando prever a alteração de vida dos personagens dos jogadores por meio da regressão.

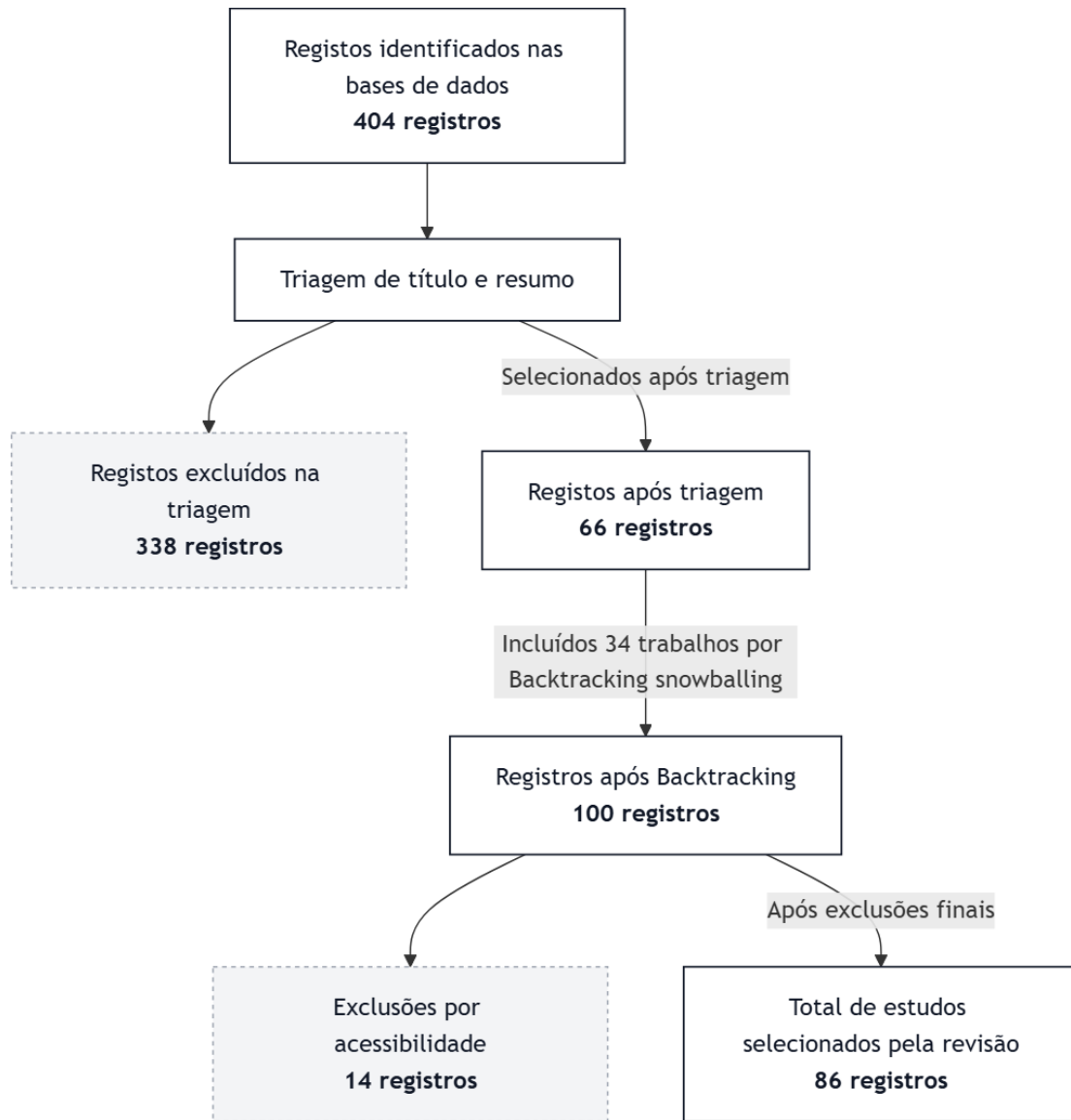


Figura 2.4: Fluxograma do processo da revisão sistemática rápida.

Como foi retratado no trabalho YANG e ROBERTS (2013), algumas pesquisas encontradas pela revisão da literatura não apresentam como principal objetivo de estudo a predição de vitória ou eventos em jogos MOBA. Por exemplo, os estudos CHEN *et al.* (2018a), CONLEY e PERRY (2013), LEE *et al.* (2022), HANKE e CHAIMOWICZ (2017) e CHEN *et al.* (2021) empregam preditores de vitória a fim de definir as melhores composições de personagens em seus sistemas de recomendação de composições de times. Enquanto o trabalho SMIT (2019) estrutura um sistema de recomendação de itens por meio de preditores de vitória; e SMERDOV *et al.* (2021) utiliza predição de vitória de encontros na partida com o propósito de detectar fadiga dos jogadores.

A maior parte dos documentos encontrados aborda a predição de vitória de jogos MOBA, enquanto apenas 11 trabalhos abordam a predição de eventos, sendo

que quatro destes também analisam a predição de vitória. Certos trabalhos investigam a predição de vitória durante a seleção de personagens apenas pela informação de quais foram selecionados pelas equipes CONLEY e PERRY (2013) CARDOSO (2019) CHEN *et al.* (2021). Em contrapartida, outros acrescentam informações a mais para o processo da seleção de personagens, como histórico de performance de jogadores e/ou equipes DO *et al.* (2021), estatísticas e funções dos personagens por especialistas CHAN *et al.* (2020), e a relação entre os personagens KINKADE e LIM (2015). Alguns estudos tentam prever a vitória antes da seleção dos personagens por meio do histórico de performance dos jogadores e equipes JADOWSKI e CUNNINGHAM (2022) LEE *et al.* (2022). Já os trabalhos ZHANG (2021) e JOHANSSON e WIKSTROM (2015) abordam a predição de vitória depois de decorrido certo tempo da partida após a seleção de personagens, as quais analisam vários estados distintos de tempo de partidas para realizar a predição de vitória.

Após a leitura e análise de todos os trabalhos encontrados na revisão, estabelecemos as respostas para cada uma das perguntas definidas anteriormente, sendo elas descritas nas subseções a seguir.

2.6.1 Métodos na literatura

De acordo com a tabela 2.1, a maioria dos trabalhos de predição de vitória utilizam principalmente os métodos: Regressão logística KLEINBAUM *et al.* (2002), Random Forest HO (1995), Redes neurais BISHOP e NASRABADI (2006) e Máquina de vetores de suporte (SVM) BOSER *et al.* (1992). Para cada método, há um leque de aplicações devido às suas características: a Regressão Logística é interpretável e adequada para cenários binários; o Random Forest é resiliente ao sobreajuste e gerencia bem variáveis heterogêneas; as Redes Neurais identificam interconexões não lineares complexas; e a Máquina de Vetores de Suporte é eficaz em dados de alta dimensionalidade com margens de decisão claras. Observou-se que tais métodos foram utilizados devido à sua simplicidade, eficiência e rapidez de treinamento, além de apresentarem desempenho satisfatório em problemas de classificação. Consequentemente, esses métodos foram utilizados como referência para a avaliação comparativa de desempenho em relação a outras abordagens, especialmente a Regressão Logística e a Random Forest.

Ademais, demonstra uma diversidade grande de metodologias utilizadas na predição de vitória de jogos MOBA por apresentar 22 métodos de predição, sendo o último a junção de 17 metodologias que foram aplicadas apenas por um trabalho. Dessa forma, teriam 38 métodos distintos aplicados a predição de vitória em jogos MOBA na literatura.

Para a predição de eventos em jogos MOBA, as metodologias principais aplicadas

são Redes neurais BISHOP e NASRABADI (2006) e Regressão logística KLEIN-BAUM *et al.* (2002), sendo esta empregada para calcular eventos de variações de parâmetros durante a partida, enquanto aquela trata de eventos mais complexos. Tendo a metodologia aplicada de cada trabalho informada na tabela A.1 para pesquisas que avaliam predição de vitória, e na tabela 2.2 retrata os trabalhos que realizam predição de eventos. O capítulo 3 apresenta a descrição detalhada dos principais métodos listados na tabela 2.1.

Tabela 2.1: Métodos empregados para a predição de vitória em jogos MOBA divididos por ano.

Métodos	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	Total
Regressão logística(KLEINBAUM <i>et al.</i> , 2002)	1	2	3	5	4	5	1	4	5	7	2	1	40
Random Forest(HO, 1995)	0	0	3	0	2	2	3	3	4	6	1	1	25
Redes neurais(BISHOP e NASRABADI, 2006)	0	1	0	1	1	3	2	3	3	5	2	0	21
SVM(BOSER <i>et al.</i> , 1992)	0	1	1	0	1	3	1	0	2	5	4	2	20
kNN(PETERSON, 2009)	1	0	0	0	1	1	0	1	4	3	4	1	16
GBDT(FRIEDMAN, 2001)	0	0	0	1	1	2	4	0	4	1	2	0	15
Classificador Naive Bayes(RISH <i>et al.</i> , 2001)	0	1	1	0	4	1	0	1	1	2	2	2	15
LSTM(HOCHREITER, 1991)	0	0	0	0	0	2	0	0	1	5	1	0	9
Árvores de decisão(BHARGAVA <i>et al.</i> , 2013)	1	1	0	0	1	2	0	0	2	2	0	0	9
XGBoost(BHATTACHARYA <i>et al.</i> , 2020)	0	0	0	0	0	0	2	3	0	1	2	0	8
MLP(TAUD e MAS, 2018)	0	0	0	1	1	1	1	2	1	0	1	0	8
Factorization Machine(RENDLE, 2010)	0	0	0	0	1	1	0	1	1	1	0	0	5
AdaBoost(FREUND e SCHAPIRE, 1997)	0	0	1	0	1	1	1	0	0	1	0	0	5
Rede neural GRU(CHO <i>et al.</i> , 2014)	0	0	0	0	0	2	0	0	0	2	0	0	4
Bradley-Terry(BRADLEY e TERRY, 1952)	0	0	1	1	0	0	0	1	1	0	0	0	4
Transformer(VASWANI <i>et al.</i> , 2017)	0	0	0	0	0	0	0	0	0	2	1	0	3
OptMatch(GONG <i>et al.</i> , 2020)	0	0	0	0	0	0	0	1	1	1	0	0	3
TSSTN(YANG <i>et al.</i> , 2022b)	0	0	0	0	0	0	0	0	0	2	0	0	2
Extra tree(GEURTS <i>et al.</i> , 2006)	0	0	0	0	0	0	1	0	0	1	0	0	2
NeuralAC(GU <i>et al.</i> , 2021)	0	0	0	0	0	0	0	0	1	1	0	0	2
TS(HERBRICH <i>et al.</i> , 2006)	0	0	0	0	1	0	0	0	1	0	0	0	2
Outros métodos empregados apenas 1 vez	0	2	2	1	1	2	1	1	1	3	3	0	17

2.6.2 Quais eventos são analisados nos trabalhos da literatura

Dos 86 trabalhos retratados pela revisão apenas 11 deles abordam a predição de eventos em partidas de jogos MOBA, existindo uma variedade de eventos distintos tratados por esses trabalhos. KATONA *et al.* (2019), YANG *et al.* (2023), RINGER *et al.* (2023), JIANG *et al.* (2020) e LIE *et al.* (2019) estudam eventos de morte dos jogadores, mas YANG *et al.* (2023) e LIE *et al.* (2019) investigam também a morte de NPCs de destaque, já WONG *et al.* (2022) tenta prever qual equipe ira realizar a primeira eliminação de um jogador.

Outrossim, JIANG *et al.* (2020) também realizam a predição de performance dos jogadores, tentando prever antes da partida para cada jogador o engajamento dele, suas assistências para eliminações de outros personagens, quantos personagens eliminou e quantas vezes foi eliminado. Outros trabalhos analisam a variação de atributos dos jogadores, PEDRASSOLI CHITAYAT *et al.* (2020) verifica a predição de alterações de ouro, enquanto CLEGHERN *et al.* (2017) estuda alterações de vida, além da pesquisa LIE *et al.* (2019) analisar a predição de dano em jogadores, edifícios e NPCs. Ademais, TOT *et al.* (2021) averiguá a predição da ocorrência de encontros entre os times, ao passo que SMERDOV *et al.* (2021) prediz qual equipe ganharia o encontro, e ZHANG *et al.* (2020) prediz, por meio de redes neurais, eventos de destaque da partida para a plateia.

Dessa forma, os eventos analisados para predição na literatura seriam mortes dos jogadores, estruturas e de NPCs principais de cada jogo; alteração de vida e ouro dos jogadores durante a partida; e a ocorrência e resultado de eventos. A tabela 2.2 destaca quais eventos são analisados por cada trabalho que realiza a predição de eventos de jogos MOBA e suas informações principais.

A escassez de estudos e a diversidade nos eventos analisados limitam comparações padronizadas e a criação de referências. Isso evidencia a necessidade de pesquisas que explorem a conexão entre eventos intermediários e o desfecho das partidas, além da elaboração de conjuntos de dados uniformes que garantam a replicabilidade e comparabilidade dos resultados. Entretanto, este trabalho se dedica à previsão de vitória em jogos MOBA, motivado pelo interesse do autor e pela importância de antecipar resultados de partidas, já que no cenário competitivo, a capacidade de prever desfechos é valiosa para diversas partes interessadas. Embora esta pesquisa não se dedique diretamente à predição de eventos, o estudo de trabalhos que abordam esse tópico auxiliou na identificação de aspectos importantes para a previsão de vitória de jogos MOBA, como bases de dados, metodologias e métricas.

2.6.3 Momento de coleta de dados e predição

Os jogos da categoria MOBA geram inúmeros dados por partida em virtude da enorme quantidade de mecânicas complexas distintas e heróis com habilidades e estatísticas diferentes. Além disso, têm em média 10 jogadores por partida que por meio de suas interações possibilitam partidas variadas. Portanto, modelos de predição para eventos ou desfechos de partidas podem utilizar muitos dados distintos como entrada de treinamento de modelos de predição de vitória e de eventos. No intuito de facilitar a análise de como são coletadas e quais informações são empregadas na literatura, pode dividir os trabalhos definidos pela revisão em três categorias. Essa divisão é baseada em qual momento da partida os estudos realizam a predição e concentram seus esforços na captura dos dados, em relação à etapa de seleção dos personagens.

A primeira categoria engloba a coleta de dados antes da seleção de personagens. Essa categoria compreende estudos que necessitam de informações prévias dos jogadores ou equipes para inferir o resultado da partida XIA *et al.* (2019) JADOWSKI e CUNNINGHAM (2022) HITAR-GARCÍA *et al.* (2023). Esses trabalhos desenvolvem metodologias que descrevem relatórios de performance dos jogadores ou equipes, que podem ser utilizados para balanceamento de equipes, identificação de jogadores de alto nível ou trapaceiros JADOWSKI e CUNNINGHAM (2022).

A segunda categoria trata da coleta de dados durante a seleção de personagens. A maior parte dos trabalhos nessa categoria utilizam combinações de personagens de cada equipe para realizar a predição de vitória CARDOSO (2019) ANSHORI *et al.* (2018) SEMENOV *et al.* (2017). Alguns deles desenvolvem sistemas de recomendação de seleção de personagens por meio da predição de vitória CHEN *et al.* (2018a) HANKE e CHAIMOWICZ (2017) CONLEY e PERRY (2013). Apenas o trabalho WONG *et al.* (2022) que realiza predição de eventos está presente nessa categoria, pois realiza a predição de qual equipe eliminaria primeiro um adversário com base nas combinações de personagens pelas equipes durante a seleção de personagens.

A última categoria aborda estudos que realizam a coleta após a seleção de personagens, com intuito de realizar predição de eventos no decorrer do jogo SMERDOV *et al.* (2021) KATONA *et al.* (2019) ZHANG *et al.* (2020), ou predição de vitória no decorrer da partida HODGE *et al.* (2019) KE *et al.* (2022) TIAN *et al.* (2022). Além disso, Esses dados podem ser estruturadas em séries de tempo com a finalidade de definir estados distintos durante a partida HODGE *et al.* (2019) JUNG e KIM (2022). As tabelas A.1 e 2.2 demonstram qual categoria cada trabalho é definido.

2.6.4 Bases de dados empregadas na literatura

Na literatura, os conjuntos de dados são frequentemente estruturados com base em informações provenientes de torneios públicos, dados disponibilizados por empresas de jogos ou APIs desenvolvidas por jogadores. Em muitos casos, jogadores entusiasmados e pesquisadores independentes desenvolvem ferramentas para coleta sistemática de dados em tempo real ou históricos de partidas, contribuindo significativamente para o volume e diversidade de dados disponíveis. Em virtude da alta popularidade de jogos como *Dota 2* e *League of Legends*, e da disponibilização de dados de partidas por parte de suas respectivas desenvolvedoras, existe uma grande variedade de bases de dados disponíveis desses jogos. Ademais, a comparação direta de resultados entre diferentes estudos na área revela-se desafiadora, em virtude da escassez de pesquisas que disponibilizam os bancos de dados utilizados em seus experimentos. Tal cenário resulta em uma vasta diversidade de bases de dados na literatura, enquanto poucos trabalhos se dedicam à análise dos mesmos conjuntos de dados. A padronização de protocolos experimentais é necessária para possibilitar comparações mais consistentes entre os estudos, uma vez que a diversidade de bases de dados empregadas e a carência da padronização dos experimentos restringem, até o presente momento, uma avaliação consolidada da evolução dos métodos aplicados na área.

2.6.5 Métricas abordadas na literatura

Para aferir a performance dos modelos desenvolvidos na literatura, vários estudos empregam métricas estatísticas estabelecidas no domínio do aprendizado de máquina. Dentre estas, a acurácia e a Área sob a Curva ROC (AUC - Area Under the Curve) são as mais recorrentes. A acurácia expressa a razão entre as predições corretas efetuadas pelo modelo e o número total de observações, contudo, em situações de desequilíbrio entre classes, justifica a preferência pela AUC nesses casos. Isso ocorre, pois AUC examina a habilidade do modelo em discriminar entre as classes em distintos limiares de decisão, apresentando uma avaliação mais sólida, sobretudo quando a base de dados não é balanceada.

Adicionalmente a essas métricas, a literatura também utiliza indicadores complementares para conduzir análises mais detalhadas e sensíveis ao contexto de previsão de eventos e partidas de MOBA. Entre esses, incluem-se a revocação (recall), F1-Score, erro médio absoluto, erro quadrático médio e o tempo de execução, já que o emprego dessas métricas auxilia na identificação de compromissos entre a precisão e o custo computacional, além de simplificar a adequação dos modelos a diferentes propósitos práticos. As tabelas A.1 e 2.2 descrevem quais métricas são empregadas nos experimentos dos trabalhos selecionados pela revisão da literatura.

Tabela 2.2: Síntese dos trabalhos que abordam a predição de eventos em jogos MOBA, especificando a metodologia adotada, o tipo de dados considerados e os eventos previstos.

Trabalhos	Métodos	Eventos	Momento de predição	Métricas
CLEGHERN <i>et al.</i> (2017)	Autoregressive moving-average (ARMA), Regressão logística e Regressão linear	Alteração de vida dos jogadores	Após a seleção de personagens	Acurácia e erro médio
LIE <i>et al.</i> (2019)	AdaRank e k-Nearest Neighbour (kNN)	Destruição ou dano do “Roshan”, heróis ou edifícios.	Após a seleção de personagens	Acurácia
KATONA <i>et al.</i> (2019)	Rede neural	Evento de morte durante a partida	Após a seleção de personagens	AUC, precisão e revocação
ZHANG <i>et al.</i> (2020)	Rede neural e Rede neural recorrente long short-term memory(LSTM)	Predição de eventos de destaque nas partidas.	Após a seleção de personagens	Precisão, revocação e F1-Score

Continua na próxima página

Continuação da Tabela 2.2

Trabalhos	Métodos	Eventos	Momento de predição	Métricas
PEDRASSOLI CHI-TAYAT <i>et al.</i> (2020)	Regressão logística	Predição de alterações de ouro.	Após a seleção de personagens	Correlação entre atributo e resultado esperado
JIANG <i>et al.</i> (2020)	Fatoração de tensores de incorporações neurais individualizadas com reconhecimento de contexto (NICE), Rede neural, Random Forest e Extreme Gradient Boosting (XGBoost)	Predição de mortes de campeões, assistências, 'KDA' e engajamento de usuário	Antes da seleção de personagens	AUC, erro quadrático médio e erro quadrático médio normalizado
TOT <i>et al.</i> (2021)	Rede neural	Predição da ocorrência de encontros entre os times	Após a seleção de personagens	Acurácia e F1-Score

Continua na próxima página

Continuação da Tabela 2.2

Trabalhos	Métodos	Eventos	Momento de predição	Métricas
SMERDOV <i>et al.</i> (2021)	Transformer e Rede neural Gated Recurrent Unit (GRU), Regressão logística e k-Nearest Neighbour (kNN)	O time que ganharia o encontro	Após a seleção de personagens	AUC, precisão média e revocação
WONG <i>et al.</i> (2022)	Classificador Naive Bayes, Regressão logística, k-Nearest Neighbour (kNN), Máquina de vetores de suporte(SVM) e Rede neural	Qual equipe irá realizar a primeira eliminação	Durante a seleção de personagens	Acurácia, precisão, revocação e F1-Score
RINGER <i>et al.</i> (2023)	Rede neural	Evento de morte durante a partida	Após a seleção de personagens	Precisão, precisão média, revocação e F1-Score

Continua na próxima página

Continuação da Tabela 2.2

Trabalhos	Métodos	Eventos	Momento de predição	Métricas
YANG <i>et al.</i> (2023)	Rede neural recorrente long short-term memory(LSTM) e Transformer	Evento de morte durante a partida e qual time derrotou o Tyrant	Após a seleção de personagens	Acurácia

Capítulo 3

Técnicas de Aprendizado de Máquina

Este capítulo revisa as técnicas de aprendizado de máquina predominantes na literatura para a previsão de resultados e eventos em jogos do gênero Multiplayer Online Battle Arena.

As técnicas abordadas neste capítulo são categorizadas com base na distinção entre modelos discriminativos, que visam delimitar diretamente a fronteira de decisão ou estimar a probabilidade condicional $P(y|x)$, e modelos generativos, que caracterizam a distribuição conjunta $P(x, y)$ ou $P(x|y)$ e $P(y)$, permitindo tanto a predição quanto a geração de novas instâncias.

3.1 Modelos discriminativos

3.1.1 Regressão Logística

A Regressão Logística é um modelo de aprendizado de máquina de natureza discriminativa, sendo amplamente adotado em tarefas de classificação binária. cuja formulação clássica é apresentada em KLEINBAUM *et al.* (2002). O modelo estima a probabilidade da classe positiva com base em um vetor de atributos x_i , correlacionando-os linearmente com os parâmetros do modelo (\mathbf{w} e b) e utilizando uma função logística para gerar valores entre zero e um.

A Regressão Logística é reconhecida por sua simplicidade, interpretabilidade e eficiência computacional, estabelecendo-a como uma técnica de referência em análises comparativas com modelos mais sofisticados. Em virtude disso, torna a regressão logística o método mais empregado na literatura de predição de vitórias em jogos MOBA.

3.1.2 Máquina de Vetores de Suporte (SVM)

A Máquina de Vetores de Suporte (SVM) constitui um paradigma de aprendizado de máquina discriminativo, concebido em BOSER *et al.* (1992), que se destaca em problemas de classificação. Seu objetivo primordial é determinar o hiperplano ótimo que maximiza a margem de separação entre diferentes classes de dados, visando aprimorar a capacidade de generalização do modelo. Para cenários onde os dados não admitem uma separação linear, emprega funções de perda que oferecem tolerância a erros de classificação.

A característica central das Máquinas de Vetores de Suporte reside na sua capacidade de empregar funções de *kernel*, que efetivamente mapeiam os dados de entrada para um espaço de características de dimensionalidade superior. Essa projeção permite a separação linear de conjuntos de dados que, em sua representação original, não são linearmente separáveis. Consequentemente, a SVM demonstra ser uma abordagem adaptável, apta a tratar classificações intrincadas em diversas áreas de aplicação, como na previsão de desfechos em jogos digitais.

3.1.3 k-Nearest Neighbours (kNN)

O algoritmo k-Nearest Neighbours (kNN), detalhado por PETERSON (2009), é uma técnica discriminativa que opera com base em instâncias. A classificação de uma nova observação é determinada pela regra da maioria de votos entre os seus K vizinhos mais próximos no espaço de atributos, sendo a distância Euclidiana uma métrica comum para definir a proximidade.

Embora simples, o kNN pode incorrer em custos computacionais elevados para conjuntos de dados de alta dimensionalidade, dado que cada nova predição exige o cálculo de distâncias em relação a todas as amostras de treinamento. Não obstante, sua característica não paramétrica o estabelece como um método de referência útil para problemas de classificação.

3.1.4 Árvores de Decisão e Ensembles

As Árvores de Decisão (BHARGAVA *et al.*, 2013) constituem modelos que segmentam recursivamente o espaço de atributos em regiões homogêneas, com base na classe prevalente. Os nós internos representam decisões condicionadas a um atributo específico, ao passo que as folhas indicam as predições finais. A seleção dos pontos de partição é orientada por métricas de pureza, como o índice de Gini ou a entropia da informação, que quantificam a heterogeneidade das classes em cada nó.

Ao associar várias árvores de decisão em um modelo de ensemble, permite criar os modelos Random Forest (HO, 1995) e o Extra Trees (GEURTS *et al.*, 2006) que

mitigam o sobreajuste e aprimorar a robustez preditiva. No Random Forest, a construção de cada árvore baseia-se em subconjuntos de dados e atributos amostrados aleatoriamente, fomentando a diversidade entre os modelos. Seu princípio fundamental reside na mitigação do risco de *overfitting*, explorando a diversidade introduzida pela amostragem aleatória dos dados e das variáveis preditoras em cada árvore. O resultado final é obtido por meio da agregação dos votos individuais das árvores ou pelo cálculo da média de suas predições. O Extra Trees adota um princípio similar, porém emprega divisões mais aleatórias, o que acelera o processo de treinamento e reduz a correlação entre as árvores.

Ademais, os métodos de Boosting, incluindo o Gradient Boosted Decision Trees (GBDT)(FRIEDMAN, 2001), XGBoost (BHATTACHARYA *et al.*, 2020) e AdaBoost (FREUND e SCHAPIRE, 1997), desenvolvem árvores de decisão de maneira iterativa, onde cada modelo subsequente visa retificar as imprecisões dos modelos precedentes. O GBDT emprega o gradiente da função de perda para direcionar essas correções, ao passo que o XGBoost refina essa abordagem por meio de regularização e otimizações de desempenho. O AdaBoost, por sua vez, aloca pesos incrementais a instâncias que foram classificadas incorretamente, direcionando o modelo para focar em observações de maior complexidade. Tais metodologias estabeleceram-se como *benchmarks* em problemas de classificação de dados tabulares.

3.1.5 Redes Neurais

As Redes Neurais Artificiais constituem uma classe de modelos de aprendizado de máquina, inspirados na arquitetura e funcionamento do cérebro humano. Elas são compostas por camadas de nós interconectados, ou neurônios, que processam e transformam dados de entrada. As principais implementações de redes neurais voltadas para a predição de vitórias em jogos MOBA serão detalhadas a seguir.

O Perceptron Multicamadas(MLP) TAUD e MAS (2018), um modelo de rede neural, consiste em múltiplas camadas de neurônios interconectados, onde cada camada executa uma transformação linear seguida por uma função de ativação não linear, como ReLU ou tanh. Essa arquitetura permite que o MLP aprenda representações de dados intrincadas, sendo um alicerce para abordagens neurais contemporâneas em tarefas de classificação e regressão.

As Redes Neurais Recorrentes foram concebidas para o processamento de dados sequenciais, contudo, enfrentam desafios inerentes como o desvanecimento e a explosão de gradientes. Para mitigar essas dificuldades, arquiteturas aprimoradas foram propostas, como a Long Short-Term Memory (LSTM) (HOCHREITER, 1991) e a Gated Recurrent Unit (GRU) (CHO *et al.*, 2014). Apesar de suas vantagens, essas arquiteturas especializadas que empregam mecanismos de "portas" para gerenciar

o fluxo de informações e manter dependências de longo prazo, podem apresentar um custo computacional de treinamento mais elevado e ser superadas por modelos baseados em atenção em tarefas que envolvem sequências de dados extremamente longas.

3.1.6 Transformer

Os Transformers representam uma arquitetura de aprendizado de máquina de significativa influência recente, tendo sido introduzidos por VASWANI *et al.* (2017). Inicialmente desenvolvidos para tradução automática, sua aplicabilidade foi expandida para abranger visão computacional, séries temporais e, mais recentemente, dados tabulares. A inovação central do Transformer reside na substituição de mecanismos recorrentes e convolucionais pelo *self-attention*, permitindo interações diretas entre todos os elementos de entrada e a captura eficiente de dependências de longo alcance.

No Transformer, a entrada é representada como uma sequência de vetores $X = \{x_1, x_2, \dots, x_n\}$, os quais são projetados em três espaços distintos: consultas (Q), chaves (K) e valores (V). A atenção escalada por produto interno é calculada pela fórmula:

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \quad (3.1)$$

onde d_k representa a dimensão dos vetores de chave. Este mecanismo gera combinações ponderadas dos valores, onde os pesos refletem as relações de similaridade entre consultas e chaves.

Para aprimorar a expressividade, o Transformer emprega múltiplas cabeças de atenção que processam paralelamente subespaços distintos dos vetores de entrada, seguidas por conexões residuais, normalização por camadas e camadas totalmente conectadas. Adicionalmente, como não há uma noção implícita de ordem sequencial, codificações posicionais são adicionadas aos vetores de entrada, permitindo que o modelo distinga a posição relativa dos elementos.

A arquitetura Transformer é construída através do empilhamento desses blocos de atenção e *feed-forward*, que formam os codificadores e decodificadores. Essa estrutura modular permite sua adaptação a diferentes domínios de dados, consolidando-o como o estado da arte em diversas tarefas.

FT-Transformer

Apesar do sucesso expressivo do Transformer em tarefas de manipulação de texto e análise de imagens, sua adaptação direta para dados tabulares apresenta desafios.

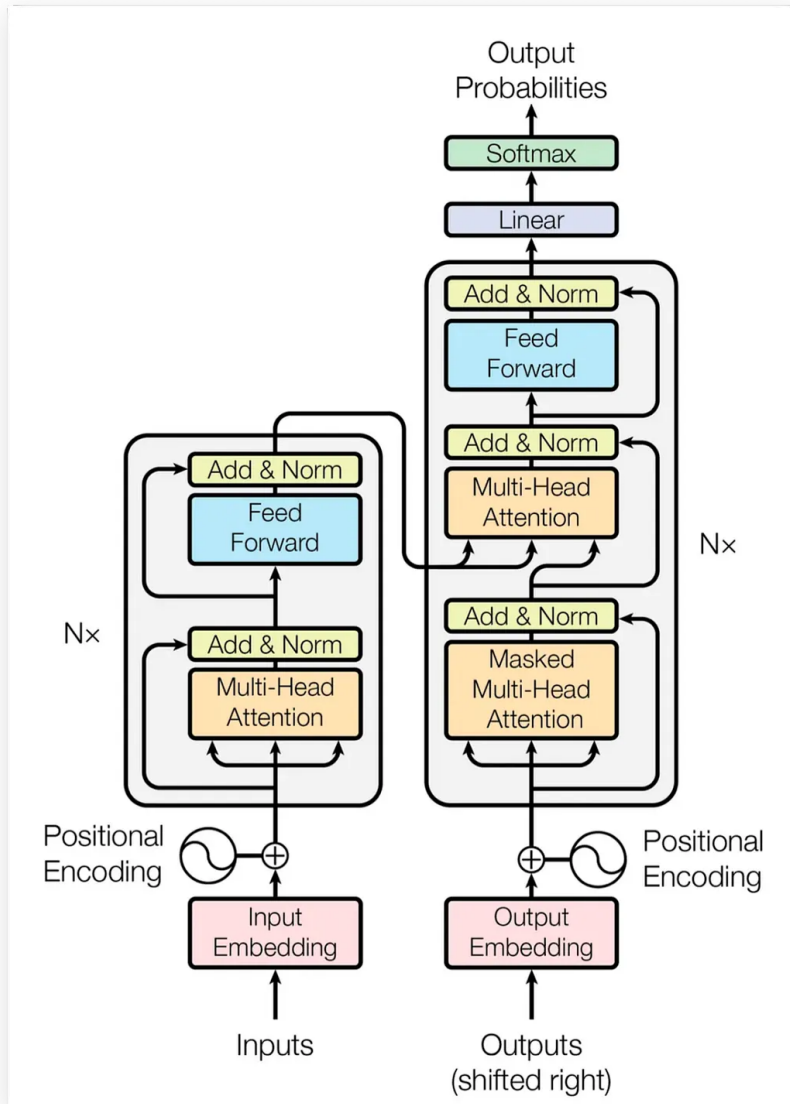


Figura 3.1: Arquitetura do Transformer. Fonte: VASWANI *et al.* (2017).

Isso se deve à ausência de uma ordem intrínseca entre as colunas, bem como à natureza mista dos atributos, que podem ser tanto numéricos quanto categóricos. Para solucionar essas limitações, GORISHNIY *et al.* (2021) introduziram o FT-Transformer, uma adaptação do Transformer projetada especificamente para dados tabulares. Contudo, até o presente momento, a literatura não registra a aplicação desta metodologia ao problema de predição em jogos do gênero MOBA.

A abordagem fundamental do FT-Transformer consiste em tratar cada atributo como um *token* individual. Atributos numéricos são convertidos em representações vetoriais densas através de camadas lineares, enquanto atributos categóricos são mapeados para *embeddings* aprendidos, de forma análoga ao seu uso em processamento de linguagem natural. Adicionalmente, um *embedding* posicional específico é atribuído a cada coluna, permitindo que o modelo discerne entre variáveis distintas.

Uma formulação simplificada do processo de tokenização é apresentada a seguir:

$$z_j = \text{Embed}(x_j) + e_j, \quad j = 1, \dots, m, \quad (3.2)$$

onde x_j é o valor do atributo j , $\text{Embed}(\cdot)$ é a função de *embedding* (linear para numéricos ou vetorial para categóricos), e e_j é o *embedding* da coluna correspondente.

Um *token* especial [CLS] é empregado para consolidar a informação proveniente dos atributos restantes através das camadas de atenção, culminando na predição da classe ou valor alvo pelo modelo. Essas escolhas posicionam o modelo de forma competitiva em relação a métodos baseados em árvores de decisão, superando-os em alguns cenários para tarefas de classificação e regressão em dados de alta dimensionalidade.

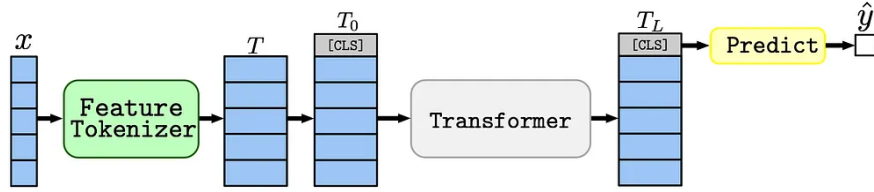


Figura 3.2: Arquitetura geral do FT-Transformer. Fonte: GORISHNIY *et al.* (2021).

Concluindo, o FT-Transformer capitaliza as vantagens do Transformer, ao mesmo tempo que aborda as desvantagens inerentes à sua aplicação em conjuntos de dados tabulares. Essa adaptação o torna especialmente adequado para desafios preditivos em jogos MOBA, onde a presença de atributos numéricos e categóricos é proeminente.

3.1.7 TSSTN

A Two-Stage Spatial-Temporal Network(TSSTN) (YANG *et al.*, 2022b) foi desenvolvida para apreender concomitantemente as interdependências espaciais e temporais inerentes a dados de séries temporais complexas. Sua arquitetura em duas fases integra a extração de padrões espaciais locais com a modelagem sequencial global, tornando-a particularmente apta à análise de partidas em jogos digitais, onde a dinâmica temporal das ações desempenha um papel crucial. Apesar de seu potencial, a aplicação da TSSTN permanece restrita a investigações pontuais.

3.2 Modelos generativos

3.2.1 Naive Bayes

O classificador Naive Bayes (RISH *et al.*, 2001) é um modelo probabilístico que opera sob a premissa de independência condicional entre os atributos, dado a classe. Esta simplificação metodológica resulta em alta eficiência e facilidade de implementação, viabilizando seu uso em grandes volumes de dados. Não obstante a suposição de independência, o Naive Bayes demonstra desempenho satisfatório em tarefas de classificação de texto, servindo como um ponto de referência em estudos comparativos.

3.2.2 Factorization Machines

As Máquinas de Fatoração (RENDLE, 2010) foram introduzidas com o propósito de modelar interações entre atributos em cenários de alta dimensionalidade. Sua formulação probabilística possibilita a decomposição de interações de segunda ordem em produtos escalares de vetores latentes, o que se mostra particularmente vantajoso para atributos categóricos com alta cardinalidade. Tais características consolidam as Máquinas de Fatoração como uma técnica eficiente para tarefas como sistemas de recomendação e predição em dados esparsos.

3.2.3 Modelos de Ranking

Modelos de ranking constituem uma classe de abordagens generativas que visam estimar as habilidades latentes de jogadores ou equipes a partir dos resultados de confrontos. A premissa básica é que cada entidade possui um parâmetro de habilidade subjacente que influencia diretamente a probabilidade de vitória em uma partida.

O modelo Bradley–Terry (BRADLEY e TERRY, 1952), uma formulação estatística para confrontos em pares, estima a probabilidade de sucesso de uma entidade sobre outra com base em suas habilidades latentes. Esta abordagem é aplicada para modelar resultados de partidas diretas em contextos esportivos e de jogos eletrônicos.

O TrueSkill (HERBRICH *et al.*, 2006), uma extensão bayesiana do modelo Bradley–Terry, foi desenvolvido para ambientes dinâmicos. Ele aprimora o ranqueamento em jogos online competitivos ao atualizar iterativamente as distribuições de habilidade dos jogadores com base em novos resultados observados.

3.2.4 Outros modelos

Para além das metodologias generativas convencionais, têm sido investigadas abordagens inovadoras para solucionar desafios específicos em competições e sistemas de ranqueamento no domínio dos jogos eletrônicos, conforme detalhado a seguir.

O OptMatch (GONG *et al.*, 2020) é uma metodologia que emprega correspondência probabilística para refinar comparações entre grupos ou jogadores. Embora menos frequente, oferece uma perspectiva valiosa para problemas específicos de equilíbrio competitivo.

O NeuralAC (GU *et al.*, 2021) introduz uma abordagem generativa para modelagem de competições, utilizando redes neurais para capturar distribuições de habilidade e interações complexas. Embora sua aplicação seja ainda emergente, sinaliza uma tendência crescente no uso de arquiteturas neurais para tarefas de ranqueamento.

Capítulo 4

Método Proposto

A Revisão da literatura demonstra uma lacuna no estudo do método Transformer para predição de vitória de jogos MOBA durante a partida. Até o momento, as únicas pesquisas que empregam o método seriam SMERDOV *et al.* (2021), LEE *et al.* (2022), TIAN *et al.* (2022) e YANG *et al.* (2023). Entretanto SMERDOV *et al.* (2021), TIAN *et al.* (2022) e YANG *et al.* (2023) não realizam a otimização de hiperparâmetros utilizados pelo Transformer. Enquanto LEE *et al.* (2022) foca na predição durante a fase de seleção de personagens, implementando um sistema de recomendação para a escolha de personagens com maior probabilidade de vitória. Neste contexto, o presente estudo conduziu experimentos com o objetivo de investigar a aplicação do modelo Transformer, com otimização de hiperparâmetros, na predição dos resultados de partidas de jogos do tipo MOBA durante sua execução.

O Transformer (VASWANI *et al.*, 2017) é uma arquitetura de rede neural que se distingue pelo uso exclusivo de mecanismos de atenção, em vez de abordagens baseadas em recorrência ou convolução. Essa arquitetura demonstra capacidade de identificar relações de dependência entre elementos dentro de uma sequência, independentemente da distância que os separa. Ademais, constatou-se a possibilidade de estruturar os dados em formato tabular, o que viabilizou a aplicação conjunta do modelo Transformer e de sua variação FT-Transformer (GORISHNIY *et al.*, 2021). A escolha pelo FT-Transformer justifica-se por se tratar de uma adaptação do Transformer especificamente desenvolvida para lidar com dados tabulares, apresentando resultados superiores aos obtidos pelo Transformer original nesse tipo de tarefa.

A seção a seguir detalha o método proposto para a investigação, delineando as etapas que estruturam o processo de análise, com objetivo de fornecer uma visão geral da lógica seguida, desde a coleta de dados até a avaliação dos modelos. Isso visa demonstrar a sequência metodológica adotada para garantir a consistência e transparência da pesquisa.

4.1 Visão Geral da Metodologia

A Figura 4.1 ilustra o fluxograma metodológico adotado nesta investigação. O propósito é compilar de maneira organizada as fases cruciais que abrangem a seleção e o pré-processamento dos dados até a comparação avaliativa dos modelos estudados. Cada seção do diagrama sinaliza um componente essencial do procedimento, que será detalhado de forma geral nas seções subsequentes. A descrição minuciosa de cada fase, compreendendo implementações, algoritmos, conjuntos de dados e os resultados alcançados, será apresentada no Capítulo 5.

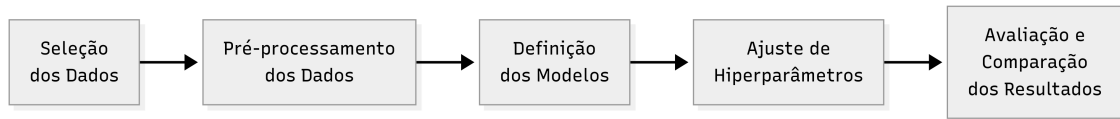


Figura 4.1: Fluxograma da metodologia proposta neste trabalho.

4.1.1 Seleção dos Dados

A etapa inicial da metodologia envolve a seleção das bases de dados que fornecerão o alicerce para esta investigação, as quais podem ser desenvolvidas para este estudo ou obtidas a partir da literatura. Na literatura, os conjuntos de dados empregados em estudos sobre jogos MOBA são, em grande parte, estruturados a partir de diferentes fontes: torneios públicos, bases disponibilizadas oficialmente pelas empresas desenvolvedoras ou ainda por meio de APIs desenvolvidas pela comunidade de jogadores. Pesquisadores independentes e jogadores entusiastas também desempenham papel relevante nesse contexto, ao desenvolver ferramentas para a coleta sistemática de dados em tempo real ou de históricos de partidas, contribuindo significativamente para o aumento da disponibilidade e diversidade dos dados utilizados em investigações científicas.

A proeminência de jogos do gênero MOBA, como *Dota 2* e *League of Legends*, juntamente com a disponibilização de dados por suas empresas desenvolvedoras, fomentou a existência de um vasto acervo de bases de dados acessíveis à comunidade acadêmica. Não obstante, a comparação direta entre estudos permanece restrita, visto que uma parcela reduzida de trabalhos compartilha publicamente os conjuntos de dados utilizados em seus experimentos. Essa conjuntura resulta em uma diversidade de bases de dados, frequentemente não compartilhadas, o que compromete a replicabilidade e limita a consolidação de achados na área.

A formulação da seleção de dados nesta investigação deve, portanto, priorizar tanto a qualidade e a relevância das informações quanto a acessibilidade dos conjuntos de dados pertinentes. A uniformização de protocolos de pesquisa é um elemento

benéfico, visto que facilita a comparação entre distintos estudos, mitigando as restrições decorrentes da variabilidade nos conjuntos de dados empregados. Para assegurar que os achados desta pesquisa possuam validade externa e contribuam para o avanço do conhecimento na área, a seleção das bases de dados é cuidadosamente alinhada aos objetivos metodológicos estabelecidos. Dessa forma, duas bases de dados do jogo *League of Legends* foram selecionadas da literatura para esta pesquisa, as quais serão detalhadas no próximo capítulo.

4.1.2 Pré-processamento dos Dados

Após a seleção das bases, o pré-processamento dos dados é realizado para converter os dados brutos das bases de dados em formatos adequados para análise, minimizando inconsistências e vieses. As tarefas comuns incluem a normalização de escalas numéricas, o manejo de valores ausentes, a exclusão de atributos redundantes e a definição dos atributos categóricos, a fim de assegurar que os modelos de aprendizado de máquina recebam dados consistentes e comparáveis, fortalecendo a robustez das análises. Para ambas as bases de dados selecionadas necessitou apenas excluir os atributos redundantes e definir os atributos categóricos para o experimento com o modelo FT-Transformer.

4.1.3 Definição dos Modelos

A etapa de definição dos modelos envolve a seleção das arquiteturas de aprendizado de máquina a serem testadas. Nesta investigação, prioriza-se o Transformer e sua adaptação para dados tabulares, o FT-Transformer, por serem os alvos de pesquisa desse trabalho. Adicionalmente, foram considerados os principais métodos discriminativos e generativos da literatura para servirem de base comparativa e de validação para os resultados obtidos, permitindo uma avaliação exaustiva do desempenho dos modelos Transformer. Além disso, o método XGBoost foi selecionado por obter resultados relevantes para problemas que envolvem dados tabulares, no intuito de servir como comparação para o modelo FT-transformer.

4.1.4 Ajuste de Hiperparâmetros

Modelos que utilizam mecanismos de atenção, como o Transformer e o FT-Transformer, possuem uma variedade de hiperparâmetros que impactam significativamente sua performance. A otimização desses hiperparâmetros é crucial na metodologia, assegurando que os modelos sejam avaliados em condições otimizadas e equitativas. Este processo envolve não apenas a seleção de valores específicos, mas também a definição de estratégias de busca, métricas de avaliação e métodos

de controle de complexidade, garantindo que os resultados apresentados reflitam a capacidade intrínseca dos modelos, e não deficiências de configuração. Em virtude disso, realizou uma busca em grade dos hiperparâmetros do FT-transformer e aplicou avaliação independente dos hiperparâmetros do Transformer, enquanto para os demais modelos, utilizou-se os valores padrão de seus hiperparâmetros.

4.1.5 Avaliação e Comparação dos Resultados

A fase final do estudo concentra-se na análise crítica dos resultados obtidos com os modelos selecionados, contemplando tanto o processo de busca dos hiperparâmetros quanto a avaliação final das predições realizadas. As métricas de avaliação predefinidas são aplicadas de forma sistemática para aferir a eficácia de cada modelo e compará-los em distintos contextos, além de examinar as limitações inerentes aos modelos e identificar tendências recorrentes, fornecendo bases sólidas para inferências sobre o potencial das arquiteturas de atenção na previsão de resultados em jogos do gênero MOBA.

Capítulo 5

Experimentos

Neste capítulo, detalhamos a implementação da metodologia delineada no Capítulo 4, especificando os procedimentos experimentais. Iniciamos com a seleção das bases de dados e seu respectivo pré-processamento, seguido pela definição dos modelos e o ajuste de seus hiperparâmetros, que englobam o modelo Transformer e sua variante para dados tabulares, o FT-Transformer. Concluímos com a exposição e análise dos resultados, permitindo uma comparação estruturada das abordagens investigadas e fornecendo a base para a discussão crítica subsequente.

5.1 Seleção e Pré-processamento dos dados

Para realizar os experimentos desta pesquisa, duas bases de dados do jogo *League of Legends*, empregadas em outros estudos da literatura, foram selecionadas devido à sua capacidade de análise da vitória durante a partida e acessibilidade. A seguir, serão apresentadas as duas bases de dados selecionadas: a primeira, que abrange jogos ranqueados nos quais jogadores participam de partidas classificatórias por meio de um sistema de emparelhamento automático; e a segunda, que consiste em jogos profissionais disputados em torneios oficiais por equipes de alta performance.

5.1.1 Partidas Rankeadas

A primeira base de dados (MA, 2021) compreende um total de 9879 jogos distintos, cada um dos quais é caracterizado por 40 atributos que fornecem informações sobre as equipes após os 10 minutos iniciais de jogo. As partidas que integram a base de dados são caracterizadas por um nível de ranqueamento elevado, assegurando um nível técnico mais consistente e minimizando a influência de comportamentos de jogadores inexperientes. Desses, 38 atributos são igualmente distribuídos entre as duas equipes, enquanto os 2 atributos restantes referem-se à identificação da partida e à determinação da equipe vencedora. O atributo alvo dos modelos de predição,

denominado blueWins, indica a vitória da equipe azul e é representado por um valor binário. A classe representando a derrota da equipe azul(0) contém 4.949 instâncias (50,0962%), enquanto a classe a vitória(1) possui 4.930 instâncias (49,9038%). Essa pequena diferença indica que a base de dados pode ser considerada balanceada e dispensa técnicas de tratamento de desbalanceamento de classes. Adicionalmente, a ausência de amostras com atributos nulos e corrompidos elimina a exigência de tratamento de erros e valores ausentes na base.

A literatura existente propõe diversas abordagens para a seleção de atributos nesta base de dados, visando a construção de modelos de predição de vitória. Entretanto, algumas metodologias empregadas em determinados estudos podem comprometer a capacidade de generalização dos modelos. No estudo conduzido por SHEN (2022), os atributos foram inicialmente ajustados pela diferença entre os times e, em seguida, filtrados por mapa de calor, resultando na seleção de apenas cinco atributos. Entretanto, os resultados experimentais desta pesquisa revelaram que o desempenho obtido com este conjunto restrito de atributos foi inferior ao alcançado com os valores originais, o que sugere que o processo de seleção pode ter levado à exclusão de informações relevantes.

LAUREL *et al.* (2023) adotam uma metodologia de seleção de atributos que se fundamenta em critérios subjetivos, carecendo de uma justificativa técnica aprofundada. Uma abordagem metodológica similar manifesta-se no trabalho conduzido por OMAR *et al.* (2024), o qual explora exclusivamente os atributos da equipe azul, depois aplica a seleção por meio do mapa de calor. Em virtude disso, o experimento perde a oportunidade de comparar a performance entre as equipes para cada atributo.

A pesquisa ZHANG (2021) realiza a seleção dos atributos por meio da análise da relevância dos atributos na predição de vitória, utilizando um mapa de calor para identificar os atributos considerados pertinentes. Contudo, exclui a quantidade de ouro obtida pelas equipes como medida de treinamento e teste dos modelos construídos, o que demonstra uma ação contrária a literatura, já que outras pesquisas têm confirmado a relevância do montante total de ouro acumulado pelas equipes na determinação do resultado em jogos MOBA(GAINA e NORDMOEN, 2018; YANG *et al.*, 2022b).

Em contraposição às metodologias de seleção dos atributos adotadas em estudos prévios, o presente estudo incluiu o máximo de atributos independentes e que não identificam jogadores, equipes ou partidas. Tal decisão visa evitar a exclusão de variáveis que, em conjunto, possam aprimorar o desempenho preditivo dos modelos, enquanto simultaneamente previne potenciais vieses de identificação. Todos os atributos selecionados para o experimento da base de partidas ranqueadas estão listados na tabela 5.1.

No presente estudo, a necessidade de identificar atributos categóricos surgiu exclusivamente para a aplicação do modelo FT-Transformer. Essa exigência advém da arquitetura do modelo, que distingue explicitamente entre variáveis numéricas e categóricas durante o processo de representação em *embeddings*. Em virtude disso, os atributos subsequentes, extraídos do conjunto de dados de partidas ranqueadas, foram classificados como categóricos por representarem eventos binários ou contagens discretas de objetivos:

- `blueFirstBlood` – ocorrência do primeiro abate pelo time azul (binário);
- `blueDragons` – número de dragões abatidos pelo time azul;
- `redDragons` – número de dragões abatidos pelo time vermelho;
- `blueHeralds` – número de Arautos abatidos pelo time azul;
- `redHeralds` – número de Arautos abatidos pelo time vermelho;
- `blueTowersDestroyed` – número de torres destruídas pelo time azul;
- `redTowersDestroyed` – número de torres destruídas pelo time vermelho.

5.1.2 Partidas Profissionais

A segunda base de dados (EPHRON—) utilizada neste estudo é composta por um conjunto de partidas profissionais do jogo League of Legends, realizadas entre os anos de 2015 e 2017. Essa base contempla confrontos provenientes de ligas regionais, como a liga norte-americana, europeia, sul-coreana e a liga brasileira, além de torneios internacionais de destaque, como o Campeonato Mundial e o Torneio Internacional do Meio de Temporada. Cada entrada nesta base de dados representa uma partida distinta, incorporando variáveis contextuais e estatísticas dos jogadores e equipes de toda a duração da partida.

Dois estudos empregaram esta base de dados; o mais antigo, conduzido por SILVA *et al.* (2018), não especifica os atributos utilizados e relata que a base de dados original continha 53 atributos, em contraste com os 57 listados na base. Além disso, realiza a investigação de predição de vitória considerando os dados no intervalo entre 10 e 15 minutos de jogo. Enquanto HITAR-GARCÍA *et al.* (2023) o trabalho realiza a predição considerando todos os dados coletados durante a partida inteira.

No presente estudo, optou-se por selecionar exclusivamente atributos que representassem informações quantitativas e estratégicas das partidas, preservando a independência entre as variáveis e evitando dados que pudessem identificar diretamente jogadores, equipes ou partidas específicas. Assim, foram mantidos indicadores

Tabela 5.1: Atributos selecionados para os experimentos do conjunto de dados de partidas ranqueadas de League of Legends.

Atributo	Descrição
blueWins	Indica se o time azul venceu (1) ou não (0).
blueWardsPlaced	Número de sentinelas colocadas pelo time azul.
blueWardsDestroyed	Número de sentinelas inimigas destruídas pelo time azul.
blueFirstBlood	Indica se o time azul realizou o primeiro abate da partida.
blueKills	Número de abates realizados pelo time azul.
blueAssists	Número de assistências realizadas pelo time azul.
blueDragons	Número de dragões abatidos pelo time azul.
blueHeralds	Número de Arautos abatidos pelo time azul.
blueTowersDestroyed	Número de torres destruídas pelo time azul.
blueTotalGold	Total de ouro acumulado pelo time azul.
blueAvgLevel	Nível médio dos campeões do time azul.
blueTotalExperience	Experiência total acumulada pelo time azul.
blueTotalMinionsKilled	Total de minions abatidos pelo time azul (CS).
blueTotalJungleMinionsKilled	Total de monstros da selva abatidos pelo time azul.
redWardsPlaced	Número de sentinelas colocadas pelo time vermelho.
redWardsDestroyed	Número de sentinelas inimigas destruídas pelo time vermelho.
redKills	Número de abates realizados pelo time vermelho.
redAssists	Número de assistências realizadas pelo time vermelho.
redDragons	Número de dragões abatidos pelo time vermelho.
redHeralds	Número de Arautos abatidos pelo time vermelho.
redTowersDestroyed	Número de torres destruídas pelo time vermelho.
redTotalGold	Total de ouro acumulado pelo time vermelho.
redAvgLevel	Nível médio dos campeões do time vermelho.
redTotalExperience	Experiência total acumulada pelo time vermelho.
redTotalMinionsKilled	Total de minions abatidos pelo time vermelho (CS).
redTotalJungleMinionsKilled	Total de monstros da selva abatidos pelo time vermelho.

de desempenho geral das equipes além dos campeões utilizados por cada jogador. Os atributos empregados fornecem um contexto competitivo detalhado, excluindo variáveis irrelevantes para a predição, como duração da partida e banimento de personagens pelas equipes, além de atributos derivados de outros já incluídos. Além de excluir atributos susceptíveis a introduzir viés no modelo por poderem identificar as melhores equipes e jogadores. A tabela 5.2 apresenta os atributos selecionados para o experimento com a base de dados de partidas profissionais.

A variável alvo de predição de vitória ‘bResult’ indica o resultado da partida para a equipe azul. Ela é representada por um valor binário, com 3.474 instâncias na classe 0 (derrota) e 4.146 na classe 1 (vitória). Essa distribuição, correspondendo a 45,6% para a classe 0 e 54,4% para a classe 1, evidencia um leve desbalanceamento de classes. No entanto, essa diferença é considerada pequena, minimizando o risco

de viés nos modelos preditivos e por isso dispensa a necessidade de técnicas de balanceamento. A inexistência de instâncias com atributos nulos ou corrompidos dispensa a necessidade de procedimentos de tratamento para erros e valores ausentes na base de dados.

Para a análise preditiva do resultado de partidas, o ponto de observação foi estabelecido no 15^o minuto de jogo. Nesse intervalo temporal, os atributos referentes à acumulação de ouro por cada equipe (goldblue e goldred) foram processados para refletir o total acumulado até aquele momento, selecionando-se o 15^o registro de cada lista. Enquanto, para os atributos bKills, bTowers, bInhibs, bDragons, bBarons, bHeralds, rKills, rTowers, rInhibs, rDragons, rBarons e rHeralds, que registram a ocorrência de eventos ao longo da partida, foram alterados. Para cada um deles, calculou-se a quantidade total de eventos ocorridos até o 15^o minuto.

No que se refere aos atributos que indicam os campeões de *League of Legends* selecionados pelos jogadores, eles foram modificados para representá-los por meio de identificadores numéricos únicos ao invés de seus nomes. Essa conversão foi realizada para otimizar o processamento pelo modelo de aprendizado de máquina, minimizando a complexidade do tratamento de variáveis textuais e aprimorando a eficiência computacional. Para tal, um mapeamento entre os nomes dos campeões e identificadores inteiros foi criado, utilizando um conjunto ordenado de todos os campeões presentes nos dados. Esse mapeamento foi aplicado a cada instância, substituindo os nomes dos campeões por seus respectivos códigos numéricos. O dicionário de correspondência foi armazenado para permitir sua reutilização futura ou aplicação em novos dados.

Para a base de partidas profissionais, a distinção entre atributos numéricos e categóricos foi estabelecida para viabilizar a aplicação do FT-Transformer. Os atributos categóricos identificados consistem em marcadores discretos para a destruição de estruturas ou a obtenção de objetivos neutros:

- **bTowers** – torres destruídas pelo time azul;
- **bInhibs** – inibidores destruídos pelo time azul;
- **bDragons** – dragões abatidos pelo time azul;
- **bBarons** – Barões Nashor abatidos pelo time azul;
- **bHeralds** – Arautos abatidos pelo time azul;
- **rTowers** – torres destruídas pelo time vermelho;
- **rInhibs** – inibidores destruídos pelo time vermelho;
- **rDragons** – dragões abatidos pelo time vermelho;

- **rBarons** – Barões Nashor abatidos pelo time vermelho;
- **rHeralds** – Arautos abatidos pelo time vermelho.

5.2 Definição dos Modelos

O foco desse trabalho é o estudo do modelo Transformer por meio de duas implementações, a primeira desenvolvida com *TensorFlow/Keras*, seguindo a arquitetura do trabalho VASWANI *et al.* (2017); já a outra seguindo o trabalho GORISHNIY *et al.* (2021). Para fins de avaliação comparativa dos modelos Transformer, foram implementados os métodos de regressão logística, Random Forest, classificador Naive Bayes e Extreme Gradient Boosting (XGBoost). A regressão logística e a Random Forest foram selecionadas por personificarem modelos discriminativos, enquanto o classificador Naive Bayes foi incluído como exemplo de modelo generativo (NG e JORDAN, 2001), por serem os principais métodos listados na tabela 2.1 que encaixam nessas classificações. Por fim, o XGBoost foi escolhido por sua eficiência no tratamento de dados tabulares (SHWARTZ-ZIV e ARMON, 2022).

5.3 Ajuste de Hiperparâmetros

Nesta seção, serão detalhados os modelos de referência para esta pesquisa, incluindo sua implementação e configuração de hiperparâmetros.

5.3.1 Transformer

O modelo Transformer, foco central deste estudo, foi adaptado para a tarefa de prever o resultado de partidas de jogos MOBA. Originalmente desenvolvido para o campo do processamento de linguagem natural (VASWANI *et al.*, 2017), o modelo foi remodelado para efetuar a classificação binária de dados tabulares. A implementação foi feita com *TensorFlow/Keras*, criando camadas que adaptam o Transformer para dados tabulares. Isso envolveu tratar os atributos como uma sequência ordenada, similar ao processamento de sentenças em linguagem natural.

Em termos específicos, cada registro tabular foi expresso como uma sequência ordenada de seus atributos constituintes. Essa representação sequencial possibilitou a aplicação do mecanismo de atenção pelo modelo ao longo dos atributos, permitindo a interpretação das interdependências existentes entre eles de forma contextualizada. De forma análoga ao processo de aprendizado das relações entre palavras em uma frase, realizado pelo Transformer tradicionalmente empregado para prever o termo

seguinte em uma sequência, no presente estudo, o termo subsequente da sequência representa a equipe vencedora.

Arquitetura do transformer

A arquitetura do modelo pode ser sumarizada em três seções: primeiramente, a camada de *embedding*; subsequentemente, o bloco Transformer adaptado; e, finalmente, camadas densas para classificação, conforme detalhado a seguir.

A camada de *embedding* é responsável por transformar os dados de entrada (discretos e categóricos) em representações vetoriais densas, que capturam as relações semânticas e posicionais entre as variáveis. Para realizar essa transformação, realiza duas operações: o *embedding* de *tokens*, que mapeia cada valor inteiro para um vetor de dimensão predefinida durante o treinamento, e o *embedding* posicional, que codifica explicitamente a posição dos atributos na sequência, permitindo que o modelo discrimine atributos com representações idênticas, mas posições distintas. A saída da camada é obtida pela soma dos dois *embeddings*, integrando, assim, informação semântica e posicional.

Neste trabalho, a arquitetura adotada integra um bloco Transformer, customizado para a análise de dados tabulares. No início do bloco transformer aplica a *multi-headed attention*, implementado de acordo com VASWANI *et al.* (2017), essa operação permite que o modelo aprenda diferentes relações entre os atributos dos dados, utilizando múltiplas cabeças de atenção em paralelo. A saída da *multi-headed attention* é submetida a uma camada de *Dropout* (taxa de 10%), que funciona como um regularizador. Em seguida, aplica-se uma conexão residual, somando essa saída à entrada original do bloco, o que auxilia na preservação de informações e otimiza o fluxo de gradientes. A normalização em camadas é aplicada à soma resultante da etapa anterior, o que contribui para a estabilidade do treinamento e para uma convergência mais rápida do modelo. A sequência subsequente envolve a aplicação de uma rede neural densamente conectada, compreendendo duas camadas densas, a primeira camada emprega a função de ativação *ReLU*, enquanto a segunda utiliza uma ativação linear. E no fim do bloco transformer realiza mais uma camada de *Dropout* (taxa de 10%) e, subsequentemente, agrega à entrada anterior da rede neural densamente conectada para ser normalizado por camada.

Após o processamento na camada do bloco Transformer, os vetores resultantes são agregados por meio da operação de *Global Average Pooling*, o que resulta em um vetor de tamanho fixo, independente do comprimento da sequência de entrada. Subsequentemente, aplicam-se duas camadas densas, sendo a primeira uma camada intermediária, composta por 128 unidades e função de ativação *ReLU*, juntamente com uma taxa de *dropout* de 20%. A segunda camada é a camada final, composta por duas unidades e função de ativação *softmax*, a fim de retornar o resultado da

classificação binária representando que time venceu.

Configuração de Treinamento e Hiperparâmetros do Transformer

Para a compilação do modelo, foi empregado o otimizador Adam, com uma taxa de aprendizado estabelecida em 0.001. A função de perda utilizada foi a entropia cruzada esparsa, e a métrica de acurácia foi monitorada durante o processo de otimização. O treinamento do modelo foi realizado em 100 épocas, com um tamanho de lote de 64.

Os hiperparâmetros do modelo Transformer foram definidos da seguinte forma: a dimensão do vetor de *embedding* dos *tokens* foi estabelecida como o menor valor possível, desde que fosse superior ao número de atributos na base de dados e, simultaneamente, múltiplo do número de cabeças de atenção, resultando em 32 para ambas as bases de dados. O número de cabeças de atenção paralelas foi variado entre 4, 8 e 16. A dimensão da camada intermediária da rede neural dentro do bloco Transformer apresentou uma influência mínima sobre os resultados, por isso, para os experimentos, ela foi mantida em 4. Ademais, foram avaliadas quatro estratégias de *embedding* de posição. A primeira é a organização alternada, onde os atributos das equipes são intercalados, de modo que os valores de cada variável para cada equipe se sucedem. A segunda organização agrupa os atributos de cada equipe sequencialmente. A terceira consiste na disposição aleatória dos atributos, sem ordenação predefinida. E por fim a quarta é a ausência de *embedding* de posição, sem informação explícita de posição.

5.3.2 FT-Transformer

O FT-Transformer, proposto por GORISHNIY *et al.* (2021), adapta a arquitetura Transformer para dados tabulares. Diferente do Transformer tradicional, que processa sequências ordenadas, o FT-Transformer foi desenvolvido para modelar dados heterogêneos, com variáveis categóricas e numéricas sem ordenação inerente.

Para a realização dos experimentos neste estudo, o modelo FT-Transformer foi implementado por meio da biblioteca *PyTorch Tabular*. De forma similar ao modelo Transformer, o vetor de *embedding* foi dimensionado em 32, e o treinamento do modelo foi conduzido ao longo de 100 épocas, empregando um tamanho de lote de 64. Uma vez que o FT-Transformer dispensa a codificação posicional, a avaliação de diferentes posicionamentos das colunas foi considerada desnecessária.

Com o propósito de refinar o modelo FT-Transformer para a análise de dados tabulares em questão, conduziu-se um processo de otimização de hiperparâmetros via busca em grade. Tal técnica envolve a avaliação sistemática de diversas combinações predefinidas de valores de hiperparâmetros, visando identificar a configuração

que oferece o desempenho superior em relação à métrica de validação. Foram selecionados dois hiperparâmetros para a busca em grade: o número de cabeças no mecanismo de atenção multi-cabeça, cujos valores testados foram 4, 8, 16; e o número de blocos de atenção empilhados na rede, com os valores testados de 4, 6, 8.

5.3.3 Regressão Logística

A regressão logística é um modelo estatístico comum para classificação binária. Ele modela a probabilidade de um evento usando uma função sigmoide aplicada a uma combinação linear das variáveis de entrada. Os coeficientes podem ser interpretados como pesos que indicam a influência de cada variável preditora.

No presente estudo, a implementação foi conduzida por meio da biblioteca *Scikit-learn*, com um ajuste para um máximo de 1000 iterações, em decorrência da presença de múltiplos atributos nas bases de dados, o que pode dificultar a convergência do algoritmo em um período de tempo adequado.

5.3.4 Classificador Naive Bayes

O Classificador Naive Bayes é um classificador probabilístico que se baseia no teorema de Bayes, presumindo independência condicional entre os atributos. Apesar dessa simplificação, o modelo destaca-se pela sua eficiência, simplicidade e desempenho competitivo em diversas aplicações, particularmente em conjuntos de dados com alta dimensionalidade.

No presente estudo, foi adotado o modelo Gaussian Naive Bayes, uma das variações mais difundidas do classificador Naive Bayes. Essa versão assume que os atributos contínuos seguem uma distribuição normal dentro de cada classe. A implementação valeu-se da versão padrão do classificador, disponibilizada pela biblioteca *Scikit-learn*, preservando todos os hiperparâmetros em suas configurações originais.

5.3.5 Random Forest

O Random Forest é um método de aprendizado de conjunto que combina múltiplas árvores de decisão para criar uma classificação mais robusta. Ele funciona gerando diferentes subconjuntos de dados e atributos para treinar cada árvore, o que reduz a variância do modelo individual e o risco de sobreajuste.

No contexto desta pesquisa, foram treinadas 150 árvores de decisão, cada uma com uma profundidade máxima limitada a 10 níveis, buscando um equilíbrio entre desempenho e complexidade. Para um controle mais rigoroso da generalização, estabeleceu-se que cada divisão de nó interno requer um mínimo de 10 amostras, e

que cada folha terminal contenha pelo menos 5 amostras. Tais restrições têm como objetivo evitar a criação de árvores excessivamente complexas, mantendo a robustez do modelo em face de dados ruidosos ou desbalanceados. A implementação foi conduzida utilizando a biblioteca *Scikit-learn*.

5.3.6 Extreme Gradient Boosting (XGBoost)

O Extreme Gradient Boosting é uma abordagem de aprendizado de conjunto que utiliza o paradigma de *boosting*, desenvolvendo árvores de decisão de maneira sequencial, onde cada árvore subsequente visa retificar as falhas das anteriores. Sua proeminência se deve à sua elevada acurácia preditiva, eficiência computacional e a incorporação de múltiplos mecanismos de regularização, que mitigam o risco de sobreajuste.

Neste estudo, o modelo foi implementado com a biblioteca XGBoost e configurado para treinar 100 árvores de decisão, com profundidade máxima de 6 níveis e uma taxa de aprendizado de 0.1. Tais parâmetros foram escolhidos com base em recomendações amplamente utilizadas na literatura, equilibrando o tempo de treinamento e a capacidade preditiva.

5.4 Avaliação e Comparação dos Resultados

Diante da ausência de partições pré-definidas para treino e teste nas duas bases de dados, todos os experimentos foram conduzidos mediante a aplicação da técnica de validação cruzada *k-fold*, com $k = 5$. Este procedimento envolve a segmentação do conjunto de dados em k subconjuntos mutuamente exclusivos (*folds*). Em cada iteração, um desses subconjuntos é designado como conjunto de teste, enquanto os subconjuntos remanescentes são empregados para o treinamento, assegurando que todos os exemplos sejam utilizados tanto para treinamento quanto para teste em diferentes rodadas. Todos os modelos foram treinados e avaliados sob as mesmas condições experimentais, utilizando as mesmas partições de treino e teste das duas bases de dados.

Os experimentos foram implementados no ambiente disponibilizado pelo Google Colaboratory (Google Colab), com a seguinte configuração: um processador Intel Xeon CPU @ 2.20 GHz (12 núcleos lógicos, arquitetura *x86_64*, cache de 38,5 MiB, instruções vetoriais avançadas como AVX e AVX512), 83,5 GB de memória RAM, uma GPU NVIDIA A100-SXM4 (40 GB de memória, arquitetura Ampere, CUDA 12.4) e o interpretador Python 3.11.11. As principais bibliotecas utilizadas para a implementação dos modelos e condução dos experimentos foram TensorFlow/Keras, PyTorch Tabular, XGBoost e *Scikit-learn*.

Para realizar uma avaliação comparativa do desempenho dos modelos, foram empregadas as seguintes métricas, amplamente difundidas na literatura: acurácia, revocação (recall), F1-Score e tempo de execução. A acurácia avalia a proporção de predições corretas em relação ao total de instâncias analisadas. A revocação, por sua vez, mede a capacidade do modelo em identificar corretamente as instâncias positivas. O F1-Score, que é a média harmônica entre precisão e revocação, proporciona uma avaliação mais balanceada do desempenho. Adicionalmente, o tempo de execução foi considerado como um critério de avaliação complementar, permitindo a comparação não apenas da qualidade preditiva, mas também da eficiência computacional de cada método. A métrica AUC não foi considerada neste estudo, em virtude das bases de dados utilizadas não apresentarem desbalanceamento exacerbado e a acurácia desempenhar a mesma função.

Tabela 5.2: Atributos selecionados para os experimentos do conjunto de dados de partidas profissionais de League of Legends.

Atributo	Descrição
bResult	Resultado da partida para o Time Azul
goldblue	Valor total de ouro do Time Azul nos 15 minutos da partida.
bKills	Valor total de abates do Time Azul nos 15 minutos da partida.
bTowers	Valor total de torres destruídas pelo Time Azul nos 15 minutos da partida.
bInhibs	Valor total de inibidores destruídos pelo Time Azul nos 15 minutos da partida.
bDragons	Valor total de dragões abatidos pelo Time Azul nos 15 minutos da partida.
bBarons	Valor total de Barões Nashor abatidos pelo Time Azul nos 15 minutos da partida.
bHeralds	Valor total de Arautos do Vale abatidos pelo Time Azul nos 15 minutos da partida.
goldred	Valor total de ouro do Time Vermelho nos 15 minutos da partida.
rKills	Valor total de abates do Time Vermelho nos 15 minutos da partida.
rTowers	Valor total de torres destruídas pelo Time Vermelho nos 15 minutos da partida.
rInhibs	Valor total de inibidores destruídos pelo Time Vermelho nos 15 minutos da partida.
rDragons	Valor total de dragões abatidos pelo Time Vermelho nos 15 minutos da partida.
rBarons	Valor total de Barões Nashor abatidos pelo Time Vermelho nos 15 minutos da partida.
rHeralds	Valor total de Arautos do Vale abatidos pelo Time Vermelho nos 15 minutos da partida.
blueTopChamp	Código do campeão do Time Azul na posição topo.
blueJungleChamp	Código do campeão do Time Azul na posição selva.
blueMiddleChamp	Código do campeão do Time Azul na posição do meio.
blueADCCChamp	Código do campeão do Time Azul na posição atirador (ADC).
blueSupportChamp	Código do campeão do Time Azul na posição suporte.
redTopChamp	Código do campeão do Time Vermelho na posição topo.
redJungleChamp	Código do campeão do Time Vermelho na posição selva.
redMiddleChamp	Código do campeão do Time Vermelho na posição do meio.
redADCCChamp	Código do campeão do Time Vermelho na posição atirador (ADC).
redSupportChamp	Código do campeão do Time Vermelho na posição suporte.

Capítulo 6

Resultados

Este capítulo detalha a análise dos resultados experimentais obtidos com os modelos previamente descritos no Capítulo 4. Inicialmente, abordam-se os experimentos com o modelo Transformer, explorando variações no número de cabeças de atenção e nas estratégias de incorporação de *embedding* de posição. Subsequentemente, são apresentados os resultados do FT-Transformer, com ênfase no processo de calibração de hiperparâmetros mediante validação cruzada aninhada. Finalmente, uma análise comparativa abrangente dos modelos avaliados é realizada, salientando suas respectivas fortalezas, fragilidades e relevância para a predição de resultados em jogos do gênero MOBA.

6.1 Resultados do Transformer

Esta seção apresenta uma avaliação sistemática da aplicação do modelo Transformer às duas bases de dados. O estudo compreendeu duas etapas: a análise do impacto do número de cabeças de atenção no desempenho e a investigação do papel das estratégias de *embedding* de posição na natureza tabular dos atributos. Em ambos os cenários, a validação cruzada foi aplicada utilizando as mesmas bipartições para assegurar a comparabilidade direta entre as configurações testadas. O reuso das bipartições minimiza a variação decorrente do particionamento e possibilita comparações pareadas entre as condições.

Ademais, foram conduzidos testes exploratórios com o objetivo de avaliar o impacto da variação na dimensão dos vetores de *embedding* e na dimensão da camada *feed-forward* interna do Transformer. Contudo, tais modificações não resultaram em alterações significativas nas métricas de desempenho, mantendo-se dentro da margem de variação observada nos experimentos principais. Diante disso, optou-se por focar a análise nos parâmetros que demonstraram maior influência sobre os resultados, como o número de cabeças de atenção e a estratégia de incorporação posicional, em detrimento de um aprofundamento nestas últimas variáveis.

6.1.1 Impacto do número de cabeças de atenção

A investigação inicial consistiu na variação do número de cabeças de atenção entre 4, 8 e 16, mantendo os demais hiperparâmetros constantes. Em ambas as bases de dados, os resultados foram semelhantes: na base de partidas ranqueadas, as acurácias médias registradas foram 70,58%, 70,66% e 70,62%, enquanto na base de partidas profissionais, os valores foram 70,47%, 70,56% e 70,41%, respectivamente. Concluiu-se que 8 cabeças de atenção proporcionaram o melhor desempenho médio em ambos os cenários, embora a diferença em relação às demais configurações tenha sido mínima como demonstrado na Figura 6.1.

Transformer: acurácia por número de cabeças (Ranqueadas vs Profissionais)

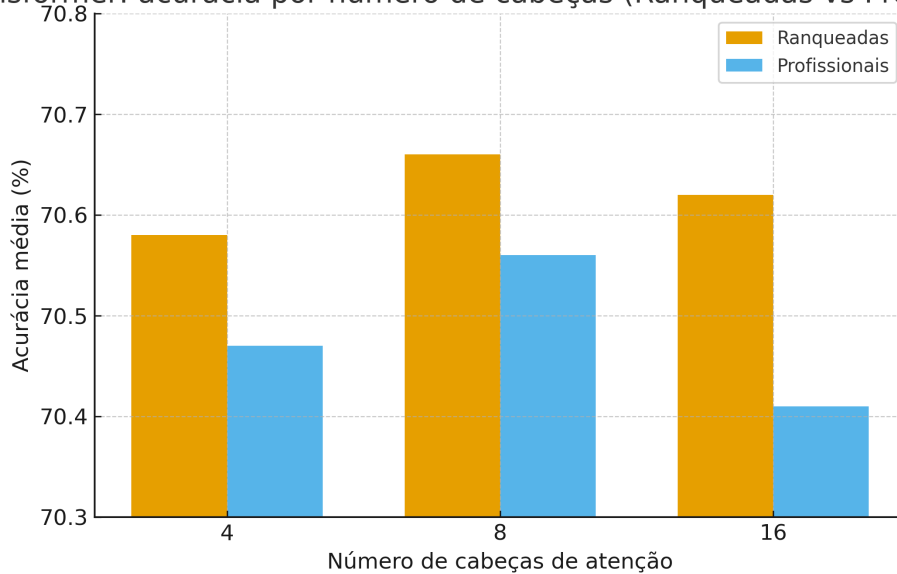


Figura 6.1: A performance média do modelo Transformer, quantificada pela acurácia, foi avaliada em relação ao número de cabeças de atenção em ambas as bases de dados (partidas ranqueadas e profissionais).

Esse comportamento sugere uma relação entre a capacidade de modelagem e a estabilidade do modelo. O aumento de 4 para 8 cabeças de atenção parece fornecer capacidade suficiente para capturar interações relevantes entre os atributos, enquanto o salto para 16 cabeças não resulta em ganho adicional. Isso pode ocorrer devido à fragmentação do espaço de atenção e/ou ao aumento da variância na estimação dos pesos sem um sinal discriminativo adicional correspondente. Em suma, com o conjunto de atributos e o regime de dados utilizados, 8 cabeças de atenção representam um ponto de equilíbrio entre expressividade e robustez.

6.1.2 Impacto do *embedding* de posição

Na etapa subsequente, com o número de cabeças de atenção fixado em 8, foram exploradas quatro abordagens distintas para o *embedding* de posição: alternado,

agrupado, aleatório e a ausência de qualquer *embedding* posicional. As Tabelas 6.1 (ranqueadas) e 6.2 (profissionais) resumizam as médias e os desvios-padrão para as métricas de Acurácia, F1-Score, Revocação e o Tempo de execução associado a cada configuração.

Em termos gerais, todas as estratégias que incorporam informação posicional resultaram em desempenhos muito semelhantes entre si, com variações na ordem de décimos de ponto percentual. Em contraste, a ausência de informação posicional levou a uma queda acentuada no desempenho em ambas as bases de dados. Por exemplo, na base de partidas ranqueadas, a acurácia diminui de aproximadamente 70,1%–70,7% para 53,75% sem a inclusão da posição. Similarmente, na base de partidas profissionais, a acurácia cai de cerca de 69,4%–70,6% para 54,24%. Este achado é consistente com a natureza intrínseca do Transformer: na ausência de informações posicionais, o mecanismo de auto atenção tende a tratar os atributos de forma permutável, desconsiderando a ordem sequencial. Consequentemente, a capacidade do modelo de capturar comparações e interdependências entre as estatísticas das duas equipes é comprometida.

Das três estratégias que incorporam informação posicional, a organização alternada apresentou a média de desempenho superior em ambas as bases de dados (apresentando acurácia de 70,66% nas ranqueadas e 70,60% nas profissionais), seguida de perto pelas abordagens agrupada e aleatória. A vantagem conceitual do arranjo alternado, neste domínio, reside na intercalação de atributos de equipes adversárias, o que estabelece uma proximidade local entre estatísticas homólogas. Isso, por sua vez, facilita que as cabeças de atenção aprendam comparações par-a-par diretamente no espaço de representações. Em contraste, no modo agrupado, a contiguidade das métricas de cada equipe pode demandar saltos maiores para comparações entre equipes, enquanto a ordem aleatória requer que a atenção descubra consistentemente pares relevantes dispersos. Contudo, a diferença entre as estratégias alternada, agrupada e aleatória é mínima, sugerindo que a simples provisão de uma estrutura posicional consistente é suficiente para que o Transformer capture uma parcela significativa das relações úteis, representado na Imagem 6.2.

Com relação às métricas de F1 e Revocação, observou-se um padrão semelhante ao da acurácia, com a organização alternada apresentando o melhor desempenho, embora de forma muito próxima às abordagens agrupada e aleatória. Adicionalmente, a ausência de um *embedding* de posição também resultou em um desempenho inferior em relação às organizações que utilizam *embedding* de posição para F1 e revocação. Os tempos médios de processamento exibiram pouca variação entre as diferentes estratégias de *embedding* de posição, sem que isso impactasse as conclusões gerais obtidas.

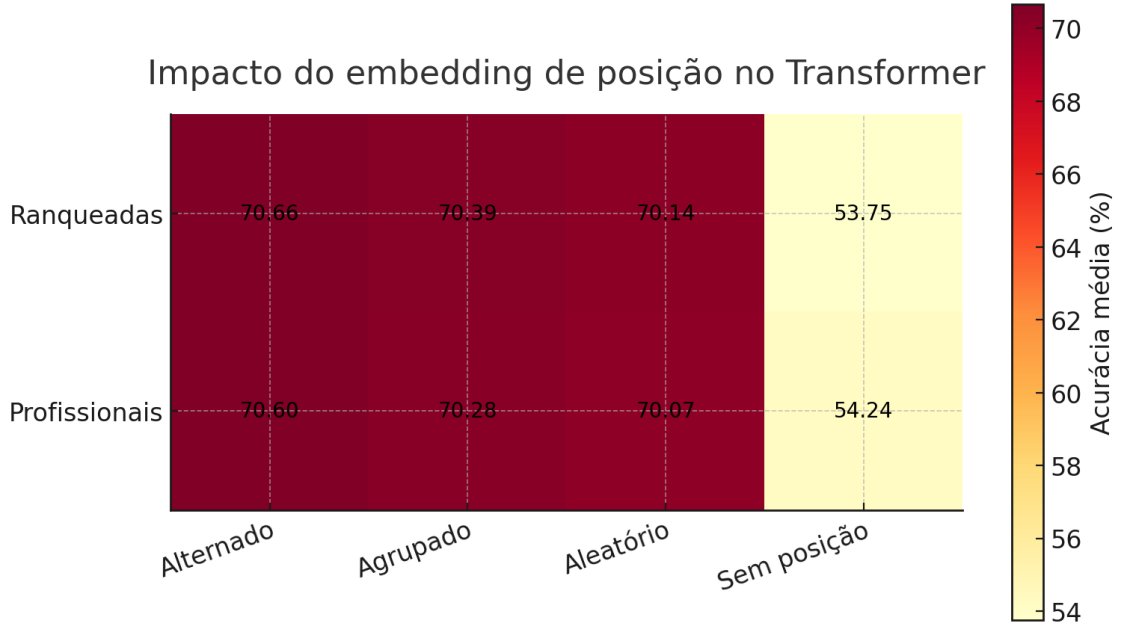


Figura 6.2: Heatmap da acurácia média do modelo Transformer considerando diferentes estratégias de *embedding* de posição (Alternado, Agrupado, Aleatório e Sem posição) nas bases de partidas ranqueadas e profissionais.

Tabela 6.1: Resultados médios (\pm desvio padrão) do modelo Transformer em termos de Acurácia(%), F1-Score(%), Revocação(%) e Tempo de execução(s) para a base de partidas ranqueadas, ao avariar a forma que realiza o *embedding* de posição.

Organização	Acurácia	F1-Score	Revocação	Tempo
Alternado	70,66 \pm 1,16	70,65 \pm 1,16	70,66 \pm 1,17	60,00 \pm 4,78
Agrupado	70,39 \pm 1,19	70,36 \pm 1,20	70,37 \pm 1,19	57,28 \pm 0,14
Aleatório	70,14 \pm 0,79	70,12 \pm 0,81	70,15 \pm 0,80	56,94 \pm 0,34
Sem posição	53,75 \pm 1,58	53,37 \pm 1,27	53,70 \pm 1,55	49,91 \pm 4,93

6.2 Resultados do FT-Transformer

Na condução dos experimentos com o modelo FT-Transformer, empregou-se a validação cruzada aninhada, com parâmetros $k = 5$ e $n = 5$, para a otimização dos hiperparâmetros. Esta abordagem metodológica foi selecionada com o intuito de assegurar um processo de avaliação robusto e minimamente enviesado, mediante a integração de duas camadas complementares de validação. A validação externa, estruturada em cinco partições, tem a função de mensurar o desempenho geral do modelo em dados inéditos, mimetizando sua aplicação em cenários práticos. Paralelamente, a validação interna, realizada em cada partição externa, direciona a busca pelos hiperparâmetros mais adequados, mitigando o risco de sobreajuste e garantindo uma estimativa mais fidedigna do potencial de generalização do modelo. Consequentemente, cada conjunto de treinamento é decomposto em cinco subconjuntos adicionais, resultando em um total de 25 avaliações distintas, o que permite

Tabela 6.2: Resultados médios (\pm desvio padrão) do modelo Transformer em termos de Acurácia(%), F1-Score(%), Revocação(%) e Tempo de execução(s) de execução para a base de partidas profissionais, ao avariar a forma que realiza o *embedding* de posição.

Organização	Acurácia	F1-Score	Revocação	Tempo
Alternado	70,60 \pm 1,48	70,20 \pm 1,54	70,14 \pm 1,54	42,89 \pm 0,57
Agrupado	70,28 \pm 0,32	69,81 \pm 0,25	69,73 \pm 0,26	42,86 \pm 0,09
Aleatório	70,07 \pm 1,14	69,45 \pm 1,38	69,38 \pm 1,37	42,69 \pm 0,23
Sem posição	54,24 \pm 0,91	36,16 \pm 1,79	49,94 \pm 0,23	43,42 \pm 5,12

que as métricas reportadas sejam não apenas médias globais, mas também representações estáveis do comportamento do modelo em diferentes divisões dos dados.

6.2.1 Impacto dos hiperparâmetros investigados

Na avaliação do FT-Transformer, dois hiperparâmetros foram priorizados devido à sua influência direta na capacidade de representação do modelo: a quantidade de cabeças no mecanismo de atenção multi-cabeça, que define os subespaços de projeção para a exploração de relações entre atributos, e o número de blocos de atenção empilhados, que determina a profundidade da arquitetura e, consequentemente, a habilidade do modelo em compor interações mais complexas com as variáveis de entrada.

A investigação sistemática desses parâmetros demonstrou que variações sutis na configuração podem exercer uma influência considerável na acurácia final, embora a magnitude da diferença entre as melhores combinações seja relativamente discreta. Isso corrobora a importância da otimização criteriosa de hiperparâmetros em contextos de dados tabulares complexos, como os inerentes a partidas de e-sports, onde múltiplos fatores interagem de forma dinâmica.

6.2.2 Resultados na base de partidas ranqueadas

Na base de partidas ranqueadas, a análise das 25 partições da validação cruzada aninhada revelou que a configuração ótima, com acurácia média de 72,95%, compreendeu 4 blocos de atenção e 16 cabeças de atenção (Tabela 6.3 e Imagem 6.3). As demais configurações apresentaram resultados muito similares, na casa dos 72,9%, indicando que o modelo atingiu um platô de desempenho dentro da faixa de hiperparâmetros explorada.

Contudo, a sutil superioridade da configuração indica que a diminuição no número de blocos, em conjunto com o acréscimo no paralelismo do mecanismo de atenção, pode otimizar a captação de relações relevantes sem impor uma complexidade excessiva à arquitetura. Tal comportamento pode estar vinculado à natureza

dos dados ranqueados, que exibem um volume maior e uma variabilidade acentuada entre as partidas, exigindo um modelo apto a explorar múltiplas relações simultaneamente, porém sem profundidade excessiva que poderia incorrer em ruído ou redundância.

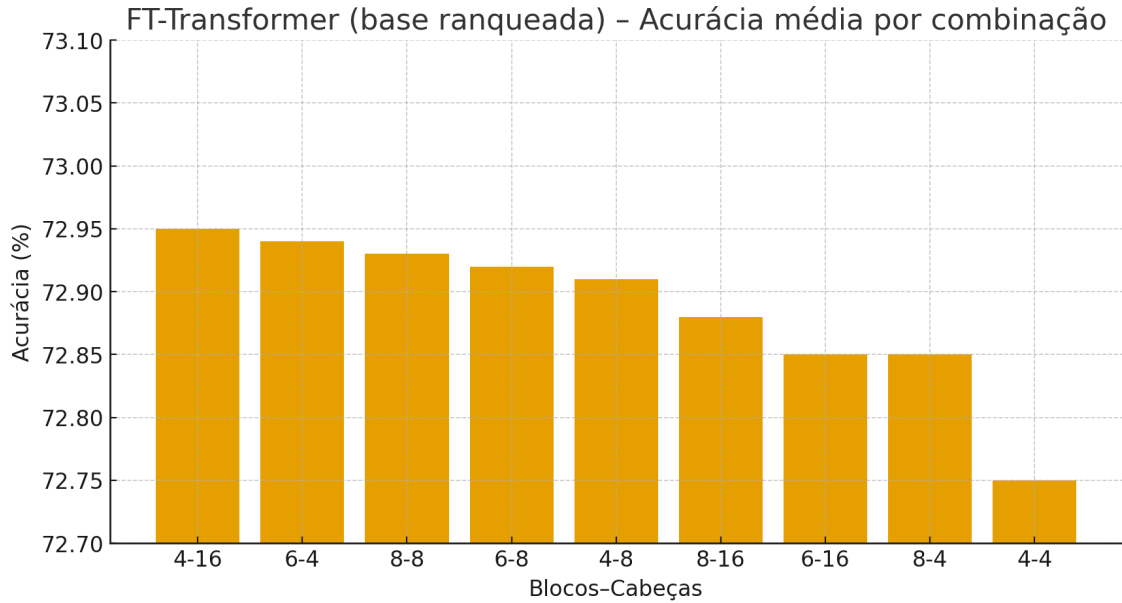


Figura 6.3: Acurácia média obtida em 25 *folds* da validação cruzada aninhada para a base de partidas ranqueadas, utilizando o modelo FT-Transformer com diferentes combinações de hiperparâmetros. Os valores estão apresentados no formato *Blocos-Cabeças* e o eixo foi ajustado para destacar pequenas variações entre as configurações.

Tabela 6.3: Resultados da acurácia média obtida em 25 *folds* para a base de partidas ranqueadas, empregando FT-Transformer e variando o número de blocos de atenção (*num_attn_blocks*) e o número de cabeças de atenção (*num_heads*).

num_attn_blocks	num_heads	Acurácia Média(%)
4	16	72,95
6	4	72,94
8	8	72,93
6	8	72,92
4	8	72,91
8	16	72,88
6	16	72,85
8	4	72,85
4	4	72,75

6.2.3 Resultados na base de partidas profissionais

No FT-Transformer, a configuração que resultou no desempenho superior na base de dados de partidas profissionais consistiu em 6 blocos de atenção e 8 cabeças de atenção, com uma acurácia média alcançada de 72,09% (Tabela 6.4 e Imagem 6.4). Nesta base específica, em contraste com os dados ranqueados, a aprofundamento da arquitetura mostrou-se mais benéfico do que a expansão no número de cabeças de atenção. Este achado sugere que, no contexto de partidas profissionais, caracterizadas por estratégias mais padronizadas e estruturadas, o modelo beneficia-se de uma maior capacidade de compor hierarquicamente padrões — um atributo de arquiteturas mais profundas. Simultaneamente, um número intermediário de cabeças de atenção mostrou-se suficiente para capturar a diversidade das interações, dispensando um paralelismo excessivo.

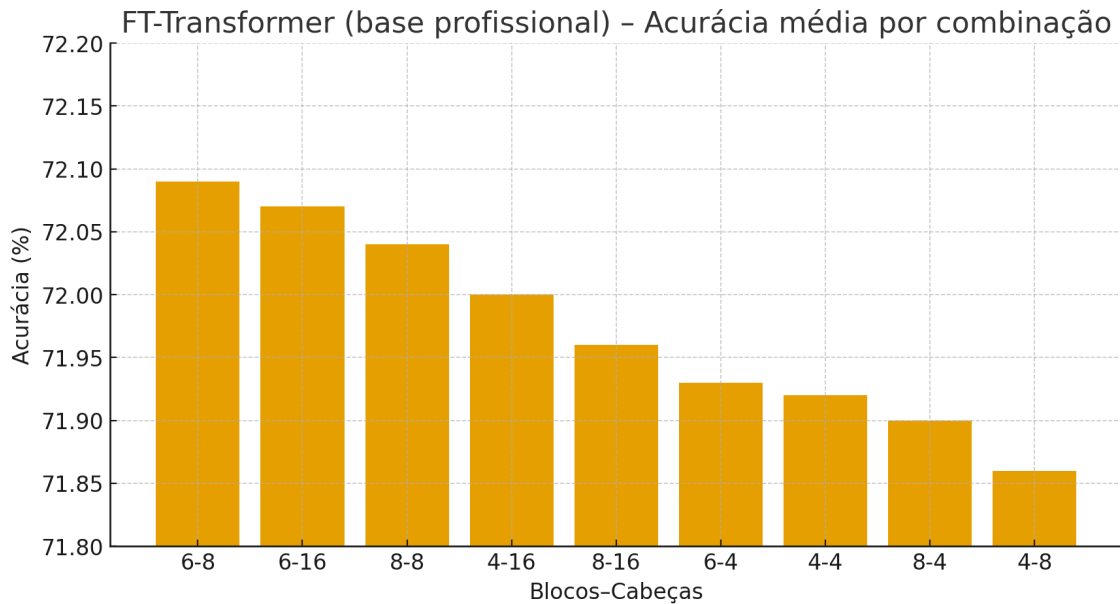


Figura 6.4: Acurácia média obtida em 25 *folds* da validação cruzada aninhada para a base de partidas profissionais, utilizando o modelo FT-Transformer com diferentes combinações de hiperparâmetros. Os valores estão apresentados no formato *Blocos-Cabeças* e o eixo foi ajustado para destacar pequenas variações entre as configurações.

6.2.4 Análise da validação cruzada interna

As Tabelas 6.5 e 6.6 fornecem uma análise detalhada dos resultados obtidos através da validação cruzada interna, destacando o desempenho de distintas configurações de hiperparâmetros em cada partição externa. Observou-se uma convergência notável para as combinações de hiperparâmetros consideradas ótimas para cada base de dados, apesar de algumas variações pontuais onde certas configurações superaram

Tabela 6.4: Resultados da acurácia média obtida em 25 *folds* para a base de partidas profissionais, empregando FT-Transformer e variando o número de blocos de atenção (*num_attn_blocks*) e o número de cabeças de atenção (*num_heads*).

num_attn_blocks	num_heads	Acurácia Média(%)
6	8	72,09
6	16	72,07
8	8	72,04
4	16	72,00
8	16	71,96
6	4	71,93
4	4	71,92
8	4	71,90
4	8	71,86

momentaneamente a média global (ilustrado nas Imagens 6.5 e 6.6).

Essa convergência consolida a robustez das seleções de hiperparâmetros, sugerindo que os resultados obtidos são consistentes e não meramente aleatórios. Ademais, a proximidade das métricas de acurácia entre as diferentes configurações indica que o FT-Transformer exibe uma considerável resiliência a modificações marginais em sua parametrização, uma qualidade desejável para modelos aplicados em cenários práticos onde restrições de recursos computacionais podem impor a necessidade de ajustes arquiteturais.

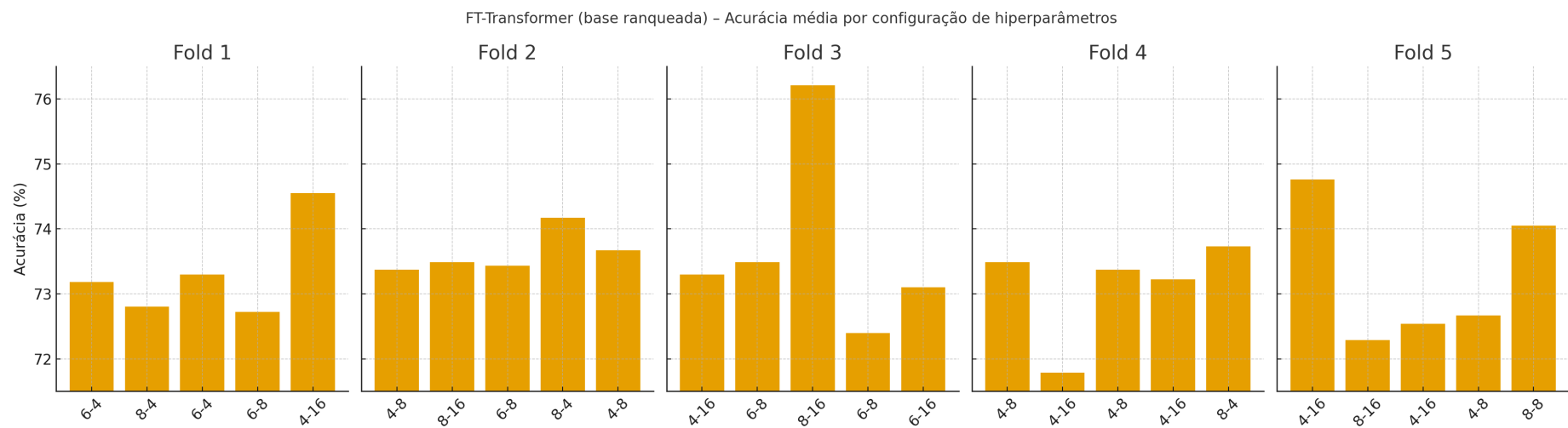


Figura 6.5: Desempenho do modelo FT-Transformer nos diferentes *folds* externos da validação cruzada aninhada para a base de dados de partidas ranqueadas.

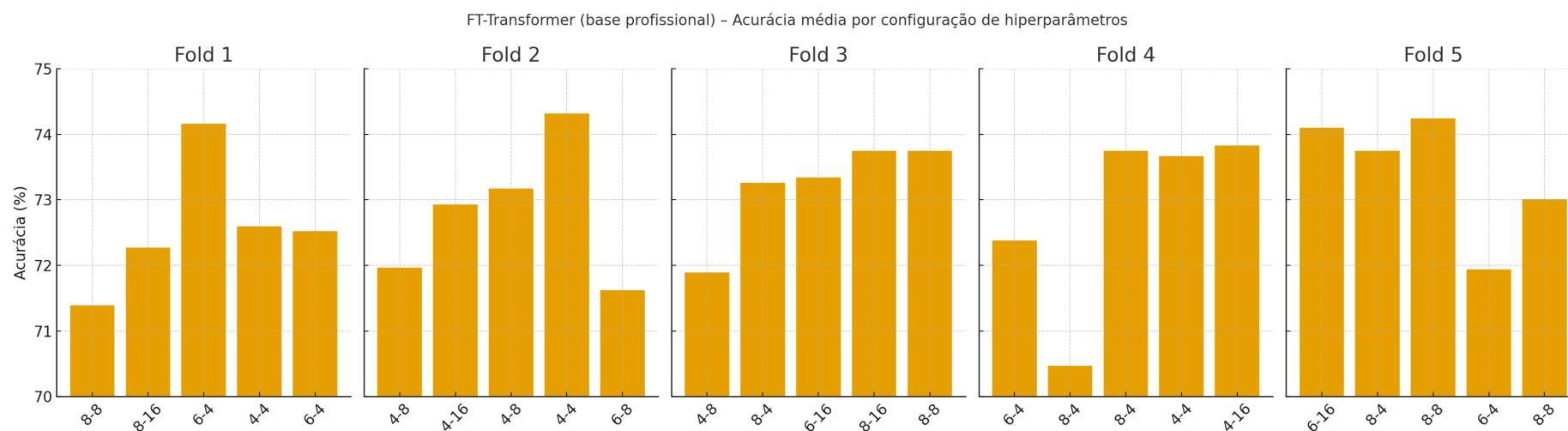


Figura 6.6: Desempenho do modelo FT-Transformer nos diferentes *folds* externos da validação cruzada aninhada para a base de dados de partidas profissionais.

Tabela 6.5: Acurácia média obtida para a melhor configuração de hiperparâmetros na validação cruzada interna, realizada em cada *fold* externo da validação cruzada aninhada, utilizando a base de partidas ranqueadas com o modelo FT-Transformer.

Fold Externo	Blocos de Atenção	Cabeças de Atenção	Acurácia(%)
1	6	4	73.18
	8	4	72.8
	6	4	73.3
	6	8	72.72
	4	16	74.55
2	4	8	73.37
	8	16	73.49
	6	8	73.43
	8	4	74.17
	4	8	73.67
3	4	16	73.3
	6	8	73.49
	8	16	76.21
	6	8	72.4
	6	16	73.1
4	4	8	73.49
	4	16	71.79
	4	8	73.37
	4	16	73.22
	8	4	73.73
5	4	16	74.76
	8	16	72.29
	4	16	72.54
	4	8	72.67
	8	8	74.05

6.2.5 Síntese dos resultados

Na base ranqueada, a abordagem mais eficiente favorece um número elevado de cabeças de atenção, com uma quantidade reduzida de blocos, o que pode ser atribuído à maior diversidade intrínseca aos dados. Na base profissional, o padrão observado é o oposto: maior profundidade arquitetural, com um número moderado de cabeças de atenção, o que se alinha com a homogeneidade estratégica típica desse conjunto de dados. Essas distinções ressaltam não apenas a importância da seleção criteriosa de hiperparâmetros, mas também a necessidade de ponderar as particularidades de cada tipo de dado. Em ambas as bases, contudo, os desempenhos convergiram para um patamar superior a 72%, indicando um aprimoramento consistente em relação a modelos de referência mais simples e validando a adequação do FT-Transformer como uma abordagem competitiva para a predição de resultados.

Tabela 6.6: Acurácia média obtida para a melhor configuração de hiperparâmetros na validação cruzada interna, realizada em cada *fold* externo da validação cruzada aninhada, utilizando a base de partidas profissionais com o modelo FT-Transformer.

Fold Externo	Blocos de Atenção	Cabeças de Atenção	Acurácia(%)
1	8	8	71.39
	8	16	72.27
	6	4	74.16
	4	4	72.60
	6	4	72.52
2	4	8	71.97
	4	16	72.93
	4	8	73.17
	4	4	74.32
	6	8	71.62
3	4	8	71.89
	8	4	73.26
	6	16	73.34
	8	16	73.75
	8	8	73.75
4	6	4	72.38
	8	4	70.47
	8	4	73.75
	4	4	73.67
	4	16	73.83
5	6	16	74.1
	8	4	73.75
	8	8	74.24
	6	4	71.94
	8	8	73.01

6.3 Discussão Comparativa dos Modelos

Após a calibração dos hiperparâmetros do modelo FT-Transformer para cada conjunto de dados, foram realizados experimentos com todos os modelos descritos no Capítulo 5, empregando validação cruzada do tipo k -fold com $k = 5$. O emprego da validação cruzada k -fold com $k = 5$ conferiu maior robustez à avaliação, mitigando o viés inerente à divisão aleatória dos dados. É relevante notar que a análise para o modelo Transformer foi previamente executada na Seção 6.1, possibilitando um contraste direto com as demais metodologias abordadas.

6.3.1 Desempenho nos dados de partidas ranqueadas

Na base de dados referente às partidas ranqueadas (Tabela 6.7), o FT-Transformer demonstrou uma performance superior em todas as métricas analisadas. Especificamente, alcançou uma acurácia média de 73,45%, um F1-Score de 73,06% e uma revocação de 72,15%. Tais resultados sublinham a capacidade do modelo em identificar padrões intrincados nos dados tabulares, superando até mesmo algoritmos reconhecidamente robustos como Random Forest e Regressão Logística. Contudo, este aprimoramento no desempenho veio acompanhado de um custo computacional consideravelmente mais alto, evidenciado por um tempo médio de execução de 476,73 segundos, que se mostrou mais de duas ordens de magnitude superior aos demais modelos testados.

Modelos mais tradicionais, como Regressão Logística e Random Forest, apresentaram métricas de desempenho comparáveis às do FT-Transformer, porém com tempos de execução substancialmente menores, configurando-se como opções mais viáveis em contextos práticos que impõem restrições de recursos computacionais. Em contrapartida, os algoritmos XGBoost e Naive Bayes destacaram-se pela agilidade computacional, com tempos de execução médios inferiores a um segundo, embora isso tenha resultado em uma redução notável no desempenho preditivo, particularmente nas métricas de F1-Score e revocação. O modelo Transformer convencional, por sua vez, registrou os resultados menos expressivos, com uma acurácia média de 70,66% e um custo computacional consideravelmente superior aos modelos clássicos, ainda que inferior ao FT-Transformer.

Esses resultados podem ser visualizados de forma mais clara na Figura 6.8 e na Figura 6.9, que comparam o F1-Score e a revocação entre os modelos. Já a relação entre custo computacional e acurácia é mostrada na Figura 6.7, evidenciando a discrepância entre o FT-Transformer e as outras abordagens.

Tabela 6.7: Resultados médios (\pm desvio padrão) dos principais modelos de classificação avaliados na base de dados de partidas ranqueadas, com base em Acurácia(%), F1-Score(%), Revocação(%) e Tempo de execução(s).

Modelo	Acurácia	Tempo	F1-Score	Revocação
Regressão Logística	73.00 \pm 0.95	1.08 \pm 0.14	72.92 \pm 1.17	72.88 \pm 1.66
Naive Bayes	72.46 \pm 0.72	0.005 \pm 0.0005	72.07 \pm 0.82	71.24 \pm 1.22
Random Forest	72.71 \pm 0.76	2.00 \pm 0.02	72.52 \pm 0.90	72.17 \pm 1.31
XGBoost	72.13 \pm 0.72	0.244 \pm 0.007	71.69 \pm 0.97	70.75 \pm 1.65
Transformer	70.66 \pm 0.12	29.89 \pm 3.84	70.64 \pm 1.22	70.65 \pm 1.22
FT-Transformer	73.45 \pm 0.71	476.73 \pm 4.84	73.06 \pm 0.45	72.15 \pm 1.07

6.3.2 Desempenho nos dados de partidas profissionais

Na base de dados das partidas profissionais (Tabela 6.8), observou-se um cenário similar, onde o FT-Transformer se consolidou como a abordagem mais eficaz em termos de métricas preditivas, alcançando uma acurácia de 73,12%, um F1-Score de 75,81% e uma revocação de 77,69%. Estes resultados confirmam sua capacidade de generalização em diferentes conjuntos de dados. Contudo, o tempo médio de execução reiterou a limitação de sua aplicabilidade prática em contextos que demandam análises em tempo real ou escalabilidade.

Nesta base de dados, a discrepância de desempenho entre os modelos Random Forest e Regressão Logística foi mais pronunciada. O Random Forest demonstrou uma ligeira superioridade em acurácia, F1-Score e revocação, mantendo um tempo de execução notavelmente rápido, inferior a um segundo. Isso indica que, em situações que exigem celeridade e um desempenho balanceado, esses algoritmos podem representar uma alternativa viável ao FT-Transformer. O Naive Bayes, apesar de apresentar métricas de acurácia inferiores, manteve sua posição como o método mais eficiente em termos de tempo, configurando-se como uma opção preferencial em aplicações de larga escala ou quando a velocidade de resposta é um fator crítico.

De forma análoga, os resultados para a base de partidas profissionais estão representados na Figura 6.11 e na Figura 6.12, que apresentam os valores de F1-Score e revocação. A relação entre custo computacional e acurácia é mostrada na Figura 6.10, destacando novamente o elevado custo do FT-Transformer frente aos outros modelos.

6.3.3 Análise comparativa e implicações práticas

Em uma análise comparativa, o FT-Transformer demonstrou ser o modelo preditivo mais eficaz. Contudo, sua elevada demanda computacional restringe sua aplicabilidade em ambientes que não dispõem de infraestrutura especializada, como GPUs de alto desempenho. Em contrapartida, os modelos Random Forest e Regressão

Tabela 6.8: Resultados médios (\pm desvio padrão) dos principais modelos de classificação avaliados na base de dados de partidas profissionais, com base em Acurácia(%), F1-Score(%), Revocação(%) e Tempo de execução(s).

Modelo	Acurácia	Tempo	F1-Score	Revocação
Regressão Logística	72.47 ± 1.07	0.77 ± 0.01	75.28 ± 1.05	77.10 ± 1.56
Naive Bayes	71.34 ± 1.28	0.004 ± 0.0002	73.44 ± 1.42	72.90 ± 2.42
Random Forest	72.56 ± 1.26	1.08 ± 0.01	75.50 ± 1.37	77.80 ± 2.64
XGBoost	71.9 ± 1.06	0.326 ± 0.201	74.53 ± 1.24	75.6 ± 2.22
Transformer	70.6 ± 0.14	42.89 ± 0.57	70.2 ± 1.54	70.14 ± 1.54
FT-Transformer	73.12 ± 1.27	523.67 ± 5.30	75.81 ± 1.99	77.69 ± 5.55

Logística ofereceram um balanço favorável entre o desempenho preditivo e a eficiência de processamento, tornando-os alternativas mais adequadas para cenários que requerem celeridade ou são executados em plataformas com capacidade de processamento limitada. O modelo Naive Bayes, apesar de sua simplicidade, manteve uma performance competitiva em relação ao seu tempo de execução, quase insignificante, sendo uma opção viável para tarefas onde a eficiência computacional é um fator primordial.

O modelo Transformer padrão apresentou os desempenhos mais modestos, sugerindo que sua aplicação direta em dados tabulares não é suficiente para capturar de forma eficaz a estrutura intrínseca das variáveis. O modo como os atributos foram organizados teve um impacto pequeno na performance, com a intercalação de atributos das duas equipes demonstrando ser a abordagem mais benéfica. Essa constatação abre possibilidade para futuras pesquisas, como a modificação arquitetônica do Transformer, com foco na exploração explícita das interdependências sequenciais dos atributos, potencialmente integrando suas capacidades de atenção com métodos que preservem a natureza tabular dos dados.

Em síntese, as evidências experimentais indicam que a seleção do modelo mais apropriado deve ponderar a acurácia preditiva em relação à eficiência computacional, em conformidade com os requisitos específicos de cada aplicação. O FT-Transformer, apesar de seu desempenho superior, demanda otimizações adicionais para viabilizar sua implementação em cenários práticos. Em contraste, as abordagens baseadas em Random Forest e Regressão Logística oferecem um balanço vantajoso entre precisão e eficiência, configurando-se como alternativas robustas e pragmáticas. Tais achados não apenas validam a relevância de modelos especializados como o FT-Transformer para o avanço do conhecimento científico, mas também enfatizam a importância contínua de modelos tradicionais em contextos práticos, onde a relação custo-benefício é um fator determinante.

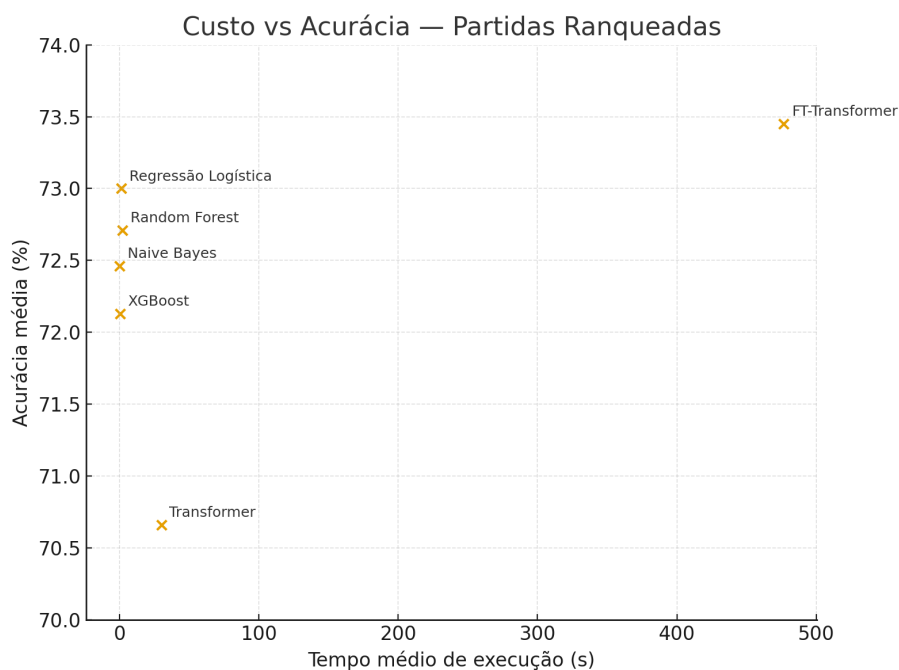


Figura 6.7: Comparação do computacional e acurácia entre os modelos na base de partidas ranqueadas.

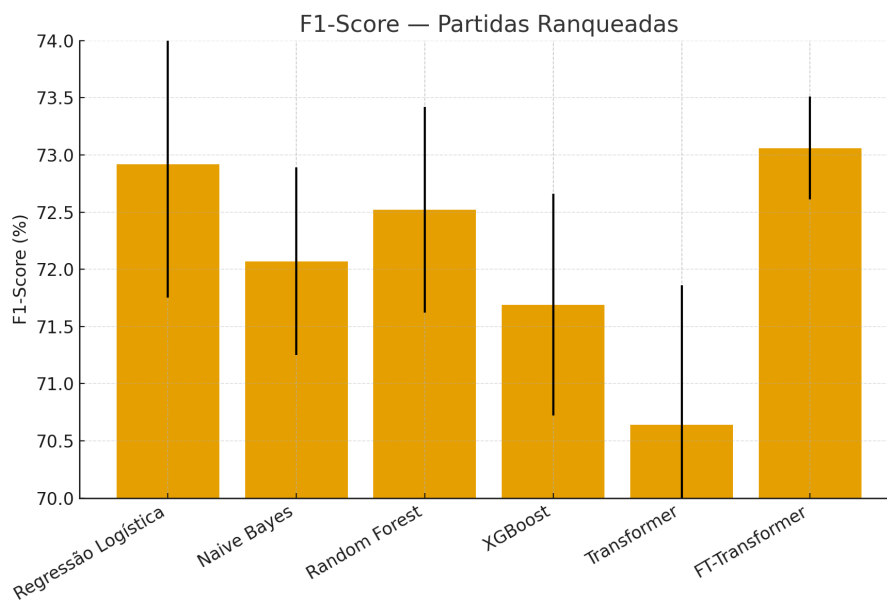


Figura 6.8: Comparação do F1-Score entre os modelos na base de partidas ranqueadas.

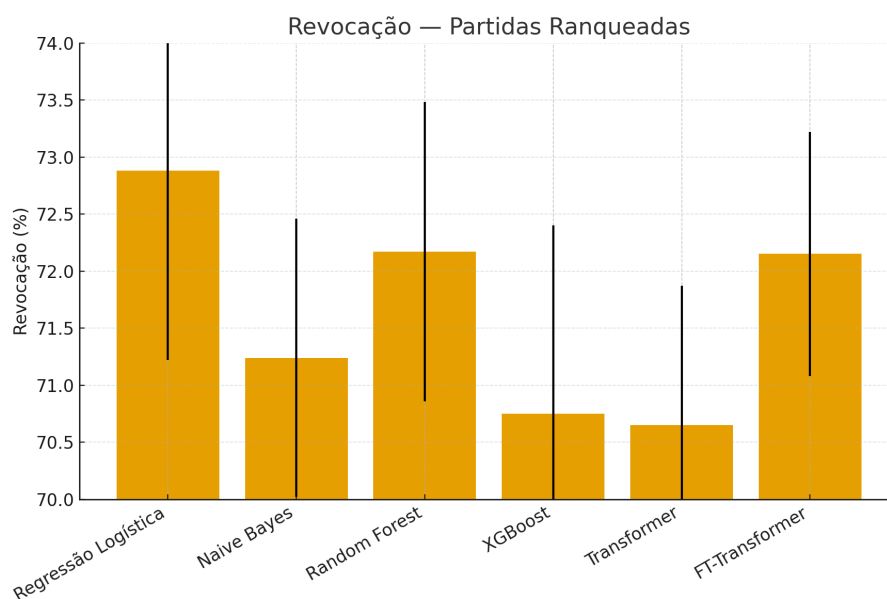


Figura 6.9: Comparação da revocação entre os modelos na base de partidas ranqueadas.

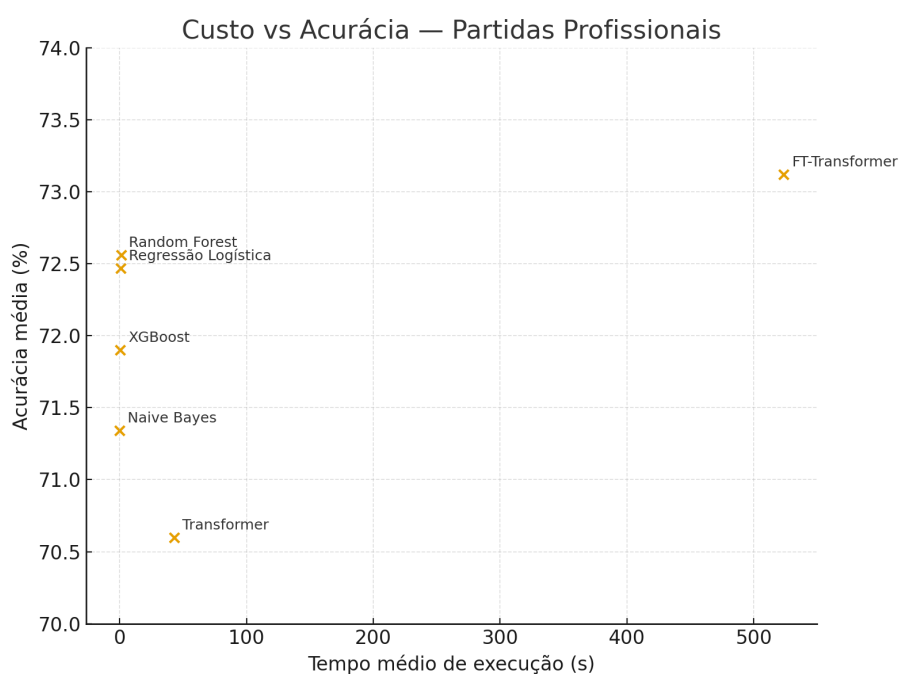


Figura 6.10: Comparação do custo computacional e acurácia entre os modelos na base de partidas profissionais.

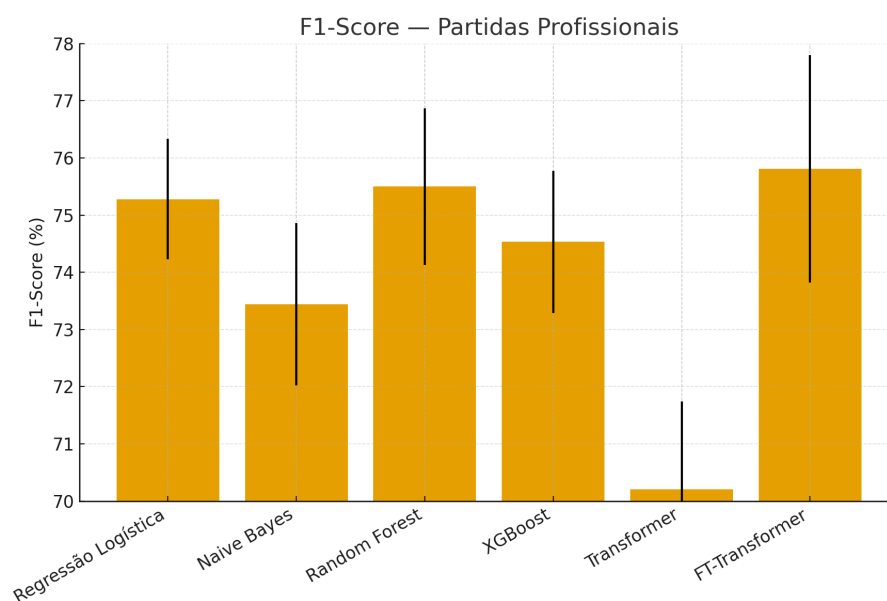


Figura 6.11: Comparação do F1-Score entre os modelos na base de partidas profissionais.

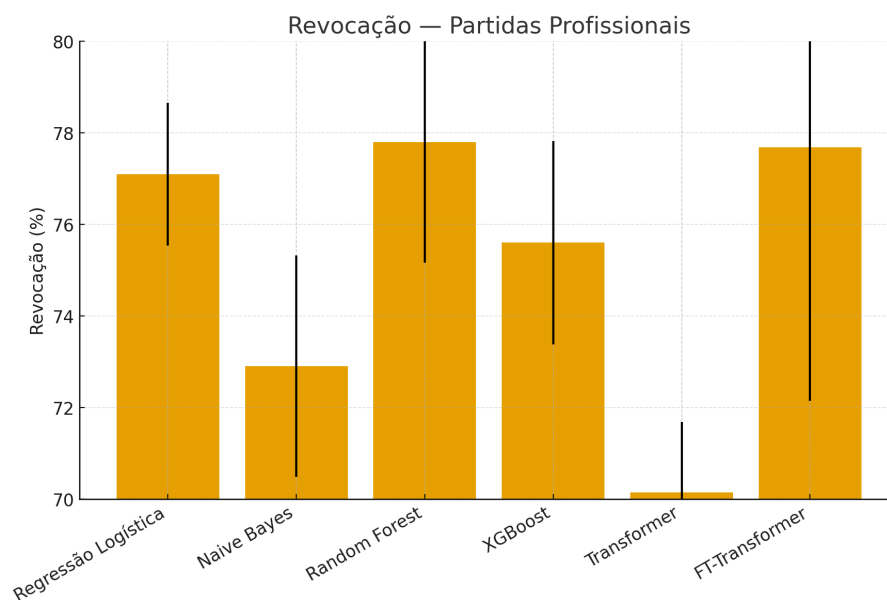


Figura 6.12: Comparação da revocação entre os modelos na base de partidas profissionais.

Capítulo 7

Conclusão

Esta pesquisa se dedicou ao estudo da previsão de vitórias e eventos em jogos do gênero MOBA, identificando uma carência na literatura referente à análise do modelo Transformer. Na literatura não apresenta um estudo profundo do modelo Transformer na predição de vitória de jogos MOBA durante a partida. Em virtude disso, este trabalho analisou duas versões do modelo Transformer adaptados para tratar de dados tabulares provenientes de dois conjunto de partidas do jogo *League of Legends*.

A primeira versão seria o modelo original do Transformer, definido em VASWANI *et al.* (2017), adaptado para considerar os atributos das partidas como *tokens* equivalentes às palavras em sentenças de linguagem natural. Cada coluna do conjunto de dados foi transformada em um *embedding*, permitindo que o modelo processasse os atributos de forma similar ao tratamento de sequências em tarefas de NLP, capturando relações e dependências entre variáveis por meio do mecanismo de atenção. Isso possibilitou analisar a possibilidade de alterar o posicionamento dos atributos para identificar se essa alteração impactaria a performance preditiva do modelo.

A segunda variante, inspirada no FT-Transformer (GORISHNIY *et al.*, 2021), incorpora uma metodologia otimizada para o tratamento de dados tabulares. Essa abordagem mantém as características intrínsecas da estrutura tabular e emprega *embeddings* específicos para cada coluna, capacitando o modelo a discernir interações entre atributos sem depender de uma ordem sequencial predefinida.

O modelo FT-Transformer apresentou um desempenho superior em ambas as bases de dados testadas, enquanto o modelo Transformer obteve os resultados menos expressivos em comparação com abordagens amplamente utilizadas na literatura. Outrossim, ambas variações de Transformer analisadas apresentaram tempo médio de execução elevados. Em contrapartida, os modelos Random Forest e Regressão Logística apresentaram resultados preditivos competitivos com menor demanda computacional. Adicionalmente, os classificadores Naive Bayes e XGBoost, apesar de não alcançarem os mais elevados índices de acurácia, distinguiram-se pela eficiência temporal, exibindo os menores tempos médios de execução entre todos os métodos

testados.

No modelo Transformer, a forma de organização dos atributos demonstrou ser um fator relevante para a performance. A abordagem alternada, que intercala variáveis de equipes distintas com temas semelhantes, resultou em uma acurácia ligeiramente superior em ambas as bases de dados. Essa disposição facilita o estabelecimento de proximidade entre estatísticas correspondentes das equipes, permitindo ao Transformer capturar relações de comparação direta entre os desempenhos das equipes durante a partida. Portanto, a estrutura dos atributos pode influenciar significativamente a eficácia do modelo em dados tabulares, ressaltando a importância de considerar a organização sequencial das variáveis ao aplicar Transformers neste tipo de problema.

7.1 Contribuições

A principal contribuição deste estudo reside na compilação exaustiva da literatura referente à previsão de eventos e resultados em jogos do gênero MOBA, estabelecendo uma base de conhecimento robusta para investigações futuras nesta área.

Os resultados experimentais indicaram que o FT-Transformer demonstrou um desempenho notável na previsão de vitórias em diferentes fases da partida, superando os demais métodos avaliados.

Adicionalmente, a análise demonstrou o impacto da organização sequencial dos atributos no desempenho do modelo Transformer aplicado a dados tabulares, indicando que a disposição desses atributos pode influenciar diretamente a eficácia do modelo.

7.2 Limitações

A principal limitação deste estudo reside na restrição da análise a dois conjuntos de dados oriundos de um único título, o *League of Legends*. Tal delimitação pode comprometer a generalização dos resultados observados, sugerindo que investigações futuras incorporem um leque mais amplo de bases de dados e avaliem outros modelos para obter uma compreensão mais abrangente do fenômeno em questão.

7.3 Trabalhos Futuros

Pesquisas futuras podem se concentrar na otimização do tempo de processamento do FT-Transformer para torná-lo mais viável em aplicações práticas. Uma direção adicional promissora envolve a integração de mecanismos no modelo que conside-

rem explicitamente a ordenação sequencial dos atributos, de maneira análoga ao Transformer original.

Adicionalmente, recomenda-se investigar abordagens alternativas para a organização e representação de características tabulares, com o objetivo de otimizar a performance de modelos baseados em mecanismos de atenção.

Essas direções de pesquisa podem catalisar avanços tanto no campo dos jogos eletrônicos quanto em outros setores que lidam com dados tabulares de natureza complexa e volátil.

Em suma, este estudo destaca o FT-Transformer como um modelo com potencial significativo para predições no contexto de jogos MOBA, ao mesmo tempo que sugere novas linhas de investigação para otimizar a eficiência, aplicabilidade e utilidade prática de arquiteturas que empregam mecanismos de atenção.

Referências Bibliográficas

- AGARWALA, A., PEARCE, M., 2014, “Learning Dota 2 team compositions”, *Sl: sn.*
- AKHMEDOV, K., PHAN, A. H., 2021, “Machine learning models for DOTA 2 outcomes prediction”, *arXiv preprint arXiv:2106.01782*.
- ALMEIDA, C. E. M., CORREIA, R. C. M., ELER, D. M., et al., 2017, “Prediction of winners in MOBA games”. In: *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6. doi: 10.23919/CISTI.2017.7975774.
- ALTAY, O., 2024. “The First MOBA: Aeon of Strife”. Disponível em: <<https://mmos.com/editorials/the-first-moba-aeon-of-strife>>. Acesso em: 22 ago. 2025.
- ANDONO, P. N., KURNIAWAN, N. B., SUPRIYANTO, C., 2017, “DotA 2 bots win prediction using Naive Bayes based on AdaBoost algorithm”. In: *Proceedings of the 3rd International Conference on Communication and Information Processing*, pp. 180–184.
- ANI, R., HARIKUMAR, V., DEVAN, A. K., et al., 2019, “Victory prediction in League of Legends using Feature Selection and Ensemble methods”. In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pp. 74–77. doi: 10.1109/ICCS45141.2019.9065758.
- ANSHORI, M., MAR’I, F., ALAUDDIN, M. W., et al., 2018, “Prediction Result of Dota 2 Games Using Improved SVM Classifier Based on Particle Swarm Optimization”. In: *2018 International Conference on Sustainable Information Engineering and Technology (SIET)*, pp. 121–126. doi: 10.1109/SIET.2018.8693204.
- ARYANATA, G. A., RAHADI, P., SUDARMOJO, Y., 2017, “Prediction of DOTA 2 match result by using analytical hierarchy process method”, *International Journal of Engineering and Emerging Technology*, v. 2, n. 1, pp. 22–25.

- ASYHRAF ZAMIR ZAMRI, M., OMAR, N. A., HAMID, I. R. A., 2024, “Online Game Outcome Prediction Model Using Weighted-Based Feature Approach”, *Fusion: Practice and Applications*, v. 15, n. 2, pp. 132 – 144. doi: 10.54216/FPA.150212. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85194182637&doi=10.54216%2fFPA.150212&partnerID=40&md5=8e7537f50701eb0e6349638c47740450>>. Cited by: 0.
- BAHROLOLLOOMI, F., SAUER, S., KLONOWSKI, F., et al., 2022, “A Machine Learning based Analysis of e-Sports Player Performances in League of Legends for Winning Prediction based on Player Roles and Performances.” In: *VISIGRAPP (2: HUCAPP)*, pp. 68–76.
- BAHROLOLLOOMI, F., KLONOWSKI, F., SAUER, S., et al., 2023, “E-Sports Player Performance Metrics for Predicting the Outcome of League of Legends Matches Considering Player Roles”, *SN Computer Science*, v. 4, n. 3. doi: 10.1007/s42979-022-01660-6. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85149244467&doi=10.1007%2fs42979-022-01660-6&partnerID=40&md5=9edc497dfaeec1cf9fbffe0831f710f7>>. Cited by: 0; All Open Access, Hybrid Gold Open Access.
- BÁNYAI, F., ZSILA, Á., GRIFFITHS, M. D., et al., 2020, “Career as a Professional Gamer: Gaming Motives as Predictors of Career Plans to Become a Professional Esport Player”, *Frontiers in Psychology*, v. 11. ISSN: 1664-1078. doi: 10.3389/fpsyg.2020.01866. Disponível em: <<https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2020.01866>>.
- BHARGAVA, N., SHARMA, G., BHARGAVA, R., et al., 2013, “Decision tree analysis on j48 algorithm for data mining”, *Proceedings of international journal of advanced research in computer science and software engineering*, v. 3, n. 6.
- BHATTACHARYA, S., MADDIKUNTA, P. K. R., KALURI, R., et al., 2020, “A novel PCA-firefly based XGBoost classification model for intrusion detection in networks using GPU”, *Electronics*, v. 9, n. 2, pp. 219.
- BIRANT, K. U., 2022, “Multi-view rank-based random forest: A new algorithm for prediction in eSports”, *Expert Systems*, v. 39, n. 2. doi: 10.1111/exsy.12857. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85118185877&doi=10.1111%2fexsy>>.

12857&partnerID=40&md5=b04dc5ee9d3185848c3c6413b29a4189>.

Cited by: 3.

- BISHOP, C. M., NASRABADI, N. M., 2006, *Pattern recognition and machine learning*, v. 4. New York, NY, USA, Springer.
- BOSER, B. E., GUYON, I. M., VAPNIK, V. N., 1992, “A training algorithm for optimal margin classifiers”. In: *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152.
- BRADLEY, R. A., TERRY, M. E., 1952, “Rank analysis of incomplete block designs: I. The method of paired comparisons”, *Biometrika*, v. 39, n. 3/4, pp. 324–345.
- CAPITAL, M., 2018, *eSports Whitepaper Series*. Relatório técnico, Mercer Capital. Relata prize pool do The International 2017: US\$ 24,787,916, com 94% via crowdfunding.
- CARDOSO, G. M. M., 2019, “Predição do resultado utilizando KNN: análise do jogo League of Legends”, *Engenharia de Computação*.
- CHAN, A., FACHRIZAL, F., LUBIS, A., 2020, “Outcome prediction using Naive Bayes algorithm in the selection of role hero mobile legend”. In: *Journal of Physics: Conference Series*, v. 1566, p. 012041. IOP Publishing.
- CHEN, S., ZHU, M., YE, D., et al., 2021, “Which Heroes to Pick? Learning to Draft in MOBA Games With Neural Networks and Tree Search”, *IEEE Transactions on Games*, v. 13, n. 4, pp. 410–421. doi: 10.1109/TG.2021.3095264.
- CHEN, S., JOACHIMS, T., 2016, “Modeling intransitivity in matchup and comparison data”. In: *Proceedings of the ninth acm international conference on web search and data mining*, pp. 227–236.
- CHEN, Z., SUN, Y., EL-NASR, M. S., et al., 2017, “Player skill decomposition in multiplayer online battle arenas”, *arXiv preprint arXiv:1702.06253*.
- CHEN, Z., NGUYEN, T.-H. D., XU, Y., et al., 2018a, “The art of drafting: a team-oriented hero recommendation system for multiplayer online battle arena games”. In: *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 200–208, a.
- CHEN, Z., XU, Y., NGUYEN, T.-H. D., et al., 2018b, “Modeling game avatar synergy and opposition through embedding in multiplayer online battle arena games”, *arXiv preprint arXiv:1803.10402*.

- CHO, K., VAN MERRIENBOER, B., GULCEHRE, C., et al., 2014, “Learning phrase representations using RNN encoder-decoder for statistical machine translation”, *arXiv preprint arXiv:1406.1078*.
- CLEGHERN, Z., LAHIRI, S., OZALTIN, O., et al., 2017, “Predicting Future States in DotA 2 Using Value-Split Models of Time Series Attribute Data”. In: *Proceedings of the 12th International Conference on the Foundations of Digital Games, FDG '17*, New York, NY, USA. Association for Computing Machinery. ISBN: 9781450353199. doi: 10.1145/3102071.3102095. Disponível em: <<https://doi.org/10.1145/3102071.3102095>>.
- CONLEY, K., PERRY, D., 2013, “How does he saw me? a recommendation engine for picking heroes in dota 2”, *Np, nd Web*, v. 7.
- COSTA, L. M., MANTOVANI, R. G., MONTEIRO SOUZA, F. C., et al., 2021, “Feature Analysis to League of Legends Victory Prediction on the Picks and Bans Phase”. In: *2021 IEEE Conference on Games (CoG)*, pp. 01–05. doi: 10.1109/CoG52621.2021.9619019.
- CRUZ, A. C., DO RÊGO, T. G., DE M FILHO, T., et al., 2021, “League of legends: an application of classification algorithms to verify the prediction importance of main in-game variables”. In: *Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pp. 329–333. SBC.
- CUNHA, R., 2021. “Guia de iniciantes – Como jogar na rota do meio em League of Legends”. Disponível em: <<https://www.fragster.com/br/rota-lol-guia/>>. Acesso em: 22 ago. 2025.
- DEJA, D., MYSLAK, M., 2015, “Topological clues for predicting outcomes of multiplayer online battle arena games”. In: *Proceedings of the International Conferences on Interfaces and Human Computer Interaction 2015, IHCI 2015, Game and Entertainment Technologies 2015, GET 2015 and Computer Graphics, Visualization, Computer Vision and Image Processing 2015, CGVCVIP 2015 - Part of the Multi Conference on Computer Science and Information Systems 2015*, p. 116 – 122. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-84969257303&partnerID=40&md5=6024f84d43f9a0e951211afc7818c002>>. Cited by: 2.
- DO, T. D., WANG, S. I., YU, D. S., et al., 2021, “Using machine learning to predict game outcomes based on player-champion experience in League

- of Legends”. In: *Proceedings of the 16th International Conference on the Foundations of Digital Games*, pp. 1–5.
- EPHRON, C., —. “League of Legends”. Kaggle dataset. Disponível em: <https://www.kaggle.com/datasets/chuckephron/leagueoflegends> [acessado em 11 de agosto de 2025].
- FREUND, Y., SCHAPIRE, R. E., 1997, “A decision-theoretic generalization of on-line learning and an application to boosting”, *Journal of computer and system sciences*, v. 55, n. 1, pp. 119–139.
- FRIEDMAN, J. H., 2001, “Greedy function approximation: a gradient boosting machine”, *Annals of statistics*, pp. 1189–1232.
- GAINA, R., NORDMOEN, C., 2018, “League of legends: A study of early game impact”, in *School of Electronic Engineering and Computer Science, Queen Mary University of London*.
- GEURTS, P., ERNST, D., WEHENKEL, L., 2006, “Extremely randomized trees”, *Machine learning*, v. 63, pp. 3–42.
- GONG, L., FENG, X., YE, D., et al., 2020, “Optmatch: Optimized matchmaking via modeling the high-order interactions on the arena”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2300–2310.
- GORISHNIY, Y. V., RUBACHEV, I., KHRULKOV, V., et al., 2021, “Revisiting Deep Learning Models for Tabular Data”, *ArXiv*, v. abs/2106.11959.
- GRUTZIK, P., HIGGINS, J., TRAN, L., 2017, “Predicting outcomes of professional dota 2 matches”, *Technical Report. Stanford University, Tech. Rep.*
- GU, Y., LIU, Q., ZHANG, K., et al., 2021, “Neuralac: Learning cooperation and competition effects for match outcome prediction”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 35, pp. 4072–4080.
- HANKE, L., CHAIMOWICZ, L., 2017, “A recommender system for hero line-ups in MOBA games”. In: *Proceedings of the 13th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2017*, p. 43 – 49. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85070805983&partnerID=40&md5=d96a5b942dbfcd845c246c05de34ecc2>>. Cited by: 18.

- HERBRICH, R., MINKA, T., GRAEPEL, T., 2006, “TrueSkill™: a Bayesian skill rating system”, *Advances in neural information processing systems*, v. 19.
- HILTSCHER, J., 2021, *Dota 2 and League of Legends: Two of the world’s most popular video games, their cultural aspects, their genre and its definition*. Tese de Doutorado, Dissertation, Bonn, Rheinische Friedrich-Wilhelms-Universität, 2021.
- HITAR-GARCÍA, J. A., MORÁN-FERNÁNDEZ, L., BOLÓN-CANEDO, V., 2023, “Machine Learning Methods for Predicting League of Legends Game Outcome”, *IEEE Transactions on Games*, v. 15, n. 2, pp. 171–181. doi: 10.1109/TG.2022.3153086.
- HO, T. K., 1995, “Random decision forests”. In: *Proceedings of 3rd international conference on document analysis and recognition*, v. 1, pp. 278–282. IEEE.
- HOCHREITER, S., 1991, “Untersuchungen zu dynamischen neuronalen Netzen”, *Diploma, Technische Universitat Munchen*, v. 91, n. 1, pp. 31.
- HODGE, V., DEVLIN, S., SEPHTON, N., et al., 2017, “Win prediction in esports: Mixed-rank match prediction in multi-player online battle arena games”, *arXiv preprint arXiv:1711.06498*.
- HODGE, V. J., DEVLIN, S., SEPHTON, N., et al., 2019, “Win Prediction in Multiplayer Esports: Live Professional Match Prediction”, *IEEE Transactions on Games*, v. 13, n. 4, pp. 368–379. doi: 10.1109/TG.2019.2948469.
- JADOWSKI, R., CUNNINGHAM, S., 2022, “Statistical Models for Predicting Results in Professional League of Legends”, *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, v. 422 LNICST, pp. 138 – 152. doi: 10.1007/978-3-030-95531-1_10. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85125223765&doi=10.1007%2f978-3-030-95531-1_10&partnerID=40&md5=b1232d3625d4fd77dff70ad2c72caffb>. Cited by: 0; All Open Access, Green Open Access, Hybrid Gold Open Access.
- JIANG, J., LERMAN, K., FERRARA, E., 2020, “Individualized Context-Aware Tensor Factorization for Online Games Predictions”. In: *2020 International Conference on Data Mining Workshops (ICDMW)*, pp. 292–299. doi: 10.1109/ICDMW51313.2020.00048.
- JOHANSSON, F., WIKSTROM, J., 2015. “Result prediction by mining replays in dota 2”. .

- JORDAN-VALLVERDÚ, V., PLAZA-NAVAS, M.-A., MARIA RAYA, J., et al., 2024, “The Intellectual Structure of Esports Research”, *Entertainment Computing*, v. 49, pp. 100628. ISSN: 1875-9521. doi: <https://doi.org/10.1016/j.entcom.2023.100628>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1875952123000836>.
- JUNG, C., KIM, H. K., 2022, “Win Prediction from the Snowball Effect Perspectives”. In: *2022 IEEE Games, Entertainment, Media Conference (GEM)*, pp. 1–6. doi: 10.1109/GEM56474.2022.10017891.
- KALYANARAMAN, K., 2014, “To win or not to win? A prediction model to determine the outcome of a DotA2 match”, *A prediction model to determine the outcome of a DotA2 match*.
- KANG, D.-K., KIM, M.-J., 2015, “Poisson Model and Bradley terry Model for predicting multiplayer online battle games”. In: *2015 Seventh International Conference on Ubiquitous and Future Networks*, pp. 882–887. doi: 10.1109/ICUFN.2015.7182671.
- KANG, S.-K., LEE, J.-H., 2020, “An E-sports video highlight generator using win-loss probability model”. In: *Proceedings of the 35th annual ACM symposium on applied computing*, pp. 915–922.
- KATONA, A., SPICK, R., HODGE, V. J., et al., 2019, “Time to Die: Death Prediction in Dota 2 using Deep Learning”. In: *2019 IEEE Conference on Games (CoG)*, pp. 1–8. doi: 10.1109/CIG.2019.8847997.
- KE, C. H., DENG, H., XU, C., et al., 2022, “DOTA 2 match prediction through deep learning team fight models”. In: *2022 IEEE Conference on Games (CoG)*, pp. 96–103. doi: 10.1109/CoG51982.2022.9893647.
- KHO, L. C., KASIHMUDDIN, M. S. M., MANSOR, M., et al., 2020, “Logic Mining in League of Legends.” *Pertanika Journal of Science & Technology*, v. 28, n. 1.
- KIM, D.-H., LEE, C., CHUNG, K.-S., 2020, “A Confidence-Calibrated MOBA Game Winner Predictor”. In: *2020 IEEE Conference on Games (CoG)*, pp. 622–625. doi: 10.1109/CoG47356.2020.9231878.
- KINKADE, N., LIM, K., 2015, “Dota 2 win prediction”, *Univ Calif*, v. 1, pp. 1–13.
- KLEINBAUM, D. G., DIETZ, K., GAIL, M., et al., 2002, *Logistic regression*. New York, NY, USA, Springer.

- LAUREL, K. E. Y., GUTIERREZ, A. N. R., TAN, K. A. S., et al., 2023, “Using Multiple AI Classification Models to Predict the Winner of a League of Legends (LoL) Game Based on its First 10-Minutes of Gameplay”. In: *2023 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*, pp. 693–694. IEEE.
- LEE, H., HWANG, D., KIM, H., et al., 2022, “DraftRec: Personalized Draft Recommendation for Winning in Multi-Player Online Battle Arena Games”. In: *Proceedings of the ACM Web Conference 2022, WWW '22*, p. 3428–3439, New York, NY, USA. Association for Computing Machinery. ISBN: 9781450390965. doi: 10.1145/3485447.3512278. Disponível em: <<https://doi.org/10.1145/3485447.3512278>>.
- LEE, S.-K., HONG, S.-J., YANG, S.-I., 2020, “Predicting Game Outcome in Multiplayer Online Battle Arena Games”. In: *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1261–1263. doi: 10.1109/ICTC49870.2020.9289254.
- LIE, H., LUKAS, D., LIEBIG, J., et al., 2019, “A novel learning-to-rank method for automated camera movement control in e-sports spectating”, *Communications in Computer and Information Science*, v. 996, pp. 149 – 160. doi: 10.1007/978-981-13-6661-1_12. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85063490204&doi=10.1007%2f978-981-13-6661-1_12&partnerID=40&md5=6af99491d718a6800377f8cc8ef12e5a>. Cited by: 1.
- LIN, L., 2016, “League of legends match outcome prediction”, *Comput. Sci. Dept., Univ. Stanford, Stanford, CA, USA, Rep.*
- MA, Y. L., 2021. “League of Legends Diamond Ranked Games (10 min)”. Disponível em: <<https://www.kaggle.com/datasets/bobbyscience/league-of-legends-diamond-ranked-games-10-min>>. Accessed: 2025-08-05.
- MAKAROV, I., SAVOSTYANOV, D., LITVYAKOV, B., et al., 2018, “Predicting winning team and probabilistic ratings in “Dota 2” and “Counter-strike: Global offensive” video games”, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 10716 LNCS, pp. 183 – 196. doi: 10.1007/978-3-319-73013-4_17. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85063490204&doi=10.1007%2f978-3-319-73013-4_17&partnerID=40&md5=6af99491d718a6800377f8cc8ef12e5a>.

0-85039435862&doi=10.1007%2f978-3-319-73013-4_17&partnerID=40&md5=9cd89d41c25348b1183b340067f66cd1>. Cited by: 32.

MARATSI, M. I., ALEXOPOULOS, C., CHARALABIDIS, Y., 2023, “Team Radian or Dire? Comparing the SVM and k-NN Classifiers upon a DotA2 Matches Dataset”. In: *Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics*, pp. 48–52.

MONDAL, J. J., ZAHIN, A., MANAB, M. A., et al., 2022, “Does A Support Role Player really Create Difference towards Triumph? Analyzing Individual Performances of Specific Role Players to Predict Victory in League of Legends”. In: *2022 25th International Conference on Computer and Information Technology (ICCIT)*, pp. 768–773. doi: 10.1109/ICCIT57492.2022.10055689.

NG, A., JORDAN, M., 2001, “On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes”, *Advances in neural information processing systems*, v. 14.

OMAR, H. I., PRAYOGO, M., MULIAWAN, V., et al., 2024, “Finding Feature Importance in Optimized Classification Model: League of Legends Ranked Matches”. In: *2024 IEEE International Conference on Artificial Intelligence and Mechatronics Systems (AIMS)*, pp. 1–5. IEEE.

ONG, H. Y., DEOLALIKAR, S., PENG, M., 2015, “Player behavior and optimal team composition for online multiplayer games”, *arXiv preprint arXiv:1503.02230*.

PEDRASSOLI CHITAYAT, A., KOKKINAKIS, A., PATRA, S., et al., 2020, “WARDS: Modelling the Worth of Vision in MOBA’s”, *Advances in Intelligent Systems and Computing*, v. 1229 AISC, pp. 63 – 81. doi: 10.1007/978-3-030-52246-9_5. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85088497655&doi=10.1007%2f978-3-030-52246-9_5&partnerID=40&md5=6614922ef9b18cde621bb638a0ed7ef4>. Cited by: 1; All Open Access, Green Open Access.

PÉREZ, J., CASTRO, M., LÓPEZ, G., 2023, “Serious Games and AI: Challenges and Opportunities for Computational Social Science”, *IEEE Access*, v. 11, pp. 62051–62061. doi: 10.1109/ACCESS.2023.3286695.

PETERSON, L. E., 2009, “K-nearest neighbor”, *Scholarpedia*, v. 4, n. 2, pp. 1883.

- POROKHNENKO, I., POLEZHAEV, P., SHUKHMAN, A., 2019, “Machine Learning Approaches to Choose Heroes in Dota 2”. In: *2019 24th Conference of Open Innovations Association (FRUCT)*, pp. 345–350. doi: 10.23919/FRUCT.2019.8711985.
- QI, Z., SHU, X., TANG, J., 2018, “DotaNet: Two-Stream Match-Recurrent Neural Networks for Predicting Social Game Result”. In: *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)*, pp. 1–5. doi: 10.1109/BigMM.2018.8499076.
- RENDLE, S., 2010, “Factorization machines”. In: *2010 IEEE International conference on data mining*, pp. 995–1000. IEEE.
- RINGER, C., MISSAOUI, S., HODGE, V. J., et al., 2023, “Time to Die 2: Improved in-game death prediction in Dota 2”, *Machine Learning with Applications*, v. 12, pp. 100466. ISSN: 2666-8270. doi: <https://doi.org/10.1016/j.mlwa.2023.100466>. Disponible em: <<https://www.sciencedirect.com/science/article/pii/S2666827023000191>>.
- RIOULT, F., MÉTIVIER, J.-P., HELLEU, B., et al., 2014, “Mining Tracks of Competitive Video Games”, *AASRI Procedia*, v. 8, pp. 82–87. ISSN: 2212-6716. doi: <https://doi.org/10.1016/j.aasri.2014.08.014>. Disponible em: <<https://www.sciencedirect.com/science/article/pii/S221267161400081X>>. 2014 AASRI Conference on Sports Engineering and Computer Science (SECS 2014).
- RISH, I., OTHERS, 2001, “An empirical study of the naive Bayes classifier”. In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*, v. 3, pp. 41–46.
- SCHUBERT, M., DRACHEN, A., MAHLMANN, T., 2016. “Esports analytics through encounter detection other sports”. .
- SEMENOV, A., ROMOV, P., KOROLEV, S., et al., 2017, “Performance of machine learning algorithms in predicting game outcome from drafts in Dota 2”, *Communications in Computer and Information Science*, v. 661, pp. 26 – 37. doi: 10.1007/978-3-319-52920-2_3. Disponible em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85014300412&doi=10.1007%2f978-3-319-52920-2_3&partnerID=40&md5=2c8ce7c387b6596152969157cff84f48>. Cited by: 43.
- SHEN, Q., 2022, “A machine learning approach to predict the result of League of Legends”. In: *2022 International Conference on Machine Learning and*

- Knowledge Engineering (MLKE)*, pp. 38–45. doi: 10.1109/MLKE55170.2022.00013.
- SHEVCHENKO, K., 2021. “As 17 melhores Arenas de e-Sports do mundo”. <https://senet.cloud/pt/blog/top-esports-arenas-in-the-world>. Acesso em: 2024-09-17.
- SHWARTZ-ZIV, R., ARMON, A., 2022, “Tabular data: Deep learning is not all you need”, *Information Fusion*, v. 81, pp. 84–90.
- SILVA, A. L. C., PAPPA, G. L., CHAIMOWICZ, L., 2018, “Continuous outcome prediction of league of legends competitive matches using recurrent neural networks”, *SBC-proceedings of SBCGames*, pp. 2179–2259.
- SMERDOV, A., SOMOV, A., BURNAEV, E., et al., 2021, “Detecting Video Game Player Burnout With the Use of Sensor Data and Machine Learning”, *IEEE Internet of Things Journal*, v. 8, n. 22, pp. 16680–16691. doi: 10.1109/JIOT.2021.3074740.
- SMIT, R., 2019, *A machine learning approach for recommending items in League of Legends*. Tese de Doutorado, Bachelors thesis.
- SONG, K., ZHANG, T., MA, C., 2015, “Predicting the winning side of DotA2”, *Sl: sn*.
- SOUZA, R. T. D., 2017, *Aplicação de algoritmos classificadores para previsão de vitória em uma partida de League of Legends*. Trabalho de conclusão de curso (engenharia de software), Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brasil. Disponível em: <<https://lume.ufrgs.br/handle/10183/179708>>.
- STANLLY, PUTRA, F. A., QOMARIYAH, N. N., 2022, “DOTA 2 Win Loss Prediction from Item and Hero Data with Machine Learning”. In: *2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 204–209. doi: 10.1109/IAICT55358.2022.9887525.
- TAUD, H., MAS, J., 2018, “Multilayer perceptron (MLP)”. In: *Geomatic approaches for modeling land change scenarios*, Springer, pp. 451–455.
- TIAN, P., LAN, W., ZHANG, X., 2022, “Hero featured learning algorithm for winning rate prediction of Honor of Kings”. In: *2022 IEEE Conference on Games (CoG)*, pp. 322–329. doi: 10.1109/CoG51982.2022.9893634.

- TOT, M., CONSERVA, M., CHITAYAT, A. P., et al., 2021, “What Are You Looking At? Team Fight Prediction Through Player Camera”. In: *2021 IEEE Conference on Games (CoG)*, pp. 1–8. doi: 10.1109/CoG52621.2021.9619038.
- TYRAN, J., CHOMATEK, L., 2021, “Influence of outliers in MOBA games winner prediction”, *Procedia Computer Science*, v. 192, pp. 1973–1981. ISSN: 1877-0509. doi: <https://doi.org/10.1016/j.procs.2021.08.203>. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1877050921016999>>. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 25th International Conference KES2021.
- UDDIN, J., FAHMIDA, I., MOYEEN, S. I., et al., 2022, “DOTA2 Winner Team Prediction based on Stacked Bidirectional LSTM Network”. In: *2022 4th International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)*, pp. 1–5. doi: 10.1109/ICECTE57896.2022.10114527.
- VASWANI, A., SHAZEER, N., PARMAR, N., et al., 2017, “Attention is all you need”, *Advances in neural information processing systems*, v. 30.
- VIGGIATO, M., BEZEMER, C.-P., 2020, “Trouncing in dota 2: An investigation of blowout matches”. In: *Proceedings of the 16th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, AIIDE 2020*, p. 294 – 300. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85102261785&partnerID=40&md5=84b8caf49d044a3e90a53485f53cee58>>. Cited by: 4.
- WANG, K., SHANG, W., 2017, “Outcome prediction of DOTA2 based on Naïve Bayes classifier”. In: *2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 591–593. doi: 10.1109/ICIS.2017.7960061.
- WANG, L., TANG, Y., LIU, J., 2020, “WPQA: A gaming support system based on machine learning and knowledge graph”, *Communications in Computer and Information Science*, v. 1157 CCIS, pp. 191 – 204. doi: 10.1007/978-981-15-3412-6_19. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85081169785&doi=10.1007%2f978-981-15-3412-6_19&partnerID=40&md5=9769dcbdf09359db9c8bbc29514749b4>. Cited by: 2.

- WANG, N., LI, L., XIAO, L., et al., 2018, “Outcome Prediction of DOTA2 Using Machine Learning Methods”. In: *Proceedings of 2018 International Conference on Mathematics and Artificial Intelligence*, ICMAI '18, p. 61–67, New York, NY, USA. Association for Computing Machinery. ISBN: 9781450364201. doi: 10.1145/3208788.3208800. Disponível em: <<https://doi.org/10.1145/3208788.3208800>>.
- WANG, T., 2018, *Predictive Analysis on eSports Games: A Case Study on League of Legends (LoL) eSports Tournaments*. Master’s paper, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA, abr. Disponível em: <https://cdr.lib.unc.edu/concern/masters_papers/8s45qd54c>.
- WANG, W., 2016, *Predicting multiplayer online battle arena (moba) game outcome based on hero draft data*. Tese de Doutorado, Dublin, National College of Ireland.
- WONG, N. H. L., KWOK, S.-I., TAN, J. L., 2022, “Predictive Analytics of First Blood and Match Outcome in Dota 2”. In: *TENCON 2022 - 2022 IEEE Region 10 Conference (TENCON)*, pp. 1–6. doi: 10.1109/TENCON55691.2022.9977786.
- XIA, B., WANG, H., ZHOU, R., 2019, “What Contributes to Success in MOBA Games? An Empirical Study of Defense of the Ancients 2”, *Games and Culture*, v. 14, n. 5, pp. 498 – 522. doi: 10.1177/1555412017710599. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-85046015106&doi=10.1177%2f1555412017710599&partnerID=40&md5=a2dc0f08dc783c51f38dd9a1131b5f7b>>. Cited by: 23.
- YANG, P., ROBERTS, D. L., 2013, “Knowledge discovery for characterizing team success or failure in (A)RTS games”. In: *2013 IEEE Conference on Computational Intelligence in Games (CIG)*, pp. 1–8. doi: 10.1109/CIG.2013.6633645.
- YANG, P., HARRISON, B. E., ROBERTS, D. L., 2014, “Identifying patterns in combat that are predictive of success in MOBA games.” In: *FDG*.
- YANG, Y., QIN, T., LEI, Y.-H., 2016, “Real-time esports match result prediction”, *arXiv preprint arXiv:1701.03162*.
- YANG, Z., PAN, Z., WANG, Y., et al., 2022a, “Interpretable Real-Time Win Prediction for Honor of Kings—A Popular Mobile MOBA Esport”, *IEEE Transactions on Games*, v. 14, n. 4, pp. 589–597. doi: 10.1109/TG.2022.3149044.

- YANG, Z., PAN, Z., WANG, Y., et al., 2022b, “Interpretable Real-Time Win Prediction for Honor of Kings—A Popular Mobile MOBA Esport”, *IEEE Transactions on Games*, v. 14, n. 4, pp. 589–597.
- YANG, Z., WANG, Y., LI, P., et al., 2023, “Predicting Events in MOBA Games: Prediction, Attribution, and Evaluation”, *IEEE Transactions on Games*, v. 15, n. 2, pp. 193–201. doi: 10.1109/TG.2022.3159704.
- YU, L., ZHANG, D., CHEN, X., et al., 2019, “MOBA-Slice: A Time Slice Based Evaluation Framework of Relative Advantage Between Teams in MOBA Games”, *Communications in Computer and Information Science*, v. 1017, pp. 23 – 40. doi: 10.1007/978-3-030-24337-1_2. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85069167235&doi=10.1007%2f978-3-030-24337-1_2&partnerID=40&md5=830e71fb1531684def125c8fe1a13272>. Cited by: 7; All Open Access, Green Open Access.
- ZHANG, C., LIU, J., WANG, Z., et al., 2020, “Look ahead at the first-mile in livecast with crowdsourced highlight prediction”. In: *IEEE infocom 2020-IEEE conference on computer communications*, pp. 1143–1152. IEEE.
- ZHANG, Y., 2021, “Prediction of Esports Game Results Using Early Game Datasets”. In: *ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application*, pp. 1–6.

Apêndice A: Tabelas Complementares

A.1 Revisão de trabalhos sobre predição de vitória em jogos MOBA

Tabela A.1: Síntese dos trabalhos que abordam a predição de vitória em jogos MOBA, destacando o momento da partida considerado, as métricas utilizadas, os métodos de predição aplicados e os principais resultados obtidos.

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
CONLEY e PERRY (2013)	Durante a seleção de personagens	Acurácia de treino e teste	Regressão logística (LR)	Obteve 69,8% de acurácia no teste

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			k-Nearest Neighbour (kNN)	Obteve 70% de acurácia no teste
YANG e ROBERTS (2013)	Após a seleção de personagens	Acurácia, revocação e taxa negativa verdadeira	Árvore de decisão	Obteve 83,5% de acurácia média, realizando a predição com base em dados durante toda a partida
AGARWALA e PEARCE (2014)	Durante a seleção de personagens	Acurácia de treino e teste	Regressão logística	Obteve 62% de acurácia
KALYANARAMAN (2014)	Durante a seleção de personagens	Acurácia de treino e teste, precisão, revocação e F1-Score	Regressão logística	Obteve 69,42% de acurácia no teste
			Regressão logística com algoritmo genético	Obteve 74,1% de acurácia no teste

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
YANG <i>et al.</i> (2014)	Após a seleção de personagens	Acurácia, revocação, taxa negativa verdadeira, e AUC	Abordagem baseada em dados para descobrir padrões de combate	Obteve 80% de acurácia, realizando a predição com base em dados no final partida
RIOULT <i>et al.</i> (2014)	Após a seleção de personagens	AUC, precisão e revocação	Árvores de decisão	Com o uso de indícios topológicos, os métodos conseguiram atingir 95% de AUC, realizando a predição no final da partida
			Classificador Naive Bayes	
			Redes neurais	
			Máquina de vetores de suporte(SVM)	

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
SONG <i>et al.</i> (2015)	Durante a seleção de personagens	Erro de teste e treino	Regressão logística	Regressão logística atingiu 0,39 como menor erro de treino
KINKADE e LIM (2015)	Durante e após a seleção de personagens	Acurácia	Regressão logística	Com dados do final da partida, Regressão logística obteve 99,58% de acurácia, enquanto na etapa de seleção de personagens atingiu 67%
			Random Forest	Com dados do final da partida, Random Forest obteve 99,81% de acurácia, enquanto na etapa de seleção de personagens atingiu 72,9%
JOHANSSON e WIKSTROM (2015)	Após a seleção de personagens	Acurácia	Random Forest	Identificado como melhor método, Random Forest foi o único analisado e obteve 88,83% de ácuracia média, realizando a predição em todo o decorrer das partidas

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			LogitBoost	Não realizaram testes com esse método
			Classificador Naive Bayes	Não realizaram testes com esse método
ONG <i>et al.</i> (2015)	Durante a seleção de personagens	Acurácia de treino e teste, e tempo de execução	Regressão logística(LR)	Obteve 68.8% de acurácia
			Análise discriminante gaussiana(GDA)	Obteve 70,1% de acurácia
			Máquina de vetores de suporte(SVM)	Obteve 70,4% de acurácia
KANG e KIM (2015)	Antes da seleção de personagens	Matrix de confusão de treino	Modelo de Poisson	Obteve 0,69 de acurácia de treino
			Modelo de Bradley-Terry	Obteve 0,39 de acurácia de treino

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
DEJA e MYSLAK (2015)	Após a seleção de personagens	Acurácia, precisão, revocação, F1-Score e matriz de confusão	Random forests	Random forests obteve acurácia de 92.01%, realizando a predição em todo o decorrer das partidas
SCHUBERT <i>et al.</i> (2016)	Após a seleção de personagens	Acurácia	Regressão logística	Regressão logística obteve acurácia de 78%, realizando a predição com base em dados durante toda a partida
WANG (2016)	Durante a seleção de personagens	Acurácia	Regressão logística	Acurácia de 61,04%
			Rede neural perceptron multicamadas	Acurácia de 58,8%
YANG <i>et al.</i> (2016)	Durante e Após a seleção de personagens	Acurácia e tempo de execução	Regressão logística	Durante a seleção de personagens, obteve 71,49% de acurácia, enquanto aos 10 minutos de partida, atingiu por volta de 75%

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Rede neural	Durante a seleção de personagens, obteve 70,46% de acurácia
			Modelo generativo Attribute Sequence Model (ASM)	Aos 10 minutos de partida, obteve por volta de 73% de acurácia
LIN (2016)	Durante e após a seleção de personagens	AUC e erro de teste e treino	Gradient boosted trees com e sem regressão logística	Obteve 95% de acurácia realizando a predição ao final da partida, enquanto durante a seleção obteve 0.433 de erro no teste
CHEN e JOACHIMS (2016)	Durante a seleção de personagens	Log-probabilidade média e acurácia	Modelo Bradley-Terry	Obteve 0,6579 de acurácia
SOUZA (2017)	Após a seleção de personagens	Precisão	Random Forest	Entre os 10 minutos iniciais da partida, obteve 70,66% de precisão

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Regressão logística	Entre os 10 minutos iniciais da partida, obteve 70,68% de precisão
ARYANATA <i>et al.</i> (2017)	Antes da seleção de personagens	Acurácia	Processo de hierarquia analítica (AHP)	AHP obteve 66,67% de acurácia
HODGE <i>et al.</i> (2017)	Durante e após a seleção de personagens	Acurácia	Random Forest	Durante a seleção de personagens, obteve 58,29% de acurácia, enquanto no decorrer dos 20 minutos iniciais da partida, atingiu 76,16% de acurácia
			Regressão logística	Durante a seleção de personagens, obteve 58,75% de acurácia, enquanto no decorrer dos 20 minutos iniciais da partida, atingiu 74,92% de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
CHEN <i>et al.</i> (2017)	Durante a seleção de personagens	Acurácia e desvio padrão	Regressão logística	Para a base de DOTA 2, obteve 52,05% de acurácia, enquanto para a base de LoL, atingiu 60,24% de acurácia
			TrueSkill	Para a base de DOTA 2, obteve 59,66% de acurácia, enquanto para a base de LoL, atingiu 55,24% de acurácia
GRUTZIK <i>et al.</i> (2017)	Durante a seleção de personagens	Acurácia	Máquina de vetores de suporte(SVM)	Obteve 61% de acurácia
			Rede neural	Obteve 60% de acurácia
WANG e SHANG (2017)	Durante a seleção de personagens	Acurácia	Classificador naive bayes	Obteve 58,99% de acurácia
ALMEIDA <i>et al.</i> (2017)	Durante a seleção de personagens	Precisão média e tempo de execução	Classificador Naive Bayes	Obteve 76,33% de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			k-Nearest Neighbour (kNN)	Obteve 69,55% de acurácia
			Árvore de decisão	Obteve 69,26% de acurácia
ANDONO <i>et al.</i> (2017)	Durante a seleção de personagens	Acurácia	Classificador naive bayes com Adaboost	Obteve 80% de acurácia
SEMENOV <i>et al.</i> (2017)	Durante a seleção de personagens	AUC e perda logarítmica	Classificador Naive Bayes	Obteve 68,5% de AUC
			Regressão logística	Obteve 68,8% de AUC
			Factorization Machine	Obteve 70,6% de AUC
			Gradient boosted decision tree (GBDT)	Obteve 70,2% de AUC

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
HANKE e CHAIMOWICZ (2017)	Durante a seleção de personagens	Acurácia	Rede neural de perceptrons (MLP) com retropropagação	Obteve 88,63% de acurácia
CHEN <i>et al.</i> (2018b)	Durante a seleção de personagens	AUC e coeficiente de correlação de pearson	Incorporação de avatar de jogo(GAE)	Obteve 71,43% de AUC para a base de dados do DOTA 2
			Regressão logística	Obteve 68,75% de AUC para a base de dados do DOTA 2
			Gradient Boosting Decision Trees (GBDT)	Obteve 70,14% de AUC para a base de dados do DOTA 2
			2-way Factorization Machine (FM)	Obteve 71,43% de AUC para a base de dados do DOTA 2

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
SILVA <i>et al.</i> (2018)	Após a seleção de personagens	Acurácia e desvio padrão	Rede neural Long Short-term Memory (LSTM)	Obteve 71,43% de acurácia analisando dados da partida no intervalo de 10 a 15 minutos
			Rede neural Gated Recurrent Unit (GRU)	Obteve 68,75% de acurácia analisando dados da partida no intervalo de 10 a 15 minutos
			Rede neural	Obteve 71,43% de acurácia analisando dados da partida no intervalo de 10 a 15 minutos
WANG (2018)	Durante e após a seleção de personagens	Acurácia	Regressão logística	Realizaram testes na seleção de personagens e no final da partida, em ambos obtiveram acurácia acima de 50%
			Árvores de decisão	Obteve por volta de 55% de acurácia na seleção de personagens

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
CHEN <i>et al.</i> (2018a)	Durante a seleção de personagens	AUC e acurácia	Rede neural	Obteve 65,34% de acurácia
			Regressão logística	Obteve 63,57% de acurácia
			Gradient Boosting Decision Trees (GBDT)	Obteve 64,17% de acurácia
QI <i>et al.</i> (2018)	Antes da seleção de personagens	Acurácia	Rede neural	Obteve 69,74% de acurácia
			Máquinas de vetores de suporte (SVM)	Obteve 68,95% de acurácia
			Sparse Autoencoder (SAE)	Obteve 68,42% de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Rede neural Gated Recurrent Unit (GRU)	Obteve 72,11% de acurácia
			Rede neural Long Short-term Memory (LSTM)	Obteve 73,68% de acurácia
ANSHORI <i>et al.</i> (2018)	Durante a seleção de personagens	Acurácia	Máquina de vetores de suporte com kernel linear, tendo os parâmetros otimizados por enxame de partículas	Obteve 60,03% de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
WANG <i>et al.</i> (2018)	Durante a seleção de personagens	Precisão, revocação e F1-Score	Regressão logística	Obteve 0,71 de F1-score
			Random Forest	Obteve 0,70 de F1-score
			Máquina de vetores de suporte (SVM)	Obteve 0,72 de F1-score
MAKAROV <i>et al.</i> (2018)	Antes da seleção de personagens	Acurácia, precisão, revocação, F1-Score e AUC	Regressão logística	Para a função de 'Carry' obteve AUC de 0,98
GAINA e NORDMOEN (2018)	Após a seleção de personagens	Acurácia	Random Forest	Todos os metodos obtiveram entre 61% a 62% de acurácia
			Rede neural perceptron multicamadas	
			AdaBoost	
			Árvores de decisão	

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
SMIT (2019)	Após a seleção de personagens	Acurácia	k-Nearest Neighbour (kNN)	
			Classificador Naive Bayes	
			Rede neural perceptron multicamadas	Obteve 81% de acurácia, analizando dados durante toda a partida
			Gradient Boosting Decision Trees (GBDT)	Obteve 81% de acurácia, analizando dados durante toda a partida
			Random Forest	Obteve 80% de acurácia, analizando dados durante toda a partida
			Extra Trees	Obteve 78% de acurácia, analizando dados durante toda a partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
ANI <i>et al.</i> (2019)	Durante e após a seleção de personagens	Acurácia	Random Forest	Obteve 99,75% de acurácia no final da partida, enquanto na seleção de personagens obteve 95,52%
			AdaBoost	Obteve 96,25% de acurácia no final da partida, enquanto na seleção de personagens obteve 57,22%
			Gradient Boosting	Obteve 97,01% de acurácia no final da partida, enquanto na seleção de personagens obteve 65,67%
			Extreme Gradient Boosting (XGBoost)	Obteve 97,21% de acurácia no final da partida, enquanto na seleção de personagens obteve 65,12%
POROKHNENKO <i>et al.</i> (2019)	Durante a seleção de personagens	AUC e tempo de execução	Gradient Boosting	Obteve 0,7409 de AUC

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Random Forest	Obteve 0,7002 de AUC
			Extreme Gradient Boosting (XGBoost)	Obteve 0,7402 de AUC
			Regressão logística	Obteve 0,7739 de AUC
			Máquina de vetores de suporte(SVM)	Obteve 0,7739 de AUC
			CatBoost	Obteve 0,7673 de AUC
			Redes neural	Obteve 0,7740 de AUC
YU <i>et al.</i> (2019)	Após a seleção de personagens	Acurácia, erro quadrático médio e erro absoluto médio	Redes neurais	MOBA-Slice obteve 0,7191 de acurácia média, realizando a predição em toda a duração da partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
XIA <i>et al.</i> (2019)	Antes da seleção de personagens	Taxa de discriminação	Classification and regression tree (CART)	A taxa de discriminação correta foi 94,6%
KHO <i>et al.</i> (2020)	Após a seleção de personagens	Erro quadrático médio, erro médio absoluto, erro percentual absoluto médio, tempo e acurácia	K método de análise reversa baseado em satisfatibilidade (kSATRA) com rede neural Hopfield (HNN-2SAT)	HNN-2SAT obteve acurácia de 75% na Série do Campeonato da Liga Norte-Americana, analisando dados durante toda a partida
KANG e LEE (2020)	Após a seleção de personagens	Acurácia, precisão, revocação e F1-Score	Rede neural perceptron multicamadas	Obteve 98,4% de acurácia no final da partida
LEE <i>et al.</i> (2020)	Após a seleção de personagens	Acurácia, precisão, revocação e F1-Score	Random forest	Obteve mais de 70% de acurácia após 15 minutos

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
KIM <i>et al.</i> (2020)	Após a seleção de personagens	Erro de calibração esperado (ECE), erro máximo de calibração (MCE), probabilidade logarítmica negativa (NLL) e acurácia	Rede neural perceptron multicamadas	Obteve 73,81% de acurácia realizando a predição durante toda a partida
JIANG <i>et al.</i> (2020)	Antes da seleção de personagens	AUC, erro quadrático médio e erro quadrático médio normalizado	Fatoração de tensores de incorporações neurais individualizadas com reconhecimento de contexto (NICE)	Obteve 0,953 de AUC, realizando a predição no final da partida
			Rede neural	Obteve 0,945 de AUC, realizando a predição no final da partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Random Forest	Obteve 0,920 de AUC, realizando a predição no final da partida
			Extreme Gradient Boosting (XGBoost)	Obteve 0,948 de AUC, realizando a predição no final da partida
VIGGIATO e BEZEMER (2020)	Durante a seleção de personagens	AUC	Extreme Gradient Boosting (XGBoost)	Obteve 0,86 de AUC
			Random Forest	Obteve 0,74 de AUC
			Regressão logística	Obteve 0,79 de AUC
WANG <i>et al.</i> (2020)	Durante a seleção de personagens	Acurácia	Regressão logística	Obteve 63% de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			k-Nearest Neighbour (kNN)	Obteve 55% de acurácia
			Classificador Naive Bayes	Obteve 55% de acurácia
PEDRASSOLI CHI-TAYAT <i>et al.</i> (2020)	Após a seleção de personagens	Correlação entre atributo e resultado esperado	Regressão logística	Identificou uma relação de 69,3% entre a vitória de um time e o total de 'WARDS' no final da partida
GONG <i>et al.</i> (2020)	Durante a seleção de personagens	Acurácia, erro médio absoluto, raiz do erro quadrático médio e acurácia de transferência	Extreme Gradient Boosting (XGBoost)	Obteve 0,613 de acurácia na base de dados do LoL
			Regressão logística	Obteve 0,599 de acurácia na base de dados do LoL
			BalanceNet	Obteve 0,624 de acurácia na base de dados do LoL

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Blade-Chest	Obteve 0,636 de acurácia na base de dados do LoL
			Factorization Machine	Obteve 0,639 de acurácia na base de dados do LoL
			OptMatch	Obteve 0,661 de acurácia na base de dados do LoL
GU <i>et al.</i> (2021)	Durante a seleção de personagens	AUC e Acurácia	NeuralAC	Obteve 0,6663 de AUC na base de dados do LoL
			Regressão logística	Obteve 0,6347 de AUC na base de dados do LoL
			Modelo Bradley-Terry	Obteve 0,6347 de AUC na base de dados do LoL
			TrueSkill	Obteve 0,6129 de AUC na base de dados do LoL
			LightGBM	Obteve 0,6411 de AUC na base de dados do LoL
			Factorization Machine	Obteve 0,6337 de AUC na base de dados do LoL

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			OptMatch	Obteve 0,6523 de AUC na base de dados do LoL
AKHMEDOV e PHAN (2021)	Após a seleção de personagens	Acurácia	Regressão linear	Obteve 82% de acurácia, realizando predição baseando em dados durante toda a partida
			Rede neural	Obteve 88% de acurácia baseando em dados durante toda a partida
			Rede neural Long Short-term Memory (LSTM)	Obteve 93% de acurácia baseando em dados durante toda a partida
TYRAN e CHOMATEK (2021)	Após a seleção de personagens	Perda de treinamento e validação, e acurácia	Rede neural perceptron multicamadas	Obteve 63% de acurácia após 5 minutos de partida
COSTA <i>et al.</i> (2021)	Durante a seleção de personagens	AUC	Regressão logística	Obteve 0,97 de AUC
			Random Forest	Obteve 0,97 de AUC

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Classificador linear	Não informou os resultados desse método
			Árvores de decisão	Não informou os resultados desse método
			Classificador Naive Bayes	Não informou os resultados desse método
			k-Nearest Neighbour (kNN)	Não informou os resultados desse método
			Máquinas de vetores de suporte (SVM)	Não informou os resultados desse método
HODGE <i>et al.</i> (2019)	Após a seleção de personagens	Acurácia	Random Forest	Obteve 77,51% de acurácia, pegando dados até os 20 minutos de cada partida
			Regressão logística	Obteve 77,35% de acurácia pegando dados até os 20 minutos de cada partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			LightGBM	Obteve 77,46% de acurácia pegando dados até os 20 minutos de cada partida
ZHANG (2021)	Após a seleção de personagens	Acurácia, precisão, revocação e matriz de confusão	k-Nearest Neighbour (kNN)	Obteve 69,1% de acurácia na predição de vitória após os 10 minutos de partida
			Regressão logística	Obteve 73,9% de acurácia na predição de vitória após os 10 minutos de partida
DO <i>et al.</i> (2021)	Durante a seleção de personagens	Acurácia, desvio padrão e erro padrão	Rede neural	Obteve 75,1% de acurácia média
			k-Nearest Neighbour (kNN)	Obteve 72,7% de acurácia média
			Máquina de vetores de suporte (SVM)	Obteve 74,3% de acurácia média
			Random Forest	Obteve 74,7% de acurácia média

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Gradient Boosting	Obteve 75,4% de acurácia média
CHEN <i>et al.</i> (2021)	Durante a seleção de personagens	Acurácia, F1-Score e AUC	Rede neural	Obteve 0,847 de acurácia
			Regressão logística	Obteve 0,725 de acurácia
			Gradient Boosting Decision Trees (GBDT)	Obteve 0,713 de acurácia
CRUZ <i>et al.</i> (2021)	Após a seleção de personagens	Acurácia e desvio padrão	Random Forest	Obteve 73,37% de acurácia
			Árvores de decisão	Obteve 73,76% de acurácia
			k-Nearest Neighbour (kNN)	Obteve 71,66% de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
BAHROLOLLOO <i>et al.</i> (2022)	Antes da seleção de personagens	Acurácia	Heurística de comparação de pontuação entre times	A heurística obteve 86% de acurácia
SHEN (2022)	Após a seleção de personagens	Acurácia de treino e teste	AdaBoost	Obteve 0,7226 de acurácia avaliando os 10 minutos iniciais de cada partida
			Gradient Boosting	Obteve 0,7199 de acurácia avaliando os 10 minutos iniciais de cada partida
			Random Forest	Obteve 0,7077 de acurácia avaliando os 10 minutos iniciais de cada partida
			Extra Tree	Obteve 0,703 de acurácia avaliando os 10 minutos iniciais de cada partida
			Máquina de vetores de suporte (SVM)	Obteve 0,7182 de acurácia avaliando os 10 minutos iniciais de cada partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Classificador Naive Bayes	Obteve 0,725 de acurácia avaliando os 10 minutos iniciais de cada partida
			k-Nearest Neighbour (kNN)	Obteve 0,7206 de acurácia avaliando os 10 minutos iniciais de cada partida
			Regressão logística	Obteve 0,7192 de acurácia avaliando os 10 minutos iniciais de cada partida
			Árvore de decisão	Obteve 0,6302 de acurácia avaliando os 10 minutos iniciais de cada partida
			Classificador de Votação (combinação dos outros métodos)	Obteve 0,7268 de acurácia avaliando os 10 minutos iniciais de cada partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
YANG <i>et al.</i> (2022a)	Após a seleção de personagens	Acurácia e desvio padrão	Two-Stage spatial-temporal network (TSSTN)	Em 10 minutos de partida obteve 78,5% de ácuracia
			Heurística de ouro	Em 10 minutos de partida obteve 78,4% de ácuracia
			Rede neural	Em 10 minutos de partida obteve 79,3% de ácuracia
			Regressão logística	Em 10 minutos de partida obteve 78,2% de ácuracia
			Rede neural Long Short-term Memory (LSTM)	Em 10 minutos de partida obteve 78,4% de ácuracia
WONG <i>et al.</i> (2022)	Durante a seleção de personagens	Acurácia, precisão, revocação e F1-Score	Classificador Naive Bayes	Obteve 44% de acurácia
			Regressão logística	Obteve 53% de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			k-Nearest Neighbour (kNN)	Obteve 50% de acurácia
			Máquinas de vetores de suporte (SVM)	Obteve 54% de acurácia
			Rede neural	Obteve 57% de acurácia
UDDIN <i>et al.</i> (2022)	Após a seleção de personagens	Acurácia, precisão, revocação, F1-Score e AUC	Rede neural Long Short-term Memory (LSTM)	Com dados do final da partida obteve 0,919 de acurácia
			k-Nearest Neighbour (kNN)	Com dados do final da partida obteve 0,731 de acurácia
			Máquinas de vetores de suporte (SVM)	Com dados do final da partida obteve 0,915 de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Random Forest	Com dados do final da partida obteve 0,911 de acurácia
STANLLY <i>et al.</i> (2022)	Após a seleção de personagens	Acurácia, precisão, revocação, tempo de execução, F1-Score e AUC	Árvore de decisão	Obteve 0,82 de F1-Score, realizando a predição com base em dados durante toda a partida
			Random Forest	Obteve 0,91 de F1-Score, realizando a predição com base em dados durante toda a partida
			Extreme Gradient Boosting (XGBoost)	Obteve 0,94 de F1-Score, realizando a predição com base em dados durante toda a partida
JUNG e KIM (2022)	Após a seleção de personagens	Acurácia	Rede neural Long Short-term Memory (LSTM)	Obteve 95,59% de acurácia para indicar qual time está na vantagem no decorrer de 1 minuto de partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Rede neural Gated Recurrent Unit (GRU)	Obteve 95,57% de acurácia para indicar qual time está na vantagem no decorrer de 1 minuto de partida
			Rede neural	Obteve 95,57% de acurácia para indicar qual time está na vantagem no decorrer de 1 minuto de partida
KE <i>et al.</i> (2022)	Após a seleção de personagens	Acurácia	Rede neural Long Short-term Memory (LSTM)	Todos os métodos obtiveram acurácia acima de 70% com base nos 32 minutos iniciais de uma partida
			Rede neural Gated Recurrent Unit (GRU)	
			Regressão logística	
			Random Forest	

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
TIAN <i>et al.</i> (2022)	Após a seleção de personagens	Acurácia	Hero Featured Network (HFN)	Obteve 81,8% de acurácia com dados até 7 minutos da partida
			Rede neural Long Short-term Memory (LSTM)	Obteve 78,1% de acurácia com dados até 7 minutos da partida
			Regressão logística	Obteve 78,1% de acurácia com dados até 7 minutos da partida
			Transformer	Obteve 78,5% de acurácia com dados até 7 minutos da partida
			Two-Stage spatial-temporal network (TSSTN)	Obteve 75,6% de acurácia com dados até 7 minutos da partida
			Máquinas de vetores de suporte (SVM)	Obteve 73,8% de acurácia com dados até 7 minutos da partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
LEE <i>et al.</i> (2022)	Durante a seleção de personagens	Acurácia e erro absoluto médio	Regressão logística	Obteve 0,5750 de acurácia na base de dados do DOTA 2
			Rede neural	Obteve 0,5748 de acurácia na base de dados do DOTA 2
			Factorization Machine	Obteve 0,5716 de acurácia na base de dados do DOTA 2
			OptMatch	Obteve 0,5751 de acurácia na base de dados do DOTA 2
			NeuralAC	Obteve 0,5739 de acurácia na base de dados do DOTA 2
			DraftRec	Obteve 0,5755 de acurácia na base de dados do DOTA 2
BIRANT (2022)	Antes da seleção de personagens	Acurácia, precisão, revocação e F1-Score	Multi-view rank-based random forest (MVRRF)	Obteve 93,32% de acurácia, realizando a predição no final da partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Random Forest	Obteve 86,38% de acurácia, realizando a predição no final da partida
JADOWSKI e CUNNINGHAM (2022)	Antes da seleção de personagens	AUC, coeficiente de correlação de Matthews, F1 Score, perda logarítmica e acurácia	Regressão logística	Obteve 67% de acurácia
MONDAL <i>et al.</i> (2022)	Antes da seleção de personagens	Coefficiente de correlação	Simulação Monte Carlo	Com 296 iterações obteve o coeficiente de correlação próximo de 0,50
YANG <i>et al.</i> (2023)	Após a seleção de personagens	Acurácia	Rede neural Long Short-term Memory (LSTM)	Obteve 0,704 de acurácia média nos diferentes tempos das partidas
			Transformer	Obteve 0,708 de acurácia média nos diferentes tempos das partidas

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
HITAR- GARCÍA <i>et al.</i> (2023)	Antes da seleção de personagens	Acurácia e tempo de execução	Extreme Gradient Boosting (XGBoost)	Obteve 0,6882 de acurácia
			Máquinas de vetores de suporte (SVM)	Obteve 0,6777 de acurácia
			Regressão logística	Obteve 0,6605 de acurácia
			Classificador Naive Bayes	Obteve 0,6896 de acurácia
			k-Nearest Neighbour (kNN)	Obteve 0,6658 de acurácia
			Redes neural	Obteve 0,6407 de acurácia
			Classificador de Votação (combinação de vários métodos)	Obteve 0,6975 de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Classificador por empilhamento (combinação de vários métodos)	Obteve 0,7041 de acurácia
BAHROLOLLOOM <i>et al.</i> (2023)	Antes da seleção de personagens	Acurácia	Heurística baseada na performance anterior de cada jogador em outras partidas	Obteve 86% de acurácia
LAUREL <i>et al.</i> (2023)	Após a seleção de personagens	Valor de validação calculado a partir da matriz de confusão	Rede neural	Obteve 69,98% de validação média com base nos dados provenientes dos 10 minutos iniciais da partida
			Regressão logística	Obteve 72,66% de validação média com base nos dados provenientes dos 10 minutos iniciais da partida

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Máquinas de vetores de suporte (SVM)	Obteve 72,53% de validação média com base nos dados provenientes dos 10 minutos iniciais da partida
			k-Nearest Neighbour (kNN)	Obteve 67,90% de validação média com base nos dados provenientes dos 10 minutos iniciais da partida
			Análise discriminante linear	Obteve 72,55% de validação média com base nos dados provenientes dos 10 minutos iniciais da partida
			Classificador Naive Bayes	Obteve 72,14% de validação média com base nos dados provenientes dos 10 minutos iniciais da partida
MARATSI <i>et al.</i> (2023)	Durante a seleção de personagens	Acurácia	Máquinas de vetores de suporte (SVM)	Obteve 57% de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			k-Nearest Neighbour (kNN)	Obteve 55% de acurácia
ASYHRAF ZAMIR ZAMRI <i>et al.</i> (2024)	Após a seleção de personagens	Acurácia, precisão, revocação e F1-Score	Máquinas de vetores de suporte (SVM)	Os dois métodos obtiveram entre 97,62% e 100% de acurácia ao utilizar ponderação nos atributos.
			Classificador Naive Bayes	
OMAR <i>et al.</i> (2024)	Após a seleção de personagens	Acurácia, precisão, revocação e F1-Score	Random Forest	Obteve 72,47% de acurácia
			Classificador Naive Bayes	Obteve 71,86% de acurácia
			Máquinas de vetores de suporte (SVM)	Obteve 72,42% de acurácia
			k-Nearest Neighbour (kNN)	Obteve 73,18% de acurácia

Continua na próxima página

Continuação da Tabela A.1

Trabalhos	Momento de predição	Métricas	Métodos	Principais resultados
			Regressão logística	Obteve 72,87% de acurácia