



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

Avaliação de Modelos Baseados em Aprendizado de Máquina para Previsão de Comportamento de Robôs Futebolísticos

Autor: Luiz Felipe Rodrigues e Silva
Orientador: Dr. Diogo Caetano Garcia

Brasília, DF
2025



Luiz Felipe Rodrigues e Silva

Avaliação de Modelos Baseados em Aprendizado de Máquina para Previsão de Comportamento de Robôs Futebolísticos

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dr. Diogo Caetano Garcia

Brasília, DF

2025

Luiz Felipe Rodrigues e Silva

Avaliação de Modelos Baseados em Aprendizado de Máquina para Previsão de Comportamento de Robôs Futebolísticos/ Luiz Felipe Rodrigues e Silva. – Brasília, DF, 2025-

51 p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Diogo Caetano Garcia

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2025.

1. Modelagem. 2. Aprendizado de Máquina. I. Dr. Diogo Caetano Garcia. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Avaliação de Modelos Baseados em Aprendizado de Máquina para Previsão de Comportamento de Robôs Futebolísticos

CDU 02:141:005.6

Luiz Felipe Rodrigues e Silva

Avaliação de Modelos Baseados em Aprendizado de Máquina para Previsão de Comportamento de Robôs Futebolísticos

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 14 de fevereiro de 2025:

Dr. Diogo Caetano Garcia
Orientador

**Dr. Cristiano Jacques Miosso
Rodrigues Mendes**
Convidado 1

Dr. Luciano Emídio neves da Fonseca
Convidado 2

Brasília, DF
2025

Este trabalho é dedicado à minha família.

Agradecimentos

Agradeço primeiramente a Deus por ter me permitido chegar até aqui. Aos meus pais e minhas irmãs, pelo amor, incentivo e apoio incondicional ao longo de toda a graduação. Aos meus amigos, colegas e professores, que fizeram parte desta jornada, compartilhando conhecimento, momentos de aprendizado e companheirismo.

*“The hardest part of ending is starting again.”
(Linkin Park, Waiting for the End)*

Resumo

Avanços em inteligência artificial, particularmente em redes neurais profundas (DNNs) e aprendizado por reforço (RL), têm impulsionado a modelagem comportamental de sistemas multiagente em robótica competitiva, demandando avaliações comparativas de arquiteturas para cenários dinâmicos, como a *Small Size League* (SSL) da RoboCup. Este trabalho propõe uma análise sistemática de modelos de aprendizado de máquina para prever comportamentos de robôs futebolísticos, criando uma base metodológica para o desenvolvimento de modelos em cenários semelhantes. A metodologia baseia-se no uso de dados históricos de partidas para treinar e validar modelos preditivos, além de comparar as diferentes arquiteturas empregadas, avaliando sua eficácia na previsão de comportamentos, padrões táticos e interações colaborativas entre agentes. Os resultados empíricos buscam identificar a abordagem mais robusta para integração em sistemas autônomos em tempo real, contribuindo com um *framework* replicável para a modelagem de oponentes em competições de robótica e estabelecendo bases para a otimização estratégica em ambientes com restrições operacionais e computacionais.

Palavras-chaves: Aprendizado de Máquina. Deep Learning. Filtro de Kalman. Estados. Robôs Futebolistas. Small Size League.

Abstract

Advances in artificial intelligence, particularly in deep neural networks (DNNs) and reinforcement learning (RL), have driven the behavioral modeling of multi-agent systems in competitive robotics, requiring comparative evaluations of architectures for dynamic scenarios such as the Small Size League (SSL) of RoboCup. This work proposes a systematic analysis of machine learning models to predict the behavior of soccer-playing robots, creating a methodological foundation for developing models in similar scenarios. The methodology is based on using historical match data to train and validate predictive models, as well as comparing different employed architectures, assessing their effectiveness in predicting behavior, tactical patterns, and collaborative interactions between agents. The empirical results aim to identify the most robust approach for integration into real-time autonomous systems, contributing to a replicable framework for opponent modeling in robotics competitions and establishing foundations for strategic optimization in environments with operational and computational constraints.

Key-words: Machine Learning. Deep Learning. Kalman Filter. States Soccer Robots. Small Size League. .

Lista de ilustrações

Figura 1 – Esquema de um neurônio artificial, onde x_i representa as entradas, w_i os pesos sinápticos, \sum a soma ponderada, f a função de ativação e y a saída.	28
Figura 2 – Exemplo de Arquitetura de uma LSTM	29
Figura 3 – Dimensões do Campo da Divisão B da SSL.	32
Figura 4 – Marcadores Visuais dos Robôs	33
Figura 5 – Robô da liga SSL-EL desenvolvido pela Titans - UnB	34
Figura 6 – Fluxograma Metodológico	35
Figura 7 – Simulação de Partida com os Dados Coletados	36

Lista de tabelas

Tabela 1 – Estados do jogo na SSL RoboCup.	33
Tabela 2 – Descrição dos tipos de dados coletados na extração de logs.	36
Tabela 3 – Cronograma de Atividades - TCC 1	39
Tabela 4 – Cronograma de Atividades - TCC 2	39

Lista de abreviaturas e siglas

CNNs	Convolutional Neural Networks
DQN	Deep Q-Network
IA	Inteligência Artificial
LSTM	Long Short-Term Memory
ML	Machine Learning
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
RNNs	Recurrent Neural Networks
RNAs	Redes Neurais Artificiais
SSL	Small Size League
SSL-EL	Small Size League Entry-Level

Sumário

1	INTRODUÇÃO	23
1.1	Contextualização	23
1.2	Objetivos	24
1.2.1	Objetivos Específicos	24
1.3	Estrutura do Documento	24
2	REFERENCIAL TEÓRICO	27
2.1	Inteligência Artificial	27
2.1.1	Aprendizado de Máquina e Aprendizado Profundo	27
2.1.2	Redes Neurais Artificiais	27
2.1.3	<i>Long Short Term Memory</i>	28
2.1.4	Aprendizado por Reforço	29
2.1.5	<i>Deep Q-Networks</i> (DQN)	30
2.1.6	<i>Policy Gradient</i>	30
2.2	A Competição <i>Small Size League</i> (SSL)	31
2.2.1	Dimensões do Campo	31
2.2.2	Sistema de Visão e Identificação dos Robôs	32
2.2.3	<i>Game Controller</i>	32
2.2.4	Estados do Jogo e Regras	33
2.2.5	Resumo das Regras	34
3	METODOLOGIA	35
3.1	Fluxograma Metodológico	35
3.2	Coleta de Dados Históricos	35
3.3	Pré-processamento e Filtragem	36
3.4	Engenharia de <i>Features</i>	37
3.5	Seleção da Arquitetura	37
3.6	Treinamento do Modelo	37
3.6.1	Validação e Ajuste de Hiperparâmetros	37
3.6.2	Comparação de Resultados	38
3.7	Materiais	38
3.7.1	Jupyter Notebook	38
3.7.2	PyTorch	38
3.7.3	Scipy, Matplotlib e Pandas	38
3.7.4	SSL-GO-TOOLS	38

4	CONCLUSÃO	39
4.1	Cronograma de Atividades do TCC1	39
4.2	Cronograma de Atividades do TCC2	39

APÊNDICES	41
------------------	-----------

APÊNDICE A – PRIMEIRO APÊNDICE	43
--	----

APÊNDICE B – SEGUNDO APÊNDICE	45
---	----

ANEXOS	47
---------------	-----------

ANEXO A – PRIMEIRO ANEXO	49
------------------------------------	----

ANEXO B – SEGUNDO ANEXO	51
-----------------------------------	----

1 Introdução

Este capítulo apresenta a [Contextualização](#) do problema, destacando estudos iniciais sobre modelagem de comportamentos em cenários competitivos, com ênfase em abordagens para robôs futebolísticos. Em seguida, são detalhados a [Justificativa](#) da pesquisa e os [Objetivos](#) propostos. Por fim, descreve-se a [Estrutura do Documento](#), oferecendo uma visão resumida da organização dos capítulos que compõem o trabalho.

1.1 Contextualização

A *Small Size League* (SSL)(??) da RoboCup emerge como um dos principais cenários para a criação de pesquisa em robótica multiagente. A liga foi criada com o objetivo de desenvolver e testar técnicas avançadas de inteligência artificial, controle e visão computacional aplicadas a robôs futebolistas. Um dos desafios centrais nesse cenário é a capacidade de estimar e modelar as ações e estratégias dos oponentes, antecipando seu próximo estado para otimizar a tomada de decisão da equipe.

A modelagem de comportamentos em cenários competitivos, remonta à teoria dos jogos, proposta por John von Neumann e Oskar Morgenstern em 1944 (??). Essa teoria estabeleceu bases matemáticas para análise de interações estratégicas entre agentes, inicialmente aplicada a jogos de soma zero. Com o tempo, a modelagem de oponentes evoluiu, incorporando técnicas de inteligência artificial (IA) e aprendizado de máquina (ML) para prever e adaptar-se a comportamentos dinâmicos em ambientes complexos.

No contexto do futebol de robôs, a modelagem de oponentes é entendida como a capacidade de inferir o estado interno e antecipar ações de agentes adversários, mesmo quando suas intenções e estratégias são parcialmente observáveis (??). Isso envolve desde estimativas de trajetórias até identificação de padrões estratégicos, como táticas agressivas ou defensivas. Sistemas de modelagem podem ser projetados para adaptar-se a oponentes arbitrários ou específicos, definindo estratégias ótimas com base em critérios como eficiência energética, precisão preditiva e tempo de resposta (??).

Técnicas de Aprendizado Profundo vêm ganhando destaque nesse contexto, graças à capacidade de aprender representações hierárquicas e abstrair padrões complexos de grandes volumes de dados (??). Arquiteturas como Redes Neurais Recorrentes (RNNs) e Redes Neurais Convolucionais (CNNs) têm sido aplicadas à previsão de trajetórias e reconhecimento de padrões em sistemas multiagentes (??). Contudo, sua aplicação no futebol de robôs enfrenta desafios como exigência de processamento em tempo real, recursos computacionais limitados e variabilidade comportamental dos adversários.

Algumas abordagens baseadas em *Reinforcement Learning* (RL) também demonstram potencial significativo para tomada de decisão em cenários dinâmicos (??). Modelos como *Deep Q-Network* (DQN) (??), *Proximal Policy Optimization* (PPO) (??) e *Deep Deterministic Policy Gradient* (DDPG) (??) são empregados no treinamento de agentes autônomos em jogos (??) e na robótica (??).

Neste trabalho, propõe-se a avaliação de modelos baseados em aprendizado de máquina para a previsão de comportamentos de robôs adversários usando dados históricos (arquivos de logs) de partidas de futebol da *Small Size League* (SSL) da RoboCup, na categoria B. O objetivo final é estabelecer uma base para a criação de modelos de previsão de comportamento na categoria de entrada da *Small Size League Entry-Level* (SSL-EL) da RoboCup Brasil, categoria na qual a equipe da Universidade de Brasília, os Titans, compete.

1.2 Objetivos

O objetivo deste trabalho é avaliar modelos baseados em aprendizado de máquina para previsão de comportamentos de robôs futebolísticos a partir de dados históricos das partidas da *Small Size League*.

1.2.1 Objetivos Específicos

- **Revisão da literatura:** Realizar um levantamento bibliográfico para identificar abordagens, modelos e técnicas aplicadas na previsão de comportamentos em robótica, com foco no contexto de futebol de robôs futebolísticos.
- **Desenvolvimento de um modelo baseado em aprendizado:** Fazer testes de diferentes arquiteturas baseadas em aprendizado para estimar comportamentos de robôs futebolísticos.
- **Avaliação de desempenho:** Analisar o modelo desenvolvido em termos de precisão, eficiência e robustez, e possibilidade de portabilidade para cenários semelhantes a categoria SSL como por exemplo, a liga SSL-EL.
- **Comparação com abordagens existentes:** Comparar o desempenho do modelo proposto com outras técnicas ou modelos já utilizados no mesmo contexto, destacando avanços e limitações observados.

1.3 Estrutura do Documento

Este documento está organizado da seguinte forma:

- **Capítulo 2:** Apresenta o referencial teórico, abordando os fundamentos de modelagem de oponentes e suas aplicações em robótica.
- **Capítulo 3:** Descreve a metodologia empregada, incluindo os procedimentos e técnicas adotados para o desenvolvimento e avaliação do modelo.
- **Capítulo 4:** Apresenta os resultados obtidos durante o treinamento e avaliação dos modelos, com análises quantitativas e qualitativas.
- **Capítulo 5:** Conclui o trabalho, discutindo os resultados alcançados, suas implicações práticas e propondo para pesquisas futuras.

2 Referencial Teórico

Este capítulo apresenta a teoria estudada para a implementação dos objetivos deste trabalho. Primeiramente, discute os princípios do aprendizado de máquina, aprendizado profundo, redes neurais artificiais e aprendizado por reforço. Por fim, aborda os desafios e trabalhos relacionados em modelos baseados em aprendizado nas competições de futebol de robôs.

2.1 Inteligência Artificial

2.1.1 Aprendizado de Máquina e Aprendizado Profundo

O aprendizado de máquina é um subcampo da inteligência artificial que busca desenvolver algoritmos capazes de aprender padrões a partir de dados, sem serem explicitamente programados (??). Dentro desse campo, o Aprendizado Profundo se destaca por utilizar redes neurais com múltiplas camadas para aprender representações hierárquicas e abstratas dos dados, sendo especialmente eficaz em tarefas complexas, como reconhecimento de padrões e previsão de comportamentos (??).

2.1.2 Redes Neurais Artificiais

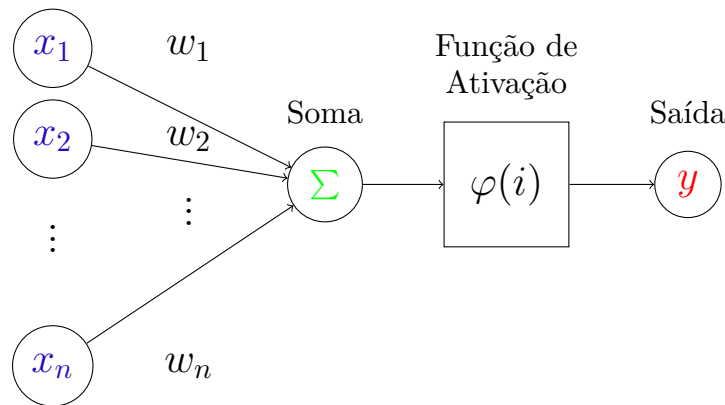
Redes Neurais Artificiais (RNAs) são modelos computacionais inspirados no funcionamento do cérebro humano, compostos por neurônios interconectados organizados em camadas (??). Cada neurônio recebe entradas, processa-as por meio de uma função de ativação e gera uma saída que é propagada para neurônios subsequentes. A estrutura básica de uma RNA inclui:

- **Camada de Entrada:** Responsável por receber os dados brutos do problema. Cada neurônio nesta camada representa uma variável de entrada do modelo, denotada por x_i , onde $i = 1, 2, \dots, n$. A quantidade de neurônios nesta camada é determinada pela dimensionalidade dos dados de entrada.
- **Camadas Ocultas:** Realizam transformações não lineares nos dados, extraindo características relevantes. Cada neurônio recebe um conjunto de entradas ponderadas pelos pesos sinápticos w_{ij} e aplica uma função de ativação para gerar uma saída. O número de camadas ocultas e a quantidade de neurônios em cada uma influenciam a capacidade de aprendizado da rede.

- **Camada de Saída:** Produz a predição final do modelo. A quantidade de neurônios nesta camada depende do tipo de problema: para classificação binária, geralmente há um único neurônio com função de ativação sigmoide; para classificação multiclasse, usa-se a função *softmax* com tantos neurônios quanto categorias; para problemas de regressão, a saída pode ser um único valor real sem ativação ou com uma ativação linear.

A escolha da função de ativação é crucial para o desempenho da rede. Funções como *ReLU* (Rectified Linear Unit) e *sigmoide* são comumente utilizadas para introduzir não-linearidades e permitir que a rede aprenda padrões complexos (??). A Figura 1 ilustra a estrutura básica de um neurônio artificial.

Figura 1 – Esquema de um neurônio artificial, onde x_i representa as entradas, w_i os pesos sinápticos, Σ a soma ponderada, f a função de ativação e y a saída.

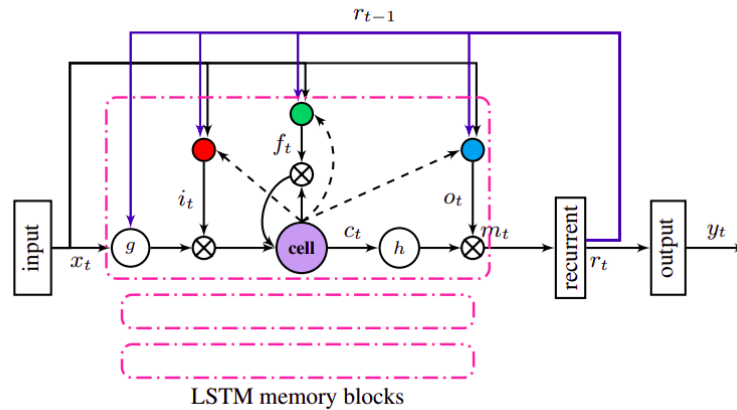


Fonte: Autor

2.1.3 Long Short Term Memory

As Redes Neurais Recorrentes (RNNs) são arquiteturas projetadas para processar dados sequenciais, porém enfrentam limitações como o desaparecimento do gradiente, que restringe o aprendizado de dependências de longo prazo (??). Para contornar esse problema, as redes LSTM (Long Short-Term Memory) foram desenvolvidas com células de memória e portas lógicas que regulam dinamicamente o fluxo de informações a Figura 2 mostra um exemplo de arquitetura desse tipo. Essa arquitetura permite que LSTMs capturem relações temporais complexas em séries de dados, superando RNNs tradicionais em tarefas como previsão de trajetórias e modelagem de sistemas não lineares (??). Estudos comparativos destacam sua eficácia em cenários onde modelos físicos são desconhecidos ou incompletos, como na filtragem adaptativa de sinais ruidosos (??).

Figura 2 – Exemplo de Arquitetura de uma LSTM



Fonte: (??)

2.1.4 Aprendizado por Reforço

O Aprendizado por Reforço (RL) é uma abordagem de aprendizado de máquina na qual um agente aprende a tomar decisões através de interações com o ambiente, buscando maximizar uma recompensa cumulativa (??). No contexto da robótica multiagente, o RL tem sido aplicado para treinar agentes a cooperar e competir em ambientes complexos, como no futebol de robôs (??). O processo de RL é dividido em 4 partes principais:

- **Agente:** Entidade que toma decisões.
- **Ambiente:** Contexto no qual o agente opera.
- **Ações:** Comportamentos que o agente pode executar.
- **Recompensas:** Feedback numérico que guia o aprendizado.

O agente aprende uma política ótima que mapeia estados do ambiente para ações, maximizando a recompensa ao longo do tempo. Técnicas como *Q-learning* e *Deep Q-Networks* (DQNs) têm sido utilizadas para resolver problemas complexos de controle e navegação em robótica (??). No entanto, a velocidade de convergência desses métodos representa um desafio significativo, especialmente em ambientes adversários, como no domínio do *RoboCup Soccer*. Se a taxa de aprendizado do agente for muito baixa, o algoritmo pode exigir muitas iterações para aprender a tarefa de forma eficiente, o que pode resultar em derrotas antes que o agente atinja uma política ótima. Para superar esse problema, técnicas como estimativas parciais do estado podem ser empregadas quando parte da dinâmica do ambiente é conhecida ou fácil de modelar, acelerando a convergência do *Q-learning* (??).

2.1.5 Deep Q-Networks (DQN)

O DQN é uma extensão do *Q-learning* que utiliza uma rede neural para aproximar a função de valor Q . A função Q é definida como:

$$Q(s, a) = \mathbb{E} \left[R_t + \gamma \max_{a'} Q(s', a') \mid s, a \right]$$

onde:

- s é o estado atual.
- a é a ação tomada.
- R_t é a recompensa imediata.
- γ é o fator de desconto (entre 0 e 1).
- s' é o próximo estado.
- a' é a próxima ação.

A rede neural é treinada para minimizar a função de perda:

$$L(\theta) = \mathbb{E} \left[\left(R_t + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta) \right)^2 \right]$$

onde:

- θ são os parâmetros da rede neural.
- θ^- são os parâmetros de uma rede alvo (*target network*), que é atualizada periodicamente para estabilizar o treinamento.

2.1.6 Policy Gradient

O método *Policy Gradient* busca otimizar diretamente a política $\pi(a|s; \theta)$, que é uma distribuição de probabilidade sobre as ações dado um estado. A função objetivo é maximizar a recompensa esperada:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^T \gamma^t R_t \right]$$

onde τ é uma trajetória gerada pela política π_θ . O gradiente da função objetivo é dado por:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tau \sim \pi_{\theta}} \left[\sum_{t=0}^T \nabla_{\theta} \log \pi(a_t | s_t; \theta) G_t \right]$$

onde $G_t = \sum_{k=t}^T \gamma^{k-t} R_k$ é o retorno descontado a partir do tempo t .

2.2 A Competição *Small Size League* (SSL)

A *Small Size League* (SSL) é uma das categorias da *RoboCup*, uma competição internacional que visa promover avanços em robótica e inteligência artificial através de desafios práticos e competitivos (??). Na SSL, dois times, compostos por 6 ou 11 robôs autônomos, competem em partidas que simulam jogos de futebol. A principal característica dessa categoria é a autonomia dos robôs, que são controlados remotamente por um computador central, responsável por processar informações do ambiente e enviar comandos para a execução de estratégias predeterminadas (??).

O ambiente de jogo é monitorado por um sistema de câmeras posicionadas acima do campo, que capturam imagens em tempo real. Essas imagens são processadas por um sistema de visão computacional unificado, que identifica a posição dos robôs, da bola e de outros elementos relevantes. Com base nessas informações, o sistema central calcula as ações ideais para cada robô, permitindo que eles se movimentem e interajam de forma autônoma no campo (??).

Os robôs utilizados na competição SSL são equipados com rodas omnidirecionais, essa configuração permite realizar movimentos em qualquer direção sem a necessidade de reorientação (??). O controle do movimento é feito ajustando as velocidades de cada roda, o que possibilita alterar a velocidade linear, angular e a orientação do robô no campo.

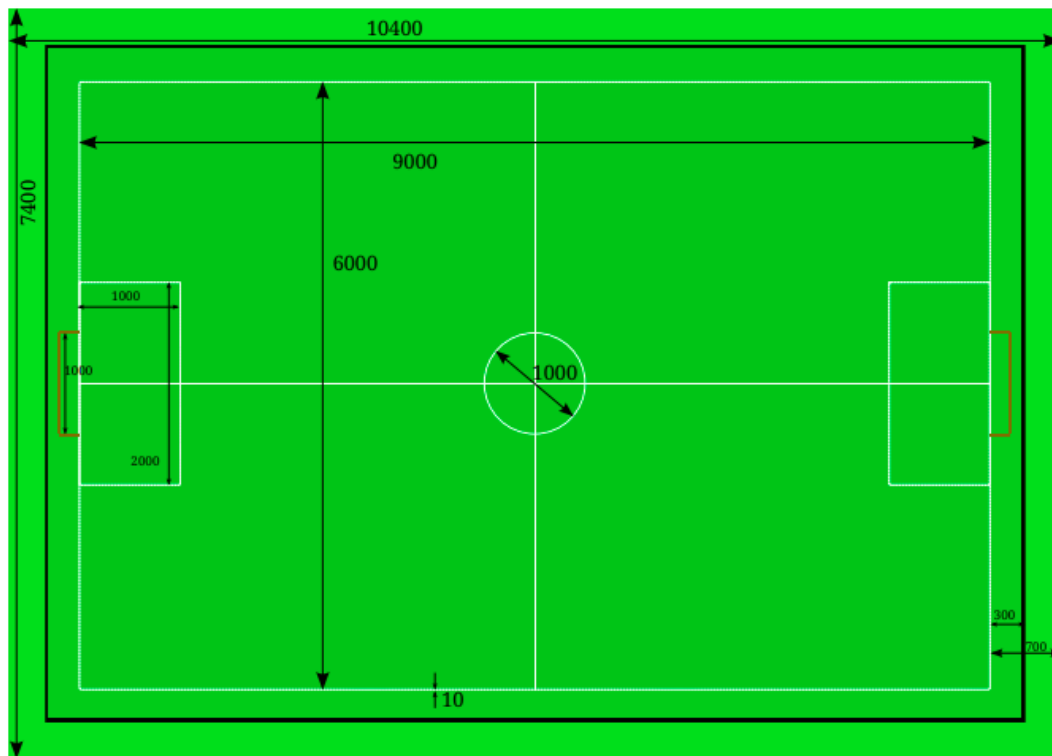
A competição destaca-se pelo uso de tecnologias avançadas em robótica, sistemas embarcados e visão computacional, proporcionando um ambiente propício para o desenvolvimento e aplicação de técnicas de controle e inteligência artificial (??).

2.2.1 Dimensões do Campo

O campo utilizado na Divisão B da SSL possui dimensões padronizadas de 10,4 metros de comprimento por 7,4 metros de largura, com uma área de jogo efetiva de 9 metros por 6 metros. As marcações do campo, como as linhas de fundo, laterais, áreas do gol e círculo central, seguem padrões definidos pela organização da competição. Pequenas variações nas dimensões lineares (até $\pm 10\%$) são permitidas para acomodar diferentes configurações de arena (??).

A Figura 3 ilustra as dimensões e marcações do campo da Divisão B, conforme estabelecido pelas regras oficiais da RoboCup(??).

Figura 3 – Dimensões do Campo da Divisão B da SSL.



Fonte: (??)

2.2.2 Sistema de Visão e Identificação dos Robôs

O sistema de visão utilizado na SSL é unificado e padronizado para todas as equipes, garantindo compatibilidade e justiça competitiva (??). Esse sistema é composto por câmeras de alta resolução posicionadas acima do campo, que capturam imagens em tempo real. As imagens são processadas por algoritmos de visão computacional para identificar a posição dos robôs, da bola e de outros elementos relevantes.

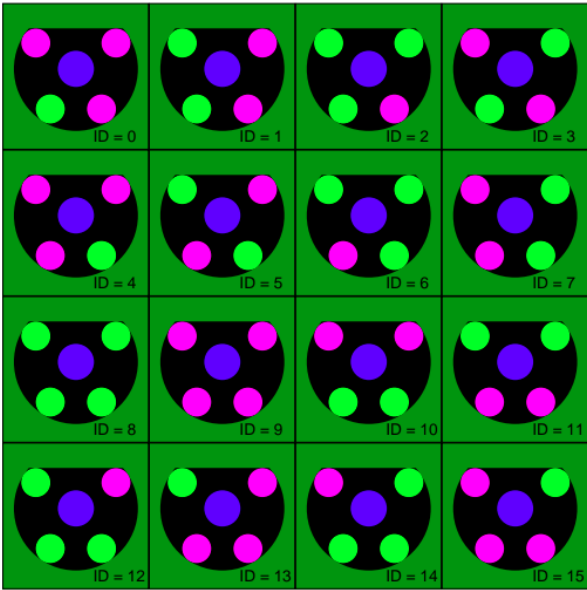
Cada robô é equipado com marcadores visuais padronizados, que permitem sua identificação única durante as partidas. Esses marcadores consistem em padrões de cores específicos, conforme ilustrado na Figura 4. A diferenciação entre os times é feita pela cor central do marcador, que pode ser azul ou amarela. As equipes devem concordar previamente sobre a escolha das cores, não sendo permitido que ambos os times utilizem a mesma cor central (??).

2.2.3 Game Controller

O *Game Controller* da RoboCup Small Size League (SSL) é um sistema automatizado que atua como árbitro central, gerenciando o fluxo do jogo e garantindo conformidade com as regras. Ele controla estados do jogo (e.g., kickoff, penalty), inicia/pausa partidas, define o placar e aplica penalidades por infrações (e.g., invasão de área), comunicando-

]

Figura 4 – Marcadores Visuais dos Robôs



Fonte: (??)

se com as equipes via protocolo de rede UDP. Sua função é padronizar a execução da partida, substituindo a intervenção humana e permitindo que os robôs operem de forma totalmente autônoma.

Integrado ao sistema SSL-Vision (que rastreia posições de robôs e bola), o Game Controller fornece dados em tempo real para as equipes, que devem interpretar seus comandos para ajustar estratégias dinâmicas. Essa integração é essencial para manter a justiça e a sincronia em ambientes competitivos, onde decisões milissegundos e precisão são críticas.

2.2.4 Estados do Jogo e Regras

Os estados do jogo são controlados pelo *Game Controller* e indicam as fases operacionais da partida. Abaixo estão os principais estados do jogo:

Estado	Descrição
Halt	Jogo parado (início ou após infrações graves).
Stop	Jogo pausado (ex.: bola saiu do campo).
Running	Jogo em andamento, com robôs operando livremente.
Kickoff	Reinício no centro do campo após gol ou início de período.
Free Kick	Cobrança de falta (defensores a 0.5 m da bola).
Penalty Kick	Cobrança de pênalti (apenas cobrador e goleiro na área).
Timeout	Pausa solicitada por uma equipe (limitado a 2 por partida).

Tabela 1 – Estados do jogo na SSL RoboCup.

2.2.5 Resumo das Regras

Cada equipe pode ter até 6 robôs, obedecendo às dimensões máximas de 22 cm de altura e 18 cm, a Figura 5 mostra um exemplo de uma categoria de entrada, a SSL-EL da Robocup Brasil desenvolvido pela equipe da Universidade de Brasília Titans.

Figura 5 – Robô da liga SSL-EL desenvolvido pela Titans - UnB



Fonte: Autor

A bola deve permanecer em movimento contínuo, e tocar robôs parados resulta em falta. Um gol é validado apenas quando a bola ultrapassa completamente a linha do gol, incluindo situações de chutes diretos ou deflexões. As infrações são categorizadas como colisões violentas (punidas com cartão amarelo ou vermelho), toque manual (que paralisa o jogo imediatamente) ou posicionamento ilegal (com remoção temporária do robô). A partida é dividida em dois períodos de 6 minutos, com intervalo de 3 minutos, e substituições são permitidas apenas durante os estados *Stop* ou *Halt*, realizadas fora da área técnica para evitar interferência no jogo.

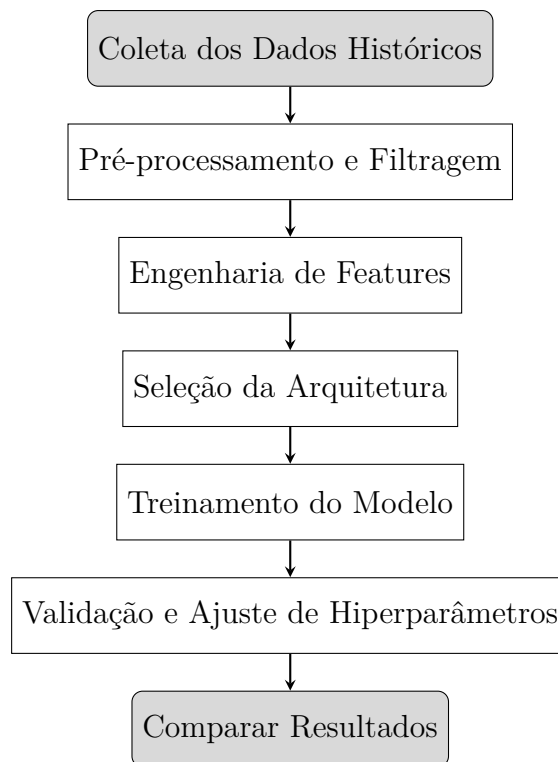
3 Metodologia

Este capítulo descreve a metodologia proposta para o desenvolvimento deste trabalho. São inicialmente apresentadas o [Fluxo do Metodológico](#) e os [Materiais](#) adotados.

3.1 Fluxograma Metodológico

De forma resumida, pretende-se seguir o seguinte fluxo de trabalho conforme mostrado na [Figura 6](#).

Figura 6 – Fluxograma Metodológico



Fonte: Autor

3.2 Coleta de Dados Históricos

Os dados utilizados são provenientes do repositório oficial da RoboCup, que armazena logs públicos de partidas da SSL-B em arquivos compactados (.gz). Cada um desses logs contém informações detalhadas sobre as posições dos robôs, estado da bola, decisões do juiz e contexto do jogo. Para extração e conversão dos dados em formatos tabulares

(CSV), utiliza-se a ferramenta *ssl-go-tools*, que permite acessar e estruturar as informações de forma adequada para análise. Foram coletados 370.486 mil dados de treino. A estrutura dos dados já processados segue o seguinte formato:

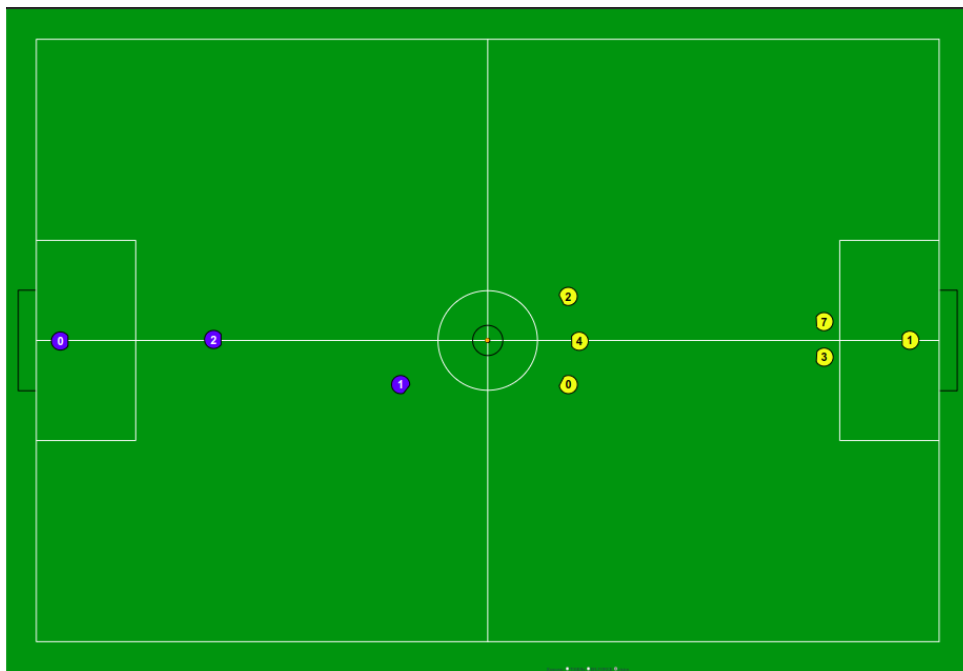
Tabela 2 – Descrição dos tipos de dados coletados na extração de logs.

Tipo de Dado	Descrição	Unidade
Timestamp	Marca temporal da coleta de dados	s
Pixel X	Posição horizontal do robô	pixels
Pixel Y	Posição vertical do robô em	pixels
X	Posição do robô no campo (coordenada X)	mm
Y	Posição do robô no campo (coordenada Y)	mm
Orientação	Orientação do robô no campo	rad

Fonte: Autor

A imagem a seguir mostra o processamento em tempo real de uma partida simulada a partir dos dados coletados utilizando um simulador.

Figura 7 – Simulação de Partida com os Dados Coletados



Fonte: Autor

3.3 Pré-processamento e Filtragem

Na etapa de pré-processamento, pretende-se aplicar um Filtro de Kalman para criar trajetórias contínuas da bola e dos robôs, visto que os dados são obtidos de forma discreta. Feito isso, os dados serão então normalizados para o intervalo $[-1, 1]$, filtram-se registros irrelevantes para manter apenas sequências de ação contínua.

3.4 Engenharia de Features

Os parâmetros de entrada do modelo representados pelas coordenadas de posição $x(t)$ e $y(t)$, o tempo t e a orientação angular $\theta(t)$, além das velocidades linear e angular dos robôs. A velocidade linear v é obtida a partir das derivadas temporais da posição, enquanto a velocidade angular ω é derivada da variação da orientação ao longo do tempo. As equações são dadas por:

$$v_x(t) = \frac{dx}{dt}, \quad v_y(t) = \frac{dy}{dt} \quad (3.1)$$

$$v(t) = \sqrt{v_x^2(t) + v_y^2(t)} \quad (3.2)$$

$$\omega(t) = \frac{d\theta}{dt} \quad (3.3)$$

onde $v_x(t)$ e $v_y(t)$ representam as componentes da velocidade linear nos eixos cartesianos, e $\omega(t)$ é a velocidade angular do robô.

3.5 Seleção da Arquitetura

A seleção de arquiteturas seguirá uma abordagem hierárquica, iniciando com modelos lineares clássicos como *baseline* e evoluindo para arquiteturas profundas especializadas em dados sequenciais. A progressão considerou um modelo inicial para estabelecer desempenho mínimo esperado. Em seguida, pretende-se fazer o uso das redes Recorrentes (LSTM/GRU), a escolha justifica-se pela natureza temporal dos dados e necessidade de balancear complexidade computacional com precisão preditiva.

3.6 Treinamento do Modelo

O treinamento será realizado no *framework* *PyTorch*, com aceleração em GPU via CUDA. Os dados são divididos em conjuntos de treinamento selecionados de forma aleatória sendo para treinamento (70%), validação (15%) e teste (15%), com *early stopping* para evitar *overfitting*.

3.6.1 Validação e Ajuste de Hiperparâmetros

O modelo é avaliado por métricas como Erro Absoluto Médio (MAE) e Erro Quadrático Médio Raiz (RMSE). Uma busca em grade (grid search) otimiza hiperparâmetros críticos, como taxa de aprendizado e profundidade da rede.

3.6.2 Comparação de Resultados

Os resultados são comparados com algoritmos clássicos, como regressão linear e filtro de partículas, que servem como *baseline*. O critério de sucesso inclui superioridade em RMSE e eficiência computacional.

3.7 Materiais

3.7.1 Jupyter Notebook

O Jupyter Notebook é um ambiente interativo de código aberto para criação e compartilhamento de documentos que combinam código executável (e.g., Python, R), visualizações de dados, texto explicativo e equações. Sua estrutura baseada em células facilita a exploração iterativa de dados, prototipagem de algoritmos e documentação de fluxos de trabalho científicos.

3.7.2 PyTorch

O PyTorch é uma biblioteca de aprendizado profundo baseada em tensores, destacando-se pela flexibilidade e grafos computacionais dinâmicos. Desenvolvido pelo Facebook AI Research (FAIR), permite a construção e treinamento de redes neurais, com suporte a aceleração de cálculos por GPU/TPU.

3.7.3 Scipy, Matplotlib e Pandas

O Scipy oferece módulos para álgebra linear, otimização e processamento de sinais. O Matplotlib é utilizado para visualização de dados 2D/3D, enquanto o Pandas proporciona estruturas de dados eficientes para manipulação de dados tabulares, integrando-se a fluxos de trabalho de análise e machine learning.

3.7.4 SSL-GO-TOOLS

O ssl-go-tool é um conjunto de pacotes em Go que facilita tarefas na RoboCup Small Size League (SSL). Oferece funcionalidades para leitura, escrita, envio, recebimento e análise de mensagens do SSL-Vision e SSL-GameController, sendo utilizado também para extrair dados de logs de partidas e rodar partidas anteriores através de um log player.

4 Conclusão

O presente trabalho

4.1 Cronograma de Atividades do TCC1

Tabela 3 – Cronograma de Atividades - TCC 1

Descrição	Prazo
Definição do tema e objetivos do TCC	Outubro/2024
Revisão bibliográfica e levantamento de referências	Outubro/2024
Elaboração do problema de pesquisa e justificativa	Novembro/2024
Definição da metodologia e coleta de dados	Dezembro/2024
Redação do projeto de TCC	Janeiro/2025
Apresentação do projeto de TCC para banca	Fevereiro/2025

4.2 Cronograma de Atividades do TCC2

Tabela 4 – Cronograma de Atividades - TCC 2

Descrição	Prazo
Filtragem e Normalização dos Dados	Fe vereio/2025
Escolha do Modelo e Implementação da Rede Neural	Abril/2025
Implementação da Rede Neural	Maio/2025
Redação dos capítulos 4 e 5 (resultados e conclusão)	Junho/2025
Revisão final e formatação do documento	Julho/2025
Apresentação do TCC para banca	Agosto/2025

Apêndices

APÊNDICE A – Primeiro Apêndice

APÊNDICE B – Segundo Apêndice

Anexos

ANEXO A – Primeiro Anexo

Texto do primeiro anexo.

ANEXO B – Segundo Anexo

Texto do segundo anexo.