Tolerância a Falhas com Aprendizado por Reforço para Tomada de Decisão em Cenários Distribuídos *

Vinícius Rodrigues Oliveira[1], **Júnia Maísa Oliveira**[12], **Daniel Macedo**[1], **and José Marcos Nogueira**[1]

[1]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brasil

[2]Department of Electrical, Electronic, and Information Engineering (DEI)
University of Bologna, Italy

{vinicius, damacedo, jmarcos}@dcc.ufmg.br, junia.deoliveira@unibo.it

Resumo. *Este trabalho apresenta contribuições ao estudo do impacto da latência em sistemas distribuídos com aprendizado por reforço. Propostas práticas incluem a repetição da última ação ou a execução de ações aleatórias para mitigar falhas de temporização. A eficácia dessas estratégias é avaliada para diferentes níveis de latência, sendo critérios considerados o tempo de convergência no treinamento, a tolerância a falhas ou atrasos e as estratégias de ação. O desempenho do aprendizado por reforço é analisado em contextos geograficamente distribuídos, considerando condições de redes de comunicação brasileiras. Modificações na biblioteca* Stable Baselines3 *simulam condições reais de comunicação, aumentando a reprodutibilidade dos resultados. Diretrizes práticas são fornecidas para aplicações em drones autônomos, redes industriais e dispositivos IoT, destacando particularidades regionais do Brasil.*

Abstract. *This work presents contributions to the study of the impact of latency in distributed systems using reinforcement learning. Practical proposals include repeating the last action or executing random actions to mitigate timing failures. The effectiveness of these strategies is evaluated across different latency levels. Furthermore, the performance of reinforcement learning is analyzed in geographically distributed contexts, considering Brazilian network conditions. Modifications to the* Stable Baselines3 *library simulate real communication conditions, enhancing the reproducibility of the results. Practical guidelines are provided for applications in autonomous drones, industrial networks, and IoT devices, highlighting the regional particularities of Brazil.*

1. Introdução

A inteligência artificial impacta de maneira positiva e significativa os sistemas autônomos inteligentes, promovendo avanços em diversas áreas, como transporte, logística, manufatura e monitoramento [Azar et al. 2021]. Sistemas autônomos podem ser definidos como aqueles que operam de forma independente, sem intervenção humana direta, em ambientes dinâmicos e frequentemente imprevisíveis [Samanta et al. 2018]. Esses sistemas encontram aplicação em variados cenários [Shakhatreh et al. 2019], como otimização de processos industriais, entrega de produtos e monitoramento de áreas específicas. Entretanto, desenvolver

*Os autores agradecem às seguintes instituições brasileiras pelo apoio: MPMG - Ministério Público de Minas Gerais, CAPES, CNPq, FAPEMIG e FAPESP/MCTI (processos 2023/13518-0, 2020/05182-3 e 2018/23097-3).

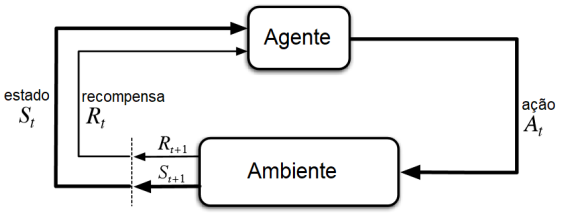

Figura 1. Fluxo do aprendizado por reforço.

soluções autônomas para operar de forma eficaz em tais cenários apresenta desafios devido à complexidade e à dinamicidade dos ambientes envolvidos. Métodos de inteligência artificial, como aprendizado por reforço, têm sido amplamente utilizados para lidar com esses desafios, possibilitando adaptações e tomadas de decisão em tempo real [Azar et al. 2021].

A Figura 1 apresenta o fluxo de interação entre um agente e o ambiente em que está inserido, em um processo de aprendizado por reforço, técnica amplamente aplicada em sistemas autônomos. Nesse modelo, o agente toma decisões com base no estado atual do ambiente (S_t), seleciona uma ação (A_t) e a envia ao ambiente. Em resposta, após executar a ação, o ambiente fornece uma nova observação do estado (S_{t+1}) e uma recompensa (R_{t+1}). Segundo [Li et al. 2019], o treinamento de modelos de aprendizado por reforço pressupõe interações repetidas entre o agente e o ambiente ao longo de várias etapas. Inicialmente, o agente recebe um estado do ambiente e seleciona uma ação com base em uma política que mapeia estados para ações. Após a execução da ação, o ambiente retorna o próximo estado e uma recompensa, formando trajetórias usadas para aprimorar a política. O objetivo do agente é maximizar a recompensa acumulada.

Dessa forma, no treinamento de modelos de aprendizado por reforço, um agente interage com um determinado ambiente repetidamente ao longo de inúmeras etapas. No início, o agente recebe um estado inicial do ambiente e então toma uma ação com base em um conjunto de regras, denominadas políticas, que mapeiam o estado atual para uma ação de um possível conjunto de ações. Após a ação selecionada e executada surtir efeito no ambiente, o próximo estado é gerado e uma recompensa é passada para o agente. Esses estados, ações e recompensas são coletados para formar uma trajetória, que é, então, utilizada para melhorar a política de recompensa. O objetivo do agente é construir políticas ou regras que maximizem as recompensas [Li et al. 2019], para que, assim, o sistema melhore seu desempenho.

É amplamente reconhecido na literatura que atrasos impactam o correto funcionamento de um serviço. Esses atrasos são denominados falhas de temporização [Avizienis et al. 2004]. Estudos recentes, como os de [Bernardo et al. 2022] e [Li et al. 2019], destacam que as falhas de temporização, que em outras palavras pode ser dito como atraso na entrega de dados, impactam no desempenho do treinamento de algoritmos de aprendizado por reforço. Diversos estudos em aprendizado por reforço têm explorado sua aplicação em cenários industriais e redes de dispositivos inteligentes [Jiang et al. 2020], [Bernardo et al. 2022], [Cheng et al. 2021], [Wu et al. 2021], [Bayerlein et al. 2020]. Há lacunas significativas na análise do impacto da latência da rede no desempenho de sistemas distribuídos, especialmente quando o agente e o ambiente estão geograficamente separados.

Os autores em [Szarski and Chauhan 2021] propuseram a utilização de aprendizado por reforço para o controle de temperatura no processo de fabricação de plástico reforçado com fibra de carbono, haja vista que a modificação das variáveis que afetam a transferência de calor durante o processamento do compósito é uma forma de otimizar o processo de fabricação. Um exemplo adicional da aplicação do aprendizado por reforço ocorre na indústria de energia

elétrica, que enfrenta o desafio de atender à demanda de energia de maneira eficiente, garantindo redes confiáveis e custos reduzidos. Os autores em [Lu et al. 2020] propõem um esquema de resposta à demanda baseado no aprendizado por reforço profundo de múltiplos agentes para o gerenciamento energético de sistemas de manufatura discretos. Entretanto, no nosso entendimento, a literatura existente ignora as possíveis falhas de temporização e sua influência direta na confiabilidade e eficácia de sistemas autônomos e distribuídos.

Diante dessas lacunas, este trabalho inova ao investigar estratégias para mitigar falhas de temporização em ambientes descentralizados, aplicando aprendizado por reforço em cenários distribuídos que refletem as particularidades da infraestrutura de redes brasileiras. Para isso, são exploradas duas abordagens principais: (i) o ambiente retém a última ação executada, repetindo-a em caso de falha na comunicação; (ii) o ambiente seleciona aleatoriamente uma nova ação para execução. Essas estratégias são avaliadas em um contexto no qual o agente e o ambiente estão geograficamente separados, possibilitando uma análise dos impactos da latência e do jitter nas tomadas de decisão.

O estudo abrange as cinco regiões do Brasil – Centro-Oeste, Nordeste, Norte, Sudeste e Sul – utilizando métricas reais de latência (tempo necessário para a transmissão de dados entre o agente e o ambiente) e jitter (variação no tempo de transmissão dos dados) para representar os desafios enfrentados em sistemas distribuídos quando operados a partir do território nacional. A abordagem apresentada neste trabalho é baseada no pressuposto de que não existe tolerância a falhas sem redundância [Gärtner 1999]. Assim, com o resultado dos experimentos, pretendemos responder às seguintes questões:

- *Qual o efeito da latência na temporização no aprendizado por reforço?*
- *Qual é o impacto de uma ação após definir um limite de tempo para a chegada do pacote?*
- *Qual é a viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro onde o agente e o ambiente estão em localidades diferentes?*

Este trabalho apresenta contribuições para o estudo do impacto da latência em sistemas distribuídos utilizando aprendizado por reforço. Primeiramente, propõe estratégias práticas para mitigar falhas de temporização, como a repetição da última ação ou a execução de ações aleatórias, e avalia sua eficácia em cenários com diferentes níveis de latência. Em segundo lugar, apresenta uma análise do desempenho do aprendizado por reforço em um contexto geograficamente distribuído, utilizando como base as condições de rede brasileiras para demonstrar sua aplicabilidade em cenários reais cuja regiões possuem infraestruturas de rede diferentes. Além disso, modifica a biblioteca *Stable Baselines3*[1] para simular condições reais de comunicação em rede, permitindo maior reprodutibilidade e aplicabilidade dos resultados. Por fim, este trabalho destaca particularidades regionais do Brasil, fornecendo diretrizes práticas para a implementação de aprendizado por reforço em sistemas distribuídos, com implicações diretas para aplicações em áreas como drones autônomos, redes industriais e dispositivos IoT.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados, as contribuições existentes na literatura e as lacunas abordadas neste estudo; a Seção 3 descreve os procedimentos executados neste trabalho; a Seção 4 apresenta o desenvolvimento do trabalho, com foco na implementação das estratégias propostas e nos experimentos realizados; na Seção 5 são apresentados e analisados os resultados obtidos, com uma discussão sobre suas implicações; por fim, a Seção 6 encerra o artigo, sintetizando as principais conclusões e sugerindo direções para trabalhos futuros.

[1] `https://jmlr.org/papers/volume22/20-1364/20-1364.pdf`

2. Trabalhos Relacionados

Na literatura, o aprendizado por reforço é aplicado de diversas formas em sistemas autônomos para tomada de decisão adaptativa. Esta seção apresenta alguns exemplos de trabalhos que utilizam essa abordagem para abordar desafios em ambientes dinâmicos e complexos. [Bayerlein et al. 2020] aborda o aprendizado por reforço em que o agente, neste caso o VANT, interage com ambiente coletando informações de dispositivos de Internet das Coisas (IoT) para planejar sua trajetória, adaptando-se aos obstáculos dinâmicos presentes um ambiente urbano. No entanto, o trabalho não aborda a comunicação de redes entre o agente e o ambiente.

Já em [Jiang et al. 2020], é proposta uma estrutura assistida por IA para redes sem fio para a otimização da latência de informações, considerando operação de multiagentes do ponto de vista de aprendizado por reforço. Foram estabelecidas métricas, como exemplo de confiabilidade de transmissão de pacotes, a partir de análises em nuvem dos desafios de otimização de latência.

[Bernardo et al. 2022] propõem uma abordagem de aprendizado por reforço onde o agente e o ambiente se encontram em localidades diferentes. No trabalho, destaca-se que a latência de rede impacta no desempenho dos algoritmos de aprendizado por reforço, assim como foi observado que à medida que a latência da rede aumenta, recompensas de menor valor são passadas ao agente. Apesar de ser avaliado o impacto da rede no aprendizado por reforço, os autores não proprõem um método para tolerar possíveis falhas presente no ambiente.

[Cheng et al. 2021] apresentam uma arquitetura em dois níveis para controlar e otimizar redes de VANTs com base em aprendizado profundo por reforço. No estudo, cada VANT interage com um ambiente composto por uma rede de VANTs, com o objetivo de aprender uma política ótima para se adaptar às mudanças na rede. Embora o trabalho apresente bons resultados na modelagem de diferentes problemas de controle de rede de VANTs, os autores sugerem como trabalhos futuros a avaliação de como o sistema pode lidar com casos de falhas.

[Wu et al. 2021] apresentam uma solução para projetar a melhor trajetória de VANTs para minimizar o tempo de distribuição dos dados de detecção gerados pelos vários VANts. Dessa forma, os autores utilizam aprendizado por reforço para mapear a melhor rota na qual os dados podem ser transmitidos. Esses dados podem ser transmitidos para estações base terrestres ou para dispositivos celulares móveis. Apesar da solução conseguir encontrar a melhor rota para fazer a transmissão dos dados, os autores não consideram problemas de comunicação entre os VANTs e o ambiente durante o treinamento do agente.

Como demonstrado nos trabalhos apresentados, nenhuma dessas investigações aborda de forma específica a avaliação da confiabilidade em sistemas de aprendizado por reforço quando as informações são transmitidas pela rede ou não propõem estratégias para tolerar falhas de comunicação ou ignoram problemas de latência e jitter durante o treinamento do agente. Diversas falhas porém ocorrem em comunicações de rede, exercendo um impacto direto na confiabilidade dos sistemas [Raposo et al. 2016]. Este trabalho propõe uma estratégia de tolerância a falhas utilizando aprendizado por reforço e avalia o seu impacto sobre as métricas de rede em cenários brasileiros.

Este trabalho complementa a literatura existente ao oferecer uma análise mais prática e focada em cenários distribuídos. Além disso, o trabalho apresenta particularidades regionais e propõe estratégias para lidar com os desafios impostos pela latência. Ao explorar o contexto brasileiro, aborda-se desafios encontrados em países em desenvolvimento, tornando-se um exemplo de aplicação adaptada às condições locais e com potencial para inspirar soluções em contextos semelhantes.

3. Metodologia

Esta seção apresenta os procedimentos experimentais desenvolvidos para analisar o impacto da latência e do jitter no aprendizado por reforço em sistemas distribuídos. O objetivo principal é avaliar como essas variáveis influenciam o desempenho e a viabilidade de estratégias de tolerância a falhas em um contexto geograficamente distribuído.

Para investigar o impacto da latência no tempo de aprendizado, utiliza-se o simulador Lunar Lander que faz parte da biblioteca OpenAI Gym [2][3]. O Lunar Lander é um ambiente de simulação que replica o desafio de controlar a descida de um módulo lunar em um terreno irregular, exigindo precisão no controle dos propulsores para pousar com segurança. Este ambiente é representativo para o estudo por envolver tarefas contínuas e discretas, além de ser amplamente utilizado como referência na avaliação de algoritmos de aprendizado por reforço devido à sua complexidade moderada e aplicabilidade a cenários reais com controle dinâmico e condições desafiadoras. Baseando-se nos estudos de [Bernardo et al. 2022], que variam a latência entre 0ms e 50ms, e de [Americas 2018], que estabelecem 100ms como o limite de sobrevivência para processos autônomos, são gerados valores de latência no intervalo de 0ms a 100ms para validar o efeito da latência no tempo de aprendizado por reforço. O jitter é fixado em 20ms, considerando sua relevância para a qualidade de serviço (QoS) em sistemas em tempo real [Kunst et al. 2019].

Neste estudo, são priorizadas as métricas de latência e jitter, pois apresentam maior impacto em serviços em tempo real. Para trabalhos futuros, considera-se a inclusão de outras métricas, como perda, corrupção e reordenamento de pacotes. O valor de latência reflete dados reais das condições de rede no Brasil, com base em análises do Ceptro.br [Ceptro.br 2021].

Para possibilitar a troca de informações por rede, a biblioteca *Stable Baselines3* é modificada. Essa alteração inclui a implementação de uma função específica para envio e recebimento de dados via sockets, estabelecendo comunicação entre o agente de aprendizado e o ambiente. Essa adaptação assegura um canal de interação eficiente e simula condições reais de comunicação distribuída.

O impacto de ações tomadas após o tempo limite de recebimento de pacotes é analisado em treinamentos realizados localmente, com atrasos simulados por valores gerados a partir de uma distribuição normal. A latência média e o jitter são utilizados como parâmetros para essa geração, enquanto o limite de 60ms, recomendado por [Americas 2018], é adotado como referência para aplicações autônomas. Durante o processo, verifica-se se o agente deve tomar uma ação aleatória, repetir a última ação ou seguir o fluxo normal, dependendo dos valores de atraso observados.

A avaliação da viabilidade do aprendizado por reforço em cenários brasileiros considera as métricas específicas das cinco regiões do país, apresentadas na Tabela 1. Essa análise empírica utiliza dados reais para refletir os desafios de infraestrutura de redes de internet públicas nas regiões Centro-Oeste, Nordeste, Norte, Sudeste e Sul [Ceptro.br 2021]. Os critérios de avaliação mensuram o impacto da latência e do jitter no desempenho do aprendizado por reforço, sendo eles:

- Tempo de Convergência: Avaliação do tempo necessário para que o agente atinja uma recompensa média próxima do valor ideal.
- Tolerância a Falhas: Análise de como o desempenho do agente é afetado em diferentes

[2]`https://gymnasium.farama.org/environments/box2d/lunar_lander/`
[3]`https://www.gymlibrary.dev/`

níveis de latência e jitter, e seu desempenho ao manter a eficiência em situações de atraso crítico.
- Estratégias de Ação: Decisão entre repetir a última ação ou tomar uma ação aleatória em cenários com falhas de temporização.

Tabela 1. Métricas latência e jitter de redes por região do Brasil [Ceptro.br 2021].

Estado	Latência (ms)	Jitter (ms)
Centro Oeste	31,7	1,24
Nordeste	46,2	1,27
Norte	61,6	1.00
Sudeste	15,1	1,10
Sul	23,2	1,12

Os resultados das medições são analisados considerando: (i) o tempo necessário para o agente atingir a convergência, medido pelo valor de uma recompensa média próxima ao valor ótimo (200 no cenário Lunar Lander); (ii) a robustez das estratégias de tolerância a falhas, avaliando o desempenho do agente em condições de rede adversas. Adicionalmente, métricas de QoS, como latência média e jitter, são utilizadas para conectar os resultados aos cenários reais simulados, permitindo uma análise detalhada da eficácia e viabilidade das estratégias propostas. Os resultados são apresentados como respostas às três questões apresentadas: " *(i) Qual é o efeito da latência na temporização no aprendizado por reforço? (ii) Qual é o impacto de uma ação após definir um limite de tempo para a chegada do pacote? (iii) Qual é a viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro onde o agente e o ambiente estão em localidades diferentes?*".

4. Desenvolvimento

Nesta seção, detalhamos o desenvolvimento do trabalho, que inclui a configuração experimental, os cenários avaliados e as estratégias implementadas para lidar com as falhas de temporização em sistemas distribuídos. São apresentados os recursos computacionais utilizados, as adaptações realizadas na biblioteca Stable Baselines3 para simular condições reais de comunicação em rede, e a implementação das estratégias propostas. Além disso, descrevemos os experimentos conduzidos para avaliar o impacto da latência e do jitter no desempenho do aprendizado por reforço, considerando diferentes condições de rede baseadas nas métricas reais das cinco regiões brasileiras.

4.1. Configuração experimental

Para a elaboração do experimento, utilizamos uma máquina virtual com sistema operacional Ubuntu 20.04, equipada com 4 GB de RAM e 4 núcleos de processamento. Para simular o ambiente com um VANT, foi utilizada a biblioteca OpenAI Gym[4] versão 0.25, com o cenário Lunar Lander. Este ambiente simula a tarefa de um VANT de realizar um pouso suave em uma superfície lunar, conforme ilustrado na Figura 2. O critério para considerar que o objetivo de pouso foi alcançado é quando a média das recompensas passam a convergir para 200, conforme detalhado em [Guttulsrud et al. 2024].

Q-function é uma medida que atribui um valor de utilidade para cada ação possível em um determinado estado. Por empregar uma arquitetura de rede neural para estimar a *Q-function* [Mnih et al. 2015], o algoritmo Deep Q-Network (DQN) foi implementado e utilizado. Ele foi

[4]`https://github.com/openai/gym`

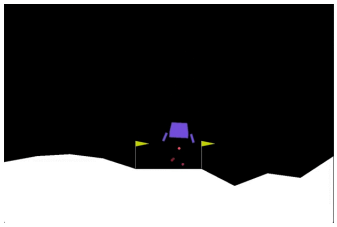

Figura 2. Lunar Lander.

escolhido pois utiliza técnicas de otimização para treinar a rede neural de forma a maximizar os valores de Q para ações que levam a recompensas maiores. Esse algoritmo e os hiperparâmetros utilizados no treinamento foram baseados em fontes prévias[5], visando assegurar resultados comparáveis e confiáveis neste estudo. O DQN é a representação do agente neste trabalho, o qual realiza as tarefas de aprendizado e tomadas de decisão no contexto considerado.

O simulador de rede foi baseado no trabalho de [Bernardo et al. 2022]; ele é composto por duas máquinas virtuais (VMs), cliente e servidor, sendo que uma contém o agente e a outra o ambiente. Foi utilizada a ferramenta chamada network emulator (NetEm)[6] do Linux, que fornece funcionalidades para emular redes visando testar propriedades de redes do mundo real, possibilitando alterar os parâmetros de rede entre as máquinas. A Figura 3 ilustra o esquema do simulador de rede.

Figura 3. Esquema de simulação de rede [Bernardo et al. 2022].

4.2. Experimentos

Neste trabalho, o agente e o ambiente estão fisicamente separados, interagindo exclusivamente por meio da rede. Uma falha de temporização ocorre quando uma ação enviada pelo agente não é recebida pelo ambiente dentro de um intervalo de tempo estabelecido. Os valores limiares para identificar falhas de temporização foram arbitrariamente estabelecidos, sendo levado em consideração o valor da latência definido no trabalho e os resultados obtidos ao longo da execução dos experimentos.

Para a estratégia de tolerância a falhas de temporização, foi adotada a técnica de redundância, amplamente reconhecida na literatura como essencial para possibilitar a tolerância falhas de sistema [Khayatian et al. 2022, Avizienis et al. 2004, Gärtner 1999]. Nesse sentido, este estudo investiga dois cenários para lidar com falhas de temporização: um em que o ambiente retém a última ação executada para ser repetida em caso de falha; um em que o ambiente seleciona aleatoriamente uma nova ação para execução. Para simular a variação da rede e o atraso, foram gerados valores aleatórios seguindo uma distribuição normal, em que a média representa a latência e o desvio padrão representa o jitter.

O algoritmo desenvolvido começa definindo a média de latência utilizada para simular a variabilidade dos atrasos de transmissão na rede. Em cada interação entre o agente e

[5] https://github.com/AbuzzBodhe/Lunar-Lander-Environment
[6] https://www.linux.org/docs/man8/tc-netem.html

o ambiente, um valor de atraso é gerado aleatoriamente dentro dos parâmetros estabelecidos. O algoritmo então verifica se esse atraso excede o valor limite predefinido como critério para identificação de falhas de temporização. Em caso afirmativo, o ambiente aguarda o tempo correspondente ao valor limite em milissegundos e executa a última ação conhecida do agente ou seleciona aleatoriamente uma nova ação. Por outro lado, se o atraso estiver dentro do valor limite, o ambiente aguarda o tempo correspondente ao atraso em milessegundos e executa a ação recebida do agente.

5. Resultados e Discussão

Esta seção apresenta os resultados e discussões das três questões de pesquisa. A RQ1 investiga o efeito da latência no tempo de aprendizado por reforço. A RQ2 explora o impacto de tomar uma ação após estabelecer um limite de tempo para a chegada do pacote. Por fim, a RQ3 aborda a viabilidade da aplicação do aprendizado por reforço em uma arquitetura descentralizada no cenário brasileiro.

RQ1: Qual é o efeito da latência no tempo de aprendizado por reforço?

O objetivo dessa pergunta foi investigar o impacto da latência no desempenho do algoritmo quando nenhuma ação é executada, ou seja, quando o agente aguarda a chegada do pacote sem realizar qualquer ação, e também determinar o tempo necessário para treinar o agente.

Os resultados obtidos, conforme ilustrado na Figura 4, demonstram que, mesmo com valores altos de atraso, o algoritmo converge para o valor ideal de recompensa, que é de 200. Portanto, é possível concluir que, ao considerar apenas a latência e desconsiderar outros fatores associados, ela não impacta significativamente o desempenho do agente. Tal resultado era esperado, uma vez que não há influência além do atraso na comunicação entre o agente e o ambiente.

Por outro lado, a Tabela 2 apresenta os valores de tempo de treinamento do agente, onde fica evidente que o tempo aumenta consideravelmente de acordo com os atrasos gerados.

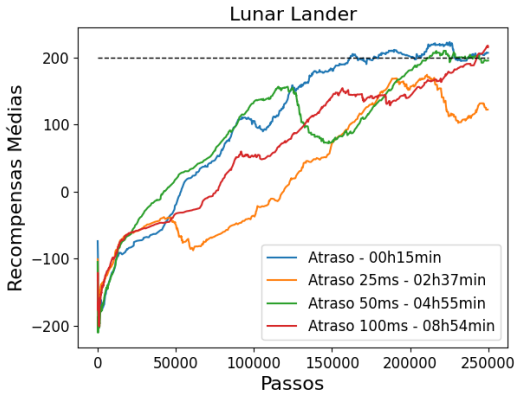

Figura 4. Impacto da latência no tempo de treinamento. Fonte:autor

Esses resultados evidenciam que a latência exerce um impacto significativo no tempo de treinamento do agente. Quanto maior a latência, maior é o tempo necessário para que o agente seja completamente treinado. Isso sugere que a alta latência pode ser um fator crítico a ser considerado em projetos de arquitetura descentralizada com algoritmos de aprendizado por reforço.

Tabela 2. Tempo de treinamento do agente

Latência (ms)	Tempo
0	00h15min
25	02h37min
50	04h55min
100	08h54min

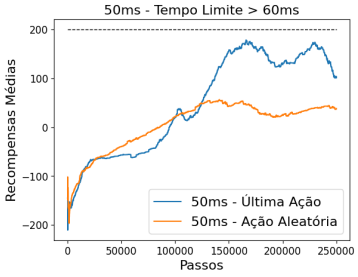

Figura 5. Tempo 50 ms. Fonte: autor

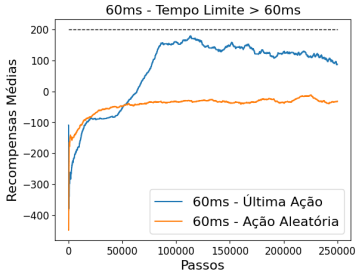

Figura 6. Tempo 60 ms. Fonte: autor

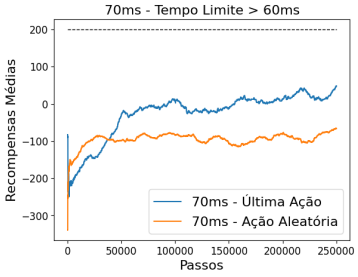

Figura 7. Tempo 70 ms. Fonte: autor

Essas descobertas têm implicações importantes para a implementação de aprendizado por reforço que dependem de interações rápidas e eficientes entre o agente e o ambiente. Considerar e mitigar os efeitos da latência pode ser crucial para garantir um desempenho adequado e eficaz desses sistemas em contextos reais.

RQ2: Qual é o impacto de tomar uma ação após definir um limite de tempo para a chegada do pacote?

É investigado aqui o impacto de tomar uma ação, seja ela aleatória ou repetir a última ação, após estabelecer um limite de tempo para a chegada do pacote, sendo esse limite definido como 60ms. A análise considera diferentes valores de latência: 50ms, 60ms e 70ms.

Nos casos em que a latência foi de 50ms e 60ms, os resultados revelaram que tomar a última ação após o limite de tempo estabelecido leva a um bom desempenho, conforme desmonstram as figuras 5 e 6. Os valores de recompensa obtidos foram próximos do valor ideal de convergência, que é de 200. Isso indica que o agente conseguiu realizar ações adequadas e maximizar sua recompensa mesmo diante da latência presente no ambiente. Por outro lado, quando o agente toma uma ação aleatória após o limite de tempo, o desempenho foi significativamente inferior, com valores de recompensa distanciados do ideal.

Em contrapartida, quando a latência aumentou para 70ms, o algoritmo apresentou um desempenho ruim, independentemente de tomar a última ação ou uma ação aleatória após o limite de tempo, conforme a Figura 7. Nesse caso, os valores de recompensa foram substancialmente baixos, indicando uma dificuldade do agente em realizar ações efetivas e alcançar resultados satisfatórios.

Essas conclusões destacam a importância de uma cuidadosa consideração do tempo limite para a tomada de ação em cenários com latência, juntamente com o desenvolvimento de estratégias e algoritmos capazes de lidar de forma eficiente com atrasos na comunicação. Fica claro que é mais viável empregar estratégias de tomadas de ação após um tempo limite em valores menores ou próximos do limite estabelecido. Essas considerações ressaltam a necessidade de um planejamento adequado e de abordagens adaptativas para garantir o

Tabela 3. Análise empírica de um abordagem descentralizada de aprendizagem por reforço no cenário brasileiro.

Região	Latência (ms)	Tempo de Treinamento Estimado	Observações
Centro-Oeste	31,7	Entre 2h30 e 4h	Viável para aplicações com tolerância moderada à latência; requer ajustes para maior eficiência em ambientes críticos.
Nordeste	46,2	Aproximadamente 5h	Aplicável em sistemas não críticos; melhorias em algoritmos poderiam reduzir o impacto do tempo de treinamento.
Norte	61,6	Entre 5h e 8h	Viável apenas para cenários não urgentes; alta latência exige estratégias adicionais para garantir confiabilidade.
Sudeste	15,1	Menos que 2h30	Altamente viável; excelente desempenho em tempo de treinamento e resposta a falhas.
Sul	23,2	Menos que 2h30	Muito viável; baixo impacto de latência e jitter possibilita alta eficiência.

desempenho desejado em ambientes com atrasos significativos de comunicação.

RQ3: Qual é a viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro onde o agente e o ambiente estão em localidades diferentes?

Os resultados obtidos na análise empírica, com base nos resultados anteriores, sobre a viabilidade da aplicação do aprendizado por reforço em um cenário brasileiro, no qual o agente e o ambiente estão localizados em regiões distintas do país, são sumarizados na Tabela 3. Esses resultados levam em consideração as métricas de rede nas cinco regiões do Brasil.

Observa-se que, no Centro-Oeste, um algoritmo de aprendizado por reforço levaria entre 2 horas e meia a 4 horas para ser totalmente treinado. No Nordeste, esse tempo seria de aproximadamente 5 horas. Na região Norte, o tempo estimado varia de 5 a 8 horas. Já nas regiões Sul e Sudeste, o tempo de treinamento seria inferior a 2 horas e meia.

É importante ressaltar que, em todos os casos, é necessário avaliar se o tempo de espera é ideal para a aplicação em questão. No entanto, com base nos resultados, pode-se inferir que, em geral, é mais vantajoso não estabelecer um tempo limite para a tomada de ação, a menos que seja uma situação crítica que exija uma resposta imediata.

Com base nessas constatações, pode-se concluir que existe viabilidade na aplicação de uma abordagem descentralizada de aprendizado por reforço no cenário brasileiro, considerando as particularidades de cada região. Essa abordagem permite adaptar o treinamento e a tomada de decisão de acordo com as métricas de rede específicas de cada região, garantindo um desempenho adequado do algoritmo em diferentes contextos geográficos.

5.1. Discussão

Os resultados deste estudo têm implicações práticas importantes para o uso de aprendizado por reforço em cenários brasileiros, especialmente em sistemas distribuídos, como redes industriais, serviços de entrega autônoma e monitoramento ambiental. Em regiões com maior latência, como o Norte e Nordeste, aplicações que exigem respostas em tempo real enfrentam desafios na redução de latência, demandando estratégias de redundância para mitigar falhas de

temporização. Essas estratégias podem incluir técnicas adaptativas baseadas em aprendizado dinâmico, bem como investimento em infraestrutura de rede para reduzir os tempos de resposta.

No contexto do agronegócio, onde drones e máquinas inteligentes operam em áreas remotas, soluções baseadas em aprendizado por reforço podem otimizar operações críticas, como irrigação de precisão e monitoramento de safras. Além disso, as condições locais de rede destacam a necessidade de sistemas robustos capazes de funcionar eficientemente mesmo sob condições adversas de comunicação.

Nas regiões Sul e Sudeste, com latências menores, a implementação de soluções descentralizadas em aplicações críticas, como controle de tráfego e monitoramento urbano, é mais viável. Essa condição permite avanços em soluções inteligentes para cidades, como gestão de semáforos e resposta a emergências em tempo real. Os resultados ressaltam a importância de personalizar algoritmos com base nas características regionais, o que também pode inspirar aplicações internacionais em cenários com infraestrutura semelhante.

Para garantir o impacto positivo dessas soluções, é essencial investir em colaborações entre governos e instituições privadas para melhorar a qualidade das redes, especialmente em regiões menos favorecidas. Além disso, iniciativas de pesquisa focadas em ampliar as aplicações práticas do aprendizado por reforço poderiam explorar novos domínios, como saúde pública e educação, onde o impacto social pode ser significativo.

Embora os resultados demonstrem a viabilidade do uso de aprendizado por reforço em sistemas distribuídos, algumas limitações foram identificadas e não foram aqui tratadas por questões de tempo e espaço. Primeiramente, o estudo concentrou-se nas métricas de latência e jitter, deixando de lado outras variáveis, como perda e reordenação de pacotes, que podem afetar significativamente a confiabilidade em redes reais. Além disso, a simulação de rede baseou-se em condições predefinidas que podem não capturar toda a dinâmica de ambientes reais com altos níveis de variabilidade. Por fim, não foram consideradas aplicações multiagente ou cenários com alterações dinâmicas significativas, o que limita a generalização dos resultados obtidos para outros contextos mais complexos que o apresentado no nosso trabalho.

6. Conclusão

Neste trabalho, foram investigadas três questões de pesquisa relacionadas à viabilidade da aplicação do aprendizado por reforço em uma abordagem descentralizada, na qual o agente e o ambiente estão geograficamente separados. As questões analisaram o impacto da latência no tempo de aprendizado por reforço, os efeitos de tomar ações após estabelecer um limite de tempo para a chegada de pacotes e a viabilidade da abordagem descentralizada em diferentes regiões do Brasil.

Na primeira questão, constatou-se que a latência não afeta significativamente o desempenho do algoritmo em termos de convergência para o valor ideal de recompensa, mesmo em cenários com atrasos elevados, como 50ms, 60ms e 70ms. No entanto, os resultados mostram que a latência impacta diretamente o tempo de treinamento, sendo necessário mais tempo para o agente completar o aprendizado à medida que a latência aumenta. Esses achados destacam a importância de mitigar os efeitos da latência no desenvolvimento de algoritmos de aprendizado por reforço para ambientes com restrições de tempo real.

Na segunda questão, os resultados indicaram que estratégias que tomam ações após um tempo limite são mais eficazes quando aplicadas a valores menores ou próximos ao limite estabelecido. Observou-se que repetir a última ação após o limite de tempo apresentou desempenho superior em comparação com a execução de uma ação aleatória, evidenciando a relevância de

estratégias robustas para lidar com falhas de temporização.

Por fim, na terceira questão, foi analisada a viabilidade da abordagem descentralizada no contexto brasileiro, considerando as particularidades das cinco regiões do país. Verificou-se que o tempo de treinamento varia significativamente conforme as métricas de rede regionais. Nas regiões Centro-Oeste, Nordeste e Norte, onde a latência é maior, o tempo de treinamento é relativamente mais longo, enquanto no Sul e Sudeste, com menor latência, os tempos são reduzidos. Apesar dessas variações, os resultados demonstram que a abordagem descentralizada é viável, permitindo a adaptação do treinamento e da tomada de decisão às condições específicas de cada região.

Essas descobertas têm implicações relevantes para a aplicação de aprendizado por reforço em cenários reais, onde latência e distribuição geográfica são fatores críticos. Projetar algoritmos que considerem a latência, estabelecer limites de tempo adequados e implementar estratégias adaptativas são aspectos essenciais para garantir desempenho eficaz em sistemas distribuídos. Além disso, a abordagem descentralizada mostrou-se promissora, oferecendo uma solução prática para lidar com as particularidades regionais e permitindo sua aplicação em cenários geograficamente distribuídos.

Como perspectivas futuras, estão a ampliação do leque de métricas, incluindo a avaliação de perda de pacotes e reordenação, a exploração de técnicas adaptativas que ajustem automaticamente os parâmetros do agente conforme as condições da rede. Investigar cenários mais complexos, envolvendo múltiplos agentes e ambientes dinâmicos, também representa uma direção promissora. Validações práticas em redes industriais brasileiras podem aproximar os resultados simulados de aplicações reais, fortalecendo a viabilidade do aprendizado por reforço em sistemas descentralizados.

Referências

Americas, G. (2018). 5g communications for automation in vertical domains.

Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. (2004). Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, 1(1):11–33.

Azar, A. T., Koubaa, A., Ali Mohamed, N., Ibrahim, H. A., Ibrahim, Z. F., Kazim, M., Ammar, A., Benjdira, B., Khamis, A. M., Hameed, I. A., and Casalino, G. (2021). Drone deep reinforcement learning: A review. *Electronics*, 10(9).

Bayerlein, H., Theile, M., Caccamo, M., and Gesbert, D. (2020). Uav path planning for wireless data harvesting: A deep reinforcement learning approach. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6.

Bernardo, G., Jr., G. M., and Macedo, D. (2022). Analysis of network performance over deep reinforcement learning control loops for industry 4.0. In *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 1–14, Porto Alegre, RS, Brasil. SBC.

Ceptro.br (2021). Covid-19 impactos na qualidade da internet no brasil. [Online]. Available: `https://www.ceptro.br/assets/publicacoes/pdf/2021.01.05-relatorio-covid.pdf`.

Cheng, H., Bertizzolo, L., D'oro, S., Buczek, J., Melodia, T., and Bentley, E. S. (2021). Learning to fly: A distributed deep reinforcement learning framework for software-defined uav network control. *IEEE Open Journal of the Communications Society*, 2:1486–1504.

Gärtner, F. C. (1999). Fundamentals of fault-tolerant distributed computing in asynchronous environments. *ACM Comput. Surv.*, 31(1):1–26.

Guttulsrud, H., Sandnes, M., and Shrestha, R. (2024). Solving the lunar lander problem with multiple uncertainties using a deep q-learning based short-term memory agent. In *Proceedings of the 2023 12th International Conference on Computing and Pattern Recognition*, ICCPR '23, page 27–33, New York, NY, USA. Association for Computing Machinery.

Jiang, Z., Fu, S., Zhou, S., Niu, Z., Zhang, S., and Xu, S. (2020). Ai-assisted low information latency wireless networking. *IEEE Wireless Communications*, 27(1):108–115.

Khayatian, M., Mehrabian, M., Andert, E., Grimsley, R., Liang, K., Hu, Y., McCormack, I., Joe-Wong, C., Aldrich, J., Iannucci, B., and Shrivastava, A. (2022). Plan b: Design methodology for cyber-physical systems robust to timing failures. *ACM Trans. Cyber-Phys. Syst.*, 6(3).

Kunst, R., Avila, L., Binotto, A., Pignaton, E., Bampi, S., and Rochol, J. (2019). Improving devices communication in industry 4.0 wireless networks. *Engineering Applications of Artificial Intelligence*, 83:1–12.

Li, Y., Liu, I.-J., Yuan, Y., Chen, D., Schwing, A., and Huang, J. (2019). Accelerating distributed reinforcement learning with in-switch computing. In *Proceedings of the 46th International Symposium on Computer Architecture*, pages 279–291.

Lu, R., Li, Y.-C., Li, Y., Jiang, J., and Ding, Y. (2020). Multi-agent deep reinforcement learning based demand response for discrete manufacturing systems energy management. *Applied Energy*, 276:115473.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Raposo, D., Rodrigues, A., Silva, J. S., Boavida, F., Oliveira, J., Herrera, C., and Egas, C. (2016). An autonomous diagnostic tool for the wirelesshart industrial standard. In *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–3.

Samanta, S., Mukherjee, A., Ashour, A. S., Dey, N., Tavares, J. M. R., Karaa, W. B. A., Taïar, R., Azar, A. T., and Hassanien, A. E. (2018). Log transform based optimal image enhancement using firefly algorithm for autonomous mini unmanned aerial vehicle: An application of aerial photography. *Int. J. Image Graph.*, 18:1850019:1–1850019:25.

Shakhatreh, H., Sawalmeh, A. H., Al-Fuqaha, A., Dou, Z., Almaita, E., Khalil, I., Othman, N. S., Khreishah, A., and Guizani, M. (2019). Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *IEEE Access*, 7:48572–48634.

Szarski, M. and Chauhan, S. (2021). Composite temperature profile and tooling optimization via deep reinforcement learning. *Composites Part A: Applied Science and Manufacturing*, 142:106235.

Wu, F., Zhang, H., Wu, J., Han, Z., Poor, H. V., and Song, L. (2021). Uav-to-device underlay communications: Age of information minimization by multi-agent deep reinforcement learning. *IEEE Transactions on Communications*, 69(7):4461–4475.