UNIVERSIDADE FEDERAL DO RIO DE JANEIRO

INSTITUTO DE COMPUTAÇÃO

CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

FLAVIO RIBEIRO TEIXEIRA NETO

ESTUDO DE APRENDIZADO POR REFORÇO PROFUNDO APLICADO AO MERCADO DE CRIPTOMOEDAS

RIO DE JANEIRO

2021

FLAVIO RIBEIRO TEIXEIRA NETO

ESTUDO DE APRENDIZADO POR REFORÇO PROFUNDO APLICADO AO MERCADO DE CRIPTOMOEDAS

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. João Carlos P. da Silva

RIO DE JANEIRO

2021

CIP - Catalogação na Publicação

Teixeira Neto, Flavio Ribeiro
N469e ESTUDO DE APRENDIZADO POR REFORÇO PROFUNDO APLICADO AO MERCADO DE CRIPTOMOEDAS / Flavio Ribeiro Teixeira Neto. -- Rio de Janeiro, 2021.
75 f.

Orientador: João Carlos Pereira da Silva.
Trabalho de conclusão de curso (graduação) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Bacharel em Ciência da Computação, 2021.

1. Aprendizado por reforço. 2. Aprendizado de máquina. 3. Sistema autônomo. 4. Criptomoedas. 5. Mercado financeiro. I. Silva, João Carlos Pereira da, orient. II. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/628

FLAVIO RIBEIRO TEIXEIRA NETO

ESTUDO DE APRENDIZADO POR REFORÇO PROFUNDO APLICADO AO MERCADO DE CRIPTOMOEDAS

Trabalho de conclusão de curso de graduação apresentado ao Instituto de Computação da Universidade Federal do Rio de Janeiro como parte dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 04 de novembro de 2021.

BANCA EXAMINADORA:

Prof. João Carlos Pereira da Silva, D.Sc. (UFRJ)

Documento assinado digitalmente
gov.br JULIANA VIANNA VALERIO
Data: 24/11/2021 07:19:44-0300
Verifique em https://verificador.iti.br

Profª. Juliana Vianna Valério, D.Sc. (PUC-RJ)

Prof. Daniel Sadoc Menasche, PhD. (UMASS)

AGRADECIMENTOS

Gostaria de agradecer a todos que me apoiaram, não apenas na confecção deste trabalho, mas também durante toda a graduação. Em especial, agradeço ao meu orientador, João Carlos, que desprendeu de seu tempo para me guiar durante todo o processo. Agradeço a todos os amigos que hoje fazem parte de um grupo chamado "Code Fairy", que me deram muita ajuda durante toda a graduação e também neste trabalho. Agradeço também a minha namorada Ana Carolina, que me deu a energia necessária para continuar trabalhando no projeto dia após dia. E por último, mas não menos importante, agradeço aos meus pais, Flavio e Janaina, que deram todo o suporte necessário para que eu chegasse onde cheguei. Sem eles, nada disso seria possível.

RESUMO

O mercado financeiro é um campo de possibilidades infinitas que seduz cada vez mais indivíduos a tentarem ser capazes de se beneficiar de suas oscilações. Um mercado específico que ganhou bastante força nos últimos anos foi o mercado de criptomoedas. A principal característica que o torna atrativo em relação a mercados mais tradicionais, como por exemplo a bolsa de valores, é sua grande volatilidade. Este trabalho tem como principal objetivo explorar técnicas de aprendizado por reforço, uma subárea de aprendizado de máquina, de forma a tentar entender a viabilidade de sua utilização para construir um sistema capaz de operar no mercado de Bitcoin, uma criptomoeda bastante utilizada para este fim. O sistema será treinado utilizando principalmente dados históricos da cotação de Bitcoin, e será validado utilizando parte desses dados em rodadas de teste onde sua meta será obter algum lucro ao término do experimento e, de forma complementar, manter uma evolução patrimonial suave ao longo do teste, de forma que transpareça consistência em sua forma de operar e traga confiança de que seria capaz de negociar em um ambiente real.

Palavras-chave: mercado financeiro; investimento; bolsa de valores; ativos financeiros; criptomoedas; Bitcoin; sistema autônomo; aprendizado por reforço; aprendizado de máquina.

ABSTRACT

The financial market is a field of infinite possibilities that attracts more and more individuals to try to be able to benefit from its fluctuations. A specific market that has gained a lot of traction in recent years is the cryptocurrency market. The main feature that makes it attractive compared to more traditional ones, such as the stock market, is its high volatility. The main objective of this work is to explore reinforcement learning techniques, a sub-area of machine learning, in order to try to understand the feasibility of its use to build a system capable of operating in the Bitcoin market, a widely used cryptocurrency for this purpose. The system will be trained using mainly historical Bitcoin pricing data, and will be validated using part of this data in test rounds where its goal will be to obtain some profit at the end of the experiment and, in a complementary way, maintain a smooth equity evolution throughout the test , so that it shows consistency in its way of operating and brings confidence that it would be able to trade in a real environment.

Keywords: financial market; investment; stock exchange; financial assets; cryptocurrencies; Bitcoin; autonomous system; reinforcement learning; machine learning.

SUMÁRIO

1 INTRODUÇÃO

O mercado financeiro é um campo de possibilidades infinitas que seduz cada vez mais indivíduos a tentarem ser capazes de se beneficiar de suas oscilações. A forma mais tradicional de interação com o mercado é o investimento de longo prazo, onde a esperança de valorização futura de certos ativos financeiros impulsionam a compra e posse prolongada desses ativos.

O espaço de trocas de ativos diárias ou intra-diárias tem se tornado muito popular, principalmente entre os jovens, e gera uma discussão muito grande sobre a dualidade entre investimento pautado em lógica de mercado versus apostas pautadas em sorte. São poucos aqueles que conseguem gerar lucros de forma constante e previsível negociando ativos diariamente, mas a promessa de dinheiro rápido tem atraído cada vez mais investidores para o mercado.

Um mercado específico que ganhou bastante força nos últimos anos foi o mercado de criptomoedas. A principal característica que o torna atrativo em relação a mercados mais tradicionais, como por exemplo a bolsa de valores, é sua grande volatilidade. Devido a grande oscilação do preço das criptomoedas em contraste com ativos mais tradicionais, este mercado se mostra um ambiente perfeito para aqueles que estão se preocupando apenas em maximizar o lucro obtido com tal variação, onde o ativo em si é abstraído e se torna apenas um veículo para especulações financeiras.

Em uma era onde o acesso à informação está mais fácil do que nunca, é de se esperar que sistemas autônomos de negociação de ativos financeiros fossem proliferar, a ponto de hoje representar uma parcela considerável das negociações sendo realizadas. Tais sistemas variam bastante em complexidade, desde aqueles que se baseiam em uma estratégia específica que observam sinais preestabelecidos de compra e venda, até sistemas mais robustos baseados em técnicas avançadas de aprendizado de máquina, que têm como objetivo uma abordagem mais genérica e abrangente.

Este trabalho tem como principal objetivo explorar técnicas de aprendizado por reforço profundo, uma subárea de aprendizado de máquina, de forma a tentar entender a viabilidade de sua utilização para construir um sistema capaz de operar no mercado de Bitcoin, uma criptomoeda bastante utilizada para este fim.

Esta técnica foi escolhida devido a algumas de suas características que mostraram potencial de se encaixar bem com o problema proposto. Um exemplo disso seria a forma

como seus componentes básicos se relacionam analogamente com os componentes de um cenário real, onde temos um agente (trader) interagindo com um ambiente (mercado financeiro) a fim de aprender iterativamente a alcançar um objetivo predeterminado (adquirir lucro a partir de negociações de ativos). Um outro exemplo seria a forma como o agente tenta projetar ganhos futuros na tomada de decisão presente, o que é bastante útil neste problema onde o seu retorno financeiro não se dá de imediato, e sim de a partir de uma sucessão de instantes com intervalo indeterminado. Tais características específicas do aprendizado por reforço, alinhados com um interesse pessoal, foram determinantes na escolha da técnica como principal foco de exploração do trabalho.

O sistema será treinado utilizando principalmente dados históricos da cotação de Bitcoin, e será validado utilizando parte desses dados em rodadas de teste onde sua meta será obter algum lucro ao término do experimento e, de forma complementar, manter uma evolução patrimonial suave ao longo do teste, de forma que transpareça consistência em sua forma de operar e traga confiança de que seria capaz de negociar em um ambiente real.

Este trabalho está estruturado da seguinte maneira: no capítulo 2, serão apresentados alguns conceitos básicos sobre mercado financeiro e aprendizado de máquina, proporcionando uma base conceitual apropriada para a leitura do restante do trabalho; no capítulo 3, serão descritos todos os procedimentos realizados para a construção da base de dados sobre Bitcoin que será utilizada na parte inicial dos nossos experimentos; no capítulo 4, será descrito todo o processo de construção de um ambiente simulado em um sistema baseado em aprendizado por reforço; no capítulo 5, será descrito o funcionamento do processo de tomada de decisão, onde o agente escolhe qual ação irá realizar de acordo com a observação feita; no capítulo 6, serão realizados diversos experimentos a fim de tentar validar os resultados do agente; e no capítulo 7, será apresentada a conclusão do trabalho, onde serão avaliados os resultados dos experimentos de forma a tentar entender o quão efetivo o agente conseguiu ser em relação aos objetivos preestabelecidos.

2 CONCEITOS BÁSICOS

Este segmento tem como objetivo descrever brevemente alguns conceitos que são citados ao longo do trabalho, mas que não possuem uma explicação extensa.

2.1 MERCADO FINANCEIRO

É importante entender o conceito de ativo como sendo um recurso com valor econômico que um indivíduo, empresa ou país possui ou controla com a expectativa de que ele irá prover um benefício futuro. Investidores compram e vendem ativos financeiros através de corretoras, que atuam como intermediárias de transações de algum mercado específico. No contexto de compra e venda de ações, por exemplo, as corretoras são utilizadas para dar acesso ao investidor individual de comprar e vender ativos negociados na bolsa de valores (LANGAGER, 2021).

No contexto de criptomoedas, o cenário é um pouco diferente. Não existe um mercado centralizado. Ao invés de existirem diversas corretoras que interagem com um mesmo mercado, o que existe são mercados independentes chamados de *exchanges* onde, através de suas plataformas, investidores individuais têm acesso a comprar e vender criptomoedas. Por se tratarem de mercados independentes, cada exchange pode apresentar leves variações no preço e liquidez das criptomoedas negociadas (FRANKENFIELD, 2021).

Ao realizar algum tipo de operação em alguma corretora ou exchange, o investidor pode estar abrindo ou fechando alguma posição em relação a algum determinado ativo. Dois tipos de posições comuns que serão mencionadas neste trabalho são: comprado ou vendido.

Estar comprado, ou se posicionar comprado, ou operar comprado, significa que o investidor comprou alguma quantidade de algum ativo na esperança de que o preço aumente no futuro, de forma que eventualmente ele se desfaça de sua posição, ou seja, venda esses ativos, a fim de lucrar com a diferença de preços.

Estar vendido, ou se posicionar vendido, ou operar vendido, significa que o investidor vendeu alguma quantidade de algum ativo na esperança de que o preço diminua no futuro, de forma que eventualmente ele se desfaça de sua posição, ou seja, compre esses ativos, a fim de lucrar com a diferença de preços. Existem dois tipos de posição de venda. Em uma delas, o investidor possui originalmente os ativos que está vendendo ao abrir sua posição de venda. Na outra, que será a que o trabalho irá se referir ao mencionar o posicionamento vendido, o

investidor vende ativos que não estão em sua posse.

Este segundo tipo de posição vendida é possível pois o que está acontecendo na realidade é que o investidor está vendendo um ativo alugado de outro investidor. A recompra do ativo vendido é repassada para o dono original do ativo, junto com um valor a mais que representa o preço do aluguel. Isso tudo é transparente para ambos os investidores, pois tudo é realizado automaticamente pela plataforma utilizada, que exige uma garantia de quem está alugando para não oferecer riscos a quem aluga. Este é um dos principais mecanismos utilizados por investidores que procuram se aproveitar de uma queda momentânea de um determinado ativo, sem ter a necessidade de possuí-lo previamente.

No contexto de investimento de longo prazo, é interessante citar algumas estratégias comuns, como por exemplo o Buy & Hold. Ela é uma estratégia de investimento passiva onde um investidor compra algum ativo e segura ele por um período longo de tempo independente da flutuação do mercado pois acredita na valorização no longo prazo do ativo.

Um outro exemplo interessante seria a Dollar-Cost Averaging (DCA), comumente utilizada em conjunto com a Buy & Hold, onde o investidor divide o total de dinheiro que será investido em um ativo específico em compras periódicas de forma a tentar reduzir o impacto da volatilidade do preço do ativo na compra total. As compras ocorrem periodicamente independente do preço do ativo de acordo com a premissa de que seria muito difícil prever a flutuação do mercado de forma a comprar no melhor momento possível.

2.2 APRENDIZADO DE MÁQUINA

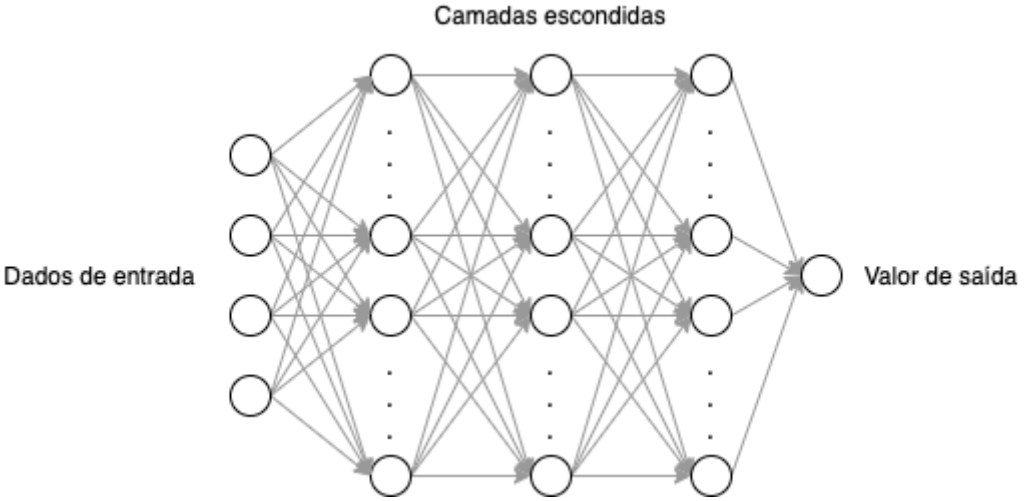
O principal tema de aprendizado de máquina utilizado neste trabalho é o aprendizado por reforço profundo. Os detalhes da implementação serão explicados ao longo do trabalho. De forma geral, a cada passo de uma simulação de um sistema que utiliza essa técnica, um agente recebe informações do ambiente e realiza alguma ação baseada nelas. De acordo com algum objetivo pré-estabelecido, o agente recebe recompensas de acordo com o quanto a ação tomada o faz chegar mais perto de cumprir tal objetivo. O algoritmo tenta maximizar essas recompensas utilizando uma rede neural como principal fonte de tomada de decisão do agente.

Uma rede neural artificial é um modelo de processamento de dados que apresenta semelhanças de alto nível com sua inspiração biológica. Tais semelhanças produzem analogias conceituais que refletem nas terminologias utilizadas para descrever seus

componentes. Ela é composta principalmente por neurônios, que são estruturas capazes de receber e transformar informações, determinando se ocorrerá ou não uma ativação, ou seja, se serão ou não passadas informações adiante para um novo neurônio (ALOM, 2019).

Uma rede neural é normalmente dividida em camadas, onde a primeira representa os dados de entrada, ou seja, os argumentos recebidos através de algum componente externo. As camadas seguintes, chamadas de camadas escondidas, representam o processamento interno da rede, onde a entrada é transformada através de diversas operações matemáticas que ocorrem de acordo com o estado atual de cada neurônio interno da rede, representados por seus pesos. Por fim, a última camada da rede representa o resultado final gerado pelo processamento, ou seja, é o valor de saída correspondente ao valor de entrada utilizado. A **Figura 1** representa visualmente os componentes básicos de uma rede neural.

Figura 1: Componentes básicos de uma rede neural.



A utilidade de uma rede neural artificial se dá no fato de que elas são aproximadores universais de funções, o que significa que dado os parâmetros apropriados, elas podem se moldar de forma a se aproximar de qualquer função com precisão arbitrária. Além disso, elas são funções matemáticas diferenciáveis, o que significa que para um determinado conjunto de parâmetros, entradas e saídas, é possível encontrar o gradiente de tais parâmetros de forma a minimizar uma função de erro pré definida, alcançando resultados cada vez mais próximos do seu objetivo proposto.

O tipo mais comum de rede neural é chamado de rede neural totalmente conectada ou FC (fully connected). Ela é um tipo de rede neural artificial onde todos os neurônios em uma camada estão conectados a todos os neurônios da camada seguinte.

Um tipo mais específico de rede neural, que será explorada durante o trabalho, é a rede neural recorrente. Sua principal característica é a sua capacidade de reconhecer padrões em informações sequenciais através de uma memória contextual. Elas são aplicadas a diversos tipos diferentes de informações sequenciais como por exemplo textos, fala, vídeos e música (CHANGHWAN, 2019).

Um tipo específico de rede recorrente é a rede neural de memória longa de curto prazo ou LSTM (long short-term memory). Ela se utiliza de componentes específicos para gerar contextos para a forma como a rede recebe entradas e produz saídas. A rede possui esse nome pois seus componentes tentam se utilizar de processamento de memória de curto prazo para tentar criar memórias de longo prazo. Ela é uma rede muito utilizada para classificar, processar e prever séries temporais com intervalos de tempo desconhecidos.

Para mais detalhes sobre aspectos gerais de redes neurais e aprendizado profundo, consulte as bibliografias adequadas (GOODFELLOW, 2016). Para mais detalhes sobre aspectos gerais de aprendizado por reforço, consulte as bibliografias adequadas (RICHARD, 2018).

3 BASE DE DADOS

O treinamento de um agente em um sistema baseado em Aprendizado por Reforço pode ocorrer de diversas formas. É possível treiná-lo diretamente no ambiente real, de forma que suas experiências durante o treinamento sejam as mais autênticas possíveis. No entanto, é mais comum que seu treinamento ocorra em algum tipo de ambiente simulado que recrie da melhor forma possível as características reais do problema, ao mesmo tempo que sua execução ocorra em uma velocidade superior, de forma que seja possível gerar muito mais experiências do que em um ambiente real com a mesma quantidade de tempo.

Existem algumas possíveis abordagens na construção de um ambiente simulado que são limitadas pela natureza do problema a ser resolvido. Se as leis que regem o ambiente real do problema são conhecidas, mesmo que parcialmente, é possível recriar um ambiente simulando tais leis. O problema que este projeto se propõe a explorar se encontra em um contexto de mercado financeiro, que é influenciado por incontáveis variáveis dependentes de eventos reais relacionados à política, economia e diversos outros setores. Pela grande complexidade em criar um modelo que fielmente descreva o ambiente real, a criação de um ambiente simulado neste projeto irá optar pela abordagem de recriação de eventos passados de acordo com dados históricos obtidos através de alguma fonte externa.

3.1 FONTE DOS DADOS

Os dados foram obtidos através de uma sucessão contínua de consultas à uma API da exchange de criptomoedas Foxbit. Cada consulta nos fornece diversos dados sobre 23 diferentes pares de criptomoeda/exchange. Alguns exemplos de criptomoedas disponibilizadas pela API são Bitcoin, Ethereum e Litecoin. A exchange representa por onde tais moedas estão sendo negociadas, como por exemplo Foxbit, Mercado Bitcoin ou BrasilBitcoin.

Para cada par em uma consulta, são disponibilizados o preço atual, a data e hora da consulta, os valores de máximo, mínimo, variação e volume de negociações nas últimas 24 horas, e uma imagem associada à criptomoeda. Durante este passo de construção da base de

dados, são coletadas todas essas informações para que, em um segundo momento, seja decidido o que será de fato utilizado no projeto.

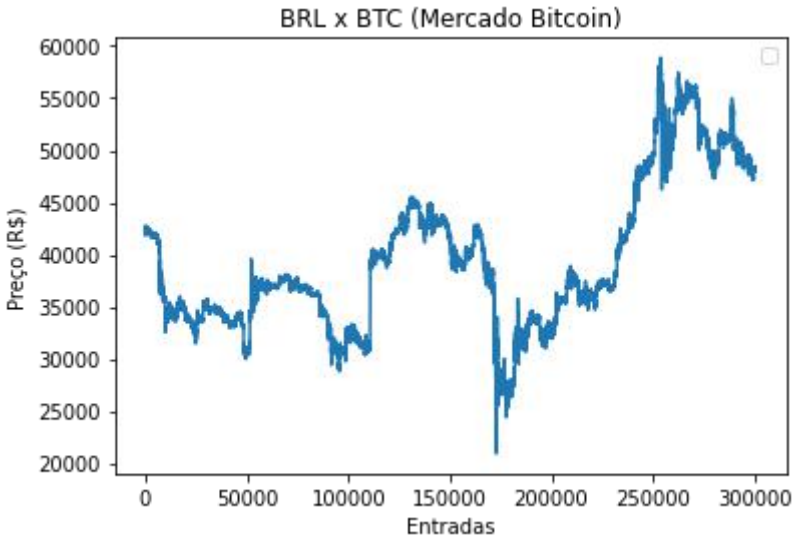
Foi realizada uma consulta a cada dois minutos a partir de 19 de Setembro de 2019 até 10 de Junho de 2020. Durante todo o processo, as informações disponibilizadas para cada par de criptomoeda/exchange foram armazenadas formando uma base de dados com cerca de 7 milhões de registros.

3.2 FILTRO

Em um primeiro momento, foi optado por apenas serem utilizados os registros relativos à moeda Bitcoin negociadas em Real Brasileiro na exchange Mercado Bitcoin. A escolha de uma única moeda tem como objetivo manter o escopo de treinamento mais restrito inicialmente, para que o agente tenha apenas que aprender o padrão de um ativo. A escolha do Mercado Bitcoin se deve a uma possibilidade futura de utilizar o agente treinado para realizar transações reais neste mercado através de sua API de negociação, de forma a tentar validar sua performance em um ambiente real caso o agente apresente bons resultados no ambiente simulado.

Após remover os dados não relacionados ao par Bitcoin/Mercado Bitcoin, a base de dados foi reduzida para cerca de 300 mil registros, representados pela **Figura 2**.

Figura 2: Preço (R$) do Bitcoin no Mercado Bitcoin ao longo dos registros restantes após o filtro.



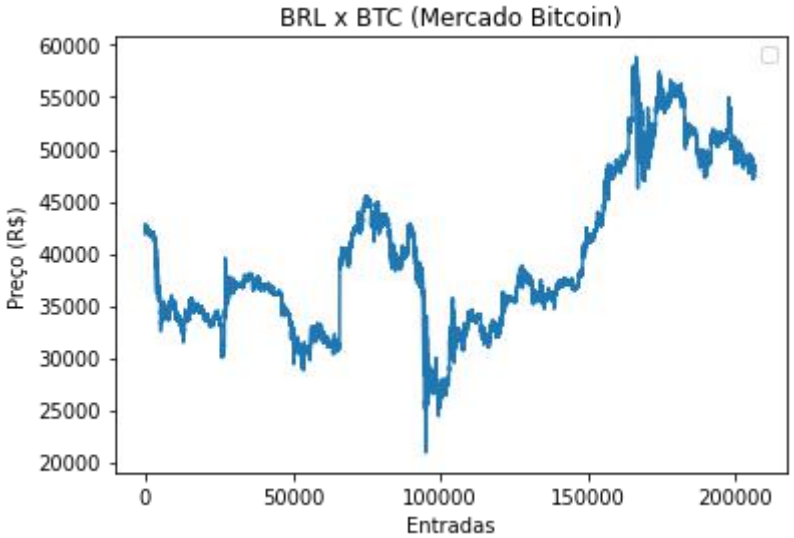
3.3 LIMPEZA

A base de dados atual contém muitas informações que não são de interesse deste projeto no momento. Dados como preço máximo e mínimo são desejados, no entanto, eles se referem ao período das últimas 24h a partir do instante da consulta, o que não é necessariamente muito útil para o agente.

Foram descartadas todas as informações que não foram utilizadas, restando apenas o preço e a data e hora da consulta. Através do preço, será possível derivar algumas das informações que foram descartadas, tendo a liberdade de escolher o período que se adeque melhor ao projeto.

Além disso, devido a recorrentes atrasos na API em atualizar os dados consultados, algumas informações repetidas foram registradas. Ao remover tais repetições, a base de dados foi reduzida a cerca de 200 mil registros, representados graficamente pelo preço na **Figura 3**.

Figura 3: Preço (R$) do Bitcoin no Mercado Bitcoin ao longo dos registros restantes após a limpeza.



3.4 TRANSFORMAÇÃO

No momento, para cada entrada da base, é possível acessar o preço da moeda no instante em que tal entrada foi consultada na API. Ou seja, temos o preço instantâneo em um intervalo de dois minutos ao longo de todo o período de coleta.

É comum que, em análises gráficas de ativos financeiros, seja utilizado um gráfico de vela ao invés do mais tradicional gráfico de linha. Nele são representados intervalos de tempo ao invés de instantes. Para um determinado intervalo, é dada uma representação gráfica, denominada vela, do valor inicial, máximo, mínimo e final de uma variável. É possível derivar a informação de preço instantâneo de forma a apresentar para o agente tais valores em um intervalo de tempo específico.

Para realizar a conversão, o primeiro passo é dividir a base de dados em grupos de n registros sequenciais, com a regra de que a última entrada de cada grupo seja duplicada para ser a primeira entrada do próximo grupo, de forma que não existam espaços de tempo entre os grupos. Inicialmente, a base foi dividida em grupos de 4, capaz de gerar um gráfico de velas com período de aproximadamente 6 minutos, pois existe um intervalo de 2 minutos entre cada entrada de um mesmo grupo.

Quanto mais registros são utilizados em um grupo, mais "informação" aquele grupo carrega, embora o total de registros na tabela final fique menor. Por esse motivo o número de 4 registros por grupo foi escolhido, de forma a não reduzir muito esse total, que já é limitado.

O próximo passo é gerar uma entrada para cada grupo, em uma nova base de dados, contendo o preço inicial, máximo, mínimo e final do grupo, como exemplificado na **Figura 4**. O "corpo" da vela é representado pelos valores inicial e final, enquanto os "pavios" são representados pelos valores mínimo e máximo. Além disso, cada entrada também deve possuir uma data e hora associada, que deve ser igual a que estava previamente associada ao valor final do grupo, que deve representar o instante em que o agente observa a vela em questão durante a simulação. Ao final do processo, a nova base de dados é formada com cerca de 70 mil registros, representados na íntegra pela **Figura 5**. Para maior compreensão da operação realizada, a **Figura 6** expõe uma seção aleatória dos dados após a transformação. A coloração do gráfico é realizada automaticamente pela biblioteca utilizada para gerá-lo, onde uma vela verde representa um intervalo onde o valor inicial é menor do que o final, ou seja, o preço do ativo sobe ao longo do período da vela, enquanto que uma vela vermelha representa o oposto.

Figura 4: Exemplo de conversão de 4 registros de preço para uma entrada no gráfico de velas.

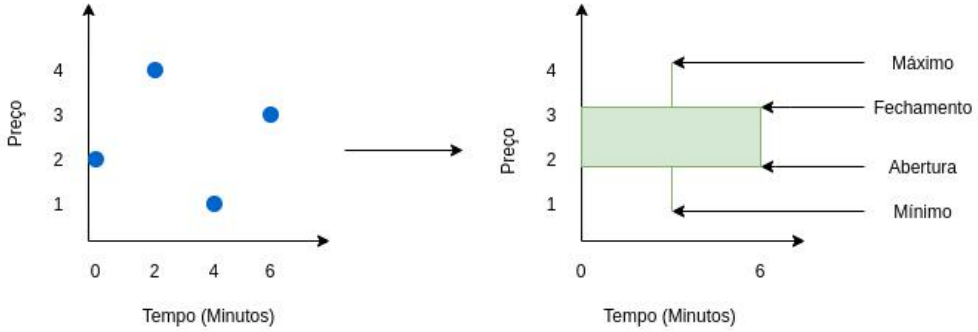


Figura 5: Gráfico de velas do preço (R$) do Bitcoin no Mercado Bitcoin.

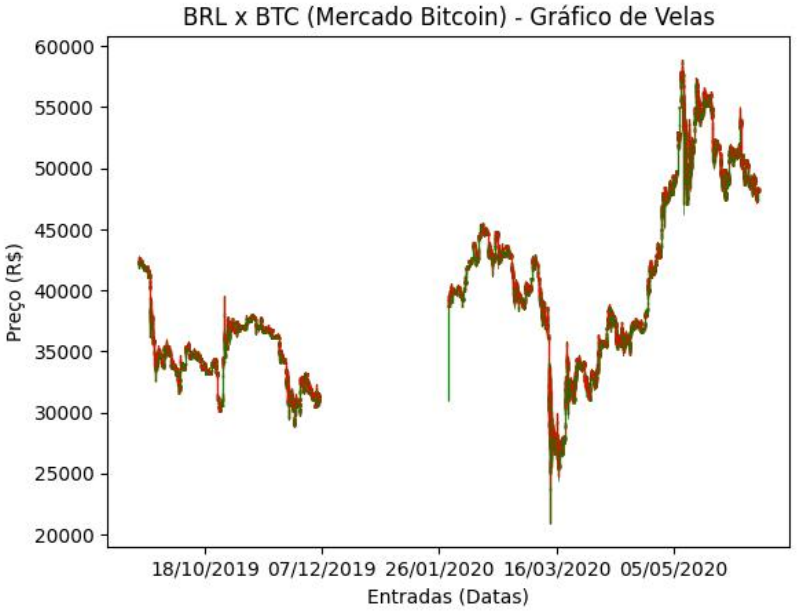
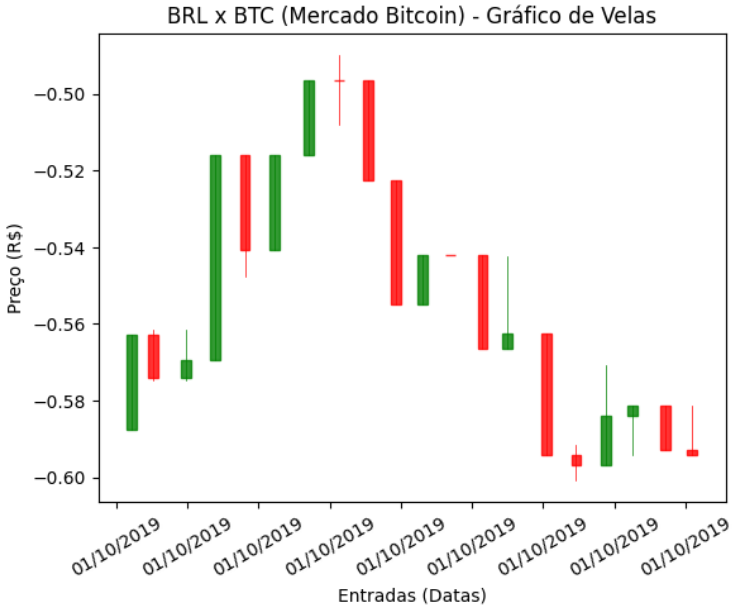


Figura 6: Seção aleatória do gráfico de velas para melhor compreensão dos resultados.



3.5 ESCALONAMENTO

A base de dados será utilizada para alimentar uma rede neural artificial que se encontra no centro do processo de aprendizado do agente. Por isso, um passo importante em seu tratamento é o escalonamento dos dados de forma que sua média se aproxime de zero, otimizando assim alguns processos computacionais que envolvem a rede.

Todos os valores obtidos no momento se encontram em uma mesma escala, pois todos derivam do preço do Bitcoin. No entanto, caso houvesse dados em escalas diferentes, o escalonamento também seria importante para garantir uma mesma ordem de grandeza entre os diversos tipos de dados, de forma que todos tenham o mesmo impacto na rede neural.

O procedimento que será utilizado para atingir o efeito desejado se chama padronização (*standardization*). Ele consiste em escalonar um grupo de dados de forma que a média entre eles seja igual a 0 e o desvio padrão seja igual a 1. Para cada valor a ser modificado, é aplicada a fórmula (1).

$$z = (x - \mu) / \sigma \qquad (1)$$

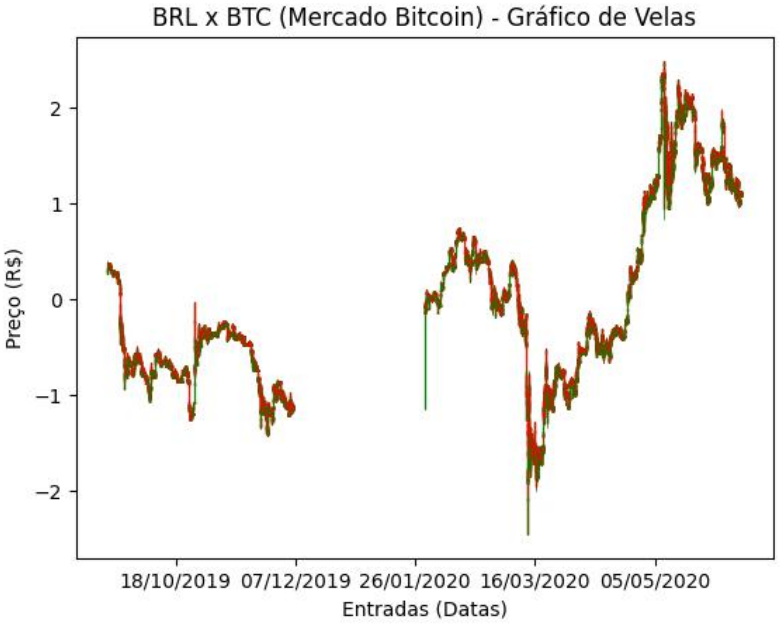
z = Valor escalonado

x = Valor original

μ = Média do conjunto original

σ = Desvio padrão do conjunto original

Figura 7: Gráfico de velas do preço (R$) do Bitcoin no Mercado Bitcoin após o redimensionamento.



3.6 SEGMENTAÇÃO

Ao analisar a **Figura 7** relativa aos dados após o escalonamento, é possível perceber que, embora a maior parte das velas do gráfico estejam dispostas de forma contínua, eventualmente ocorrem espaçamentos indesejados. Um exemplo discrepante é o período de janeiro de 2020, que não possui nenhum registro.

Para manter uma consistência no ambiente simulado, a base será segmentada de forma que o intervalo entre velas de um mesmo segmento seja de no máximo 15 minutos. Além disso, serão descartados segmentos que por ventura tenham menos do que 1000 registros, de modo que cada simulação dure no mínimo aproximadamente 4 dias.

Ao final do processo, a base foi dividida em 6 segmentos que juntos possuem cerca de 68 mil registros, representados graficamente por cores diferentes na **Figura 8**. Para uma melhor compreensão de todo o processo de tratamento do banco de dados, a **Figura 9** demonstra um fluxo das operações contendo os detalhes mais importantes de cada passo realizado.

Figura 8: Preço (R$) do Bitcoin no Mercado Bitcoin após a segmentação.

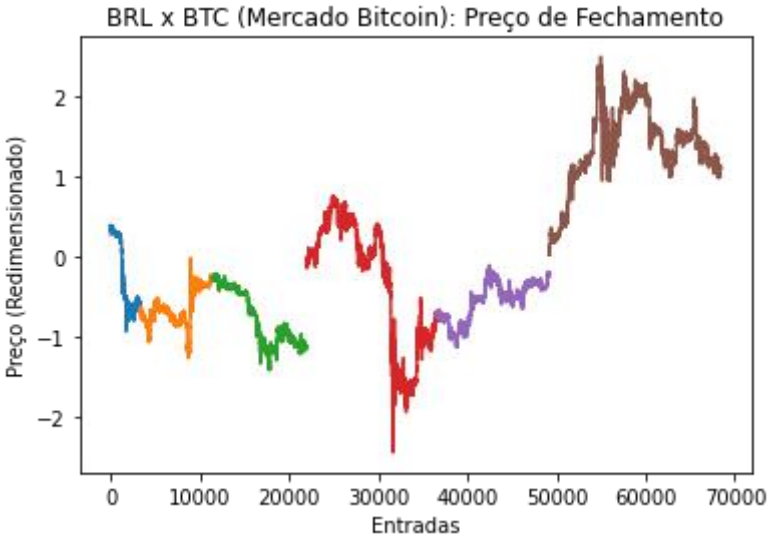
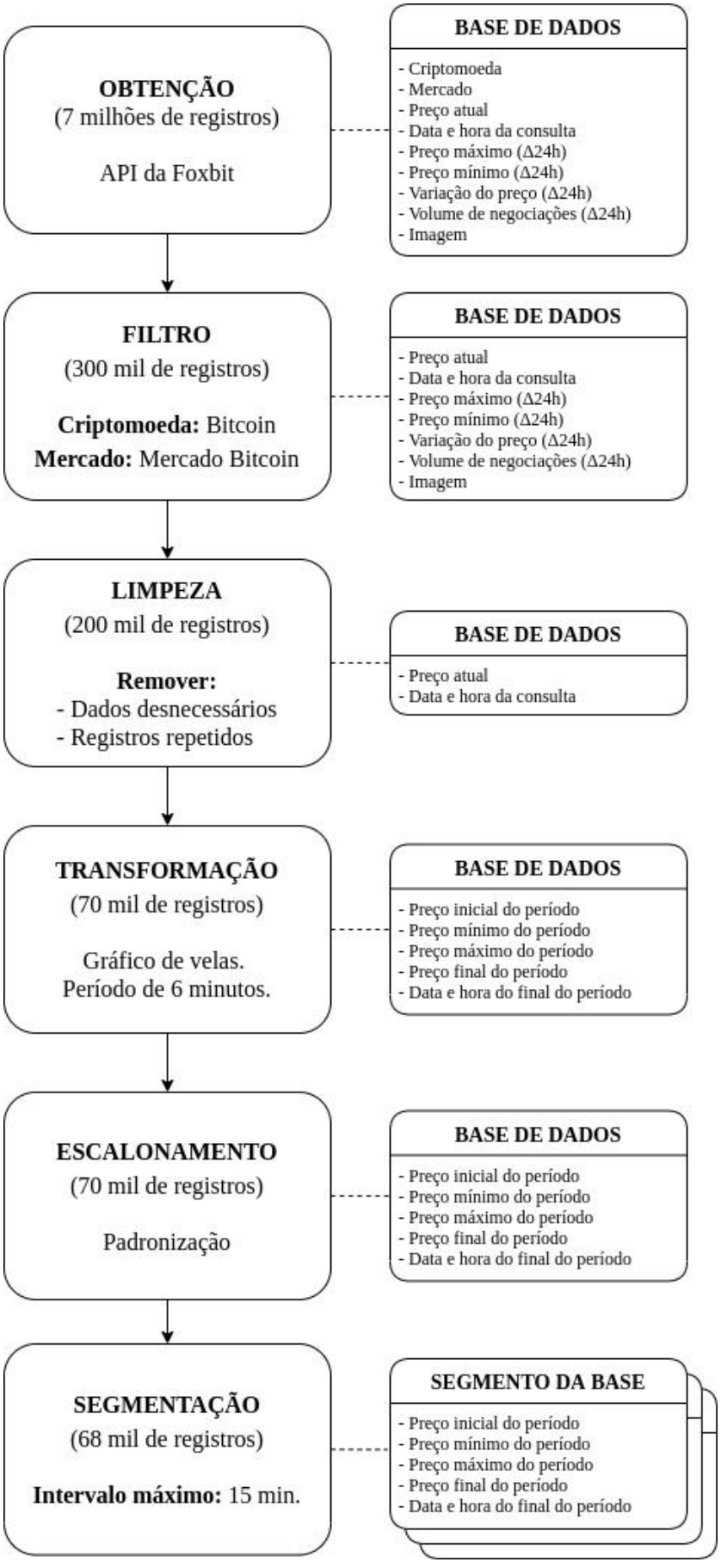


Figura 9: Fluxo de todos os passos realizados no tratamento do banco de dados.



4 AMBIENTE SIMULADO

Em um sistema baseado em Aprendizado por Reforço, um dos componentes principais é o ambiente responsável por prover ao agente estados observáveis ao longo de um episódio, seja ele real ou simulado. Além disso, o ambiente deve aceitar interações do agente no formato de ações discretizadas, no qual devem ser recompensadas a fim de incentivar um comportamento desejado.

Neste projeto, o ambiente é completamente simulado através dos dados históricos contínuos representados pela base de dados construída. Cada entrada da base nos fornece um instante no tempo em que o agente deve ser capaz de observar, agir e ser recompensado de acordo. A sucessão desses instantes somada às alterações provocadas pelo agente são o que constituem um **episódio de simulação**.

4.1 MODELAGEM

O ambiente é responsável por determinar todas as variáveis contidas dentro da simulação, assim como a relação entre elas e o que pode ou não ser observado e manipulado pelo agente.

Neste projeto, a simulação consiste basicamente de uma sucessão de unidades de tempo, representado por cada entrada na base de dados, onde o agente é capaz de observar os dados históricos relacionados a um ativo financeiro até o instante atual, e decidir se ele irá comprar ou vender unidades negociáveis deste ativo. Não serão levados em consideração os eventuais atrasos durante o processo de observação, tomada de decisão, e envio de ordem que normalmente ocorrem no ambiente real, de forma que é considerado que a execução da ordem do agente seja realizada no exato instante de observação das informações associadas à entrada atual.

No início de cada episódio da simulação, o agente possui R$10.000 que representam dinheiro em sua suposta conta no Mercado Bitcoin. A cada transação realizada, seja de compra ou venda, será aplicada a taxa de 0,7% sobre o total da operação. Não será contabilizada nenhuma taxa relativa a transferência do dinheiro para a conta de banco pessoal

do agente, embora tal taxa exista no ambiente real. Caso o agente reduza o dinheiro em sua conta a um valor negativo, a simulação é encerrada e uma nova irá começar no instante seguinte, com todas as variáveis reiniciadas.

4.2 OBSERVAÇÃO

No mundo real, existe um número incontável de informações capazes de serem utilizadas por um agente para dar suporte a sua tomada de decisão. Em um ambiente virtual, precisamos delimitar o que será de fato informado, e como esta informação será apresentada.

A simulação está limitada de acordo com as informações contidas na base de dados adquirida e o que é possível de ser derivado a partir delas. Como ponto de partida para os primeiros testes do ambiente simulado, a cada instante da simulação serão disponibilizados como observação para o agente os valores inicial, máximo, mínimo e final do preço do Bitcoin em Reais, negociadas no Mercado Bitcoin, dos últimos 6 minutos do instante que a entrada representa.

Se toda a base de dados fosse utilizada em um mesmo momento, uma simulação completa seria composta de 6 segmentos isolados, onde cada um representa um conjunto de instantes sucessivos a serem observados, totalizando cerca de 68 mil estados observáveis. No entanto, a base de dados será dividida posteriormente de forma que uma parte será utilizada para o treinamento do agente, enquanto outra será usada para a validação de seu aprendizado.

4.3 AÇÕES

O agente precisa ser capaz de interagir com o ambiente de forma a tentar atingir seu objetivo. Tais interações são definidas através de um grupo de ações distintas que são modeladas de acordo com o ambiente no qual o agente está interagindo.

É possível realizar tal modelagem de inúmeras formas, e isso faz parte do processo de exploração deste projeto. Em um primeiro momento, tais interações foram definidas de forma a determinar qual é a posição do agente imediatamente após o estado observado. As possíveis posições foram discretizadas em valores inteiros pertencentes ao intervalo [-3, 3]. Cada valor

a deste intervalo possui a seguinte interpretação:

$a \in [-3, -1]$: Estar vendido em $a * k$ Bitcoins.
$a = 0$: Sem posicionamento.
$a \in [1, 3]$: Estar comprado em $a * k$ Bitcoins.

onde k é um parâmetro a ser ajustado de forma que a compra de k unidades de Bitcoin têm seu custo na ordem de grandeza de R$100. Isso é feito para ajustar as transações do agente a um tamanho condizente com seu capital inicial de R$10.000. Inicialmente, k é igual a 0,001.

Note que, nesta modelagem, o agente não diz exatamente quantas bitcoins ele pretende comprar ou vender, e sim qual a posição que ele deseja se encontrar no instante em que interage com o ambiente. Supondo, por exemplo, que em determinado momento o agente se encontre na posição -2 (vendido em duas Bitcoins) e que no próximo instante ele escolha estar na posição 3 (comprado em três Bitcoins). Neste caso, será preciso que o ambiente execute uma ordem de compra de 5 Bitcoins para que a ação que o agente escolheu seja realizada conforme o modelo descrito.

4.4 RECOMPENSAS

Toda vez que o agente realiza uma ação, o ambiente precisa recompensar esta ação de forma a estimular algum tipo de comportamento. Esta recompensa pode ser positiva ou negativa e será utilizada durante o processo de aprendizado para que o agente correlacione padrões entre os estados observados, ações realizadas e as recompensas associadas a eles.

Assim como diversos outros componentes neste projeto, a modelagem das recompensas é algo que influencia diretamente na capacidade do agente em aprender a se comportar da forma esperada. Inicialmente, ela é expressa a partir da seguinte fórmula:

$$r = (p_{t+1} \div p_t) - 1 \qquad (2)$$

onde p_t representa o patrimônio do agente no tempo t. Patrimônio é definido aqui como a soma entre o dinheiro que o agente ainda possui na conta e o valor em Reais da quantidade de Bitcoins que possui. É possível que o agente possua uma quantidade negativa de Bitcoins, caso esteja operando vendido. Neste caso, o valor em Reais da quantidade que possui será negativo.

A interpretação da fórmula (2) é de que a recompensa representa a variação do patrimônio do agente no próximo instante de tempo em relação ao instante de tempo atual, onde a ação foi realizada. Uma variação positiva no patrimônio representa uma recompensa positiva, incentivando o comportamento que levou a este resultado.

5 AGENTE

Em um sistema baseado em Aprendizado por Reforço, o agente é responsável por interagir com o ambiente, de forma a aprender algum tipo de comportamento. Nele se concentra todo o processo de aprendizado. Por meio de suas interações através dos episódios de simulação, o agente deve tentar extrair experiências que o ajude a entender padrões para conseguir um melhor desempenho no futuro.

A simulação consiste de uma sucessão de instantes, onde a cada um, o agente observa um estado provido pelo ambiente, realiza alguma ação disponível e recebe uma recompensa ou penalidade pela ação realizada. Durante todo esse processo, o agente deve ser capaz de:

- Escolher qual a melhor ação para ser realizada dado o estado observado.
- Guardar as experiências obtidas durante a simulação para que possa utilizá-las no processo de aprendizado.
- Aprender com as experiências que armazenou, de forma a incorporá-las em seu processo de tomada de decisão.

A **Figura 10** ilustra os componentes de um sistema de Aprendizado por Reforço Profundo e suas interações. A **Figura 11** representa uma visão geral simplificada das etapas dentro de uma simulação. Em cada iteração, o ambiente provê para o agente a observação do instante atual da simulação, que é utilizado como entrada de uma rede neural para se determinar a ação que o agente irá realizar. Em seguida, o ambiente recompensa o agente pela ação escolhida, de forma positiva ou negativa, e disponibiliza o estado observável do próximo instante. O agente então registra a experiência vivenciada neste instante em uma estrutura chamada de **Memória de Repetição**. Caso a quantidade de registros salvos em tal estrutura tenha ultrapassado um valor mínimo parametrizado, o agente se utiliza de um grupo de experiências passadas registradas na estrutura para ajustar os pesos da rede neural, aprimorando assim sua capacidade de tomar decisões visando recompensas cada vez maiores, de acordo com o objetivo pré-estabelecido.

Figura 10: Componentes de um sistema de Aprendizado por Reforço Profundo.

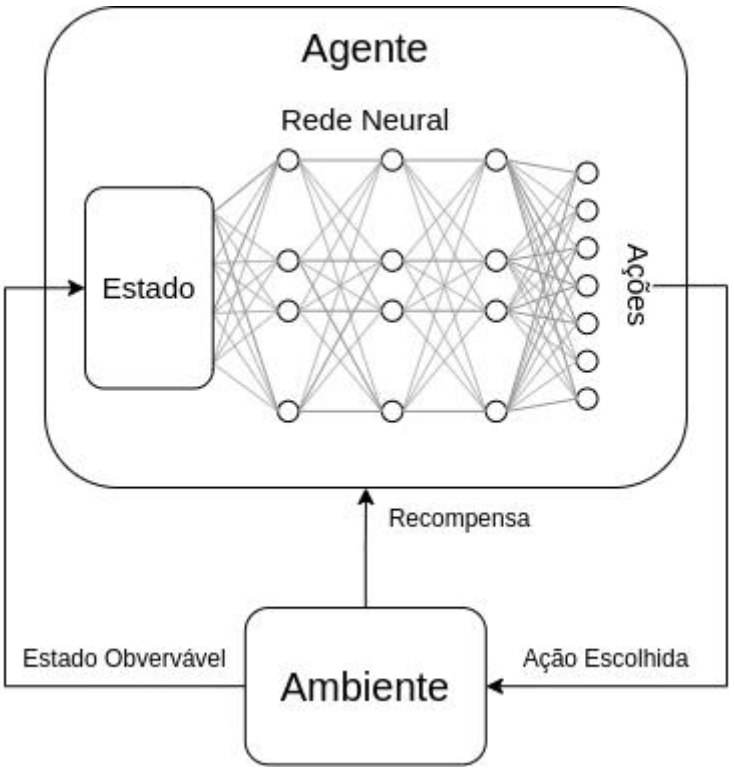
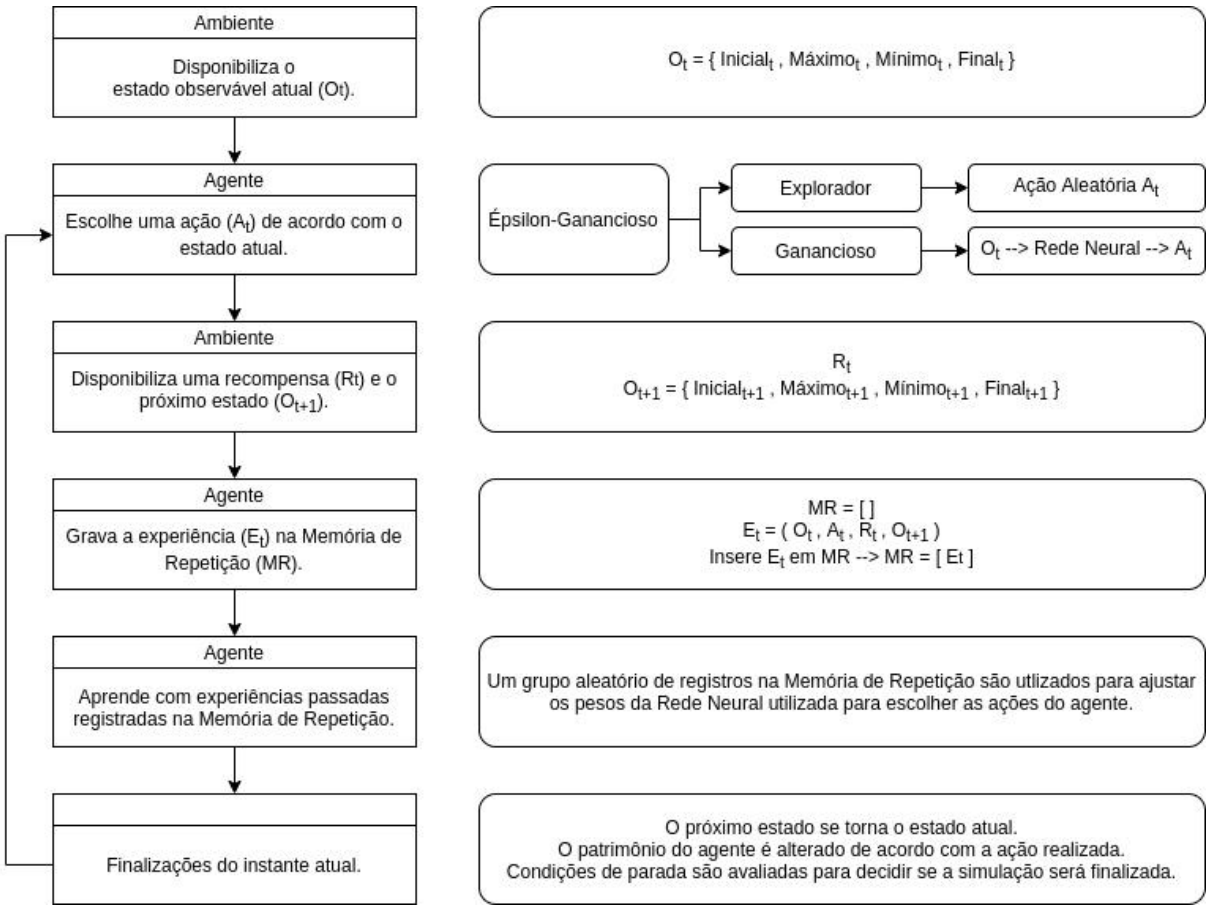


Figura 11: Visão geral simplificada das etapas de uma simulação



5.1 ESCOLHA DA AÇÃO

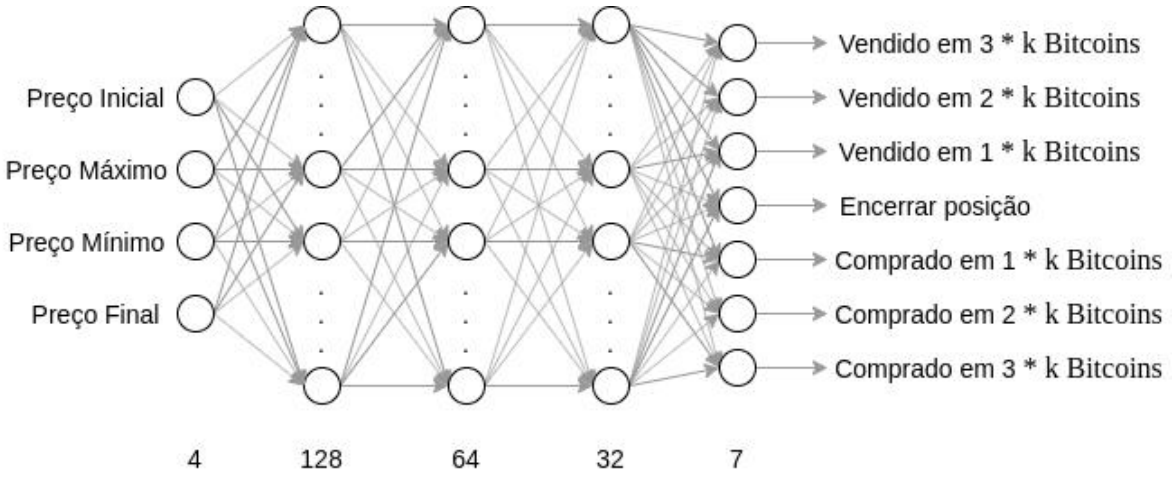
A escolha da ação a ser realizada em um determinado instante é determinada através de uma rede neural. A rede recebe como entrada o estado observado pelo agente em um determinado instante, e entrega em retorno a ação a ser realizada. No início do treinamento, tal rede terá seus pesos inicializados com algum tipo de aleatoriedade, o que significa que o agente não deve ter um desempenho melhor do que se tivesse apenas escolhendo a ação de forma aleatória. No entanto, ao longo do treinamento, o agente deve aprender com suas experiências de forma a ajustar os pesos da rede e ter alguma melhora em sua performance.

Inicialmente, optou-se por escolher uma modelagem bem comum, de forma a ter algum tipo de base inicial comparativa para futuras iterações da rede. O tipo de rede escolhida foi uma rede totalmente conectada, e seus parâmetros foram escolhidos de forma usual,

tentando manter a simplicidade do modelo. A estrutura escolhida pode ser observada pela representação gráfica da **Figura 12**, e possui as seguintes características:

- Uma camada de entrada com quatro neurônios relativos aos valores observados pelo agente: Preço inicial, máximo, mínimo e final do período relativo ao instante atual da simulação.
- Três camadas escondidas totalmente conectadas com 128, 64 e 32 neurônios respectivamente.
- Uma camada de saída contendo 7 neurônios relativos às possíveis ações a serem realizadas pelo agente, definidas pela modelagem do ambiente.
- Taxa de aprendizado de 0,0005 no processo de retropropagação de ajuste de pesos da rede.

Figura 12: Representação da modelagem inicial da rede neural.



Para permitir que o agente explore mais o ambiente, foi realizada uma modificação no passo de escolher a ação através da implementação de um algoritmo épsilon-ganancioso. Seu objetivo é permitir que o agente transite de um perfil explorador (que escolhe a ação de forma aleatória) para um perfil ganancioso (que escolhe a ação indicada pelo retorno da rede neural) ao longo do treinamento. Ele é implementado da seguinte forma:

- É definido um valor inicial no intervalo [0,1] para uma variável épsilon.
- Para escolher uma ação, primeiro é gerado aleatoriamente um número dentro do intervalo [0,1]:

    - Se o número for menor que épsilon, o agente escolhe uma ação aleatória.
    - Se o número for maior que épsilon, o agente escolhe a ação indicada pelo retorno da rede neural.
- Ao longo do treinamento, periodicamente, o valor de epsilon decresce de acordo com alguma taxa até atingir um valor mínimo pré definido.

Neste projeto, o valor inicial de épsilon foi definido como 1, e ele decresce a cada instante da simulação, logo após a escolha da ação pelo agente, de tal forma que atinja o valor mínimo 0,001 ao alcançar o instante que representa uma porcentagem parametrizada, definida inicialmente como 50%, do total de instantes da simulação do treinamento.

A ideia do algoritmo é permitir que o agente explore a maior quantidade de cenários possíveis durante a fase inicial do treinamento, para evitar que o agente se prenda rápido demais a algum comportamento subótimo, de forma a tentar aumentar a probabilidade de encontrar padrões melhores de tomada de decisão.

5.2 GUARDAR EXPERIÊNCIAS

Ao longo do treinamento, o agente deve registrar as experiências que obtém de forma a poder aprender com elas no futuro. Este processo se chama **Repetição de Experiência**, e tem como consequência a separação lógica entre processo de aprendizado e a simulação no qual o agente adquire as experiências. É um processo opcional que permite reutilizar quantas vezes desejar as experiências passadas, ao invés de apenas a experiência do instante atual da simulação.

As experiências são armazenadas em uma estrutura chamada de Memória de Repetição, onde cada experiência registrada contém:

- O estado observado pelo agente no instante atual.
- A ação que o agente realizou a partir da observação.
- A recompensa adquirida de acordo com a ação realizada.
- O estado observado pelo agente no próximo instante da simulação.

A vantagem de se registrar as experiências desta forma, é que cada uma pode ser utilizada de forma isolada, em qualquer ordem desejada, pois cada registro contém tudo que é necessário para realizar um passo isolado de ajuste de pesos de uma rede neural, como será evidenciado no passo seguinte.

Um outro detalhe sobre esta estrutura é seu tamanho total. A estrutura poderia crescer de forma indefinida, no entanto, isto pode representar uma sobrecarga em termos de memória. Foi estabelecido inicialmente que seu tamanho máximo seria de 50.000 registros, e que um novo registro irá substituir o mais antigo ao ser adicionado na estrutura após ela alcançar tal tamanho.

A partir deste conjunto de experiências, o agente poderá realizar o processo de aprendizado, no momento que for mais apropriado, e utilizando os registros que quiser, de acordo com o algoritmo escolhido para acessá-los.

5.3 APRENDER COM EXPERIÊNCIAS PASSADAS

A Repetição de Experiência deve ocorrer periodicamente durante o treinamento. A determinação da periodicidade faz parte da modelagem, e inicialmente será escolhido realizar uma vez a cada instante da simulação, logo após a escolha da ação.

O primeiro passo é determinar quantas e quais experiências contidas na Memória de Repetição serão utilizadas nesta rodada do processo. Existem diversas abordagens possíveis de serem escolhidas, e neste primeiro momento, optou-se por utilizar 256 registros escolhidos de forma aleatória. Enquanto não houverem registros o suficiente para selecionar as 256 experiências, o processo não irá ocorrer. Ou seja, apenas quando a simulação chegar ao instante 256 que o agente irá começar a aprender com suas experiências passadas.

Para cada registro, é realizado o seguinte procedimento:

- O estado observado no instante atual (do registro, e não da simulação) é utilizado como entrada da rede. O retorno recebido é uma lista (L_1) contendo um valor numérico para cada ação possível de ser realizada. Esses valores são

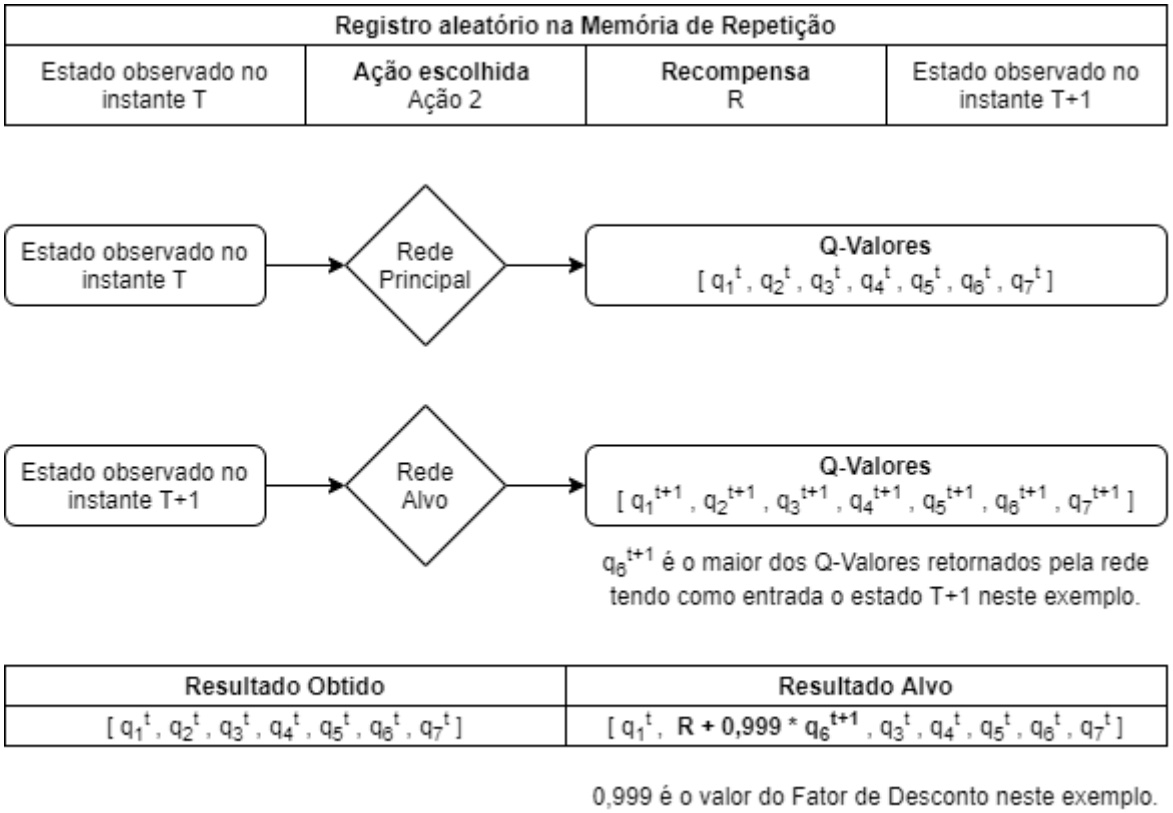
denominados Q-Valor. Quanto maior o Q-Valor de um par estado/ação, maior é a confiança da rede de que tal ação é a melhor a ser tomada para este estado.

- O estado observado no próximo instante (do registro, e não da simulação) é também utilizado como entrada da rede, gerando uma lista (L_2) como resultado. Essa lista representa os Q-Valores para as ações relativas ao próximo instante.
- O objetivo é calcular o erro do resultado obtido pela rede neural ao gerar L_1 com um resultado alvo que será calculado à parte. Este resultado alvo representa uma atualização do resultado original com algumas informações obtidas através da experiência do agente. Para calculá-lo, é primeiramente feita uma cópia de L_1, que chamamos de L_a. Nela, o Q-Valor referente à ação que o agente escolheu será substituído pela recompensa recebida somado ao maior Q-Valor de L_2 multiplicado por um parâmetro chamado de Fator de Desconto, com seu valor escolhido inicialmente igual a 0,999. A ideia deste passo é substituir o Q-Valor retornado pela rede referente a ação que o agente escolheu pela própria recompensa que o ambiente deu ao agente ao escolher tal ação, somado da expectativa de recompensa para o próximo instante de acordo com um peso pré-determinado.
- L_a é então utilizada para calcular o erro e realizar o processo de retro-propagação, atualizando assim os pesos da rede neural.

Como o resultado alvo da rede, utilizado para calcular o erro do resultado original, está utilizando a própria rede para ser calculado (através dos Q-Valores do estado do próximo instante), ao atualizar os pesos da rede, o resultado alvo para um mesmo estado muda de valor. Portanto, a atualização dos pesos tenta aproximar a saída da rede de um alvo que está em movimento, causando oscilações no treinamento. A solução utilizada para amenizar esse problema foi a de utilizar uma segunda rede neural, chamada de Rede Alvo, inicializada com os mesmos pesos da rede principal, que é atualizada periodicamente, ao invés de a cada rodada do processo. Inicialmente, foi escolhida a periodicidade de 1.000 iterações para realizar a atualização. O resultado alvo agora é calculado utilizando essa Rede Alvo, ao invés da rede principal. Com um alvo que se move pouco, a otimização se torna mais estável, melhorando o aprendizado.

Um detalhe importante é que embora os passos anteriores sejam realizados para cada registro dentro do lote de 256 experiências escolhidas aleatoriamente, eles são realizados de forma matricial, e não iterativa, de forma que todos os cálculos são realizados ao mesmo tempo para todos os registros. A **Figura 13** representa o processo descrito para facilitar a compreensão.

Figura 13: Representação do processo de Repetição de Experiência, para um registro



5.4 FINALIZAÇÃO DO INSTANTE ATUAL

Após a realização do passo anterior, é necessário alguns pequenos procedimentos antes de dar início a próxima iteração da simulação. Em primeiro lugar, o estado observável entregue pelo ambiente após a escolha da ação do agente (que foi registrado na Memória de Repetição como o estado do próximo instante) toma a posição de estado atual, pronto para ser utilizado como base na tomada de decisão do agente na próxima iteração. É verificado se tal estado realmente existe, ou seja, se ele não é nulo. Caso seja nulo, isso significa que a simulação chegou ao fim, e os resultados obtidos são disponibilizados.

Caso o estado não seja nulo, o patrimônio do agente é ajustado de acordo com a ação realizada neste instante e o valor atual do ativo negociado. Se o agente tiver esgotado seu patrimônio, como por exemplo ao tentar comprar um montante de Bitcoins sem ter o valor necessário em caixa para realizar tal operação, a simulação é encerrada neste instante. Os resultados obtidos são disponibilizados e uma nova simulação terá início neste instante, onde a anterior foi finalizada, onde o patrimônio do agente será restaurado ao seu valor inicial.

Caso a simulação ainda esteja em execução, ou seja, o estado observável não é nulo e o agente ainda possui dinheiro em caixa, é preciso decair o valor de épsilon, do algoritmo epsilon ganancioso que governa a tomada de decisão do agente. Tal valor decai segundo uma taxa pré-estabelecida, de forma que o agente caminhe para um perfil cada vez mais ganancioso, em contraste com o perfil exploratório que possui no início da simulação. Além disso, é preciso atualizar os pesos da Rede Alvo igualando-os aos da Rede Principal, caso a periodicidade de atualização pré-estabelecida tenha sido alcançada. Após tais ajustes, o agente está pronto para realizar uma nova iteração da simulação.

5.5 EXEMPLO

Para uma melhor compreensão de todos os processos descritos nesta seção, considere o seguinte exemplo hipotético:

- Suponhamos que o agente possua um patrimônio total de R$1.000 e que não possui nenhuma posição aberta no atual instante de simulação.
- Neste instante, o estado observado pelo agente foi:
    - (40 , 42 , 39 , 41). Ou seja, o preço inicial, máximo, mínimo e final do Bitcoin no período relativo ao instante atual de simulação foi de R$40, R$42, R$39 e R$41 respectivamente.
- Ao utilizar tal estado como entrada na rede neural principal, ela retornou os seguintes Q-Valores:
    - (-0.03 , -0.01 , 0.01 , 0 , 0.001 , 0.002 , 0.1). Ou seja, a rede acredita que o patrimônio do agente no próximo instante sofrerá uma variação de:
        - -3% caso ele escolha estar vendido em $3 * k$ Bitcoins.
        - -1% caso ele escolha estar vendido em $2 * k$ Bitcoins.

        - 1% caso ele escolha estar vendido em $1 * k$ Bitcoins.
        - 0% caso ele escolha encerrar sua posição atual.
        - 0.1% caso ele escolha estar comprado em $1 * k$ Bitcoins.
        - 0.2% caso ele escolha estar comprado em $2 * k$ Bitcoins.
        - 10% caso ele escolha estar comprado em $3 * k$ Bitcoins.
    - Suponhamos que k seja igual a 1. Neste caso, a rede está recomendando que o agente escolha a ação de estar comprado em 3 Bitcoins, pois acredita que o preço irá subir no próximo instante.
- O valor de épsilon do algoritmo épsilon-ganancioso se encontra em 0.2. O número aleatório gerado para definir qual será o perfil de escolha do agente neste instante foi de 0.12. Ou seja, como o valor sorteado foi menor que o de épsilon, o agente irá escolher uma ação aleatória, ao invés da ação recomendada, seguindo o perfil explorador.
- A ação aleatória tomada pelo agente foi a ação de ficar vendido em 2 Bitcoins.
    - Com isso, seu patrimônio passa a ser de R$1082, porém com uma dívida de 2 Bitcoins a ser paga no futuro.
- O ambiente então revela o próximo instante para o agente, assim como a sua recompensa.
    - O próximo instante revelado foi (41 , 41 , 32 , 35). Ou seja, o preço inicial, máximo, mínimo e final do Bitcoin no período relativo ao próximo da simulação foi de R$41, R$41, R$32 e R$35 respectivamente.
    - Como o preço do Bitcoin diminuiu, ao contrário do que havia sido previsto pela rede, a dívida de 2 Bitcoins do agente também diminuiu, gerando um patrimônio ajustado de R$1.012 (R$1082 - R$70).
    - Com isso, a recompensa dada ao agente pelo ambiente foi de 0.012. Ou seja, seu patrimônio valorizou 1,2%.
- O agente registra a experiência vivenciada em sua Memória de Repetição, com o seguinte formato:
    - ((40 , 42 , 39 , 41) , 1, 0.012 , (41 , 41 , 32 , 35))
    - (40 , 42 , 39 , 41): Estado do instante atual.
    - 1: Ação escolhida.
        - O número 1 foi registrado pois existe um total de 7 ações numeradas de 0 a 6, onde a ação número 1, escolhida pelo agente, é a de estar vendido em 2 Bitcoins.

    - 0.012: Recompensa dada ao agente.
    - (41 , 41 , 32 , 35): Estado do próximo instante.
- Neste exemplo, vamos supor que o agente já havia previamente registrado um número de experiências maior que 256, de forma a poder prosseguir com a etapa de aprendizado ainda neste instante de simulação.
- São selecionados 256 registros aleatórios na Memória de Repetição que são processados de forma matricial no ajuste dos pesos da rede. No entanto, iremos considerar que a experiência registrada no instante atual foi um dos selecionados neste passo e iremos apenas olhar para ele de forma a facilitar a compreensão.
- A entrada do instante atual deste registro é utilizada como entrada da rede neural principal, e ela retorna o seguinte resultado:
    - (-0.03 , -0.01 , 0.01 , 0 , 0.001 , 0.002 , 0.1).
    - Note que foi o mesmo resultado gerado durante a simulação em si, mas isso não é sempre o caso pois este registro pode ser utilizado em um momento posterior, onde os pesos da rede já foram ligeiramente ajustados, gerando um resultado ligeiramente diferente.
- A entrada do próximo instante deste registro é utilizada como entrada da rede alvo, e ela retorna o seguinte resultado:
    - (0.07 , 0.02 , 0 , 0 , -0.02 , -0.04 , -0.09).
    - Este conjunto de Q-Valores representa uma expectativa de recompensa futura para o agente. Iremos selecionar o valor 0.07 pois é o maior valor deste resultado. Ele representa o fato de que após a escolha da ação 1 (ficar vendida em 2 Bitcoins) no instante atual, o agente se encontrará no próximo instante indicado no registro. Assim, existe uma expectativa de que o agente ganhará a recompensa de 0.07 caso escolha a ação 0 (ficar vendida em 3 Bitcoins) neste próximo instante. Ou seja, existe uma projeção de ganhos futuros de acordo com a ação escolhida no presente.
- Com isso, temos que a recompensa ajustada para ação 1 escolhida pelo agente é de:
    - 0.012 + 0.07*0,999 = 0.08193.
    - Ou seja, é a recompensa dada ao agente pelo ambiente somado da expectativa de recompensa futura, associada a um Fator de Desconto de 0,999.
- Para realizar o passo de ajustes de peso da rede neural principal, é comparado o erro entre:
    - (-0.03 , -0.01 , 0.01 , 0 , 0.001 , 0.002 , 0.1). A saída original da rede.

- (-0.03 , **0.08193** , 0.01 , 0 , 0.001 , 0.002 , 0.1). A saída alvo.
- Note que através da experiência vivenciada pelo agente, foi possível perceber que a rede fez uma avaliação ruim da entrada. Ela previu um retorno negativo caso o agente escolhesse a ação 1. No entanto, através da experiência simulada, foi constatado que ele obteve uma recompensa positiva, e tal ajuste nos pesos é feito de forma a tentar aproximar o resultado original ao resultado alvo.

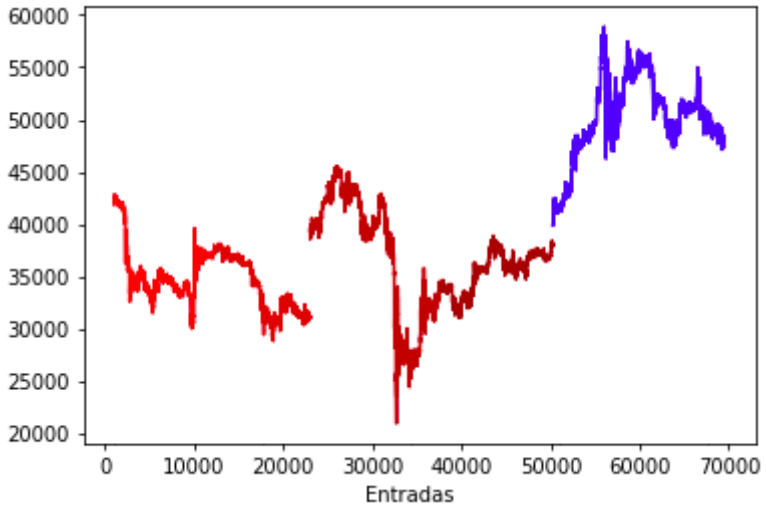
- Por fim, são realizados algumas verificações e ajustes:
  - O estado do próximo instante não é nulo, seu valor é (41 , 41 , 32 , 35). Portanto, a simulação dará continuidade e tal estado se torna o estado atual da próxima iteração.
  - O patrimônio do agente é ajustado de acordo com os valores do instante atual, ou seja:
    - Seu patrimônio deixa de ser de R$1.000.
      - R$1.082 - R$82 da dívida, pois está vendido em 2 Bitcoins e o valor do Bitcoin era de R$41.
    - E passa a ser: R$1.012.
      - R$1.082 - R$70 da dívida, pois ainda se encontra vendido em 2 Bitcoins, mas o valor do Bitcoin passou a ser R$35.
  - Como o agente ainda possui dinheiro em caixa, a simulação contínua e os seguintes ajustes ocorrem:
    - O valor de epsilon do algoritmo epsilon ganancioso decai segundo alguma taxa e deixa de ser 0.2 para se tornar 0.15.
    - É verificado se já é o momento de atualizar a Rede Alvo com os pesos da Rede Principal e a operação é realizada, caso seja o caso. Por exemplo, se tal periodicidade fosse 1 atualização a cada 10 iterações, e se esta iteração fosse por exemplo a iteração de número 20, a rede seria atualizada neste passo.
  - Este instante da simulação chega ao fim, e o processo recomeça no momento do agente tomar sua próxima decisão, dado que o estado observável atual já está definido.

6 EXPERIMENTOS

Com os módulos de ambiente e agente modelados, é possível iniciar os primeiros experimentos do nosso sistema de aprendizado por reforço profundo. Esta seção será responsável por apresentar os experimentos realizados a partir desta modelagem inicial do sistema, assim como todas as modificações que foram realizadas de forma a tentar melhorar os resultados obtidos. As métricas utilizadas para medir os resultados dos testes foram evoluindo à medida que novos experimentos foram sendo realizados, portanto serão apresentadas no momento em que foram introduzidas.

Neste primeiro momento, iremos dividir nossa base de dados históricos em duas partes, onde uma será utilizada para treinar o agente, totalizando cerca de 75% dos registros, e a outra para testar seus resultados em dados inéditos, composta pelos 25% restantes. A **Figura 14** representa os segmentos da base de dados, onde estão representados em vermelho os segmentos de treino, e em azul o segmento de teste.

Figura 14: Segmentos da base de dados divididos em treino (em vermelho) e teste (em azul).



Vale ressaltar que todos os experimentos foram realizados utilizando a plataforma Google Colab em sua versão gratuita que fornece um tempo de execução máximo de 12 horas, sem garantia de execução ininterrupta, com as seguintes especificações de hardware:

- GPU: NVIDIA Tesla K8, 2496 núcleos CUDA, 12GB GDDR5 VRAM.

- CPU: Intel Xeon, 2.3Ghz, 1 núcleo, 2 threads.
- RAM: ~12.6 GB.
- Disk: ~33 GB.

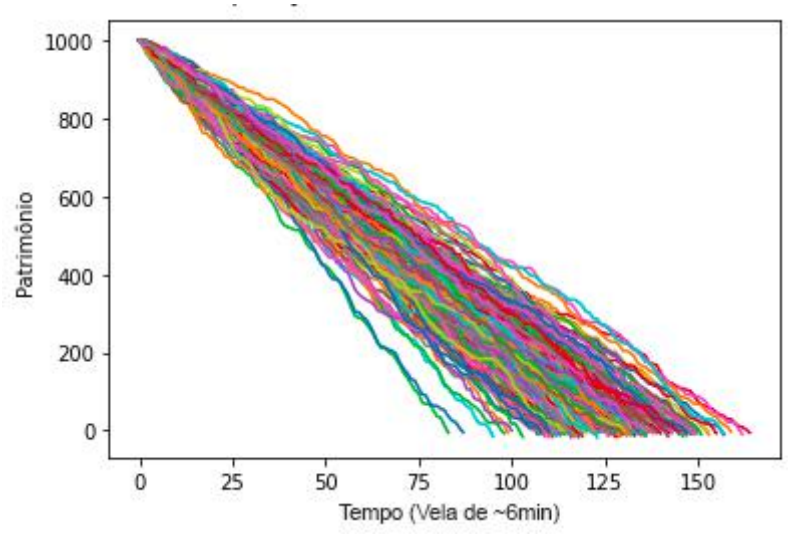
6.1 EXPERIMENTO 1 - AGENTE ALEATÓRIO

De forma a ter uma base comparativa inicial, foi realizada uma simulação utilizando a parte da base de dados designada para teste e um agente aleatório, ou seja, que a cada iteração da simulação escolhe uma ação aleatória dentre as possíveis ações disponíveis.

A **Figura 15** representa o gráfico da evolução do patrimônio em função do tempo (instantes de simulação). Note que existem muitas linhas diferentes no gráfico. Isso se deve ao fato de que o agente perde todo seu dinheiro antes de chegar ao final do segmento de teste. Toda vez que isso ocorre, seu patrimônio é reiniciado e uma nova tentativa ocorre no instante onde o anterior parou, até a base de teste chegar ao fim.

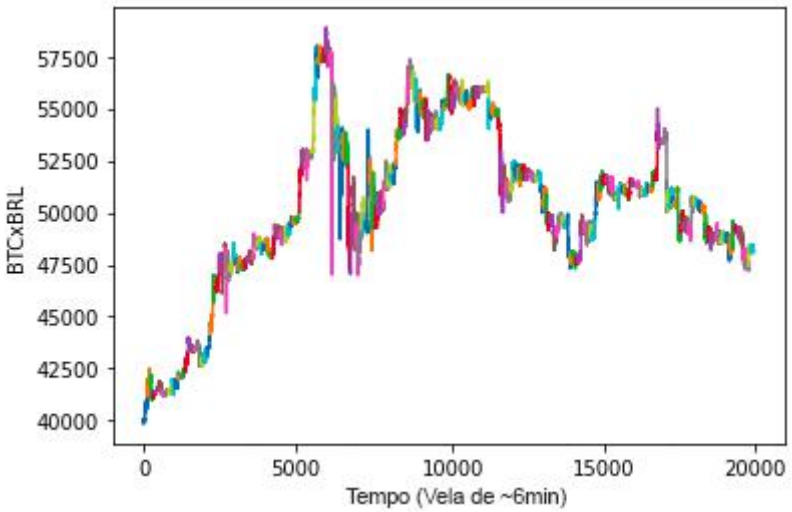
É possível perceber que seus resultados são bem ruins, como esperado, onde a cada tentativa o agente perde todo seu dinheiro em não mais do que 200 iterações.

Figura 15: Gráfico de patrimônio x tempo (número de velas) das diversas tentativas do agente aleatório.



A **Figura 16** representa a base de dados de teste dividida de forma que cada parte colorida representa o quanto um agente aleatório percorreu antes de reduzir seu patrimônio a zero. O gráfico se encontra com muitas subdivisões, o que implica que o agente não está sobrevivendo por muito tempo. São necessárias cerca de 1000 tentativas para percorrer a base de testes completamente.

Figura 16: Base de dados de teste dividida pelos agentes aleatórios que a percorreram.



Foram escolhidas três métricas para utilizar de forma a comparar os resultados obtidos durante o teste:

- **(Lucro total / Prejuízo total) * 100:**

  O lucro total se refere à soma de todos os ganhos obtidos em operações onde houve lucro enquanto o prejuízo total se refere a soma de todas as perdas sofridas em operações onde houve prejuízo. O objetivo desta métrica é tentar entender se o agente está sendo capaz de ganhar mais do que ele perde ao longo do teste realizado. Resultados positivos nesta métrica são todos os resultados acima de 100%, ou seja, quando o agente finaliza o teste com um lucro real em seu balanço total.

- **(Quantidade de negociações com lucro / Quantidade total de negociações) * 100**

  A quantidade de negociações com lucro se refere a todas as operações

realizadas que resultaram individualmente em algum ganho. O objetivo desta métrica é tentar entender a probabilidade do agente ter algum lucro ao abrir uma nova posição, independente do valor absoluto deste lucro. Resultados positivos nesta métrica são todos os resultados acima de 50%, ou seja, quando é mais provável que o agente tenha lucro do que prejuízo ao abrir uma nova posição, com base em sua performance durante o teste.

- **(Resultado médio de uma negociação com lucro / Resultado médio de uma negociação com prejuízo) * 100**

  O resultado médio de uma negociação com lucro se refere a média aritmética de todos ganhos em operações que resultaram individualmente em algum lucro, enquanto que o resultado médio de uma negociação com prejuízo se refere a média aritmética de todas as perdas em operações que resultaram individualmente em algum prejuízo. O objetivo desta métrica é tentar entender se quando o agente ganha, ele ganha em média mais do que ele perde quando ele perde. Resultados positivos nesta métrica são todos os resultados acima de 100%, ou seja, quando o lucro médio individual é mais do que o prejuízo médio individual.

O **Quadro 1** descreve os intervalos de possíveis valores onde em cada uma das métricas é considerado um resultado positivo, neutro ou negativo. O **Quadro 2** registra os resultados obtidos pelo agente aleatório no teste executado neste experimento.

Quadro 1: Métricas comparativas.

MÉTRICAS	NEGATIVO	NEUTRO	POSITIVO
(Lucro total / Prejuízo total) * 100	[0%, 100%)	100%	(100%, ∞)
(Quantidade de negociações com lucro / Quantidade total de negociações) * 100	[0%, 50%)	50%	(50%, 100%]
(Resultado médio de uma negociação com lucro / Resultado médio de uma negociação com prejuízo) * 100	[0%, 100%)	100%	(100%, ∞)

Quadro 2: Resultados do agente aleatório.

MÉTRICAS	AGENTE ALEATÓRIO
(Lucro total / Prejuízo total) * 100	0,18%
(Quantidade de negociações com lucro / Quantidade total de negociações) * 100	0,32 %
(Resultado médio de uma negociação com lucro / Resultado médio de uma negociação com prejuízo) * 100	16,26%

A partir deste primeiro experimento, foi possível estipular uma base comparativa, assim como um objetivo para a próxima iteração. O resultado mínimo esperado para que um próximo experimento seja considerado superior ao agente aleatório, é que ele seja capaz de reduzir significativamente a quantidade de tentativas necessárias para percorrer toda base de teste, de forma que suas métricas comparativas sejam superiores às do agente aleatório.

6.2 EXPERIMENTO 2 - PRIMEIRO AGENTE TREINADO (REDE FC)

Antes de descrever o experimento em si, serão formalizadas algumas definições importantes relacionadas ao treinamento de um agente que serão utilizadas durante todos os experimentos a partir deste momento:

- **Segmento**: A base de dados que contém os preços históricos do ativo negociado foi dividida em 6 segmentos contínuos. Esses segmentos, por sua vez, foram divididos de forma que os 5 primeiros serão utilizados para o treinamento do agente, e o último será utilizado para testes.
- **Episódio**: Descreve o ciclo de vida útil de um agente. Um episódio termina quando um agente perde todo seu dinheiro, ou quando o segmento da base de dados no qual o agente está observando chega ao fim. O primeiro episódio do treinamento terá início no primeiro segmento da base de dados. Caso o agente que o está percorrendo perca todo seu dinheiro em algum momento do segmento, um novo episódio terá início na parte do segmento onde ele parou. Isso significa que um segmento será percorrido por no mínimo um episódio, no caso do agente manter seu dinheiro em caixa positivo do início ao fim do segmento. Os pesos da rede neural são mantidos ao longo de todo o

treinamento, de forma que a cada novo episódio o agente irá dar continuidade ao aprendizado adquirido até o momento. Caso o agente chegue ao final de um segmento, mesmo que seu patrimônio não tenha sido completamente reduzido, o episódio será finalizado e um novo será iniciado a partir do início do próximo segmento.

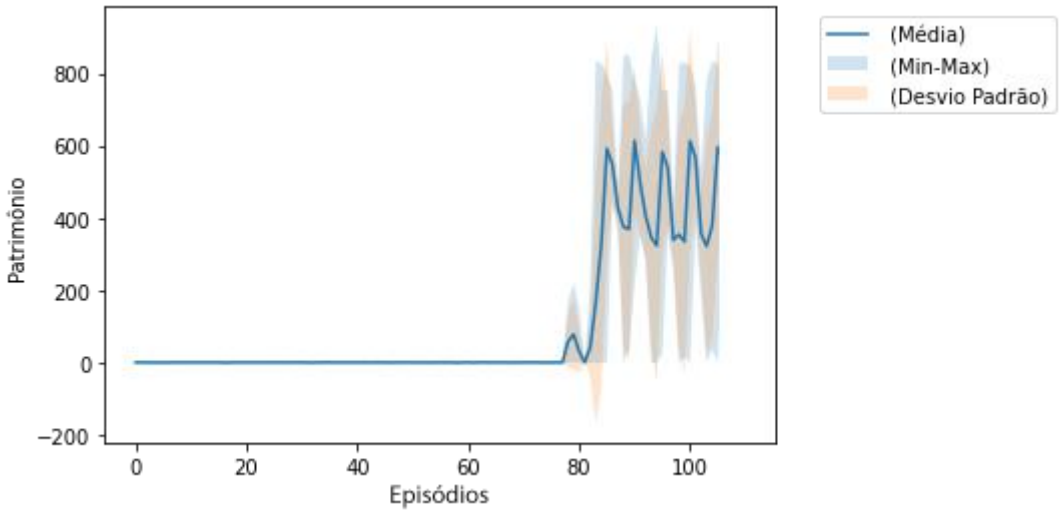
- **Época**: Descreve um ciclo dentro do treinamento onde toda a base de dados de treino, ou seja, todos os 5 segmentos, foram percorridos. Uma época tem no mínimo 5 episódios, no caso de todos os segmentos serem percorridos do início ao fim, cada um por uma instância diferente do agente. Inicialmente, foi decidido que um treinamento seria composto por 10 épocas.
- **Treinamento**: Descreve o conjunto de 10 épocas que são percorridas com o objetivo de ajustar os pesos da rede neural do agente. Um treinamento por si só já seria o suficiente para pôr à prova o agente treinado e medir seus resultados no segmento de teste. No entanto, como existem muitos fatores aleatórios dentro de um treinamento, é interessante que múltiplos treinamentos diferentes sejam realizados, para proporcionar uma confiança maior de que os resultados obtidos são consistentes e não apenas uma anomalia. Foi decidido que serão realizados 5 diferentes treinamentos, com as mesmas propriedades, a cada execução de um experimento completo. A cada novo treinamento, os pesos das redes são novamente inicializados, de forma que cada treinamento seja completamente isolado do anterior. Vale lembrar que outras técnicas poderiam ser utilizadas para trazer uma maior confiança nos resultados, por exemplo, a técnica de validação cruzada K-fold que, em especial, não foi utilizada por exigir que a base de dados seja subdividida em segmentos de tamanhos iguais, o que resultaria em uma perda substancial de dados, ou então em uma diminuição substancial no tamanho de um segmento.

Todos os gráficos apresentados para descrever alguma métrica do treinamento estarão representando todos os diferentes treinamentos realizados no experimento através da média do resultado entre os treinamentos e seus respectivos desvio padrão e valores de mínimo e máximo. No entanto, para facilitar o entendimento da análise dos gráficos, os resultados serão descritos no singular, de forma que ao descrever o comportamento de um treinamento, na realidade a descrição se refere a média de todos os treinamentos realizados no experimento.

6.2.1 Treinamento

A **Figura 17** representa o gráfico do patrimônio ao final de um episódio ao longo do treinamento. É possível perceber que no início do treinamento o patrimônio final é sempre R$0,00, e que chegando perto do fim o patrimônio passa a ser superior a R$0,00, seguindo um padrão de oscilação. Tal padrão pode ser assimilado ao fato de que, neste momento do treinamento, o agente está sendo capaz de sobreviver até o final dos segmentos, e como cada segmento possui um tamanho diferente, seu patrimônio ao final de cada um também terá um valor expressivamente diferente. Portanto, o período que reflete cada oscilação desse padrão descreve uma época do treinamento onde os agentes sobrevivem até o fim de cada um de seus segmentos.

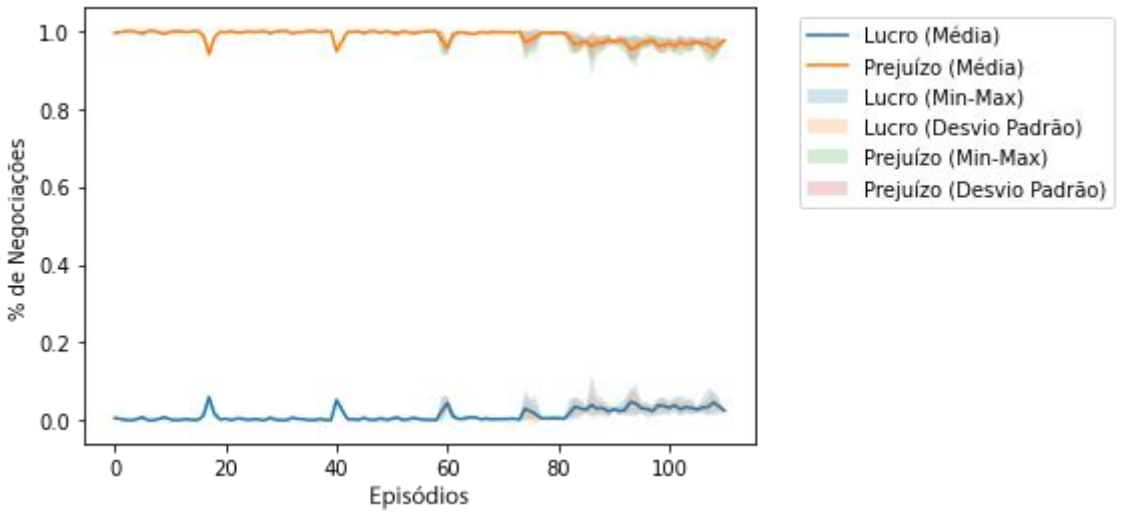
Figura 17: Gráfico do patrimônio final (R$) x episódio.



A **Figura 18** representa a porcentagem das negociações com lucro e com prejuízo por episódio realizadas ao longo do treinamento. Uma negociação é definida aqui como todo o período em que o agente passa a estar posicionado, seja essa posição comprada ou vendida, até o momento em que ele encerra tal posição. A negociação será considerada lucrativa se o patrimônio do agente ao encerrar a negociação/posição for maior do que o patrimônio no momento em que a negociação/posição foi aberta. Para facilitar a compreensão, é descrito um exemplo abaixo:

- O agente inicia o episódio sem estar posicionado, ou seja, não está comprado nem vendido em nenhuma quantidade de Bitcoins.
- Em um determinado instante, ele realiza a ação de comprar 3 Bitcoins. Com isso, ele inicia uma negociação se posicionando de forma a estar comprado em 3 Bitcoins.
- Em um outro instante, ele realiza a ação de vender 2 Bitcoins, passando a possuir 1 Bitcoin e ainda se mantendo na mesma negociação posicionado de forma a estar comprado em 1 Bitcoin.
- Em um outro instante, ele realiza a ação de vender 3 Bitcoins. Neste momento, sua ação é fragmentada em duas partes:
    - A primeira para vender 1 Bitcoin e encerrar sua posição de compra, finalizando uma negociação. Neste momento, seu patrimônio atual é comparado com o patrimônio no momento em que abriu a posição de compra, para determinar se a negociação finalizada foi lucrativa ou não.
    - A segunda para vender 2 Bitcoins e abrir uma nova posição, agora de venda, assim como uma nova negociação.

Figura 18: Gráfico da % de negociações com lucro e prejuízo x episódio.

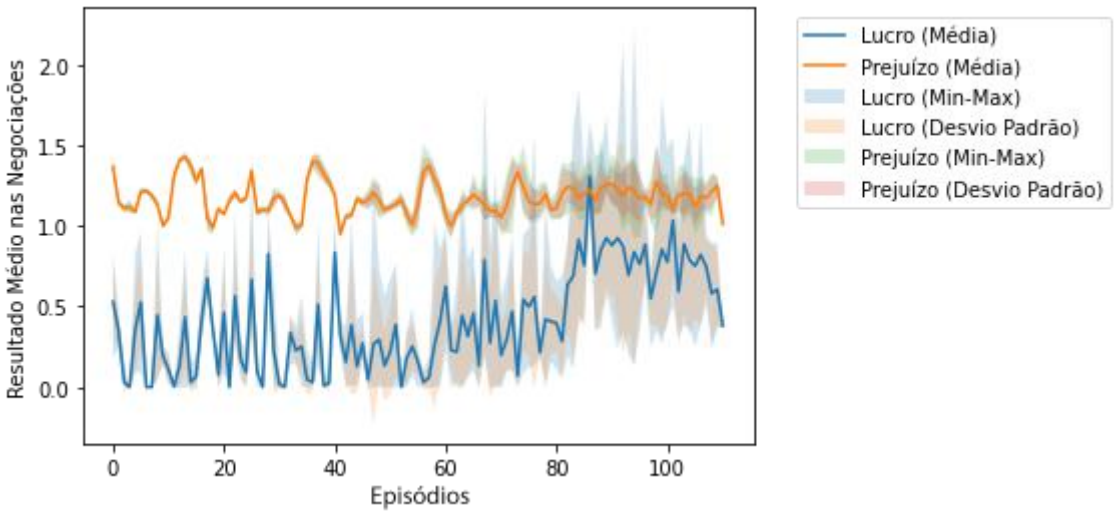


Como toda negociação é categorizada como lucro ou prejuízo, as linhas serão sempre complementares, totalizando 100%. O objetivo é que em algum momento do treinamento as linhas se cruzem, de forma que o agente realize uma porcentagem maior de negociações com lucro nos episódios mais avançados. É possível perceber que, no treinamento atual, não existe

uma evolução significativa na capacidade do agente de aumentar a quantidade de negociações com lucro. A variação ao longo do treinamento é tão pequena que pode ser desprezada.

Da mesma forma, se analisarmos a **Figura 19**, que representa o resultado médio por episódio, em reais, das negociações com lucro e prejuízo ao longo do treinamento, é possível perceber que existe apenas uma maior estabilidade no prejuízo médio e um leve aumento no lucro médio. Ou seja, a cada nova negociação com prejuízo, a quantidade de dinheiro perdido se tornou cada vez mais estável, e a cada nova negociação com lucro, a quantidade de dinheiro ganho aumentou levemente.

Figura 19: Gráfico do resultado médio (R$) de negociações x episódio.



A **Figura 20** representa o resultado total por episódio, em reais, das negociações com lucro e prejuízo de uma simulação ao longo de um treinamento. A partir deste gráfico é possível perceber que o lucro total do agente é praticamente 0 ao longo de todo o treinamento, mas que ocorre uma queda significativa do prejuízo total chegando perto do fim. Isso se deve ao fato de que o agente passa a realizar menos negociações ao final do treinamento, como evidenciado na **Figura 21**, que representa o número de negociações por episódio ao longo do treinamento, assim como o fato das negociações com lucro apresentarem um leve aumento em seu resultado médio, como evidenciado na **Figura 19**.

Figura 20: Gráfico do resultado total (R$) de negociações x episódio.

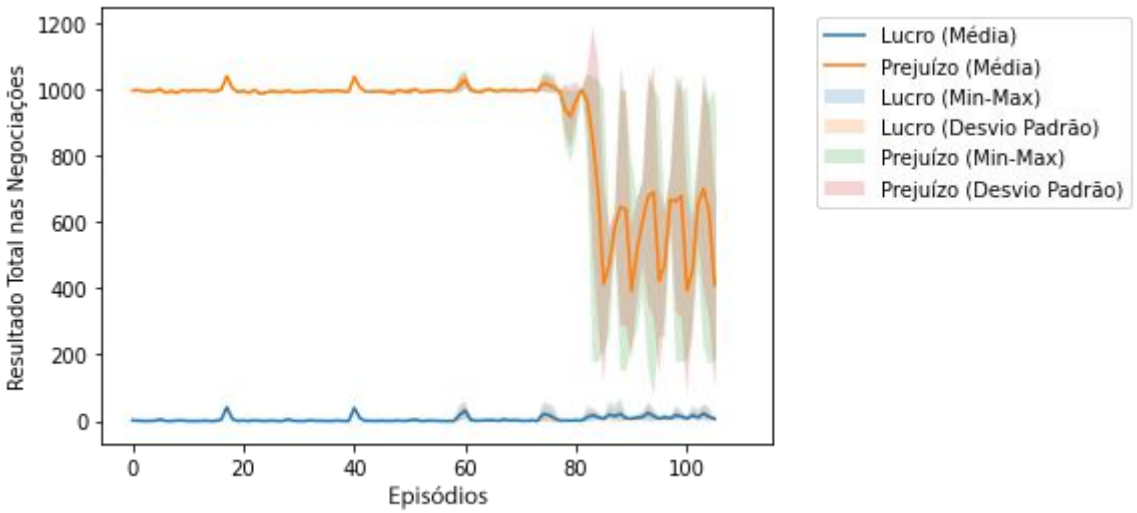
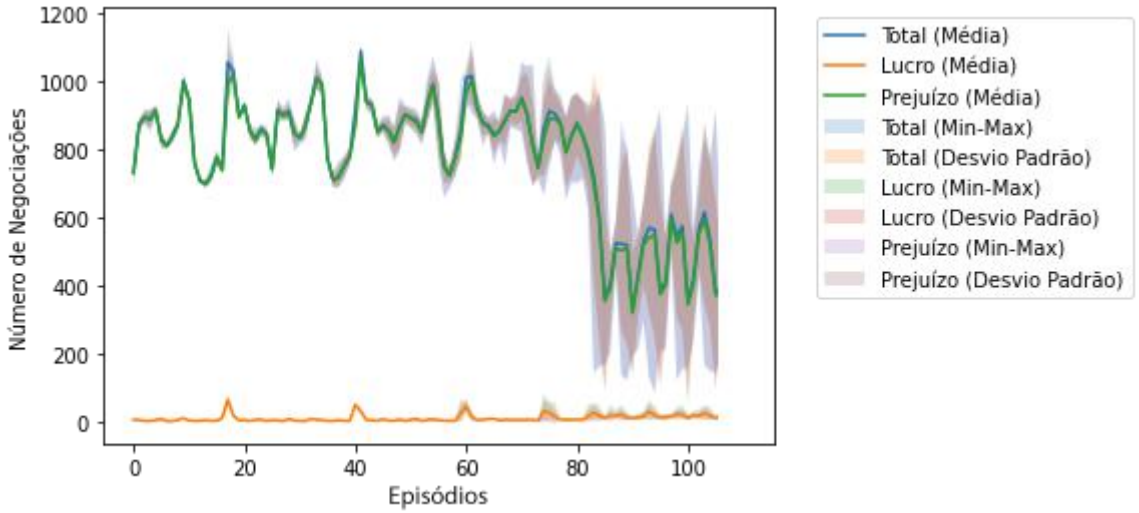


Figura 21: Gráfico do número de negociações x episódio.



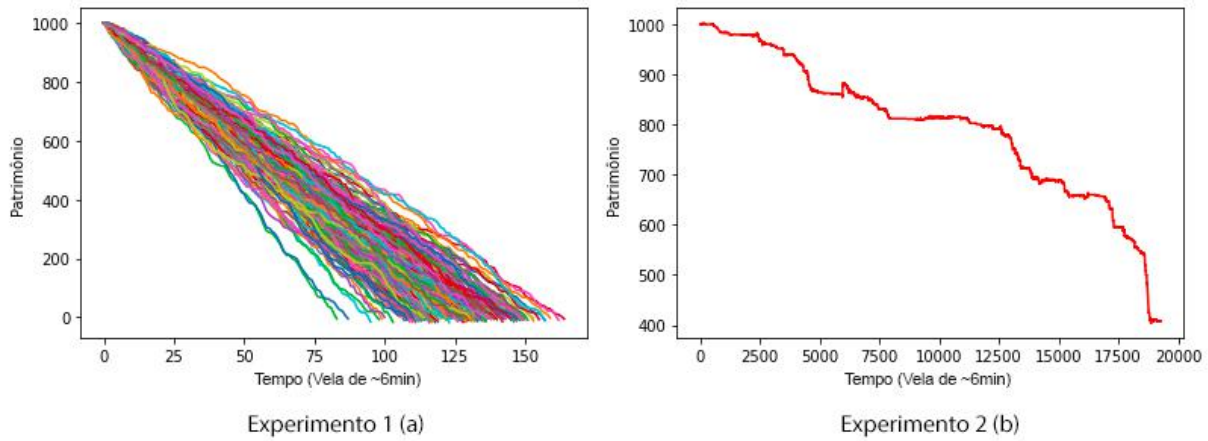
6.2.2 Teste

Ao final dos 5 treinamentos, aquele que obteve a maior média de patrimônio final ao longo dos episódios foi o escolhido para ter os pesos da rede neural correspondente salvos e utilizados na base de dados de teste, para avaliar seu resultado.

A **Figura 22.a** contém uma cópia da **Figura 15**, que representa a evolução do patrimônio do agente aleatório, em cada episódio, ao longo do segmento de teste. A **Figura 22.b** representa a evolução do patrimônio do agente treinado ao longo do segmento de teste. É

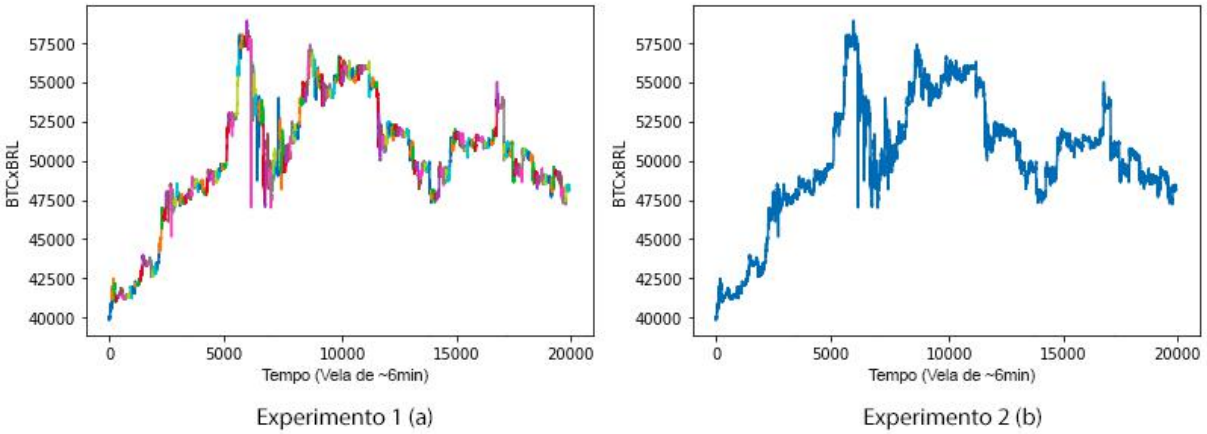
possível perceber que, assim como no final do treinamento, esse agente sobrevive até o final do segmento, com cerca de R$400, em contraste com o agente aleatório, que zera seu patrimônio diversas vezes até o fim do segmento.

Figura 22: Gráfico de patrimônio x tempo (número de velas) (agente aleatório VS agente treinado).



A partir de agora, todas as figuras que representam informações já apresentadas anteriormente, serão representadas lado a lado com a figura equivalente do experimento anterior, para facilitar a comparação dos resultados. Assim, a próxima figura representa a base de dados de teste dividida de forma que cada parte colorida representa o quanto o agente percorreu antes de reduzir seu patrimônio a zero, com um gráfico para o agente aleatório do experimento anterior na **Figura 23.a**, e um para o agente treinado do experimento atual na **Figura 23.b**. No caso, o agente treinado só possui uma cor, pois ele sobrevive até o final do segmento, em contraste com o agente aleatório, que não sobrevive por muito tempo.

Figura 23: Base de dados de teste dividida em episódios (agente aleatório VS treinado).



O **Quadro 3** representa as métricas comparativas que estamos utilizando neste momento em ambos os experimentos realizados. É possível perceber um aumento considerável em todas as métricas, como esperado, devido ao baixo desempenho do agente aleatório. Isso garantiu ao agente treinado a sobrevivência ao longo do segmento de teste, mas ainda não foi o suficiente para gerar lucros reais.

Quadro 3: Resultados do agente treinado.

MÉTRICAS	AGENTE ALEATÓRIO	AGENTE TREINADO
(Lucro total / Prejuízo total) * 100	0,18%	8,15%
Porcentagem de negociações com lucro	0,32 %	4,21 %
(Resultado médio de uma negociação com lucro / Resultado médio de uma negociação com prejuízo) * 100	16,26%	185,26%

6.3 EXPERIMENTO 3 - AGENTE LSTM

Neste novo experimento, foram realizadas algumas modificações na modelagem do sistema. A primeira diz respeito à observação do agente. Atualmente, ele apenas observa dados relacionados ao preço do ativo, sem ter acesso a informações do estado de suas próprias negociações, o que pode dificultar sua tomada de decisão. Com isso, foram adicionadas duas informações no estado observado:

- **Posição atual**: Em quantas Bitcoins o agente está comprado (representado por números positivos), vendido (representado por números negativos) ou simplesmente não posicionado (representado pelo número 0).
- **Resultado acumulado da negociação atual**: A ideia é fornecer um dado ao agente que o informe se sua posição atual está dando lucro ou prejuízo. Quando o agente não se encontra posicionado, este valor é zerado.

Além disso, foi realizada uma modificação na recompensa dada ao agente pelo ambiente. Atualmente, a recompensa leva em consideração a evolução do patrimônio de instante a instante, como pode ser observado em sua fórmula (2). Ele foi modificada de forma

a considerar a evolução do patrimônio desde o início da negociação atual em que o agente se encontra, de acordo com a fórmula abaixo:

$$r = (p_{t+1} \div p_{tina}) - 1 \qquad (3)$$

onde p_{t+1} representa o patrimônio do agente no instante seguinte à realização da ação que gerou tal recompensa, e p_{tina} representa o patrimônio do agente no início da negociação atual.

A ideia por trás dessas modificações é a de criar um ambiente mais parecido com a realidade, onde um operador não analisa a evolução de seu patrimônio de forma isolada de instante a instante, mas sim dentro do contexto de seu posicionamento atual.

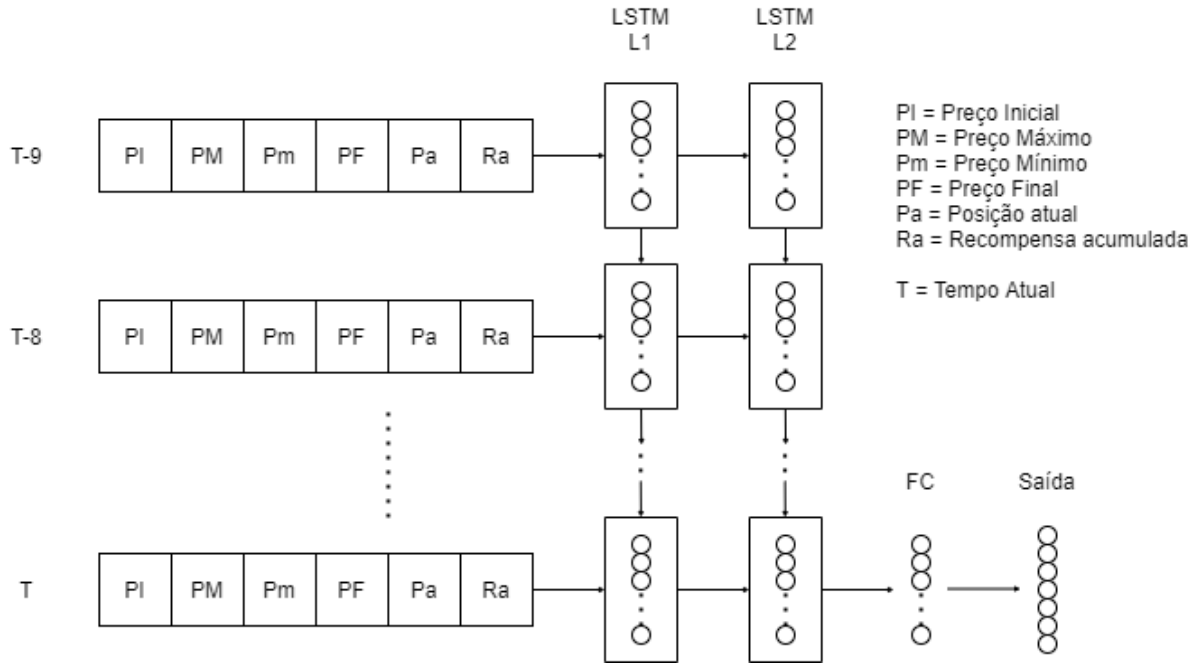
Por fim, também ocorreu uma substituição da rede neural do agente, atualmente uma rede FC (fully connected, ou completamente conectada), por uma nova rede do tipo LSTM (long short-term memory, ou memória de curto prazo longo). Uma rede LSTM é um tipo específico de RNN (Recurrent Neural Network, ou rede neural recorrente) que, ao contrário de redes FC que processam apenas um dado isolado de cada vez (como uma imagem, por exemplo), são capazes de processar sequências de dados (como um vídeo, por exemplo), o que a torna ideal para se utilizar em conjunto com uma sequência de dados temporais.

A ideia por trás da escolha dessa nova modelagem é o fato de que nossa abordagem atual, que consiste em utilizar apenas a observação mais recente do agente como entrada na rede, não costuma ser o suficiente para que alguém possa tomar uma decisão embasada sobre a flutuação futura do preço de um ativo. Seria interessante ter algum tipo de contexto histórico recente da operação. Através de uma rede LSTM será possível alimentar a rede com uma sequência de dados que contenha as últimas n observações, de forma a tentar trazer uma noção de passagem de tempo, tentando melhor embasar a tomada de decisão do agente.

A modelagem inicial escolhida para a rede LSTM utilizada pode ser observada pela representação gráfica da **Figura 24**, e possui a seguinte estrutura:

- Uma camada de entrada que contém uma sequência das 10 últimas observações do agente.
- Duas camadas LSTM onde cada célula possui internamente 128 unidades ocultas.
- A saída da última célula da segunda camada LSTM é utilizada como entrada em uma camada FC de 128 neurônios.
- Uma camada de saída contendo 7 neurônios relativos às possíveis ações a serem realizadas pelo agente, definidas pela modelagem do ambiente.

Figura 24: Representação da modelagem da rede neural LSTM.

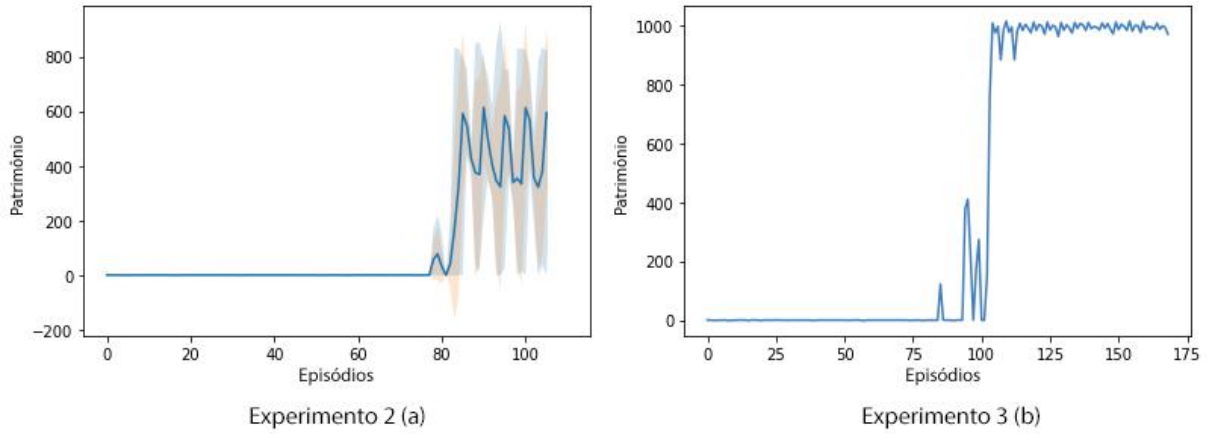


6.3.1 Treinamento

A **Figura 25.b** representa o gráfico do patrimônio ao final de um episódio ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 25.a**. É possível perceber que no início do treinamento o patrimônio final é sempre R$0,00, e que chegando perto do fim passa a ser aproximadamente R$1.000,00, o que corresponde ao patrimônio inicial do agente. Isso significa que o agente está sendo capaz de sobreviver até o fim de um segmento sem realizar prejuízos ou lucros significativos.

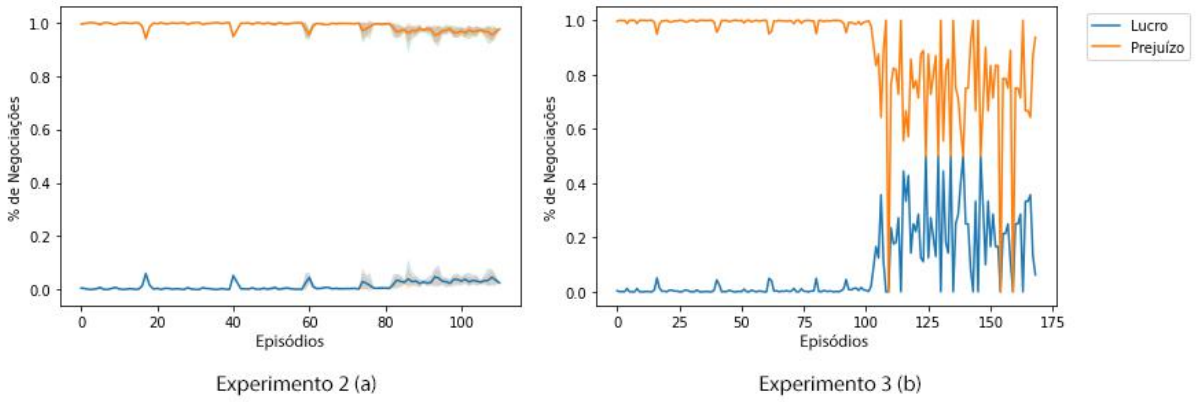
De forma a deixar os resultados mais claros, todos os gráficos deste experimento foram representados apenas com a linha da média dos treinamentos realizados, deixando de lado os preenchimentos de desvio padrão e valores máximos e mínimos. Por tanto, ao comparar um gráfico do experimento anterior com o do experimento atual, desconsidere tais preenchimentos.

Figura 25: Gráfico do patrimônio final (R$) x episódio.



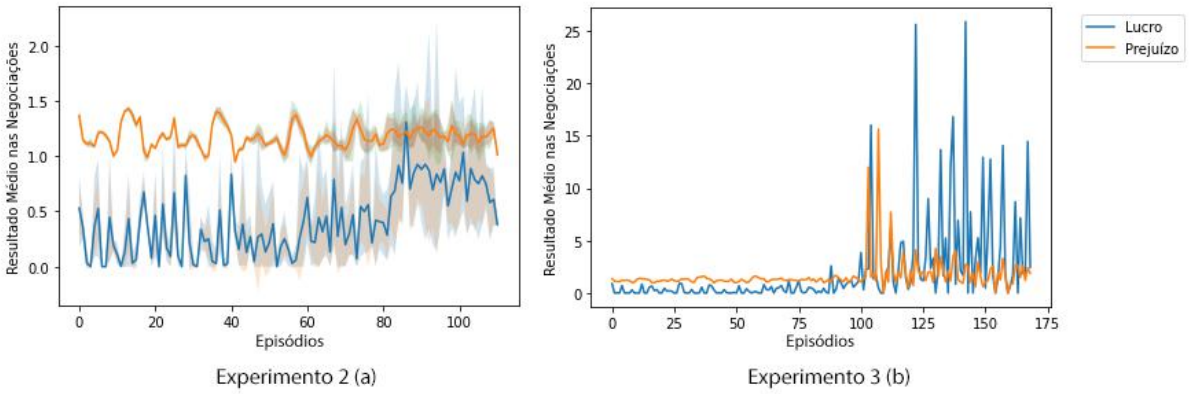
A **Figura 26.b** representa a porcentagem das negociações com lucro e com prejuízo por episódio realizadas ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 26.a**. É possível perceber uma variação muito maior do que no experimento anterior, no entanto, os valores são muito instáveis ao longo de todo o treinamento.

Figura 26: Gráfico da % de negociações com lucro e prejuízo x episódio.



A **Figura 27.b** representa o resultado médio por episódio, em reais, das negociações com lucro e prejuízo ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 27.a**. É possível perceber que no início do treinamento o prejuízo se mantinha maior do que o lucro, mas que ao final do treinamento ocorre uma inversão expressiva, mas que também é acompanhada por uma grande instabilidade.

Figura 27: Gráfico do resultado médio (R$) de negociações x episódio.



A **Figura 28.b** representa o resultado total por episódio, em reais, das negociações com lucro e prejuízo de uma simulação ao longo de um treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 28.a**. É possível perceber que o lucro total do agente é praticamente R$0,00 ao longo de todo o treinamento, e que o prejuízo também é reduzido a R$0,00 chegando perto do fim. Isso se deve principalmente pelo fato de que o agente passa a realizar um número ínfimo de negociações ao final do treinamento, como evidenciado na **Figura 29.b**, que representa o número de negociações por episódio ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 29.a**.

Figura 28: Gráfico do resultado total (R$) de negociações x episódio.

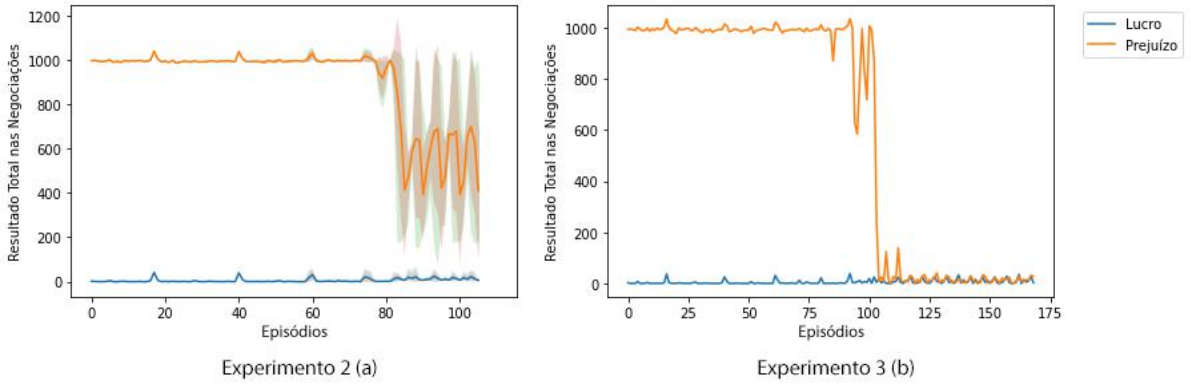
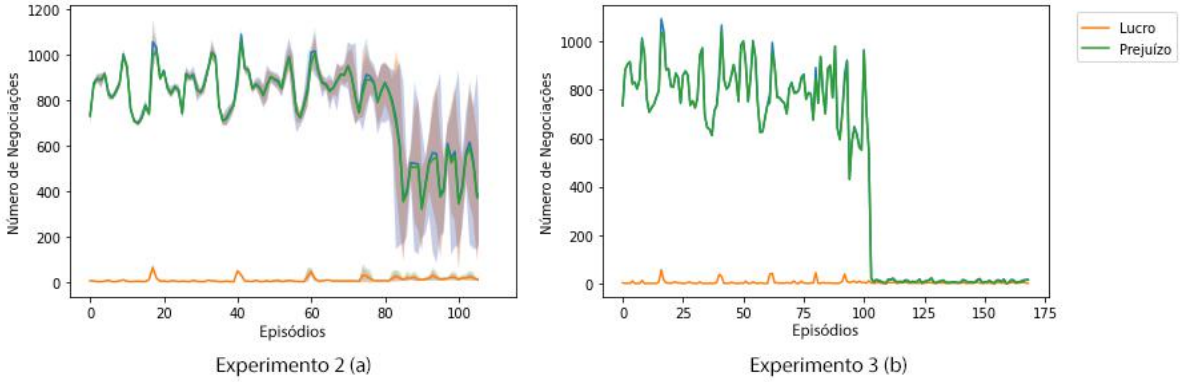


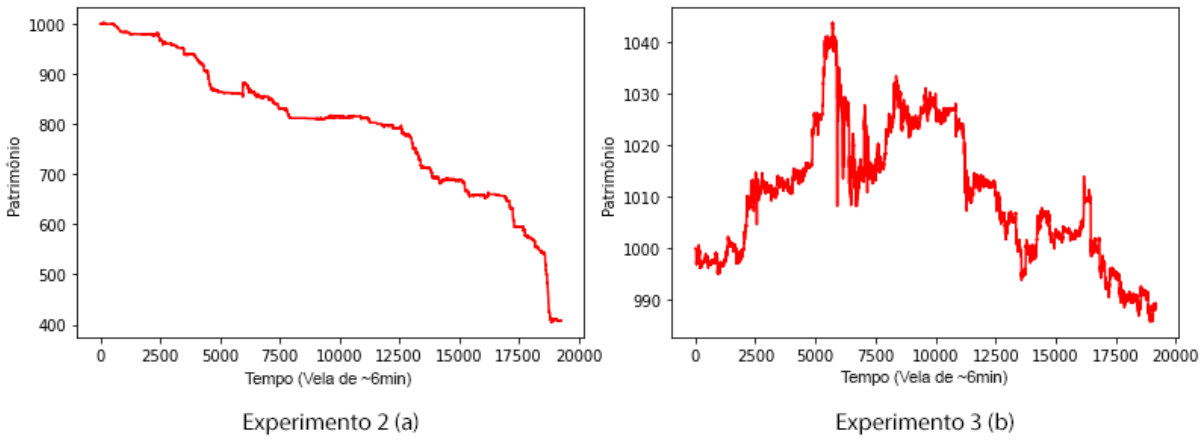
Figura 29: Gráfico do número de negociações x episódio.



6.3.2 Teste

A **Figura 30.b** representa a evolução do patrimônio do agente treinado ao longo do segmento de teste, lado a lado com o gráfico correspondente do experimento anterior na **Figura 30.a**. É possível perceber que, embora o agente apresente algum lucro em certos momentos do segmento, seus resultados estão diretamente relacionados com a flutuação do preço do Bitcoin, de forma que seu patrimônio aumenta quando o preço também aumenta, e diminui quando o preço também diminui.

Figura 30: Gráfico de patrimônio x tempo (número de velas) (experimento 2 VS experimento 3).



O **Quadro 4** representa as métricas comparativas que estamos utilizando neste momento nos últimos dois experimentos realizados. É possível perceber um aumento considerável em todas as métricas, o que em um primeiro momento indicaria uma melhora na performance em relação ao experimento anterior. No entanto, ao analisar o comportamento do agente durante o teste, foi observado que ele realizou apenas 15 negociações, um número bem pequeno se comparado com os experimentos anteriores que realizaram em média 500 negociações. Tal comportamento torna o resultado do agente extremamente atrelado à flutuação específica do preço durante o segmento de teste, pois ele permanece a maior parte do tempo com o mesmo posicionamento, sendo assim bem mais influenciado pelas altas e baixas naturais do mercado. Desta forma, se o preço tivesse apresentado uma tendência de queda, o resultado do teste teria sido pior do que o do experimento anterior.

Surge então a necessidade de se obter uma base de dados mais extensa, de forma que o teste do agente se baseie em uma maior variedade de padrões e tendências de mercado. Alguns dos experimentos seguintes ainda serão realizados utilizando apenas este segmento de teste, devido a dificuldade em se obter tais dados. No entanto, no **Experimento 5**, já será possível realizar tal comparação.

Quadro 4: Resultados do agente treinado.

MÉTRICAS	EXPERIMENTO 2	EXPERIMENTO 3
(Lucro total / Prejuízo total) * 100	8,15%	84,12%
Porcentagem de negociações com lucro	4,21 %	20%
(Resultado médio de uma negociação com lucro / Resultado médio de uma negociação com prejuízo) * 100	185,26%	336,39%

6.4 EXPERIMENTO 4 - VELA IDEAL (REDE LSTM)

Atualmente, uma vela da base de dados utilizada pelo agente para ter acesso às informações relativas à cotação do Bitcoin abrange um período de 6 minutos. Isso significa que cada iteração da simulação equivale a 6 minutos em um ambiente real, e que as informações dos valores inicial, máximo, mínimo e final da cotação são relativas a esse período de 6 minutos que se passaram entre uma iteração e outra.

Embora esse período seja um valor perto do comumente utilizado por traders reais, ele exige uma estratégia cujo posicionamento precisa ser mantido por diversas velas para que se possa ter algum ganho real, não apenas por conta da baixa variação no preço em um período tão pequeno, mas também por causa da taxa de corretagem aplicada a cada transação.

Para tentar simplificar o problema através da eliminação da necessidade de estratégias de negociação de longo prazo, o experimento atual tem como objetivo encontrar o período ideal para as velas da base de dados de forma que o agente possa realizar ordens de compra e venda a toda iteração minimizando a penalidade relacionada a baixa variação do preço e da taxa de corretagem.

Para isso, foi desenvolvido um agente especial, capaz de ver as informações contidas na vela seguinte ao momento em que se encontra. Assim, ele será capaz de trapacear para que a toda iteração ele esteja posicionado de forma que:

- Se o preço do Bitcoin for maior na vela seguinte, ele irá se posicionar de forma a estar comprado na maior quantidade de Bitcoins permitida.
- Se o preço do Bitcoin for menor na vela seguinte, ele irá se posicionar de forma a estar vendido na maior quantidade de Bitcoins permitida.

Esse agente serve para nos ilustrar o melhor cenário possível para um determinado período de vela, onde nosso objetivo é que o agente treinado realize decisões sobre seu posicionamento a toda iteração, eliminando a necessidade de desenvolver estratégias de longo prazo, tentando assim simplificar o problema.

Foi realizado um teste com esse agente especial através da base de dados em diversos períodos de velas. O primeiro período a ser testado foi o de 24 horas, por fazer sentido em um contexto de trading real. A partir daí, foram realizados testes em alguns outros períodos acima e abaixo de 24 horas para tentar encontrar algum em que o agente se destacasse. Não foram realizados testes em períodos muito acima de 24h para evitar diminuir ainda mais a base de dados.

O **Quadro 5** ilustra os resultados dos testes realizados, onde a métrica utilizada para medir a efetividade do agente foi o patrimônio ao final do teste. O período de 21h foi o que obteve o melhor resultado, e foi assim escolhido para ser o novo período utilizado para treinar o agente deste experimento. Tal escolha reduziu a base de dados para cerca de 300 entradas, dividida entre dois segmentos de treinamento e um segmento de teste.

Quadro 5: Teste de vela ideal.

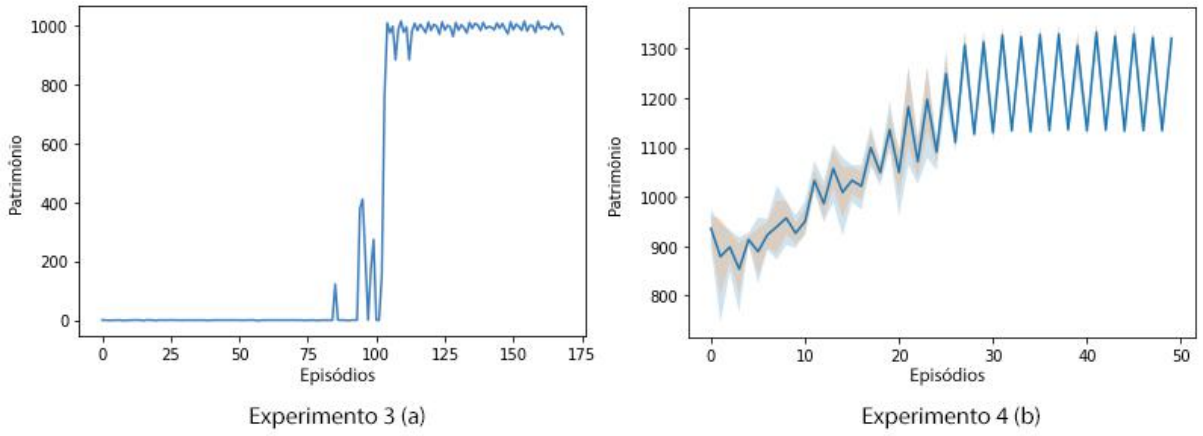
PERÍODO DO GRÁFICO DE VELAS	PATRIMÔNIO AO FINAL DO TESTE
0,5 dia - 12h	R$ 17.175,00
0,625 dia - 15h	R$ 17.917,00
0,75 dia - 18h	R$ 18.637,00
0,875 dia - 21h	**R$ 18.789,00**
1 dia - 24h	R$ 18.399,00
1,25 dias - 30h	R$ 17.827,00

1,5 dias - 36h	R$ 17.642,00
2 dias - 48h	R$ 18.017,00
3 dias - 72h	R$ 16.587,00
4 dias - 96h	R$ 15.786,00
5 dias - 120h	R$ 15.791,00

6.4.1 Treinamento

A **Figura 31.b** representa o gráfico do patrimônio ao final de um episódio ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 31.a**. É possível perceber um aumento gradual do patrimônio ao longo do treinamento, que alcança uma certa estabilidade a partir do trigésimo episódio. A oscilação do patrimônio é devido a diferença de tamanho entre os dois segmentos que agora compõem a base de dados reduzida.

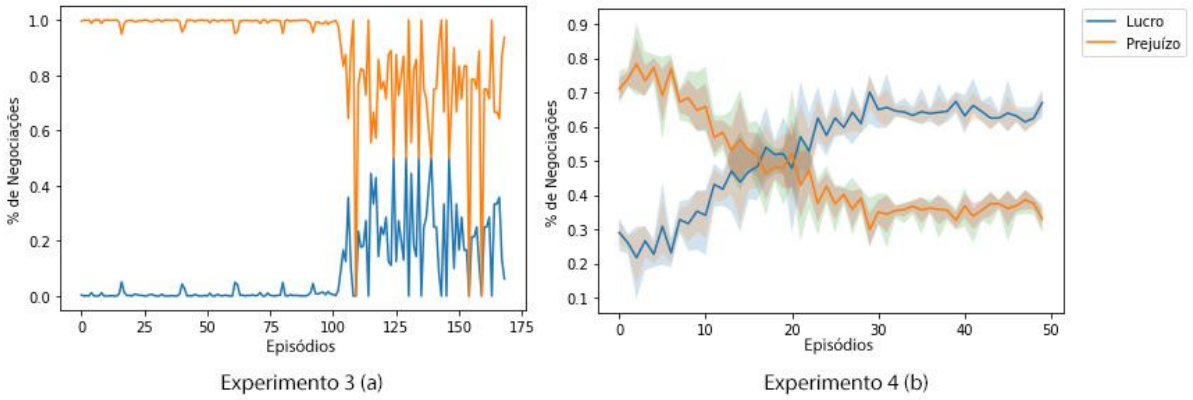
Figura 31: Gráfico do patrimônio final (R$) x episódio.



A **Figura 32.b** representa a porcentagem das negociações com lucro e com prejuízo por episódio realizadas ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 32.a**. Pela primeira vez é possível perceber um cruzamento claro entre as duas linhas do gráfico, indicando que ao longo do treinamento o agente foi

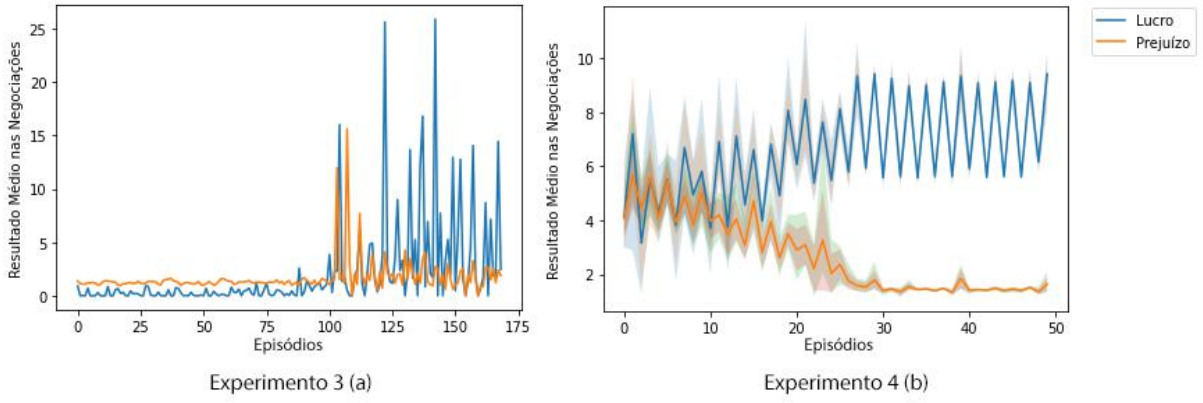
capaz de aprender a realizar uma quantidade maior de negociações com lucro do que com prejuízo.

Figura 32: Gráfico da % de negociações com lucro e prejuízo x episódio.



A **Figura 33.b** representa o resultado médio por episódio, em reais, das negociações com lucro e prejuízo ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 33.a**. Mais uma vez é possível observar uma melhora considerável em relação ao treinamento anterior, onde ao longo do treinamento o lucro médio cresce até estabilizar em um patamar maior do que o prejuízo.

Figura 33: Gráfico do resultado médio (R$) de negociações x episódio.



A **Figura 34.b** representa o resultado total por episódio, em reais, das negociações com lucro e prejuízo de uma simulação ao longo de um treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 34.a**. É possível perceber que, além de reduzir o seu prejuízo total como nos últimos experimentos, o agente é capaz de

aumentar seu lucro total. Ele faz isso sem comprometer a quantidade total de negociações que realiza, que se mantém estável ao longo do treinamento, como evidenciado na **Figura 35.b**, que representa o número de negociações por episódio ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 35.a**. Note que, nos experimentos anteriores a quantidade de negociações com lucro era tão desprezível, que não era possível distinguir o total de negociações (linha azul) do total de negociações com prejuízo (linha verde).

Figura 34: Gráfico do resultado total (R$) de negociações x episódio.

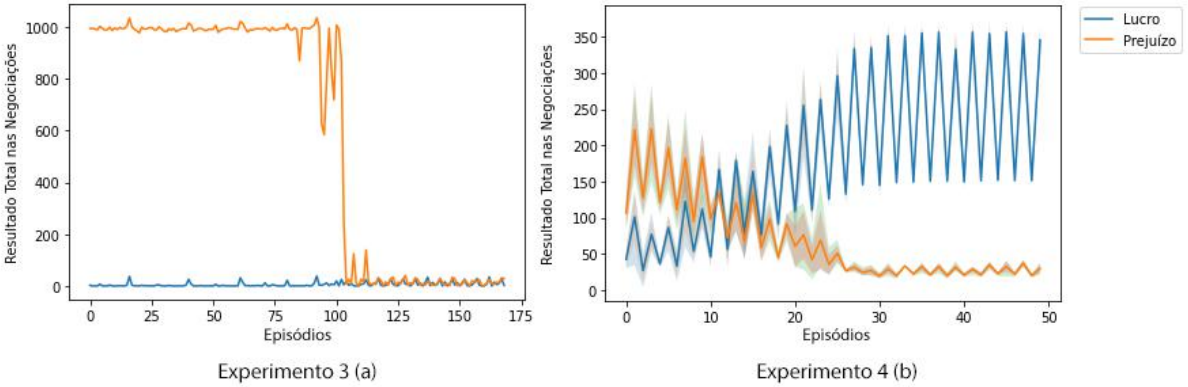
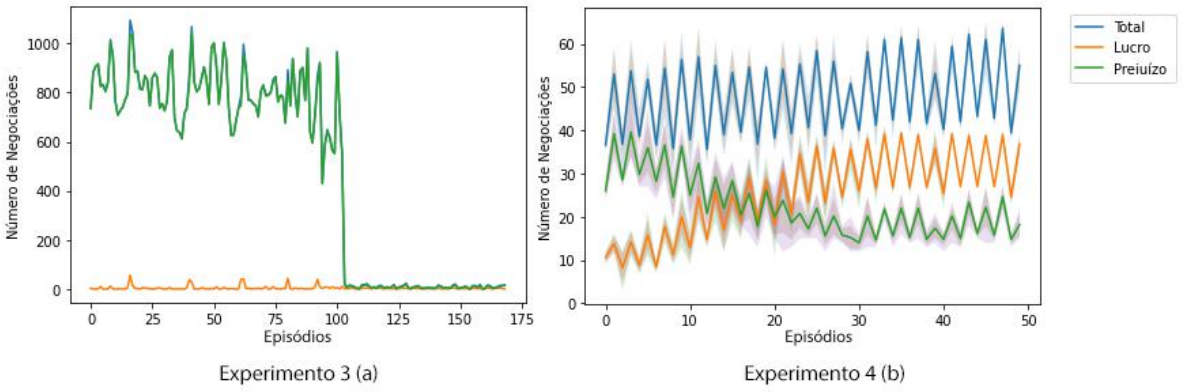


Figura 35: Gráfico do número de negociações x episódio.

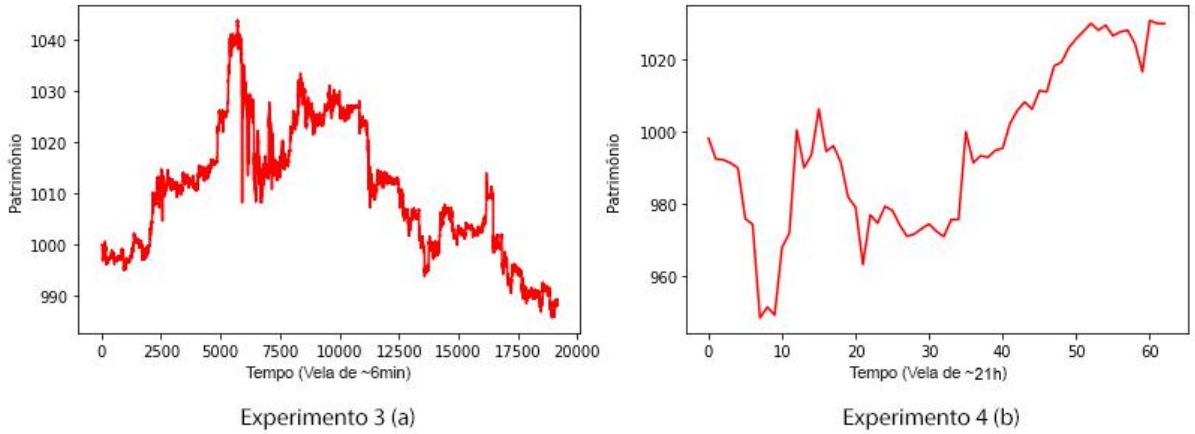


6.4.2 Teste

A **Figura 36.b** representa a evolução do patrimônio do agente treinado ao longo do segmento de teste, lado a lado com o gráfico correspondente do experimento anterior na **Figura 36.a**. É possível perceber que, pela primeira vez, o agente finalizou o segmento de teste com lucro. Não só isso, mas a variação de seu patrimônio não parece ter relação direta

com a flutuação do preço do Bitcoin, como no experimento anterior. No entanto, embora tenhamos um aparente progresso, a evolução do patrimônio ainda se encontra muito instável, e é evidente cada vez mais a necessidade de aumentar a base de dados de teste, para poder validar o agente de forma mais assertiva.

Figura 36: Gráfico de patrimônio x tempo (número de velas) (experimento 3 VS experimento 4).



O **Quadro 5** representa as métricas comparativas que estamos utilizando neste momento nos últimos dois experimentos realizados. É possível perceber que a primeira métrica ultrapassou a marca dos 100% evidenciando que o agente foi capaz de finalizar o teste com algum lucro real pela primeira vez. Além disso, a segunda métrica ultrapassou a marca de 50%, o que indica que o agente está realizando mais negociações com lucro do que com prejuízo, o que é ideal para aumentar sua confiabilidade. Houve uma queda na terceira métrica, indicando que o resultado médio das negociações com lucro e com prejuízo estão mais próximos, mas como ainda está acima de 100%, não é um fator muito preocupante.

Quadro 5: Resultados do agente treinado.

MÉTRICAS	EXPERIMENTO 3	EXPERIMENTO 4
(Lucro total / Prejuízo total) * 100	84,12%	315,31%
Porcentagem de negociações com lucro	20%	66%
(Resultado médio de uma negociação com lucro / Resultado médio de uma negociação com prejuízo) * 100	336,39%	157,66%

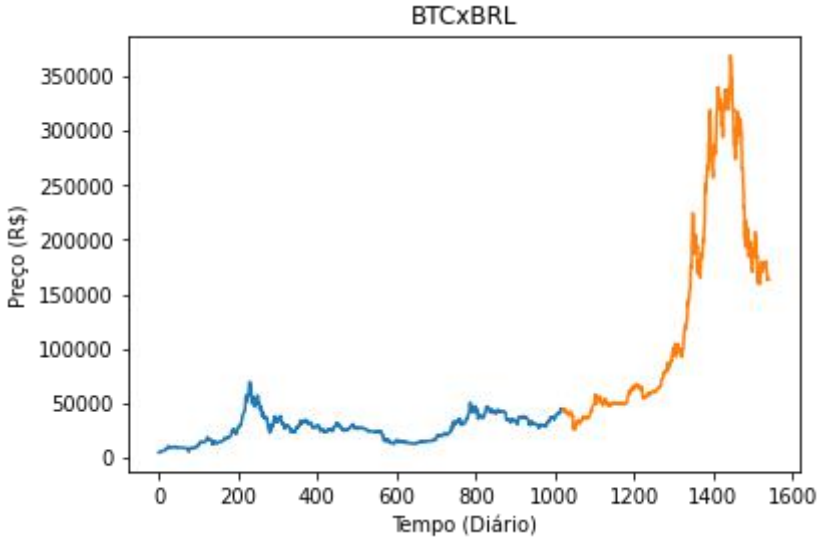
6.5 EXPERIMENTO 5 - NOVA BASE DE DADOS (REDE LSTM)

Ao realizar a modificação no período da vela para 21 horas, nossa base de dados foi reduzida drasticamente. Isso gera problemas tanto para o treinamento, que agora possui menos padrões de dados para tentar absorver, quanto para o teste, que possui um tempo menor de validação de seus resultados.

Originalmente a base de dados foi coletada de forma manual devido a necessidade da obtenção de dados com um intervalo na ordem de minutos. No entanto, agora que está sendo utilizada uma vela de 21 horas, tal restrição não é mais importante. Se o período da vela for novamente flexibilizado para um período de 24 horas, fica muito fácil encontrar bases de dados gratuitas pela internet para ser utilizada no experimento.

Este experimento tem como objetivo essencialmente a substituição da base de dados atual por uma nova base de dados, com período diário. Foram utilizados dados adquiridos gratuitamente através do site https://investing.com/ que abrangem informações diárias dos valores de preço inicial, máximo, mínimo e final de Bitcoin (em reais) desde 1 de maio de 2017 até 19 de julho de 2021. Quando aplicáveis, os dados adquiridos passaram pelos mesmos tratamentos que os da base de dados original. A **Figura 37** representa o gráfico do preço final diário ao longo de toda a base de dados, dividida entre segmentos de treinamento, com cerca de 1000 entradas, e de teste, com cerca de 600 entradas.

Figura 37: Nova base de dados segmentada para treinamento e teste.

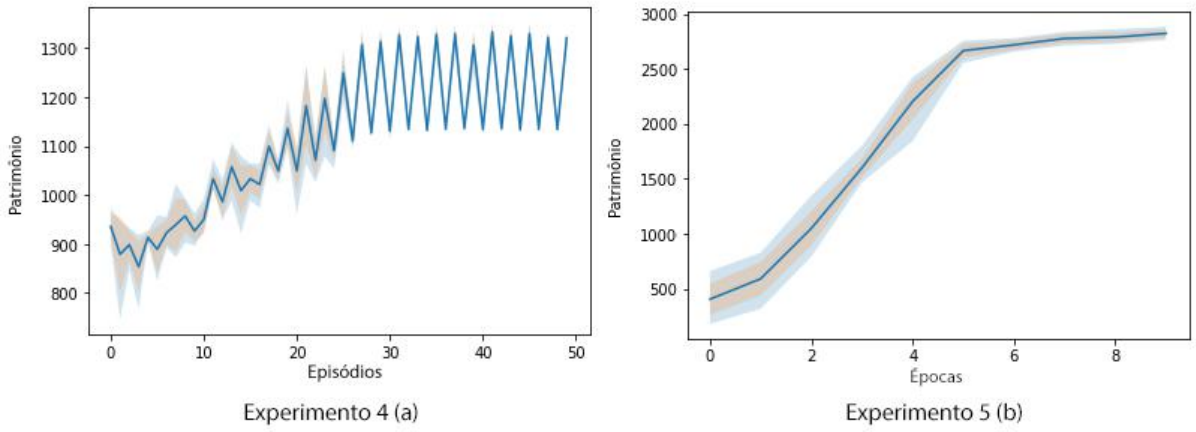


Uma outra mudança foi a redução do número de épocas em um treinamento de 25 para 10. A ideia por trás da mudança é a de manter o tempo de execução do treinamento em um intervalo aceitável de acordo com as restrições de uso do Google Colab, assim como a percepção de que, com uma base de dados maior, não serão necessárias tantas épocas quanto na base de dados original.

6.5.1 Treinamento

A **Figura 38.b** sofreu uma pequena mudança em relação às anteriores. Onde antes era representado o patrimônio ao final de um episódio ao longo do treinamento, agora está sendo representado a média dos patrimônios ao final dos episódios de uma época ao longo do treinamento. Essa mudança foi realizada com o objetivo de remover o aspecto serrilhado do gráfico, para facilitar a compreensão dos resultados. Tal mudança será aplicada em todos os gráficos a partir de agora. A **Figura 38.a** corresponde ao gráfico relacionado do experimento anterior. É possível perceber que houve um aumento no patamar de estabilização do patrimônio ao final do treinamento. Provavelmente está relacionado ao fato de que, como a base de dados é maior, o agente possui mais tempo para realizar mais negociações e, portanto, consegue alcançar valores maiores de patrimônio final.

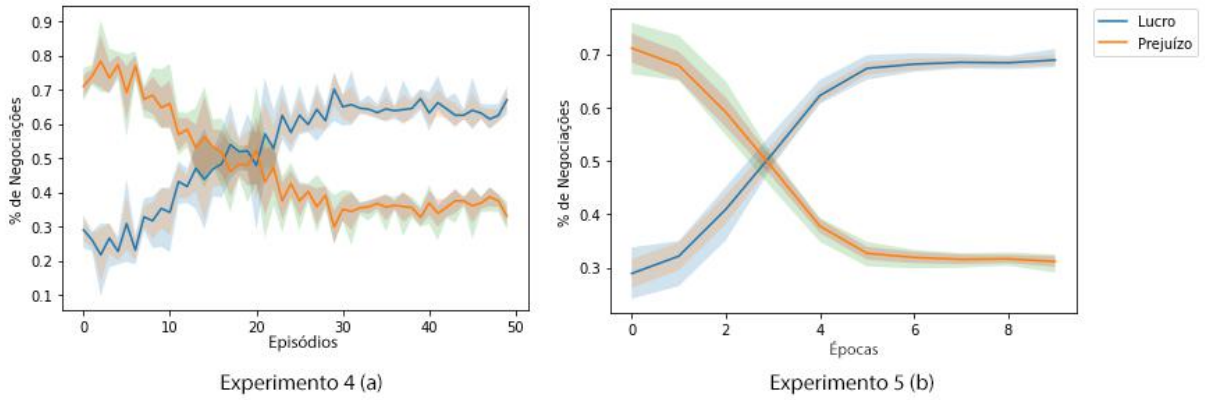
Figura 38: Gráfico do patrimônio final (R$) x episódio.



A **Figura 39.b** representa a porcentagem das negociações com lucro e com prejuízo por época realizadas ao longo do treinamento, lado a lado com o gráfico correspondente do

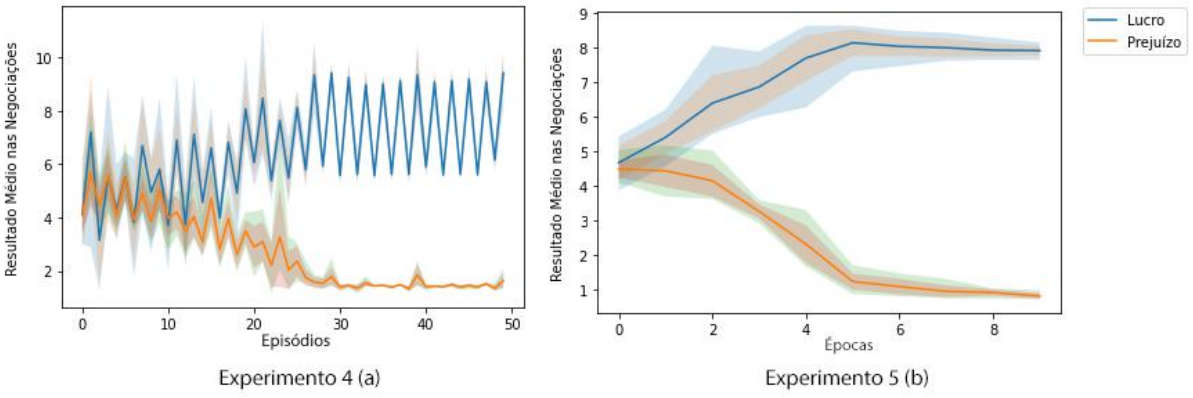
experimento anterior na **Figura 39.a**. Comparando os gráficos, não parece haver nenhuma mudança significativa. Ambos os treinamentos estabilizaram entre 60% e 70%.

Figura 39: Gráfico da % de negociações com lucro e prejuízo x episódio.



A **Figura 40.b** representa o resultado médio por época, em reais, das negociações com lucro e prejuízo ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 40.a**. Mais uma vez parece não haver muita diferença entres os treinamentos, onde ambos estabilizam em patamares semelhantes.

Figura 40: Gráfico do resultado médio (R$) de negociações x episódio.



A **Figura 41.b** representa o resultado total por época, em reais, das negociações com lucro e prejuízo de uma simulação ao longo de um treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 41.a**. Assim como no gráfico do patrimônio, houve um aumento substancial provocado pelo maior tempo de negociação. No entanto, a proporção da diferença entre as quantidades de negociações parece se manter

semelhante ao experimento anterior, como evidenciado na **Figura 42.b**, que representa o número de negociações por época ao longo do treinamento, lado a lado com o gráfico correspondente do experimento anterior na **Figura 42.a**.

Figura 41: Gráfico do resultado total (R$) de negociações x episódio.

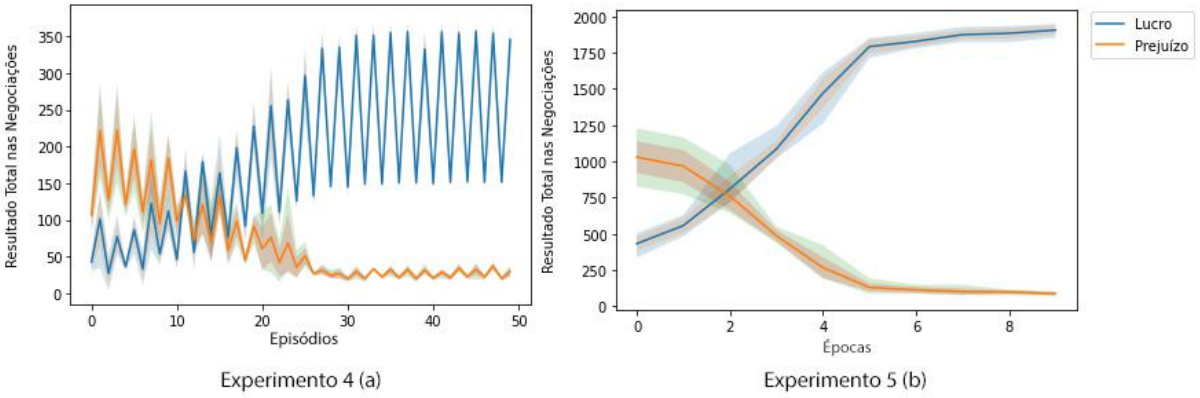
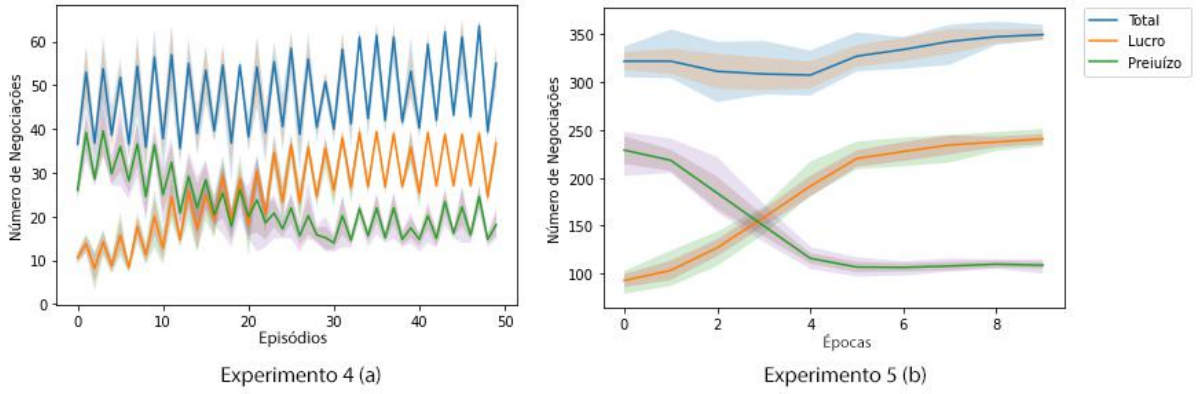


Figura 42: Gráfico do número de negociações x episódio.

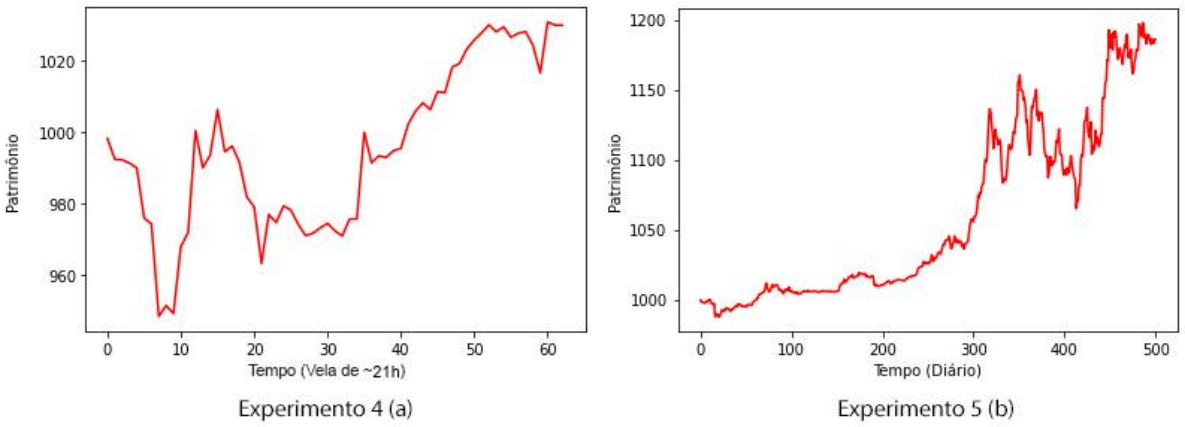


6.5.2 Teste

A **Figura 43.b** representa a evolução do patrimônio do agente treinado ao longo do segmento de teste, lado a lado com o gráfico correspondente do experimento anterior na **Figura 43.a**. O agente finalizou o teste com cerca de R$1200,00 em contraste com os R$1020,00 do experimento anterior. No entanto, é preciso levar em consideração o fato de que o segmento de teste é bem maior e a base de dados é diferente em relação ao experimento anterior, potencializando os possíveis ganhos do agente. Além disso, assim como no

experimento anterior, o agente ainda possui alguns pontos de instabilidade ao longo do teste, o que enfraquece a confiança em seus resultados.

Figura 43: Gráfico de patrimônio x tempo (número de velas) (experimento 4 VS experimento 5).



O **Quadro 6** representa as métricas comparativas que estamos utilizando neste momento nos últimos dois experimentos realizados.

Quadro 6: Resultados do agente treinado.

MÉTRICAS	EXPERIMENTO 4	EXPERIMENTO 5
(Lucro total / Prejuízo total) * 100	315,31%	1421,92%
Porcentagem de negociações com lucro	66%	80%
(Resultado médio de uma negociação com lucro / Resultado médio de uma negociação com prejuízo) * 100	157,66%	355,48%

6.5.3 Outros testes

Com a finalidade de entender melhor a efetividade do agente no mercado, foram realizados outros testes com segmentos de bases de dados de outros ativos financeiros, também adquiridos gratuitamente através do site https://investing.com/. Foram escolhidos arbitrariamente os seguintes ativos:

- Ações do Itaú (ITUB4), ações da Ambev (ABEV3), ações da B3 (B3SA3), ações da Petrobras (PETR4).

As **Figuras 44 a 48** representam os gráficos de cada um desses testes, assim como um para o Bitcoin, como base comparativa. Em cada figura, o gráfico azul representa o preço final diário do ativo em questão ao longo do segmento adquirido e o gráfico em vermelho representa o patrimônio agente ao longo deste segmento.

Figura 44: Teste final do agente - Bitcoin.

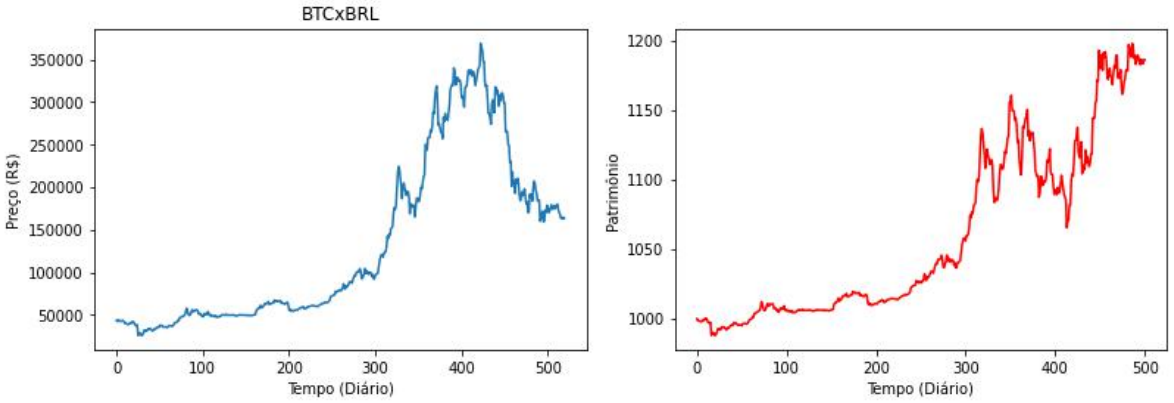


Figura 45: Teste final do agente - Itaú (ITUB4).

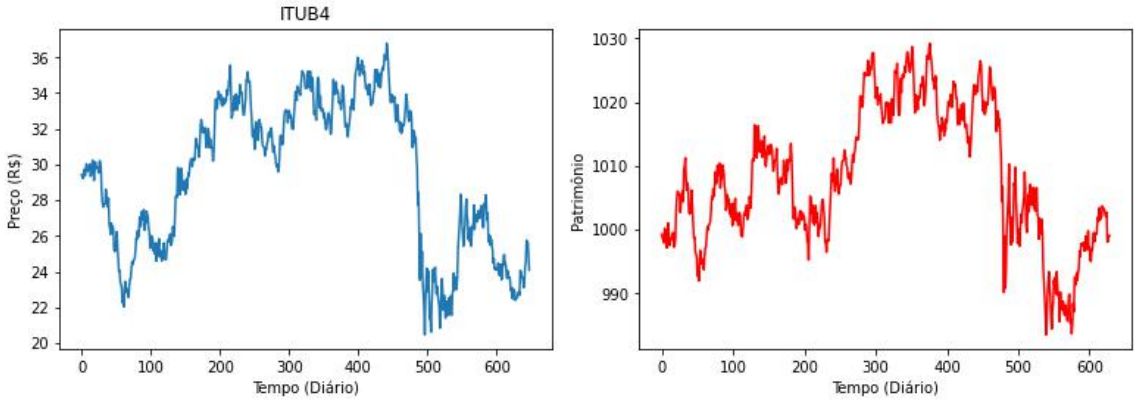


Figura 46: Teste final do agente - Ambev (ABEV3).

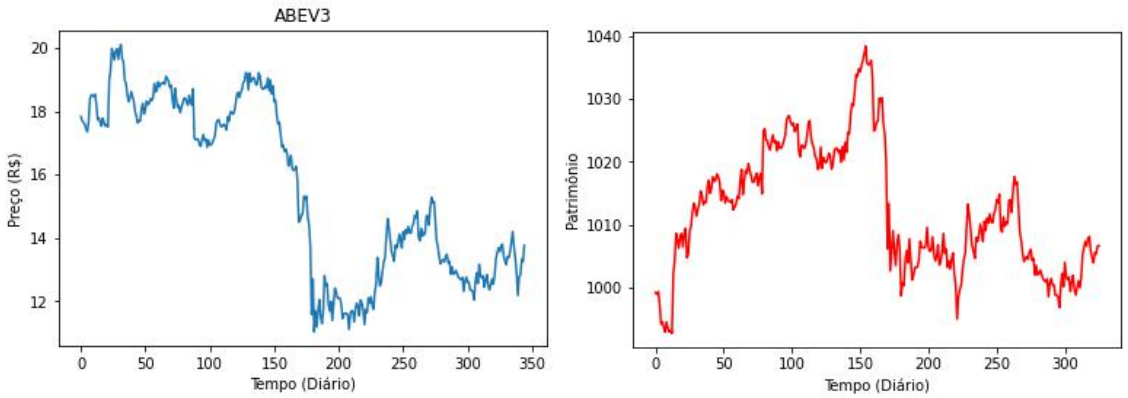


Figura 47: Teste final do agente - B3 (B3SA3).

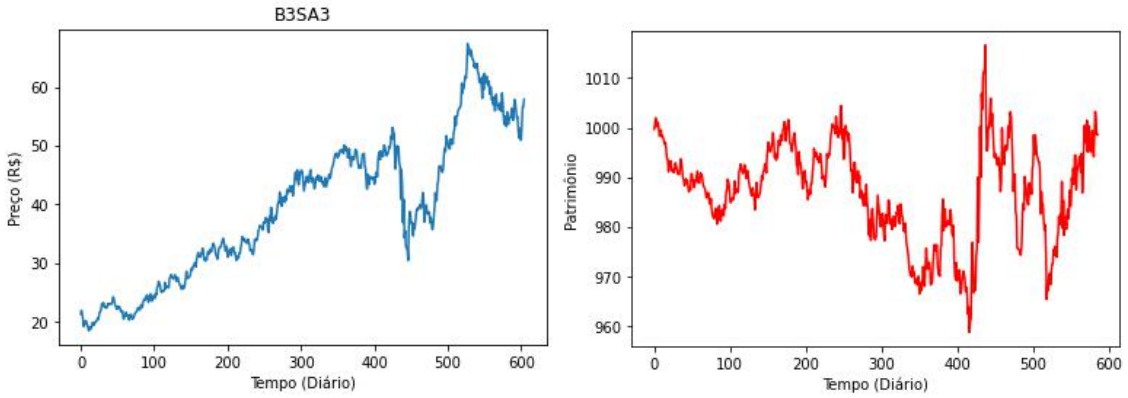
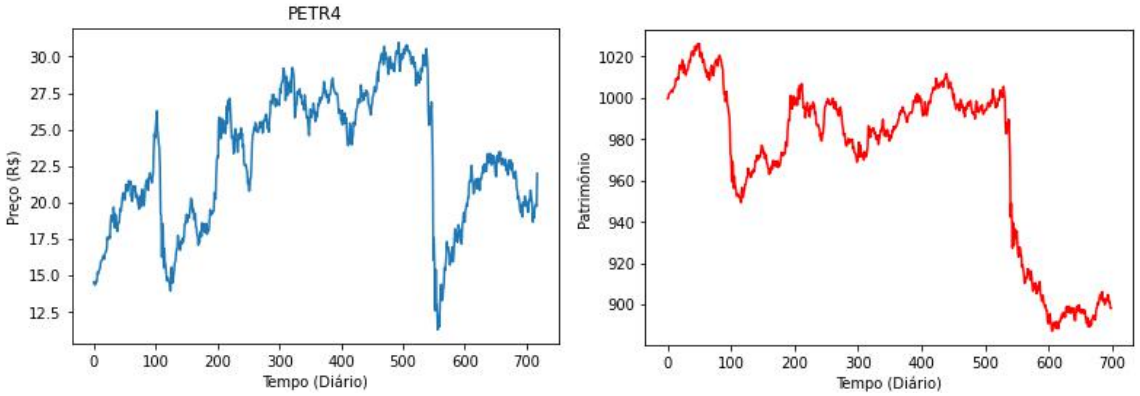


Figura 48: Teste final do agente - Petrobrás (PETR4).



Quadro 7: Resultados do agente treinado.

MÉTRICAS	BTC	ITUB4	ABEV3	B3SA3	PETR4
(Lucro total / Prejuízo total) * 100	**1422%**	**88%**	**100%**	**93%**	**39%**
Porcentagem de negociações com lucro	**80%**	**53%**	**70%**	**33%**	**45%**
(Resultado médio de uma negociação com lucro / Resultado médio de uma negociação com prejuízo) * 100	**355%**	**75%**	**43%**	**186%**	**48%**

O **Quadro 7** apresenta as métricas utilizadas para medir os resultados obtidos em todos os testes. De um modo geral, apenas o experimento utilizando Bitcoin se mostrou promissor. Todos os demais experimentos obtiverem resultados ou negativos ou permeando a neutralidade. As exceções seriam os testes com a ABEV3 e B3SA3 que obtiveram cada uma um resultado positivo em apenas uma das métricas.

6.6 RESUMO DOS EXPERIMENTOS

No **Quadro 8**, o resultado de todos os experimentos são apresentados de forma a viabilizar uma comparação geral. É possível perceber que, cada novo experimento traz uma melhora significativa em todas as métricas utilizadas.

No segundo experimento, embora os resultados gerais ainda fossem bem ruins, o resultado médio de uma negociação com lucro em relação ao de uma negociação com prejuízo já apresentou uma melhora significativa se comparado aos agentes aleatórios. Tal resultado, embora tenha oscilado bastante ao longo dos experimentos, se manteve sempre acima da faixa dos 100% em todos os demais testes com Bitcoin.

Ao analisar a evolução do lucro total em comparação com o prejuízo total e a porcentagem de negociações com lucro, é possível perceber que cada experimento traz uma melhora gradual aos resultados. No entanto, é apenas no quarto experimento que ambos os resultados ultrapassam suas respectivas faixas mínimas de 100% e 50% respectivamente, e passam a apresentar resultados positivos, que foram ainda mais potencializados no experimento final ainda nos testes com Bitcoin.

No entanto, embora o teste com Bitcoin do quinto experimento tenha apresentado resultados muito promissores, ao testar o agente com outros ativos financeiros foi possível concluir que sua capacidade de generalização de aprendizado para outros padrões de oscilação do mercado não foi suficiente para que fosse capaz de atingir resultados satisfatórios.

Experimento	(Lucro total / Prejuízo total) * 100	Porcentagem de negociações com lucro	(Resultado médio de uma negociação com lucro / Resultado médio de uma negociação com prejuízo) * 100
1: Agente aleatório	**0,18%**	**0,32 %**	**16,26%**
2: Primeiro agente treinado	**8,15%**	**4,21 %**	**185,26%**
3: Agente LSTM	**84,12%**	**20%**	**336,39%**

4: Vela ideal	**315,31%**	**66%**	**157,66%**
5: Nova base de dados	**1421,92%**	**80%**	**355,48%**
5: ITUB4	**88%**	**53%**	**75%**
5: ABEV3	**100%**	**70%**	**43%**
5: B3SA3	**93%**	**33%**	**186%**
5 PETR4	**39%**	**45%**	**48%**

7 CONCLUSÃO

Este trabalho se propôs a explorar conceitos na área de aprendizado de máquina no contexto do mercado financeiro. Seu objetivo principal foi o de tentar desenvolver um agente baseado em aprendizado por reforço profundo que fosse capaz de comprar e vender Bitcoins com o objetivo de gerar lucro através dessas negociações. Como parte de sua validação, buscou-se entender também sua efetividade em outros ativos do mercado financeiro, de forma a verificar a viabilidade da generalização de sua utilização, assim como tentar trazer uma maior confiabilidade para seus resultados.

De um modo geral, o agente apresentou resultados bastante otimistas nos testes realizados com Bitcoin, embora não tenha sido capaz de manter tal padrão com os demais ativos. No entanto, mesmo dentro do contexto da criptomoeda, é preciso levar em consideração o fato de que existem simplificações na modelagem do problema que em um ambiente real resultam em custos adicionais, depreciando os resultados obtidos. Um desses custos, por exemplo, é o da transferência do dinheiro que se encontra na exchange de volta para a conta bancária pessoal do agente. Um outro exemplo seriam os custos associados a transformar tal agente em um sistema completamente autônomo, hospedando-o em algum servidor de forma que diariamente fossem executadas negociações através de uma API paga de alguma exchange. Por fim, é possível também citar os impostos em cima da venda com lucro que também impactam de forma negativa no resultado final.

Os experimentos realizados ao longo do trabalho mostraram que é possível utilizar técnicas de aprendizado por reforço profundo dentro do contexto de mercado financeiro e atingir resultados interessantes. É claro que em ambientes cada vez mais próximos do real, os desafios ficam cada vez maiores, o que irá demandar um esforço bem maior em tentar experimentar técnicas mais robustas que possam gerar resultados ainda mais promissores.

7.1 DIFICULDADES ENFRENTADAS

Uma das principais dificuldades enfrentadas durante os experimentos do projeto foi o ambiente onde tais experimentos foram realizados. O tempo de execução máxima de 12 horas fornecido gratuitamente pelo Google Colab restringiu o quão ambicioso era possível ser com

a modelagem do sistema. No entanto, o maior problema com a plataforma foi a falta de garantia de execução ininterrupta. Diversas vezes os experimentos foram interrompidos durante sua execução, sendo necessário reiniciar o ambiente e começá-los do zero novamente.

Outra dificuldade enfrentada foi tentar treinar um agente com uma pequena base de dados. O conceito de aprendizado por reforço profundo é baseado na exploração exaustiva de simulações em um ambiente virtual que tenta simular com fidelidade de extensividade os possíveis cenários a serem encontrados pelo agente no ambiente real. Portanto, quanto maior é a quantidade de casos explorados durante o treinamento do agente, maiores são as chances de que ele seja capaz de reconhecer padrões familiares ao enfrentar cenários a princípio inéditos. Com uma baixa quantidade de dados sendo utilizados para gerar as simulações do treinamento do agente, a tarefa proposta pelo trabalho se tornou ainda mais desafiadora.

7.2 TRABALHOS FUTUROS

Um aspecto interessante de ser trabalhado melhor no futuro é o da utilização de técnicas de otimização de hiperparâmetros. Devido ao problema de recorrente interrupção da execução dos experimentos, evitou-se utilizar tais técnicas devido ao tempo necessário para se testar diferentes combinações de hiperparâmetros. A abordagem utilizada neste trabalho foi puramente empírica, o que sinaliza uma grande oportunidade de melhoria.

Outro ponto a ser analisado é a modelagem do sistema como um todo. Existem várias escolhas realizadas que impactam diretamente em como o agente se comporta, e portanto, no seu potencial de atingir grandes resultados. Um exemplo disso seriam as possíveis ações que o agente é capaz de realizar a cada iteração da simulação. Da forma como foi modelado, não é possível que o agente simule um comportamento *buy & hold* em conjunto com *Dollar-Cost Averaging*, por exemplo. Ele está limitado a se posicionar de forma a estar comprado ou vendido em uma quantidade predeterminada de Bitcoins, impossibilitando-o de continuar a comprar ou vender mais do que tal quantidade. Esta modelagem é bastante limitante, e representa uma boa oportunidade de melhoria.

Um outro exemplo de modelagem do sistema que pode ser refinado é a observação provida ao agente pelo ambiente. É bastante comum que pessoas reais utilizem informações

extras, além das que já estão sendo utilizadas atualmente no projeto, para tomar suas decisões de compra e venda. Um exemplo disso seria a utilização de notícias do mundo real relacionadas ao ativo financeiro negociado. Outro exemplo seriam indicadores estatísticos sobre o preço do ativo, como por exemplo uma média móvel. Essas e outras informações adicionais poderiam ser incorporadas na observação do agente de forma a tentar prover mais contexto para sua tomada de decisão.

Explorar modelos mais complexos de redes neurais seria interessante. Este foi um tópico pouco explorado neste trabalho. As redes aplicadas foram pouco customizadas às necessidades particulares do projeto. Seria interessante um estudo mais aprofundado sobre o impacto das mudanças de modelagem na rede do agente, assim como a utilização de técnicas mais avançadas.

Por fim, para lidar com o problema relacionado à pequena base de dados, seria interessante experimentar formas de aumentá-los. Existem técnicas específicas para aumentar artificialmente um mesmo conjunto de dados, mas seria interessante também testar a integração de outros ativos financeiros além do Bitcoin para compor a base original. Entender qual seria uma boa combinação de ativos para treinar o agente, de forma que ele seja capaz de generalizar melhor casos inéditos pode ser a chave para alcançar resultados realmente impressionantes.

REFERÊNCIAS

Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press, 2016.

Richard S. Sutton and Andrew G. Barto. 2018. Reinforcement Learning: An Introduction. A Bradford Book, 2018.

Théate T, Ernst D. An application of deep reinforcement learning to algorithmic trading. **Expert Systems with Applications**, vol. 173, n. 114632, 2021.
Disponível em: https://doi.org/10.1016/j.eswa.2021.114632. Acesso em: 14 out. 2021.

Yang L, Wanshan Z, Zibin Z. Deep Robust Reinforcement Learning for Practical Algorithmic Trading. **IEEE Access**, vol. 7, 2019.
Disponível em: https://doi.org/10.1109/ACCESS.2019.2932789. Acesso em: 14 out. 2021.

Ponomarev ES, Oseledets IV, Cichocki AS. Using Reinforcement Learning in the Algorithmic Trading Problem. **J. Commun. Technol. Electron**, vol. 64, 2019.
Disponível em: https://doi.org/10.1134/S1064226919120131. Acesso em: 14 out. 2021.

Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, Hasan M, Van Essen BC, Awwal AAS, Asari VK. A State-of-the-Art Survey on Deep Learning Theory and Architectures. **Electronics**, vol. 8, n. 3, 2019.
Disponível em: https://doi.org/10.3390/electronics8030292. Acesso em: 14 out. 2021.

Changhwan L, Yeesuk K, Young SK, Jongseong J. Automatic Disease Annotation From Radiology Reports Using Artificial Intelligence Implemented by a Recurrent Neural Network. **American Journal of Roentgenology**, vol. 212, n. 4, 2019.
Disponível em: https://www.ajronline.org/doi/10.2214/AJR.18.19869. Acesso em: 14 out. 2021.

Langager C, Mansa J. How to Start Investing in Stocks: A Beginner's Guide. **Investopedia**, 2021.
Disponível em: https://www.investopedia.com/articles/basics/06/invest1000.asp. Acesso em: 14 out. 2021.

Frankenfield J, Mansa J. Guide to Bitcoin. **Investopedia**, 2021.
Disponível em: https://www.investopedia.com/terms/b/bitcoin.asp. Acesso em: 14 out. 2021.