

Faculdades Integradas de Taquara – Faccat  
Av. Oscar Martins Rangel, 4.500  
Taquara, RS, CEP 95600-000

*Curso de Sistemas de Informação*

## UTILIZAÇÃO DE REDES NEURAIS PARA A COMPRA E VENDA DE *BITCOINS*

Daniel Fleck

Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil  
danielfleck@sou.faccat.br

Flávia Pereira de Carvalho

Professora Orientadora  
Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil  
fpereira@faccat.br

### Resumo

Este artigo apresenta os resultados de uma pesquisa experimental que teve por finalidade o desenvolvimento de um *software* que automatiza a compra e venda de *Bitcoins* e utiliza o algoritmo de Inteligência Artificial *Long Short Term Memory* (LSTM), mensurando o desempenho do algoritmo na predição do preço desta moeda, em um sistema que utiliza dados históricos, simulando assim, sua rentabilidade quando aplicada ao mercado. Como objetivo específico, definiu-se para o estudo, medir a margem de erro do algoritmo LSTM nos dados analisados. Para esta pesquisa foram utilizados os dados provenientes de negociações efetuadas entre 30 de junho de 2016 à 30 de junho de 2017, por uma corretora de *Bitcoins* alocada em São Francisco, EUA. Para análise dos dados utilizou-se o algoritmo LSTM em um ambiente teste, sendo 80% dos dados aplicados no treinamento de redes neurais e 20% na realização dos testes de compra e venda de *Bitcoins*. Observou-se nos resultados de performance do algoritmo, que o agrupamento para análise dos dados em 1 minuto possui maior eficiência no que se refere a predição do preço e consequentemente menor risco para perda de capital financeiro. O agrupamento de 24 horas, demonstrou-se menos eficaz, por possuir a maior média de erro e o maior erro dentre todos os períodos. Concluiu-se que a aplicação do algoritmo LSTM é eficiente para análise preditiva sobre uma série temporal de preços do *Bitcoin*, sendo sob o ponto de vista do algoritmo, o agrupamento de 1 minuto, a melhor estratégia para análise.

**Palavras-chave:** *Bitcoin*. Criptomoeda. Aprendizado de máquina. Comércio algorítmico.

## AUTOMATING BITCOIN TRADING WITH DEEP LEARNING

### Abstract

*This paper presents the results of an experimental research brought forth from the development of a software that automates the purchase and sale of Bitcoins and use the Long Short Term Memory (LSTM) Artificial Intelligence algorithm to measure the performance of the algorithm in the prediction of the price of this cryptocurrency, in a system that uses historical data, thus simulating its profitability when applied to the market. A specific*

*objective for this research was to measure the margin of error of the LSTM algorithm applied in the data analyzed. This research utilized data from negotiations between June 30th, 2016 and June 30th, 2017, by an exchange located in San Francisco, USA. The LSTM algorithm results data was analyzed in a test environment, with 80% of the data applied in the training of neural networks and 20% remaining data utilized for testing of Bitcoin's price prediction. It was observed in the performance results of the algorithm that the data in 1 minute time series is more efficient to the prediction of the price and consequently presents lower risks for loss of financial capital. The prediction in time periods of 24 hours was shown to be less effective, since it has the highest mean error and the highest error among all periods. It was concluded that the application of the LSTM algorithm is efficient for predictive analysis on a time series of prices of Bitcoin, being from the point of view of the algorithm, the grouping of 1 minute, the best strategy for analysis.*

**Key-words:** *Bitcoin. Criptocurrency. Machine learning. Algorithmic trade.*

## 1 INTRODUÇÃO

No mercado atual existem pessoas dedicadas em tempo integral para a negociação de ativos (INVESTOPEDIA, 2018, tradução nossa). Segundo o Glossário do Banco Central do Brasil (2018) *Day-trade* é “o equivalente a comprar e vender, no mesmo dia, a mesma quantidade de títulos de uma empresa, utilizando para isso a mesma corretora e o mesmo agente de compensação.” Entendemos através disto que a diferença entre o preço de compra e o de venda é o resultado do *Day-trade*.

Com a criação do *Bitcoin* surgiram também as corretoras de moedas virtuais (chamadas de criptomoedas), o que possibilitou a iniciação de várias pessoas à prática do *Day-trade*. A atividade atraiu indivíduos que fazem dela não necessariamente sua profissão de tempo integral, mas a utilizam para um suplemento de renda ou até mesmo como um *hobby*. A atividade de *Day-trade* de *Bitcoins* tem se tornado algo cada vez mais popular, agora não somente restrito ao perfil dos praticantes do *Day-trade* da época pré-criptomoedas. Surge então a necessidade de criação de ferramentas que auxiliam o *Trader* (praticante do *Day-trade*) a comprar e vender com uma maior consciência de resultados. Estes instrumentos de apoio fornecem ao *Trader* uma maior inteligência em suas decisões, assim como um maior controle sobre seus ativos. Nesse contexto, a inteligência artificial (IA) tem alcançado significativos avanços, contribuindo para a utilização de *softwares* inteligentes que automatizam trabalhos de rotina, como entender a fala ou imagens, fazer diagnósticos na medicina e apoiar pesquisas científicas básicas (GOODFELLOW; BENGIO; COURVILLE, 2016, tradução nossa).

Desta forma, o presente estudo, apresenta os resultados de uma pesquisa experimental que teve por finalidade o desenvolvimento de um *software* que automatiza a compra e venda de *Bitcoins* e mensura o desempenho do algoritmo de Inteligência Artificial *Long Short Term Memory* (LSTM)<sup>1</sup>, na predição do preço desta moeda, em um sistema que utiliza dados históricos de transação, aplicando tais dados no treinamento de uma rede neural que utiliza regressão linear para prever se o valor do *Bitcoin* aumentará ou diminuirá, em um determinado período. Com base nesta previsão, o sistema aplica uma ordem de compra ou de venda dentro de um ambiente de testes, simulando assim, sua rentabilidade quando aplicada ao mercado.

O tema deste estudo refere-se ao uso do aprendizado de máquina direcionado ao comércio algorítmico de criptomoedas, assim como análise de dados na predição de resultados, com o intuito de responder aos seguintes questionamentos: É possível estimar o preço de um ativo olhando somente para seus padrões de movimentação histórica? Baseado nesta predição, é possível desenvolver um sistema automatizado de compra e venda que seja rentável? Qual a sua margem de erro e qual a sua rentabilidade?. Estabeleceu-se como hipótese, que a combinação linear de dados em séries temporais existentes, é capaz de predizer a tendência de preço futuro do *Bitcoin*, resultando na criação de um modelo de negócio, cujas métricas de performance podem ser quantificadas e a sua metodologia estudada.

Para a melhor compreensão dos dados, este artigo está estruturado da seguinte maneira: a seção 2 traz o referencial teórico, que descreve os assuntos abordados; a seção 3 explana a metodologia e as tecnologias utilizadas para o desenvolvimento do *Software*; a seção 5 descreve os testes realizados e os resultados alcançados e a seção 6 traz as conclusões obtidas com este trabalho.

## **2 REFERENCIAL TEÓRICO**

Nesta seção, serão apresentados os principais conceitos relevantes ao tema proposto, buscando direcionar a pesquisa com embasamento teórico previamente publicado.

---

<sup>1</sup> Longa Memória de Curto Prazo

## 2.1 Criptomoedas

“Uma criptomoeda é uma forma digital de moeda, que incorpora técnicas criptográficas para fins de segurança” (NAKAMOTO, 2009, p. 1, tradução nossa). A atuação de uma criptomoeda é limitada ao protocolo delineado em seu código, que determina sua função, geração e transferência. Uma criptomoeda difere distintamente de outras moedas por não possuir uma entidade reguladora, tornando-se mais resistente à intervenção e regulação do governo. Outra grande diferença entre uma moeda tradicional e uma criptomoeda, é que ela não possui um valor inerente. Enquanto o valor de uma moeda tradicional é creditada através do ouro, ou de ações do governo, uma criptomoeda só tem o valor que as pessoas concordam que ela tem. Como não é apoiada por um banco central e não tem valor intrínseco, o valor de uma criptomoeda é puramente determinado pela oferta e demanda (TECHOPEDIA, 2018, tradução nossa). Desta forma, seu valor é muito menos estável do que uma moeda padrão e flutua de forma mais semelhante à outras comodidades.

“As criptomoedas são transferidas entre indivíduos ou partes através do uso de criptografia assimétrica para garantir a segurança e integridade de qualquer transação feita” (NAKAMOTO, 2009, p. 2, tradução nossa).

## 2.2 Bitcoin

*Bitcoin* é a implementação de uma criptomoeda. Sua primeira especificação foi publicada no formato de artigo em 2009 por Satoshi Nakamoto (BITCOIN, 2018, tradução nossa), são gerados através de um processo chamado de “mineração” de *Bitcoins*, uma atividade computacional de adivinhar valores de *hash*<sup>2</sup>. O usuário que adivinha o valor do *hash* recebe como recompensa uma quantia de *Bitcoins*. Muitos usuários unem-se em grupos para aumentar a sua capacidade computacional e sua probabilidade de sucesso (MINING, 2018, tradução nossa).

O *Bitcoin* grava todas as suas transações em uma espécie de livro-razão que é conhecido como cadeia de blocos (BLOCK CHAIN, 2018, tradução nossa). A moeda utiliza um sistema de escrituração de entrada tripla para manter a segurança e integridade da cadeia de blocos. Desta forma, o remetente, destinatário e um intermediário possuem cópias assinadas digitalmente da transação efetuada. Isso é usado como evidência no caso da

---

<sup>2</sup> Resultado de uma função matemática.

necessidade de uma transação ser contestada (FINANCIAL CRYPTOGRAPHY, 2011, tradução nossa).

O *Bitcoin* tem por característica inerente, o pseudo-anonimato e é considerado por muitos como um método de pagamento anônimo. No entanto, por possuir cada registro de operação efetuado em um livro razão público e digital, as atividades de uma carteira eletrônica podem ser facilmente rastreadas (OBER; KATZENBEISSER; HAMACHER, 2013, tradução nossa). Os *Bitcoins*, conforme mencionado anteriormente, são gerados por “mineração”, no entanto, para o propósito desta pesquisa, a mineração de *Bitcoin* não é necessária. O sistema destina-se a obter resultados comprando e vendendo *Bitcoins*, uma vez que o valor tenha aumentado além do preço de compra original.

### **2.3 Negociação algorítmica**

A negociação algorítmica (também chamada de comércio algorítmico) é a prática de negociar moedas, comodidades ou ações usando algoritmos de predição, para antever se o valor vai aumentar ou diminuir. Usando esta informação, um sistema determina o momento mais apropriado para comprar ou vender, visando maximizar o lucro (INVESTOPEDIA, 2018a, tradução nossa).

A negociação algorítmica funciona analisando as tendências de mercado e utiliza algoritmos diversos para previsões sobre a probabilidade de comportamento dos valores, seja com base nos dados históricos de negociação, influenciadores externos (mudanças nas taxas de juros, novas leis, etc.), assim como notícias relevantes sobre um determinado ativo. Um dos benefícios da negociação algorítmica em relação à negociação feita por seres humanos é que o seu funcionamento ocorre sem a interferência das emoções (TAFFLER, 2014, tradução nossa). No entanto, se o valor das ações começar a cair, com a prevalência de negociação algorítmica no momento atual, muitos dos sistemas podem acabar em um efeito cascata negativo, ou seja, alguém vende e isto leva o preço a cair, acarretando em uma nova venda pelo sistema, diminuindo o valor comercial da moeda e assim sucessivamente. Como o comércio algorítmico pode ocorrer em milissegundos, isso pode causar grandes picos e quedas no preço, como no caso da queda relâmpago de 2010 (THE ECONOMIST, 2018, tradução nossa).

Entidades reguladoras procuram evitar que este tipo de situação ocorra, introduzindo um conjunto de medidas de proteção, como disjuntores que suspendem ou permitem apenas

negociações limitadas por um determinado tempo, caso os preços caiam ou subam acima de um determinado parâmetro (INVESTOPEDIA, 2018b; SUBRAHMANYAM, 2013, tradução nossa).

Em mercados onde as negociações ocorrem dentro de um horário pré-determinado, a negociação algorítmica fornece mais benefícios ao permitir que negociações sejam executadas e monitoradas 24 horas por dia.

## 2.4 Predição de preço

Sendo o *Bitcoin* uma tecnologia com preço de mercado bastante volátil, existem poucos modelos rentáveis de previsão de preço. Os existentes possuem uma eficácia limitada para um ambiente de produção. Mais recentemente, Shah e Zhang (2014, tradução nossa) publicaram os resultados da aplicação de Regressão Bayesiana à previsão de preços do *Bitcoin*, um algoritmo que alcançou significativos resultados. O trabalho atual, no entanto, emprega a utilização do algoritmo LSTM que tem encontrado resultados muito satisfatórios quando aplicados a cenários onde há uma necessidade de manter uma “memória” em uma sequência de dados (SAK; SENIOR; BEAUFAYS, 2014, tradução nossa).

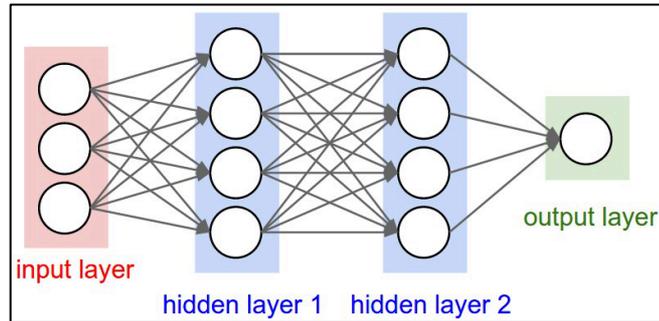
## 2.5 Redes Neurais

Segundo Greaves e Au (2015), redes neurais são uma família de métodos de aprendizagem de máquina inspirados em redes neurais biológicas, modelando um sistema de neurônios interconectados, cujos parâmetros são aprimorados baseando-se na aprendizagem iterativa. As redes neurais tem por padrão, a conexão de uma entrada multi-dimensional em uma ou mais camadas ocultas de neurônios antes de prever uma saída. O parâmetro de “esquecimento” é usado para evitar a sobrecarga do modelo. Camadas ocultas são modeladas de acordo com a necessidade da aplicação do modelo do sistema, sendo geralmente ativadas pela função matemática *Softmax*<sup>3</sup> (Figura 1).

---

<sup>3</sup> Função que calcula a distribuição de probabilidades do evento sobre eventos diferentes.

**Figura 1 – Representação de Rede neural com 3 neurônios de entrada, duas camadas ocultas e uma camada de saída**

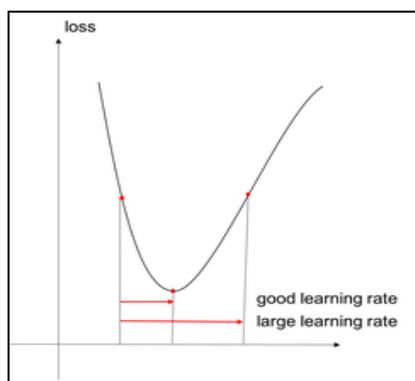


Fonte: Greaves e Au (2015)

Legenda: *input layer* (camada de entrada), *hidden layer 1 e 2* (camadas ocultas) e *output layer* (camada de saída).

Algumas características das redes são configuráveis de acordo com a necessidade do modelo a serem utilizadas. Um deles é a taxa de aprendizagem (*learning rate*). Estamos ajustando a configuração como um parâmetro que controla o quanto tá sendo ajustado os pesos das nossas rede com relação ao gradiente de perda. Quanto mais baixo o valor, mais devagar viaja-se ao longo do declive descendente. Embora isso possa ser uma boa ideia (usando uma baixa taxa de aprendizado), em termos de garantir que não perderemos nenhum local mínimo (ponto *óptimum*), também pode significar que levararemos muito tempo para convergir - especialmente se ficarmos presos em uma região de platô. Além disso, a taxa de aprendizado afeta a rapidez com que o nosso modelo pode convergir para um mínimo local (também conhecido como chegar com a melhor precisão). Assim, obtê-lo desde o início significaria menos tempo para treinarmos o modelo (ZULKIFLI, 2018, tradução nossa) (Figura 2).

**Figura 2 - Taxas de aprendizado em relação ao tempo**



Fonte: Zulkifli (2018)

Legenda: *Loss* (perda), *good learning rate* (boa taxa de aprendizado) e *large learning rate* (taxa de aprendizado excessiva).

## 2.6 Redes Neurais Recorrentes

As redes neurais possuem diferentes classificações, de acordo com sua função. A rede LSTM utilizada nesta pesquisa, é definida como uma rede neural recorrente, que por sua vez utiliza como entrada não apenas a informação da entrada atual, mas também o que perceberam anteriormente no tempo (SKYMIND, 2018, tradução nossa).

A principal diferença de uma rede neural recorrente em relação as demais, é que em uma rede neural tradicional as entradas são inseridas na rede e transformadas em uma saída. Nas redes tradicionais, as previsões não dependem dos resultados das previsões anteriores. Por exemplo, se a rede classificar um conjunto de imagens e a primeira imagem for classificada como navio, a próxima classificação de imagem não poderá ser alterada pela classificação anterior. Se a segunda imagem for um avião, ela será classificada como tal. Mas a situação é diferente em redes neurais recorrentes. A decisão que a rede neural recorrente recebe neste momento, depende da decisão que a rede obteve no momento anterior. Portanto, a saída de resultados de uma rede neural recorrente depende da saída anterior e da entrada atual (SKYMIND, 2018, tradução nossa).

Além disso, as entradas e saídas para uma rede tradicional devem ser bidimensionais e as entradas e saídas de uma rede neural recorrente devem ser tridimensionais, possuindo número de exemplos, tamanho da entrada e comprimento da série temporal (WIJEWARDHANA, 2016, tradução nossa).

Uma característica da LSTM é o parâmetro da retropropagação truncada (*Truncated Retropropagation*), onde as redes neurais recorrentes são capazes de aprender a dependência temporal através de múltiplas iterações em problemas de regressões lineares (SKYMIND, 2018, tradução nossa).

As redes neurais recorrentes modernas, como a rede LSTM, são treinadas com uma variação do algoritmo *Backpropagation Through Time*<sup>4</sup>. Este algoritmo foi melhorado em sua eficiência para problemas de predição de sequência muito longas, sendo então chamado de *Backpropagation Truncated Through Time* (BPTT)<sup>5</sup> (MAZUMDAR; HARLEY, 2008, tradução nossa).

Segundo os mesmos autores (2008, p. 3845, tradução nossa)

---

<sup>4</sup> Retropropagação através do tempo.

<sup>5</sup> Retropropagação truncada através do tempo.

A ideia de desdobramento no tempo pode ser aplicada levando em consideração o histórico dos dados de entrada e estado da rede para um número fixo de etapas de tempo. Partindo de que as conexões de retorno e recorrência em teoria, tem uma memória infinita, o BPTT deve ser truncado para levar em conta apenas um número predefinido de tempo de passos anteriores. O número de etapas é chamado de profundidade de truncamento e é geralmente denotado por  $h$ . Qualquer informação mais antiga que  $h$  nas iterações de tempo em relação a informação passada é considerada irrelevante e pode, portanto, ser ignorada.

## 2.7 Algoritmo *Long Short Term Memory* e conceito de célula de memória

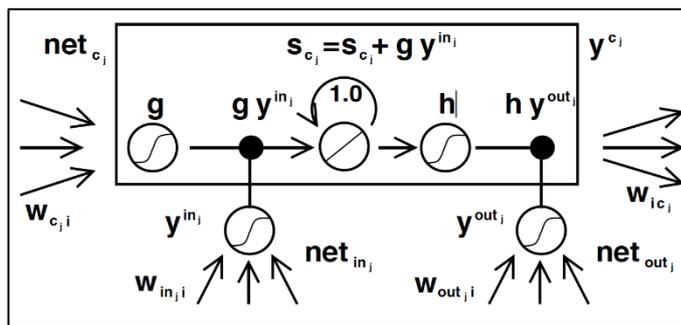
Em 1997, os pesquisadores alemães Sepp Hochreiter e Juergen Schmidhuber publicaram uma proposta de variação de redes neurais recorrentes com as chamadas *Long Short Term Memory*, ou LSTM (HOCHREITER; SCHMIDHUBER, 1997, tradução nossa).

O LSTM contém unidades especiais chamadas de blocos de memória na camada oculta recorrente de sua rede neural. Os blocos de memória contêm código que busca imitar células de memória com auto-conexões, armazenando o estado temporal da rede. Além disso ela possui unidades multiplicativas especiais chamadas de “portões” para controlar o fluxo de informações. Cada bloco de memória na arquitetura original contém um portão de entrada e um de saída. O portão de entrada controla o fluxo de ativações de entrada na célula de memória. O portão de saída controla o fluxo de saída das ativações das células para o resto da rede. Mais tarde, um terceiro portão foi adicionado ao bloco de memória chamado “portão de esquecimento” (GERS; SCHMIDHUBER; CUMMINS, 2000, tradução nossa).

Segundo Hochreiter e Schmidhuber (1997, p. 7, tradução nossa) e conforme demonstrado na Figura 3:

Para construir uma arquitetura que permita um fluxo de erro constante através de unidades conectadas uma unidade de entrada multiplicativa é introduzida para proteger o conteúdo da memória armazenado em  $j$  de interferências de outras entradas irrelevantes, e uma unidade de porta de saída multiplicativa é introduzida para proteger as outras unidades das interferências de conteúdos de memória irrelevantes em um determinado momento de tempo. A unidade resultante, é chamada de célula de memória.

**Figura 3 – Funcionamento da Célula de Memória**



Fonte: Hochreiter e Schmidhuber (1997).

Começando da esquerda para a direita, as setas triplas mostram onde a informação flui para a célula em vários pontos. Essa combinação da entrada atual e do estado passado da célula é alimentada não apenas à própria célula, mas também a cada uma de suas três portas, que decidirá como a entrada será tratada.

Os pontos pretos são os próprios portões, que determinam, respectivamente, se é possível deixar entrar novos dados, apagar o estado atual da célula e/ou deixar que esse estado cause impacto na saída da rede no momento atual. Conforme explicado por Sak, Senior e Beaufays (2014, p. 339, tradução nossa), “W representa o peso de matrizes,  $\sigma$  é a função sigmoide logística”.

A documentação sobre LSTM registrada pela empresa DeepMind explana muito bem quando explica que (SKYMIND, 2018, tradução nossa)

$s_{c_j}$  é o estado atual da célula de memória e  $g y^{in_j}$  é a entrada atual para ela. Cada porta pode ser aberta ou fechada, e elas irão recombinar seus estados de abertura e fechamento em cada iteração. A célula pode esquecer ou manter seu estado, ter suas informações sobrescritas, ou ser simplesmente lida a cada iteração de tempo.

Com a estratégia de portões evita-se conflitos de peso de entrada. Conforme Hochreiter e Schmidhuber (1997, tradução nossa) explicam através da figura 3,

O  $in_j$  controla o fluxo de erro para as conexões de entrada da célula de memória  $c_j$   $w_{c_j i}$ . Para contornar os conflitos de peso de saída de  $c_j$ ,  $out_j$  controla o fluxo de erro da saída das conexões da unidade  $j$ . Em outras palavras, a rede pode usar  $in_j$  para decidir quando manter ou substituir informações na célula de memória  $c_j$  e  $out_j$  para decidir quando acessar a célula de memória  $c_j$  e quando impedir que outras unidades sofram interferência por  $c_j$ .

### 3 METODOLOGIA

Como pode ser visto no referencial teórico, a negociação algorítmica possui por finalidade, antever se o valor de determinadas moedas ou ações, irá aumentar ou diminuir, determinando o momento mais apropriado para comprar ou vender.

Diante deste cenário, foi desenvolvido um *software* que automatiza a compra e venda de *Bitcoins*, através da utilização do algoritmo LSTM, mensurando seu desempenho na predição do preço desta moeda, em um sistema que utiliza dados históricos de transação, aplicando tais dados no treinamento de uma rede neural que utiliza regressão linear para prever se o valor do *Bitcoin* aumentará ou diminuirá, em um determinado período. Com base nesta previsão, o sistema aplica uma ordem de compra ou de venda dentro de um ambiente de testes, simulando assim, sua rentabilidade quando aplicada ao mercado.

Destaca-se que a abordagem utilizada neste projeto de pesquisa não se preocupa com o rebalanceamento provocado pelo impacto das vendas, uma vez que o desempenho do algoritmo utiliza dados históricos para o valor dos *Bitcoins*, assim como recompensa monetária fictícia pela compra e venda.

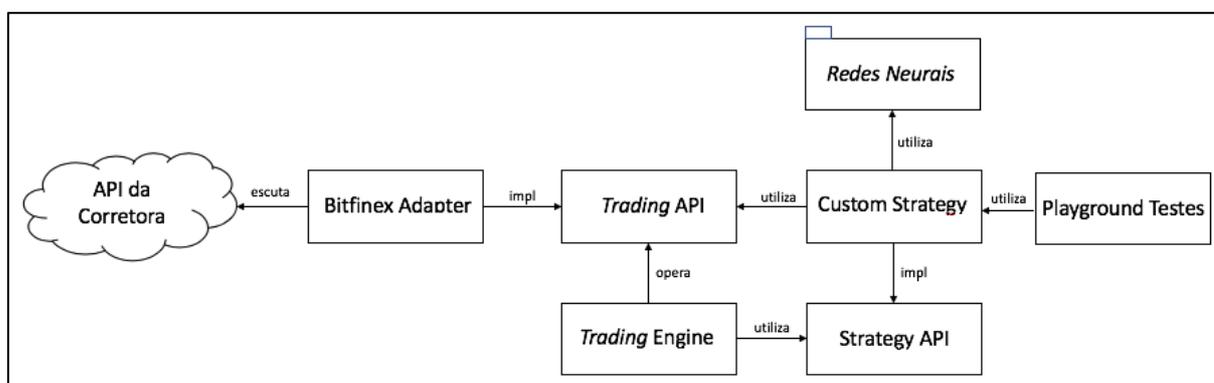
#### 3.1 Estruturação do sistema

Se ativado o modo de produção, a corretora escolhida para negociações (*Bitfinex*) possui uma Interface de Aplicação Programada (API), disponível aos seus usuários. Desta forma, pode-se estabelecer comunicação direta através do *software apresentado neste artigo*. Para este processo desenvolvemos um adaptador específico que cumpri com os protocolos de comunicação, de modo que ela seja segura, uma vez que através dela serão recebidos dados sensíveis ao funcionamento do sistema, tais como preço do *Bitcoin* e ordens de compra e venda. Esta camada de comunicação está localizada na classe *Bitfinex Adapter*, que por sua vez implementa a *Trading API*, que busca ser uma interface de mais alto nível para facilitar a possível inclusão de adaptadores para outras corretoras no futuro, sem a necessidade de duplicação de métodos que podem ser compartilhados, evitando assim código repetido no sistema e uma melhor organização.

Os adaptadores possuem a função de possibilitarem um fluxo constante de informações, que serão utilizadas pela *Trading Engine*, que se assemelha ao motor no qual o sistema opera. A *Trading Engine* provê um conjunto de métodos para integração entre os

adaptadores e as interfaces de estratégia (*Strategy API*). A *Custom Strategy* implementa uma ou mais estratégias desenvolvidas para o comércio algorítmico. Ela utiliza a interface da *Strategy API* onde é onde podemos colocar as desejadas regras de negócio que determinam o momento ideal para a tomada de decisão sobre a compra e a venda dos *Bitcoins*. A *Strategy API* usa informações provenientes do nosso modelo desenvolvido, pré-treinado pelo pacote de classes das Redes Neurais, responsável pela parametrização e treinamento das redes LSTM que fornecerão a estimativa de flutuação do preço para o próximo período. Já o *Playground de Testes* é uma classe a parte, dentro do fluxo do sistema que imita o funcionamento do sistema no mercado, utilizando-se de um banco de dados históricos do *Bitcoin* (Figura 4).

**Figura 4 – Arquitetura do sistema**



Fonte: Autor (2018).

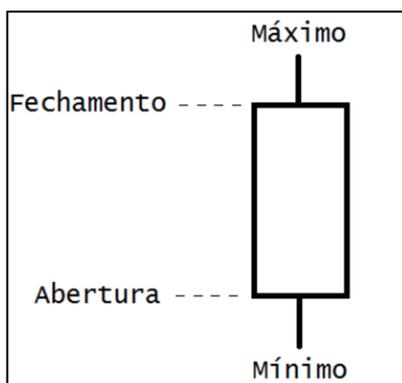
### 3.2 Coleta de dados

Para esta pesquisa foram utilizados um conjunto de dados provenientes de negociações efetuadas na corretora Coinbase, uma corretora de Bitcoins alocada em São Francisco, EUA (COINBASE, 2018). Os registros dos dados contêm informações da movimentação histórica do preço do Bitcoin em USD (*United States dollar*, moeda oficial dos Estados Unidos) e estão agrupados em períodos de um minuto, iniciando em 01 de dezembro de 2014 à 8 de janeiro de 2018. Os registros contemplam o estado do preço no momento de abertura e fechamento do minuto, seu valor máximo e mínimo registrados dentro do mesmo período e o volume de Bitcoins comercializados pela corretora no intervalo de tempo citado.

Para alimentar as redes neurais, os dados foram organizados e agrupados em uma série temporal para minimizar as preocupações com o ruído de medições de maior granularidade e menor volatilidade. Foram levados em consideração para cada período, o preço respectivo de

abertura, o preço no momento do fechamento, seu ponto alto atingido, seu ponto mais baixo, assim como o volume de vendas, conforme representado na figura 5, onde demonstra-se através de um *candlestick*<sup>6</sup> o conjunto destas cinco informações simultâneas dentro de um período “X”.

**Figura 5 – Valores informados às redes neurais**



Fonte: Prado et al., 2013.

### 3.3 Parametrização das Redes Neurais

Para fins de testes de performance do algoritmo, os resultados desta pesquisa foram desenvolvidos no ambiente de testes existente dentro do próprio sistema. Para Dias Neto (2007), “testar um *software* significa verificar através de uma execução controlada se o seu comportamento corre de acordo com o especificado”, partindo desta definição o conceito de teste remete a uma validação prévia do sistema desenvolvido antes de executar a etapa seguinte.

Diferentes estratégias e parâmetros puderam ser analisados e testados sem a necessidade de haver investimento de capital inicial, ou tampouco uma conexão externa à uma corretora. Não existe uma técnica para configuração de rede programática, ou um único método que seja padrão e aceito para configuração de rede. Seguindo apoio em pesquisas relacionadas, pode-se definir uma arquitetura de rede competente (isto é, o número e o tipo de camadas neurais e o número de neurônios compreendendo cada camada). Buscamos, desta forma, uma arquitetura otimizada através do ajuste da configuração efetuada iterativamente durante o treinamento das redes e na observação de seus resultados.

---

<sup>6</sup> Gráfico em velas que representa o conjunto de preço de um ativo, formado pelo valor de abertura, fechamento, seu ponto alto atingido e seu ponto mais baixo.

Nesta pesquisa foram utilizados 5 neurônios na camada de entrada. Cada neurônio recebeu uma das informações relacionadas ao preço do Bitcoin em determinado período sendo eles: o preço respectivo de abertura, o preço no momento do fechamento, seu ponto mais alto atingido, seu ponto mais baixo, assim como o volume total de vendas. Optamos por 1 neurônio na camada de saída, projetando em modo de regressão linear o preço de fechamento. A proporção de esquecimento<sup>7</sup> das células é de 0,2 com duas camadas ocultas de LSTMs de 256 células de memória cada, assim como uma camada densa (utilizando *softmax activation*<sup>8</sup>) contendo 32 células.

Do conjunto de dados coletados, 80% foi utilizado para treinamento das redes e 20% para teste. As redes neurais têm aprendizado supervisionado, sendo o nível de ajuste de parâmetros (*learning rate*) delimitado a 0,001, com 1 época<sup>9</sup> e a utilização de otimização de parâmetros efetuada com a estratégia *Stochastic Gradient Descent*<sup>10</sup>. Outro parâmetro considerado é a retropropagação truncada ao longo do tempo (*Truncated Backpropagation Through Time*) que neste modelo é de 22. A retropropagação truncada ao longo do tempo mantém os benefícios computacionais da retropropagação ao longo do tempo ao mesmo tempo em que alivia a necessidade de uma leitura completa em toda a sequência de dados em cada etapa (TALLEC; OLLIVER, 2017, tradução nossa).

### 3.4 Séries de tempo

Tendo como premissa que tendências de variação do preço podem ser inferidas diretamente de uma combinação linear de dados de séries temporais existentes (SHAH; ZHANG, 2014, tradução nossa), construímos para o treinamento das redes neurais, seis conjuntos de séries temporais. Estas possuem conteúdo idêntico, sendo sua diferença o agrupamento dos dados em diferentes períodos. As séries contemplam um total de 12 meses cada, sendo este período datado de 30 de junho de 2016 à 30 de junho de 2017. Na escolha de qual o melhor agrupamento a ser utilizado para o treinamento da rede neural, realizamos testes com intervalos de 1, 15 e 30 minutos, 1, 2, 4 e 24 horas, respectivamente.

Para a realização dos testes envolvendo a predição do preço do *Bitcoin*, utilizaram os mesmos agrupamentos de tempo, porém no período de 01 de maio de 2017 à 31 de maio de

---

<sup>7</sup> Parâmetro de esquecimento de dados, define qual informação será mantida e qual será descartada.

<sup>8</sup> Cálculo matemático utilizado para ativar o funcionamento do neurônio (BISHOP, 2006, tradução nossa).

<sup>9</sup> Uma passagem completa para frente e uma passagem completa para trás, de todos os exemplos de treinamento.

<sup>10</sup> Método iterativo para otimizar uma função diferenciável, objetiva uma aproximação estocástica da otimização de gradiente descendente (BOTTOU, 2010)

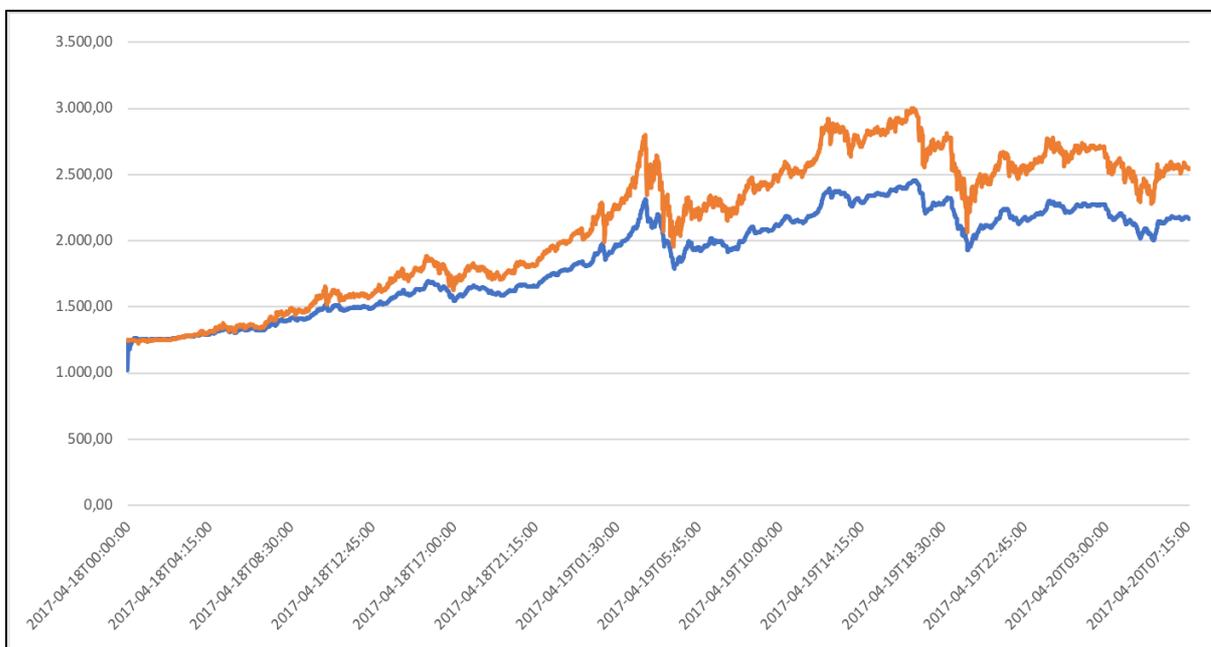
2017, nos agrupamentos de 15 minutos, 1, 2, 4 e 24 horas. Para o intervalo de 1 minuto, as datas englobadas foram de 18 de abril de 2017 à 20 de abril de 2017 e no agrupamento de 30 minutos, incluiu-se o período de 18 de abril de 2017 a 18 de maio de 2017. Para a exposição dos resultados, utilizou-se gráficos, criados com o software *Microsoft Excel*<sup>®</sup>

## 4 RESULTADOS

Esta seção apresenta os resultados obtidos com o desenvolvimento do software ao longo do projeto.

Os testes de aplicação do algoritmo LSTM para predição do valor do *Bitcoin*, resultaram em diferentes margens de erro, conforme os agrupamentos de tempo utilizados para análise dos dados. A seguir, demonstram-se nos Gráficos 1 a 7, tais resultados. O eixo X representa a linha do tempo, explanando as datas analisadas. O eixo Y, demonstra o valor do Bitcoin em USD. Os segmentos de linha representam a volatilidade do preço em relação ao tempo.

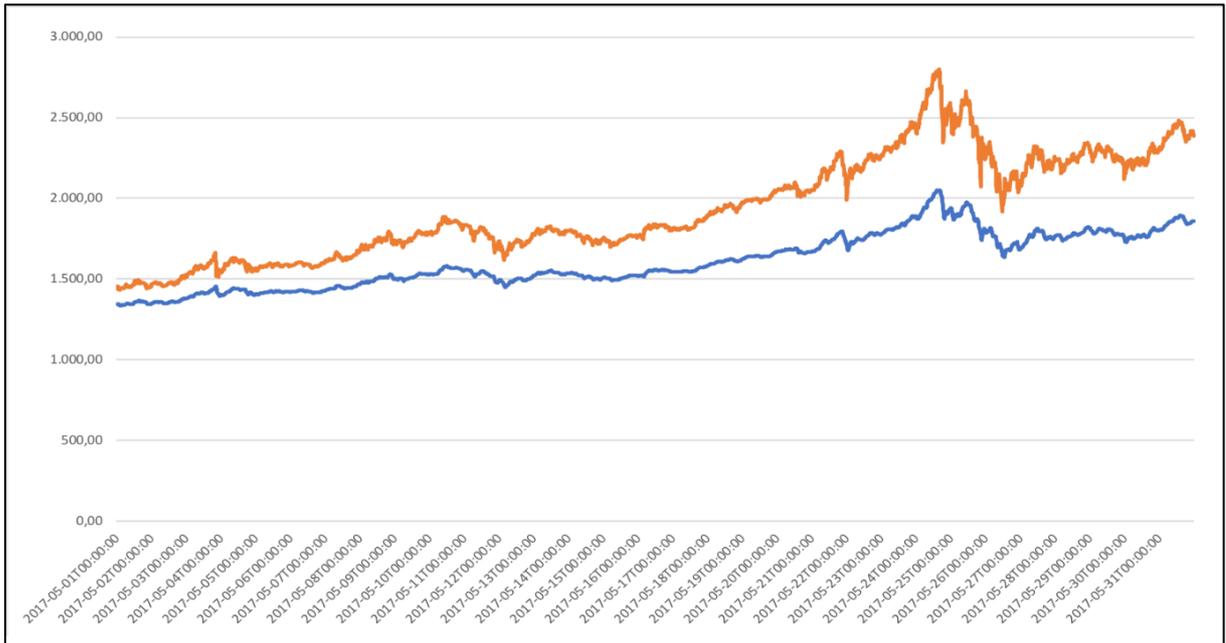
**Gráfico 1 – Predição do algoritmo na análise dos dados de minuto a minuto**



Fonte: Autor (2018).

Legenda: — predição — preço real

**Gráfico 2 – Predição do algoritmo na análise dos dados no agrupamento de quinze minutos**



Fonte: Autor (2018).

Legenda: — previsão — preço real

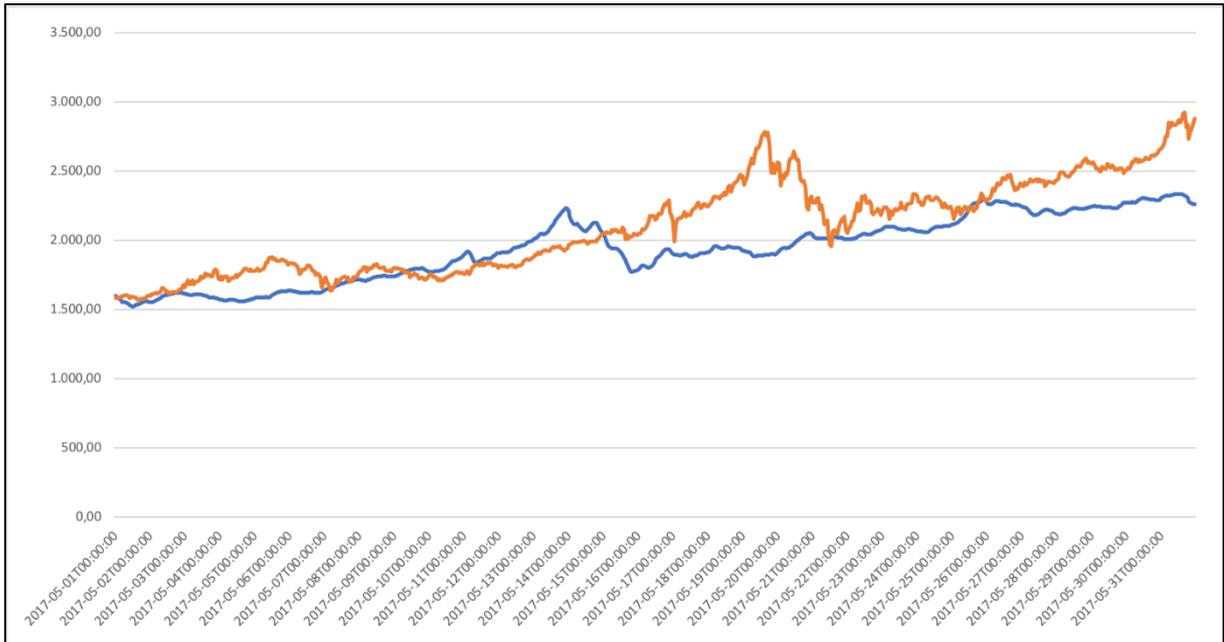
**Gráfico 3 – Predição do algoritmo na análise dos dados no agrupamento de trinta minutos**



Fonte: Autor (2018).

Legenda: — previsão — preço real

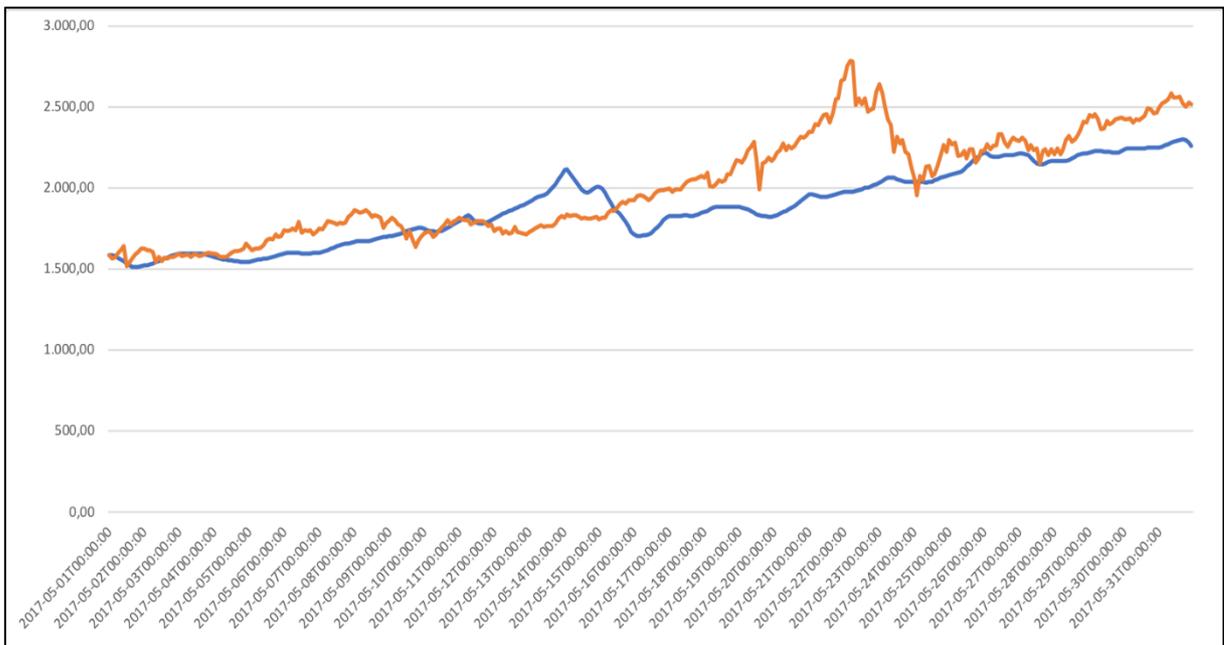
**Gráfico 4 – Predição do algoritmo na análise dos dados no agrupamento de uma hora**



Fonte: Autor (2018).

Legenda: — predição — preço real

**Gráfico 5 – Predição do algoritmo na análise dos dados no agrupamento de duas horas**



Fonte: Autor (2018).

Legenda: — predição — preço real

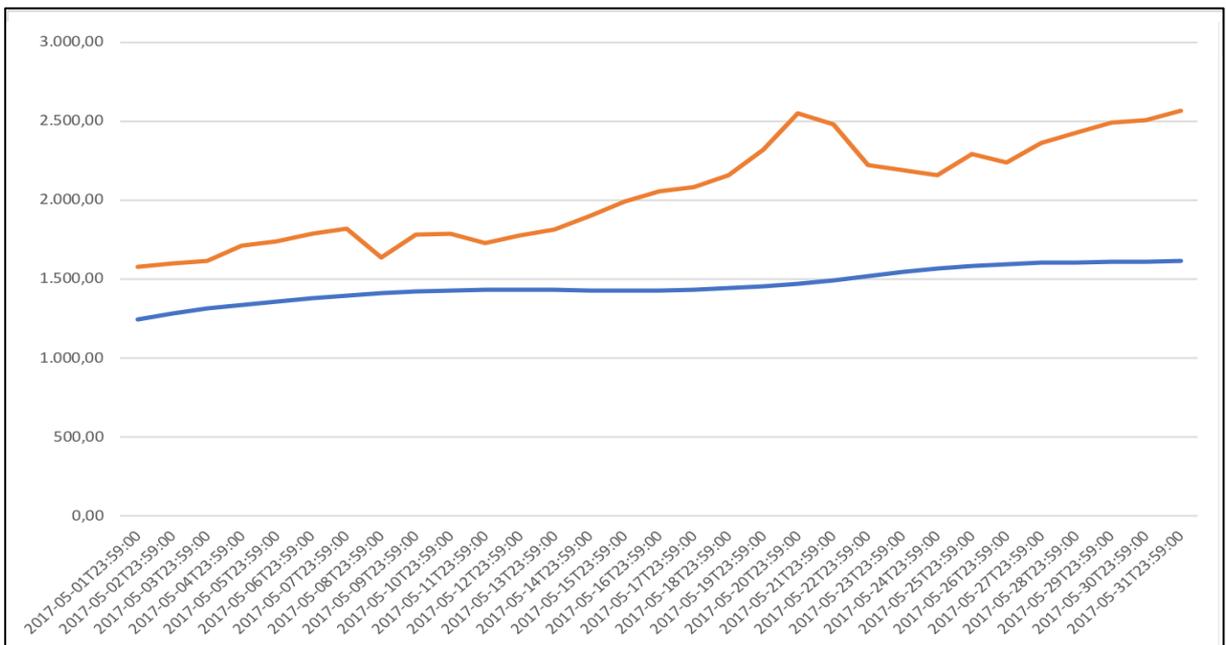
**Gráfico 6 – Predição do algoritmo na análise dos dados no agrupamento de quatro horas**



Fonte: Autor (2018).

Legenda: — predição — preço real

**Gráfico 7 – Predição do algoritmo na análise dos dados no agrupamento de vinte e quatro horas**



Fonte: Autor (2018).

Legenda: — predição — preço real

Conforme citado anteriormente, a aplicação do algoritmo resultou em diferentes margens de erro, conforme os agrupamentos de tempo utilizados para análise dos dados. A

seguir, demonstram-se na Tabela 1, a média de erro por agrupamento, o maior e o menor erro de previsão, em cada um dos períodos.

**Tabela 1 – Margens de erro por agrupamento de tempo**

<b>AGRUPAMENTO</b>	<b>MÉDIA DE ERRO</b>	<b>MAIOR ERRO</b>	<b>MENOR ERRO</b>
<b>1 minuto</b>	10.82%	19.77%	2.51%
<b>15 minutos</b>	15.22%	24.29	1.32%
<b>30 minutos</b>	14.63%	34.61%	1.44%
<b>1 hora</b>	14.63%	34.61%	1.44%
<b>2 horas</b>	16.78%	45.64%	1.82%
<b>4 horas</b>	23.49%	51.81%	0.33%
<b>24 horas</b>	32.38%	53.04%	13.96%

Fonte: Autor (2018).

Conforme demonstrado na tabela 1, a melhor performance do algoritmo foi obtida no agrupamento de 1 minuto, que possuiu as previsões mais próximas ao valor de mercado do *Bitcoin*. Tal agrupamento também demonstrou a menor taxa de maior erro cometido durante o período de teste, informação fundamental para a aplicação do *Software* em ambiente de produção, pois infere a possibilidade de maior ou menor perda de capital. Cabe ressaltar que a previsão de agrupamento de 1 minuto possui a maior quantidade de dados disponíveis para treinamento das redes.

Em relação ao agrupamento com maior ineficiência, os dados analisados em 24 horas, possuíram a maior média de erro e o maior erro dentre todos os períodos.

Quando analisado o menor erro, a análise de dados em agrupamento de 4 horas, alcançou a predição mais próxima do valor, em um momento isolado.

## **5 CONCLUSÃO**

Este projeto, resultou na criação de um *Software* que automatiza a compra e venda de *Bitcoin* e realiza previsão de tendência de variação de preço através da utilização do algoritmo LSTM. Desta maneira, o comércio algorítmico pode ser realizado sem a utilização de pesquisas manuais em diversas fontes de dados. A utilização de um sistema informatizado também auxilia o usuário na realização de cálculos e agrupamento de informações, tornando o processo mais eficiente.

Observou-se nos resultados de performance do algoritmo, que o agrupamento para análise dos dados em 1 minuto possui maior eficiência no que se refere a predição do preço e conseqüentemente menor risco para perda de capital financeiro. Por outro lado, o agrupamento de 24 horas, demonstrou-se menos eficaz, por possuir a maior média de erro e o maior erro dentre todos os períodos. Quando verificado o menor erro, a análise de dados em agrupamento de 4 horas, alcançou a predição mais próxima do valor, em um momento isolado.

O aprofundamento dos diagnósticos em múltiplas séries de tempo demonstra que o LSTM pode capturar em parte o comportamento padrão do preço, mas em última análise, gostaríamos de prever condições e eventos com antecipação, o que ainda é um processo suscetível a falhas.

Desta forma, demonstra-se que o reconhecimento de padrão de comportamento do preço através dos LSTMs, obtém vantagem em comparação com a previsão humana, minimizando a necessidade de tempo de análise, podendo ser também uma ferramenta auxiliar ao *trader*. Todavia, nossa estruturação de projeto, no que se refere a preparação de dados, não considera algumas características possuídas por criptomoedas, que poderiam influenciar diretamente o preço, tais como quantidade de moedas em circulação, status da mineração de dados e informações como notícias na mídia, que impactariam as decisões de outros compradores ou vendedores.

Embora o algoritmo produza resultados promissores, este é apenas um pequeno passo nessa linha de pesquisa, podendo ser aprimorado através da junção da predição de valor através do algoritmo LSTM à automação da compra e venda do *Bitcoin*.

Ainda como trabalhos futuros, pretende-se analisar uma maior quantidade de dados e combinações de parâmetros na tentativa de melhorar as capacidades de previsão das redes LSTMs; identificar regras adicionais necessárias para maximizar a rentabilidade no processo de mercado algorítmico e implementar variáveis indicadoras para análise de sentimento de mercado, que considere notícias provenientes da mídia através da leitura de informações e comentários postados em redes sociais. Outro fator a ser pesquisado é a inclusão de observações simultâneas da movimentação do comércio de *Bitcoins* em mais do que uma corretora, ampliando assim a visão do sistema em relação ao mercado como um todo.

Por fim, a documentação gerada neste trabalho, além de servir como meios próprios de desenvolvimento de um sistema, pode ser utilizada por outras pessoas no desenvolvimento de ferramentas similares ou mais complexas.

## 6 REFERÊNCIAS

BITCOIN. *What is Bitcoin?*. Disponível em: <<https://bitcoin.org/en/faq#what-isbitcoin>>. Acesso em: 02 out. 2018.

BISHOP, C. M. *Pattern Recognition and Machine Learning*. USA: Springer, 2006.

BLOCK CHAIN. *In: Bitcoin Wiki*. Disponível em: <[https://en.bitcoin.it/wiki/Block\\_chain](https://en.bitcoin.it/wiki/Block_chain)>. Acesso em: 03 out. 2018.

BOTTOU, L. Large-Scale Machine Learning with Stochastic Gradient Descent. *In: LECHEVALLIER, Y.; SAPORTA, G (Org.). Proceeding in Computational Statistics*. 2010. P. 177- 186. Disponível em: <<https://link.springer.com/content/pdf/10.1007%2F978-3-7908-2604-3.pdf>>. Acesso em: 21 out. 2018.

DAY-TRADE. *In: Glossário do Banco Central do Brasil*. Disponível em: <<https://www.bcb.gov.br/GlossarioLista.asp?idioma=P&idpai=GLOSSARIO>>. Acesso em: 06 ago. 2018.

DIAS NETO, A. C. Introdução a Teste de Software. *Revista Engenharia de Software*. 2007. Disponível em: <<http://www.devmedia.com.br/artigo-engenharia-de-software-introducao-a-teste-de-software/8035#ixzz2AgtFLEZ7>>. Acesso em: 15 out. 2018.

FINANCIAL CRYPTOGRAPHY. *Is Bitcoin a triple entry system?*. 2011. Disponível em: <<http://financialcryptography.com/mt/archives/001325.html>>. Acesso em: 10 out. 2018.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to forget: Continual prediction with LSTM. *Neural Computation*, EUA, v. 12, n. 10, p. 2451–2471, jan. 2000.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learnin*. USA: Press MIT, 2016.

GREAVES, A.; AU, B. *Using the Bitcoin Transaction Graph to Predict the Price of Bitcoin*. p. 1-8, dez. 2015.

HAFIDZ, Z. *Understanding Learning Rates and How It Improves Performance in Deep Learning*. Disponível em: <<https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10>>. Acesso em: 12 out. 2018.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. *Neural Computation*, USA, v. 9, n. 8, p.1735-1780, nov. 1997

HOLLMEN, J. *Learning rate*. Disponível em: <<https://users.ics.aalto.fi/jhollmen/dippa/node22.html>>. Acesso em: 12 out. 2018.

INVESTOPEDIA. *Algorithmic Trading*. Disponível em: <<http://www.investopedia.com/terms/a/algorithmictrading.asp>>. Acesso em: 05 ago. 2018a

INVESTOPEDIA. *Circuit Breaker*. Disponível em: <<http://www.investopedia.com/terms/c/circuitbreaker.asp>>. Acesso em: 29 ago. 2018b.

INVESTOPEDIA. *How to trade for a living*. Disponível em: <<https://www.investopedia.com/articles/trading/09/how-to-trade-for-a-living.asp>>. Acesso em: 15 out. 2018c.

MAZUMDAR, J.; HARLEY, R. Recurrent Neural Networks Trained With Backpropagation Through Time Algorithm to Estimate Nonlinear Load Harmonic Currents. *IEEE Transactions on Industrial Electronics*, v. 55, n. 9, p. 3484-3491, set. 2008.

MINING. In: Bitcoin Wiki. Disponível em: <<https://en.bitcoin.it/wiki/Mining>>. Acesso em:

NAKAMOTO, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*. p. 1-9, 2009. Disponível em: <<https://bitcoin.org/bitcoin.pdf>>. Acesso em: 01 out. 2018.

OBER, M., KATZENBEISSER, S.; HAMACHER, K. Structure and Anonymity of the Bitcoin Transaction Graph. *Future Internet*, Suíça, v. 5, n. 2, p. 237–250, 2013.

PRADO, H. A. et al. On the effectiveness of candlestick chart analysis for the Brazilian stock Market. *ScienceDirect*, v. 22, p. 1136-1145, 2013.

SUBRAHMANYAM, A. Algorithmic trading, the Flash Crash, and coordinated circuit breakers. *Borsa Istanbul Review*, EUA, v. 13, n. 3, p. 4–9, set. 2013.

SAK, H.; SENIOR, A.; BEAUFAYS, F. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling. *Interspeech*, EUA, p. 338-342, 2014.

SHAH, D.; ZHANG, K. Bayesian regression and Bitcoin. *Massachusetts Institute of Technology*. v. 1, p. 1-6, mar. 2014.

SKYMIND. A Beginner's Guide to LSTMs. Disponível em: <<https://skymind.ai/wiki/lstm#long>>. Acesso em: 13 out. 2018.

TAFFLER, R. *Emotional Finance: Theory and Application*. 2014.

TALLEC, C.; OLLIVIER, Y. Unbiasing Truncated Backpropagation Through Time. *Cornell University Library*, v. 23, p. 1-13, mai. 2017.

TECHOPEDIA. *What is a Cryptocurrency?*. Disponível em: <http://www.techopedia.com/definition/27531/cryptocurrency>. Acesso em: 13 out. 2018.

THE ECONOMIST. *Flash Crash 2010*. Disponível em: <[http://www.economist.com/blogs/newsbook/2010/10/what\\_caused\\_flash\\_crash](http://www.economist.com/blogs/newsbook/2010/10/what_caused_flash_crash)>. Acesso em: 15 out. 2018.

WIJEWARDHANA. *Recurrent Neural Networks Vs Feed Forward Networks*. 2016. Disponível em: <<https://thamaliwijewardhana.wordpress.com/2016/03/25/recurrent-neural-networks-vs-feed-forward-networks/>>. Acesso em: 17 out. 2018.

ZULKIFLI, H. *Understanding Learning Rates and How It Improves Performance in Deep Learning*. Disponível em: <<https://towardsdatascience.com/understanding-learning-rates-and-how-it-improves-performance-in-deep-learning-d0d4059c1c10>>. Acesso em: 12 out. 2018.