



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

RAQUEL RUFINO COSTA DA PAZ

**UTILIZANDO UM ALGORITMO LSTM PARA PREVISÃO DO
PREÇO DE UMA AÇÃO**

CAMPINA GRANDE - PB

2019

RAQUEL RUFINO COSTA DA PAZ

**UTILIZANDO UM ALGORITMO LSTM PARA PREVISÃO DO
PREÇO DE UMA AÇÃO**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharela em
Ciência da Computação.**

Orientador: Professor Dr. Rohit Gheyi.

CAMPINA GRANDE - PB

2019



P348u Paz, Raquel Rufino Costa da.
Utilizando um algoritmo LSTM para previsão do preço
de uma ação. / Raquel Rufino Costa da Paz. - 2019.

11 f.

Orientador: Prof. Dr. Rohit Gheyi.

Trabalho de Conclusão de Curso - Artigo (Curso de
Bacharelado em Ciência da Computação) - Universidade
Federal de Campina Grande; Centro de Engenharia Elétrica
e Informática.

1. Rede neural recorrente. 2. Bolsa de valores. 3.
Memória de longo prazo. 4. Algoritmo LSTM. 5. Mercado
financeiro. 6. Redes LSTM. I. Gheyi, Rohit. II. Título.

CDU:004(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

RAQUEL RUFINO COSTA DA PAZ

**UTILIZANDO UM ALGORITMO LSTM PARA PREVISÃO DO
PREÇO DE UMA AÇÃO**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharela em
Ciência da Computação.**

BANCA EXAMINADORA:

**Professor Dr. Rohit Gheyi
Orientador – UASC/CEEI/UFCG**

**Professor Dr. Elmar Uwe Kurt Melcher
Examinador – UASC/CEEI/UFCG**

**Professor Dr. Tiago Lima Massoni
Professor da Disciplina TCC – UASC/CEEI/UFCG**

Trabalho aprovado em: 25 de novembro 2019.

CAMPINA GRANDE - PB

Utilizando o algoritmo LSTM para previsão do preço de uma ação

Trabalho de Conclusão de Curso

Raquel Rufino Costa da Paz

Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Brasil
raquel.paz@ccc.ufcg.edu.br

Orientador: Rohit Gheyi, PhD

Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Brasil
rohit@dsc.ufcg.edu.br

9 de Dezembro de 2019

Resumo

A previsão do valor das ações desempenha um papel importante na definição de uma estratégia de negociação ou na determinação do momento apropriado para comprar ou vender. Na medida que a tecnologia avança, ajuda os analistas a descobrir os indicadores mais relevantes para fazer uma melhor suposição. De fato, os investidores estão muito interessados na área de pesquisa de previsão de preço das ações. Para um investimento bom e bem-sucedido, os acionistas desejam saber a situação futura do mercado de ações. Nesta perspectiva, este trabalho propõe uma abordagem de rede neural recorrente (RNN) e Memória de Longo Prazo (LSTM) para prever índices do mercado, que combina recursos aprendidos de diferentes representações dos mesmos dados. O modelo proposto de rede neural é treinado com os preços diário das ações, elementos que incluem valores importantes chamados Aberto, Alto, Baixo e Fechado, em um dataset que possui aproximadamente 55 anos de dados de preços das ações da GE nos Estados Unidos. Os resultados obtidos são promissores, atingindo uma média de erro absoluto de 0.07 ao prever o preço da ação no futuro. Essas informações devem ser muito úteis para serem usados no escritório da bolsa de valores.

Palavras-chave - Previsão, Bolsa de Valores, Rede Neural, Tecnologia.

⁰Os autores retêm os direitos, ao abrigo de uma licença Creative Commons Atribuição CC BY, sobre todo o conteúdo deste artigo (incluindo todos os elementos que possam conter, tais como figuras, desenhos, tabelas), bem como sobre todos os materiais produzidos pelos autores que estejam relacionados ao trabalho relatado e que estejam referenciados no artigo (tais como códigos fonte e bases de dados). Essa licença permite que outros distribuam, adaptem e evoluam seu trabalho, mesmo comercialmente, desde que os autores sejam creditados pela criação original.

1 Introdução

Desde os primórdios da sociedade, analisar informações do passado é uma das principais fontes de sabedoria para os maiores líderes e sábios da humanidade, porque existe uma tendência de que as ações apresentem características similares. A previsão do valor de mercado é de grande importância para ajudar a maximizar o lucro da compra de opções de ações, mantendo o risco baixo. No entanto, a complexidade e o dinamismo das bolsas de valores torna a previsão de comportamento dos preços do mercado de ações uma tarefa difícil. Existem pesquisadores de diversas áreas que buscam tentar resolver esse problema, apresentando uma variedade de abordagens para encontrar uma solução.

Uma dessas abordagens é a análise preditiva que possibilita a antecipação de resultados e ações por meio da união do avanço tecnológico, que inclui técnicas como mineração de dados, modelagem, inteligência artificial, algoritmos de cálculos estatísticos, e machine learning, com um elevado volume de informações, chamado de datasets. Nessa perspectiva, a fim de explicar, imagine que um gerente de um determinado e commerce analisando as informações dos clientes da loja, percebe que 78% dos cancelamentos são feitos por pessoas que compraram pela segunda vez, ou seja, a probabilidade de um usuário cancelar a compra do produto da loja é muito maior quando ele comprar pela segunda vez. Agora há uma tendência clara e se pode agir mais efetivamente para evitar os cancelamentos.

Este trabalho estuda um método específico usando as redes neurais recorrentes (RNN) que provaram ser um dos modelos mais poderosos para o processamento de dados seqüenciais. Essas redes têm capacidade de memória de curto prazo e a suposição a ser analisada é que esse meio pode mostrar ganhos em termos de resultados, quando comparado a outras abordagens mais tradicionais no campo de aprendizado de máquina. O algoritmo escolhido foi o LSTM, Long Short Term Memory, que apresenta a célula de memória, uma unidade de computação que substitui os neurônios artifi-

ciais tradicionais na camada oculta da rede. Com essas células de memória, as redes são capazes de associar efetivamente memórias e entradas remotas no tempo.

O objetivo deste projeto é estudar a aplicabilidade de redes neurais atuais, particularmente, as redes LSTM e avaliar o problema de previsão de movimentos dos preços do mercado de ações. Dados históricos dos preços das ações da GE serão usados como fonte de informação para a rede para fins de treinamento e validação do modelo.

Dessa forma, este trabalho foi organizado em 5 sessões, o presente artigo faz uma proposta de modelagem para previsão do valor de compra do usuário em um e commerce. A pesquisa encontra-se dividida nas seguintes seções: seção 1, introdução; segue a seção 2 com uma visão geral da fundamentação teórica que estão na base deste trabalho, tanto no contexto de mercados de ações quanto no de machine learning, a seção 3 com os detalhes do modelo proposto; a seção 4 com os resultados apresentados e discutidos; a seção 5 contém a conclusão que oferece uma relação dos resultados atingidos com os pretendidos e apresenta as considerações finais e perspectivas de trabalhos futuros.

2 Fundamentação Teórica

O mercado financeiro[1] é um ambiente que investidores podem fazer negócios de bens entre eles mesmo. Essas negociações podem ter valores mobiliários (ações, opções, títulos), câmbio (moedas estrangeiras) e mercadorias (ouro, produtos agrícolas). Nesse mercado, o investidor é a pessoa que dispõe de dinheiro sobrando e deseja multiplicá-lo, e as ações representam o direito à propriedade e lucros da empresa. O valor de uma instituição é determinado pela oferta e demanda, e pode ser considerado um reflexo das condições estruturais, de mercado e financeiras da empresa.

Quando se estuda a respeito da previsão dos valores das ações requer uma atenção maior quanto às variáveis que podem influenciar o preço, dado sua complexidade e instabilidade, no qual dificultam a predição. Existem

abordagens populares feitas em relação ao assunto que são basicamente divididas em duas categorias, a Análise Fundamental e a Análise Técnica. Considerando informações diversas como taxas de juros, preço/lucro, indicadores de mercado e parâmetros do setor onde a empresa empreende formam a base da Análise Fundamental, a partir desses aspectos ela procurar descobrir o valor da empresa e o compra com o preço de mercado, para saber se a instituição está na média do que seria o valor, hipoteticamente, correto.

A análise técnica[2] é definida como um método de previsão de tendências baseada em preços passados, volumes negociados em um determinado período de tempo, histórico de dados, ou seja, ela desconsidera quaisquer fatores externos, como políticos ou sociais. Com essas informações ela cria indicadores técnicos para achar vertentes ou pontos em que os preços se desviaram demais de suas médias, existe uma vários indicadores e com diferentes intuítos, como intensidade de uma tendência, possíveis reversões ou a direção mais provável do movimento de preço.

Redes Neurais Artificiais[3] são técnicas computacionais que apresentam um modelo matemático inspirado em uma estrutura de um cérebro biológico. De forma que os neurônios são equivalentes a uma estrutura que possui unidades e as reações químicas que são responsáveis pelo fluxo de informações entre os neurônios são representadas por funções de ativação. Cada unidade de uma rede neural possui um peso, e é dado através do treinamento, onde vários exemplos são fornecidos à rede e a mesma aprende os melhores pesos em um processo de otimização, tentando generalizar e errar o menos possível quando sujeito a exemplos desconhecidos.

Série temporal[4] é uma coleção de observações feitas sequencialmente ao longo do tempo. As séries temporais foram utilizadas nesse trabalho com redes neurais recorrentes para identificar padrões não aleatórios. Redes recorrentes possuem loops entre suas unidades, permitindo armazenar informações ao processar novas entradas. Ela é uma classe de rede neural artificial projetada para reconhe-

cer padrões em sequências de dados que emanam de sensores, bolsas de valores e agências governamentais. Esses algoritmos consideram tempo e sequência, eles têm uma dimensão temporal, ou seja, a informação não deriva com sentido único, e a saída da rede não depende mais apenas da entrada corrente, mas também das entradas anteriores.

Redes LSTM, Long Short-Term Memory (Figura 1), são um modelo profundo e recorrente de redes neurais. O LSTM foi introduzido por Hochreiter[5] e Schmidhuber, e tinha como objetivo um melhor desempenho, abordando a questão do gradiente de fuga que as redes recorrentes sofrem ao lidar com longas sequências de dados. Para isso elas fazem o fluxo do erro contínuo por de unidades especiais chamadas “portões”, que permitem ajustes de pesos e o truncamento do gradiente quando suas informações são dispensáveis, simbolizando um esquecimento.

Há vários trabalhos relacionados ao uso da inteligência artificial na previsão de preços de ações na bolsa valores [6], usando os mais diversos algoritmos, como Algoritmos Genéticos [Allen Karjalainen, 1999] [7] e Máquina de Vetor de Suporte (SVM) [jae Kim, 2003] [8], além de redes neurais que serão objeto de estudo neste trabalho. Chen et al. [2015] [9] emprega dados históricos de preço juntamente a dados de índices de ações para prever se o preço de uma ação vai subir, descer ou permanecer o mesmo em determinado dia. Luca Di Persio [2016] [6] compara os desempenhos de LSTM e MLP com o seu 15 próprio método que é proposto baseando em uma combinação de wavelets e redes neurais convolutivas que é capaz de superar ambos, contudo apresentado resultados similares aos da rede LSTM.

3 Metodologia e Desenvolvimento

Com o objetivo de realizar predição do valor de uma determinada ação da bolsa de valores, foi proposto utilizar a análise técnica baseada em dados históricos de preço e volume dessa ação para prever o valor por meio de re-

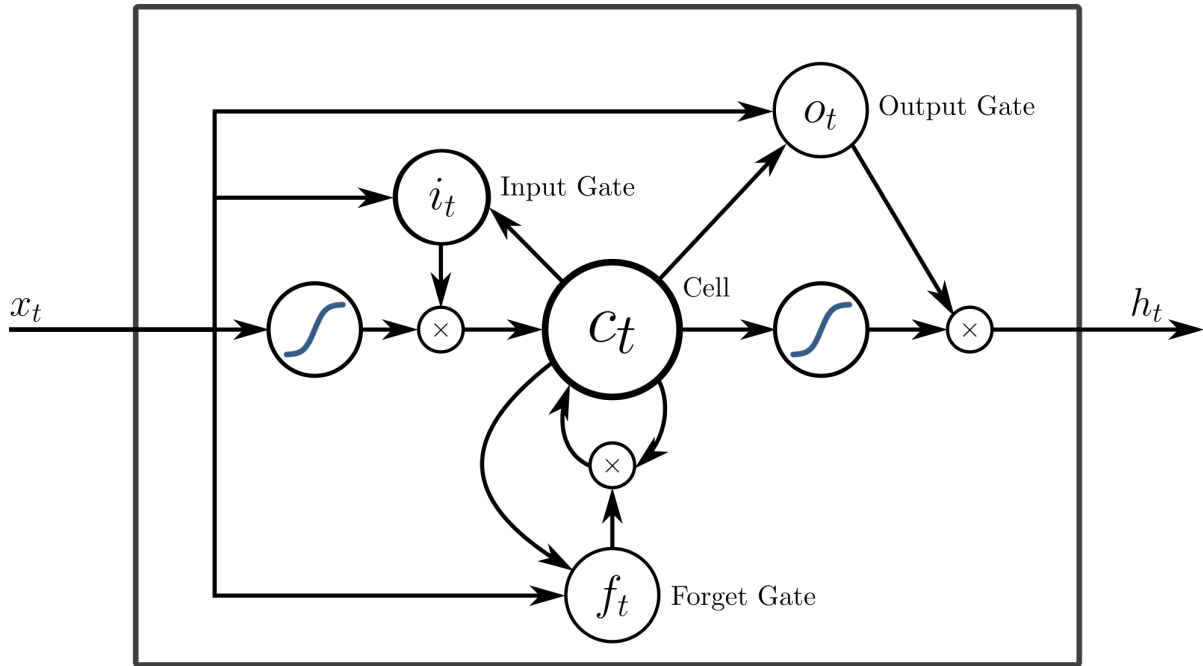


Figura 1: Redes Long short-term memory [Greff et al., 2015]

des neurais LSTM. Foi apresentado um modelo de classificação com base nas redes LSTM, e com os dados históricos de preço da empresa, o modelo será treinando para determinar o valor da ação no futuro.

Para a modelagem do problema, foi encontrada uma base de dados de tamanho considerável para ser realizado o estudo. Foi utilizado dados históricos dos preços das ações da GE, General Electric Company, no qual foi feito uma conversão em formato de uma série temporal em granularidade de um dia, ou seja, a base de dados possui dados de preços consolidados a cada um dia,. O dataset utilizado possui informações de 1962 a 2017 com informações de preço (abertura, fechamento, máximo, mínimo, volume), Figura 2, isto é, a granularidade escolhida, é o valor do preço no início do período, com o valor no final do período, além dos valores máximo e mínimo que o preço alcançou dentro daquele intervalo. Além disso, é coletado dados de volume de negociações, também com variação de um dia. Foi utilizado as bibliotecas NumPy[10] para computação científica, o Matplotlib para plotagem de gráficos e o Pandas[11] para auxiliar no carregamento e manipulação de nossos con-

juntos de dados. Com essas informações é realizado um pré-processamento, e é gerado um conjunto de indicadores técnicos que em conjunto com os dados originais serão utilizados como entrada para a rede neural [11].

A Open coluna é o preço inicial, enquanto a Close coluna é o preço final de uma ação em um determinado dia de negociação. As colunas High e Low representam os preços mais alto e mais baixo para um determinado dia.

Antes de adicionar os dados na rede neural, é preciso fazer a normalização para ajudar encontrar o mínimo local ou global com eficiência. É importante fazer esse procedimento, porque evita que o algoritmo tome decisões erradas devido ao grande intervalo de valores das variáveis utilizadas. Foi utilizado a função MinMaxScaler da biblioteca sklearn, que tem como técnica subtrair o valor em questão pelo menor valor da coluna e então divide pela diferença entre o valor máximo e mínimo, abaixo é possível a fórmula para calcular o valor mencionado. Além disso, foi feita uma exclusão nas linhas que estavam com o valor Null (sem valor).

$$(valor - Coluna.min) / (Coluna.max - Coluna.min)$$

Os dados gerados na etapa anterior, foram

	Date	Open	High	Low	Close	Volume
0	1962-01-02	0.6277	0.6362	0.6201	0.6201	2575579
1	1962-01-03	0.6201	0.6201	0.6122	0.6201	1764749
2	1962-01-04	0.6201	0.6201	0.6037	0.6122	2194010
3	1962-01-05	0.6122	0.6122	0.5798	0.5957	3255244
4	1962-01-08	0.5957	0.5957	0.5716	0.5957	3696430

Figura 2: Informações da ações

separados em subconjuntos de treinamento e validação, que tem como objetivo analisar a eficiência de generalização de cada modelo, apresentando dados para treinamento separando os dados de teste.

A próxima etapa é a criação do modelo que é uma variação de um rede neural recorrente. O objetivo desse modelo é que ele seja capaz de aprender quais informações são mais relevantes daquele conjunto de indicadores técnicos, e usá-la posteriormente para melhorar a capacidade de previsão. É composto por uma camada de entrada sequencial seguida por camadas LSTM, entre as camadas haverá uma técnica de regularização conhecida como dropout, que faz com que neurônios sejam excluídos aleatoriamente do processo de treinamento para evitar sobreadaptação (overfitting) no modelo de previsão, aumentando a capacidade de generalização.

Na extração de recursos, apenas os recursos que serão alimentados à rede neural que foram: data, alto, baixo, abertura, fechamento e volume. Após isso os dados são enviados para a rede neural e treinados para predição de atribuição de vieses e pesos aleatórios. A rede LSTM tem como entrada o tamanho da amostra, onde representa quantidade de amostras de entrada deseja que a rede neural veja antes de atualizar os pesos, etapas de tempo, que definem quantas unidades no tempo se de-

seja que a rede veja, e o número de recursos de atributos usados para representar cada etapa do tempo. Esses parâmetros foram definidos através de testes empíricos e escolhidos avaliando a qualidade dos resultados e custo de execução. Para construir o LSTM, foi importado alguns módulos do Keras :

1. Sequential para inicializar a rede neural.
2. Dense para adicionar uma camada de rede neural densamente conectada.
3. LSTM para adicionar a camada de memória de longo prazo.
4. Dropout para adicionar camadas de eliminação que impedem o ajuste excessivo.

Foi adicionado algumas Dropout camadas para evitar o ajuste excessivo. A camada LSTM tem os seguintes argumentos:

1. 100, que é a dimensionalidade do espaço de saída.
2. `return_sequences=True` que determina se deve retornar a última saída na sequência de saída ou a sequência completa.
3. `input_shapes` como forma de conjunto de treinamento.

Ao definir as Dropout camadas, foi especificado 0,4, o que significa que 40% das camadas serão descartadas. Depois disso tem a Dense camada que especifica a saída de 1 unidade. Depois disso, compila o modelo usando o popular RMSprop, que é um método utiliza a média decadente exponencial, buscando minimizar a entropia cruzada da rede, e defini a perda como mean squarred error. Isso calcula a média dos erros ao quadrado. Abaixo é mostrado o código da função que cria o modelo:

```
from keras.models import Sequential, load_model
from keras.layers import Dense, Dropout
from keras.layers import LSTM
from keras import optimizers

def create_model():
    lstm_model = Sequential()
    lstm_model.add(
        LSTM(
            100,
            batch_input_shape=(
                BATCH_SIZE,
                TIME_STEPS,
                x_t.shape[2]),
            dropout=0.0,
            recurrent_dropout=0.0,
            stateful=True,
            return_sequences=True,
            'random_uniform'))
    lstm_model.add(Dropout(0.4))
    lstm_model.add(LSTM(60, dropout=0.0))
    lstm_model.add(Dense(20, a'relu'))
    lstm_model.add(Dense(1, 'sigmoid'))
    optimizer = optimizers.RMSprop(lr=LR)
    lstm_model.compile(
        loss='mean_squared_error',
        optimizer=optimizer)
    return lstm_model
```

4 Resultados e Discussões

Foram coletadas 5 métricas de regressão utilizando o módulo Metrics do scikit-learn, no qual implementa várias funções de perda, pontuação e utilidade para medir o desempenho da regressão. Alguns dos que foram aprimorados para lidar com o caso de múltiplas

saídas: mean squared error, mean absolute error, explained variance, scoree r2 score. Essas funções têm um multi output argumento de palavra - chave que especifica a maneira como as pontuações ou perdas de cada alvo individual devem ser calculadas. O padrão é 'uniforme average', que especifica uma média ponderada uniformemente sobre as saídas. Se uma ndarray forma (n outputs) for passada, suas entradas serão interpretadas como pesos e uma média ponderada correspondente será retornada. Se multi output for 'raw values especificado, todas as pontuações ou perdas individuais inalteradas serão retornadas em uma matriz de forma (n outputs).

O r2 scoree explained variance score aceita um valor adicional 'variance weighted'para o multi output parâmetro. Essa opção leva a uma ponderação de cada pontuação individual pela variação da variável alvo correspondente. Essa configuração quantifica a variação não escalada capturada globalmente. Se as variáveis-alvo são de escala diferente, essa pontuação coloca mais importância na explicação correta das variáveis de maior variação. Multi output='variance weighted' é o valor padrão r2 score para compatibilidade com versões anteriores. Isso será alterado para uniform average no futuro. Na Tabela 1 é mostrado os resultados de cada métrica mencionada.

Foi simulado qual o retorno financeiro relativo ao tempo que é trabalhado no dataset. Os resultados foram comparados com base nos dados reais que pode ser visto na Figura 3.

Neste caso, o modelo com a rede LSTM tem resultados melhores entre os dias 1000 e 1400, onde apresenta uma pequena diferença entre os dados reais e o da predição. O fato de estar com resultados positivos na maior parte é um bom indicativo de baixo risco. O ideal é que esse valor seja o menor possível em termos absolutos, pois quanto maior, maior será o risco ao qual o investidor está exposto.

5 Conclusão

Após o estudo, constatou-se que o modelo criado apresentou uma média de erro absoluto de 0.07, que é considerado satisfatório, dado

Tabela 1: Métricas Coletadas.

Métrica	Resultado
Mean Squared Error	0.0069
Explained Variance Score	0.2961
Mean Absolute Error	0.0707
Mean Squared Log Error	0.0031
R2 Score	-0.014

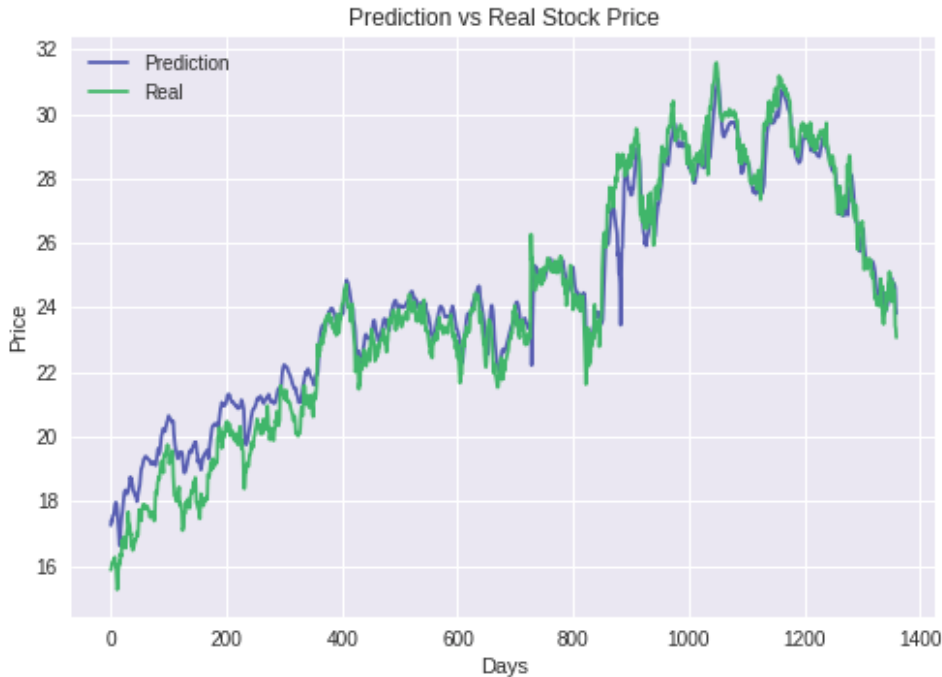


Figura 3: Comparação dos resultados do modelo com os dados reais

o grau de aleatoriedade que representa o valor de uma ação na bolsa de valores, influenciada por inúmeros fatores difíceis de se perceber. Embora o modelo proposto não apresente a melhor eficiência, em razão da existência de modelos mais sofisticados e eficazes nesse contexto de predição, acredita-se que modelos que se utilizam de ML e toda a sua matemática envolvida para gerar previsões, possuem grande potencial para seguirem otimizados e atingir uma eficiência ainda mais próxima da perfeição, desde que as entradas que alimentam o modelo sejam de fato as medidas de eficácia operacional do problema.

Com isso, sugere-se para próximos traba-

lhos, avaliar outros algoritmos de predição, como por exemplo, sugere-se combinar os resultados de entradas para o LSTM com outros métodos de previsão, como MLP, Multilayer Perceptron, e NARX, Nonlinear Autoregressive Network With Exogenous Inputs, assim como, utilizar outras técnicas de combinação linear mais avançadas para apoiar em seu resultado. Além disso, recomenda-se pesquisar outras métricas que permitam avaliar a direção das previsões e erros.

Referências

- [1] E. Fortuna, *Mercado financeiro: produtos e serviços*. Qualitymark Editora Ltda, 2008.

- [2] C. D. Kirkpatrick II and J. A. Dahlquist, *Technical analysis: the complete resource for financial market technicians*. FT press, 2010.
- [3] R. Adhikari and R. Agrawal, “A combination of artificial neural network and random walk models for financial time series forecasting,” *Neural Computing and Applications*, vol. 24, no. 6, pp. 1441–1449, 2014.
- [4] R. S. Ehlers, “Análise de séries temporais,” *Universidade Federal do Paraná*, 2007.
- [5] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [6] L. Di Persio and O. Honchar, “Artificial neural networks approach to the forecast of stock market price movements,” *International Journal of Economics and Management Systems*, vol. 1, 2016.
- [7] F. Allen and R. Karjalainen, “Using genetic algorithms to find technical trading rules,” *Journal of financial Economics*, vol. 51, no. 2, pp. 245–271, 1999.
- [8] K.-j. Kim, “Financial time series forecasting using support vector machines,” *Neurocomputing*, vol. 55, no. 1-2, pp. 307–319, 2003.
- [9] C.-H. Cheng, T.-L. Chen, H. J. Teoh, and C.-H. Chiang, “Fuzzy time-series based on adaptive expectation model for taiex forecasting,” *Expert systems with applications*, vol. 34, no. 2, pp. 1126–1132, 2008.
- [10] W. McKinney, *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. ”O’Reilly Media, Inc.”, 2012.
- [11] F. M. Raupp and I. M. Beuren, “Metodologia da pesquisa aplicável às ciências,” *Como elaborar trabalhos monográficos em contabilidade: teoria e prática. São Paulo: Atlas*, 2006.