
Aplicando redes neurais na predição de valores da moeda Bitcoin

João Vitor Pereira



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Monte Carmelo - MG
2022

João Vitor Pereira

**Aplicando redes neurais na predição de valores
da moeda Bitcoin**

Trabalho de Conclusão de Curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, Minas Gerais, como requisito exigido parcial à obtenção do grau de Bacharel em Sistemas de Informação.

Área de concentração: Redes Neurais Artificiais

Orientador: Profa. Dra. Fernanda M. da C. Santos

Monte Carmelo - MG

2022

Este trabalho é dedicado a todos aqueles que um dia sonharão em fazer a diferença na sociedade.

Agradecimentos

Agradeço aos meus pais por todo o suporte e carinho dado na minha caminhada, além de amigos e familiares que contribuíram com seu amor e amizade. Por fim, não posso deixar de agradecer a Profa. Dra. Fernanda M. da C. Santos por toda paciência e suporte me dado, sendo tal gratidão levada pelo resto da minha vida.

*“Apesar de todo o desespero, haverá sempre esperança. A vida é muito preciosa e bonita
- e ninguém deve desperdiçá-la.”
(Schloss, Eva - Depois de Auschwitz)*

Resumo

As Redes Neurais Recorrentes aplicadas à séries de dados temporais do mercado financeiro define-se por mais uma técnica de aprendizado de máquina promissora na previsão da movimentação das criptomoedas. A base de dados foi obtida por informações reais do mercado financeiro da criptomoeda Bitcoin. Assim, definiu-se modelos computacionais formados pelas redes neurais recorrentes *Long short-term memory (LSTM)* e *Multilayer Perceptron* e pelo *K Nearest Neighbor (KNN)*, obtendo melhores resultados a rede neural recorrente com 72% de acurácia.

Palavras-chave: Bitcoin, Redes neurais artificiais, Apendizado de máquinas, redes neurais recorrentes, LSTM.

Lista de ilustrações

Figura 1 – Modelo não-linear de um neurônio.	16
Figura 2 – Arquitetura de uma rede neural Perceptron Multicamadas (PMC). . .	17
Figura 3 – Comparação Rede Recorrente e Não Recorrente.	18
Figura 4 – Estrutura da LSTM.	18
Figura 5 – Time Series Forecasting.	19
Figura 6 – Representação do algoritmo KNN.	20
Figura 7 – KNN Classificado	20
Figura 8 – Etapas do modelo computacional para predição dos valores da cripto- moeda Bitcoin no mercado de ações.	22
Figura 9 – Exemplo de balanceamento de dados.	24
Figura 10 – Função de ativação Relu.	25
Figura 11 – Função de ativação Softmax.	25
Figura 12 – Representação da Matriz de Confusão.	27
Figura 13 – Conjunto de teste com 5% dos dados para o período de 24 horas em uma rede LSTM.	29
Figura 14 – Conjunto de teste com 20% dos dados para o período de 24 horas em uma rede LSTM.	29
Figura 15 – Conjunto de teste com 30% dos dados para o período de 24 horas em uma rede LSTM.	30
Figura 16 – Conjunto de teste com 5% dos dados para o período de 12 horas em uma rede LSTM.	31
Figura 17 – Conjunto de teste com 20% dos dados para o período de 12 horas em uma rede LSTM.	31
Figura 18 – Conjunto de teste com 30% dos dados para o período de 12 horas em uma rede LSTM.	32
Figura 19 – Conjunto de teste com 5% dos dados para o período de 6 horas em uma rede LSTM.	32

Figura 20 – Conjunto de teste com 20% dos dados para o período de 6 horas em uma rede LSTM.	33
Figura 21 – Conjunto de teste com 30% dos dados para o período de 6 horas em uma rede LSTM.	33
Figura 22 – Conjunto de teste com 5% dos dados para o período de 3 horas em uma rede LSTM.	34
Figura 23 – Conjunto de teste com 20% dos dados para o período de 3 horas em uma rede LSTM.	34
Figura 24 – Conjunto de teste 30% dos dados para o período de 3 horas em uma rede LSTM.	35
Figura 25 – Conjunto de teste 30% no período de 24 horas em uma rede MLP. . . .	35
Figura 26 – Resultado de medidas de avaliação do KNN sob o conjunto de teste de 5% da base de dados.	36
Figura 27 – Resultado de medidas de avaliação do KNN sob o conjunto de teste de 20% da base de dados	36

Lista de tabelas

Tabela 1 – Descrição dos atributos da base de dados.	23
Tabela 2 – Descrição dos targets de cada amostra.	23

Lista de siglas

AM Aprendizado de Máquinas

KNN K Nearest Neighbor

LSTM Long short-term memory

PMC Perceptron Multicamadas

RNA Rede Neural Artificial

RNAs Redes Neurais Artificiais

RNR Rede Neural Recorrente

RNRs Redes Neurais Recorrentes

SVM *Support Vector Machine*

Sumário

1	INTRODUÇÃO	12
1.1	Justificativa	13
1.2	Objetivos	14
1.3	Contribuições	14
1.4	Organização do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICA E REVISÃO BIBLIOGRÁFICA	15
2.1	Bitcoin	15
2.2	Redes Neurais Artificiais	15
2.3	Redes Neurais Recorrentes	17
2.3.1	LSTM	17
2.4	Times Series Forecasting	19
2.5	KNN	20
2.6	Trabalhos Relacionados	21
3	METODOLOGIA	22
3.1	Base de Dados	22
3.2	Técnicas de Pré-Processamento	23
3.3	Redes Neurais	24
3.4	Medida de Avaliação	26
4	EXPERIMENTOS E ANÁLISE DOS RESULTADOS	28
4.1	Experimentos	28
4.2	Avaliação dos Resultados	37
5	CONCLUSÃO	38
5.1	Principais Contribuições	38
5.2	Trabalhos Futuros	39

REFERÊNCIAS	40
-------------------	----

Introdução

As criptomoedas são um tipo de dinheiro, igual à outras moedas que são negociadas no mercado financeiro, porém totalmente digital. Elas podem ser usadas como meio de troca em transações comerciais, bem como ativos financeiros (ULRICH, 2017). As criptomoedas possuem características singulares de outras moedas, como não ter uma autoridade central que as administre e não estarem associadas a quaisquer ativos tangíveis (CORBET et al., 2019).

O mercado das criptomoedas, caracteriza-se por uma volatilidade alta, sem períodos de negociação fechados, uma capitalização menor e pela alta disponibilidade dos dados de mercado (FERREIRA et al., 2018). Assim, a predição do mercado de ações das criptomoedas é considerado uma tarefa desafiante e fonte de estudo tanto para a acadêmicos quanto para investidores (LAHMIRI; BEKIROS, 2018).

De acordo com pesquisas acadêmicas, os movimentos nos preços de mercado das criptomoedas não são aleatórios, mas se comportam de forma não linear e dinâmica (VALENCIA ALFONSO GÓMEZ ESPINOSA, 2019). Logo, o importante é prever a direção que o valor da moeda tomará no mercado financeiro com o objetivo de obter mais lucros.

Nesse contexto, diversos estudos estão empregando as técnicas de Aprendizado de Máquinas (AM) e da Inteligência Artificial pra predizer os cenários do mercado financeiro, principalmente, nas movimentações das criptomoedas. As principais técnicas de AM são: Redes Neurais Artificiais (RNAs) (VALENCIA ALFONSO GÓMEZ ESPINOSA, 2019; CHEN; LI; SUN, 2020), *Support Vector Machine* (SVMs) (AGGARWAL; CHANDRASEKARAN; ANNAMALAI, 2020; CHEN; LI; SUN, 2020) e *Random Forest* (RFs) (BASHER; SADORSKY, 2022; CHEN; LI; SUN, 2020).

Em (VALENCIA ALFONSO GÓMEZ ESPINOSA, 2019), os autores utilizaram a abordagem de análise de sentimentos aliado à dados reais do mercado financeiro para a predição do movimento do preço das criptomoedas *Bitcoin*, *Ethereum*, *Ripple* e *Litecoin*. A origem dos dados alusivos à análise de sentimentos foi a rede social Twitter. A aplicação das RNAs Perceptron Multicamadas na base de dados composta por dados do Twitter e do mercado das criptomoedas, obtiveram resultados superiores à outras técnicas de AM.

Neste trabalho, definiu-se um modelo com a aplicação da previsão de séries temporais (do inglês *Time Series Forecasting*) com AM com o intuito de analisar o potencial dessas ferramentas na predição dos valores das criptomoedas. As técnicas de AM empregadas foram as RNAs Recorrente *Long short-term memory*(LSTM) e Perceptron Multicamadas (PMC), além do *K Nearest Neighbor* (KNN). As Redes Neurais Recorrentes (RNRs) são altamente recomendadas para dados cujo comportamento caracterizam-se como os da séries temporais, principalmente por sua capacidade de backpropagation. Devido a isso será interessante analisar como a rede se comportará na classificação do preço do Bitcoin. Então, esse estudo irá focar exclusivamente na criptomoeda Bitcoin.

1.1 Justificativa

A utilização de redes neurais em aplicações na bolsa de valores e no mercado de ações já é bastante consolidada (WUERGES ARTUR FILIPE EWALD E BORBA, 2010). No entanto, com o surgimento das criptomoedas e, principalmente, do Bitcoin (NAKAMOTO, 2008) houve uma grande demanda em entender o funcionamento da moeda e, consequentemente, do novo mercado de *trading*, para produzir estudos capazes de explicar o seu comportamento, e seu impacto na sociedade atual.

Trading é a operação de compra e venda de ativos (ações) num curto prazo. No mercado de ações, há dois tipos mais comuns: o *swing trade* e o *day trading*. O primeiro corresponde às operações de compra e venda realizadas num intervalo de dias ou até semanas. Já o *day trading* refere-se a operações de prazo extremamente curto e que podem durar alguns minutos ou horas, mas que nunca ultrapassam o prazo de um dia.

Existe um grande mercado de *day trading* em moedas visando lucrar com as alterações do preço das mesmas. Porém, é bastante perigoso entrar nesse tipo de mercado sem estar preparado com estratégias adequadas, pois requer uma boa leitura e conhecimento do que está impactando financeiramente a moeda, para se preparar para compras ou vendas futuras.

Sabendo de todo contexto apresentado, destaca-se o estudo desenvolvido em Valencia Alfonso Gómez Espinosa (2019) que tem o objetivo de prever as movimentações das criptomoedas no mercado *day trading*, utilizando três técnicas de AM (rede neural PMC, *Support Vector Machine* (SVM) e *Random Forest* . A base de dados empregada neste trabalho foi constituída por informações oriundas do Mercado de Ações e da rede social Twitter. Os resultados demonstraram a possibilidade de prever a movimentação das criptomoedas pelas técnicas de AM juntamente com a análise de sentimentos.

Perante o exposto, propõe a utilização da rede neural recorrente aplicado à uma base de dados gerada por informações reais do mercado da criptomoeda Bitcoin num período de 24 horas. Essa abordagem é bastante interessante levando em consideração que o investidor do mercado de *day trading* terá como base o fluxo do preço no decorrer do dia,

e olhando tal fluxo a partir de gráficos, irá determinar se o preço vai ser maior ou menor do que ele está no momento.

1.2 Objetivos

O mercado de criptomedas é muito complexo e de difícil previsão, logo supõem à aplicação da rede neural recorrente LSTM sob dados reais do Bitcoin com o objetivo de verificar se o fluxo da recorrência e da aprendizagem por um maior reforço pode impactar diretamente em melhores resultados. Portanto, o objetivo deste trabalho é analisar como a LSTM atua neste cenário, em comparação a outras técnicas de aprendizado de máquina, como a rede neural PMC e KNN.

Desta forma, o modelo proposto visa prever o aumento ou a redução do valor do Bitcoin a partir de atributos diários da criptomoeda. Diante disso, metas específicas podem ser descritas:

- ❑ Criar uma base de dados com informações reais da criptomoeda Bitcoin;
- ❑ Treinar um modelo utilizando a rede neural recorrente LSTM;
- ❑ Treinar modelos utilizando a rede neural PMC e outro a KNN ;
- ❑ Formatar os dados para os modelos proposto como previsão de série temporal (*times series forecasting*);
- ❑ Comparar com os resultados gerados pelos modelos com os apresentados pelo trabalho de (VALENCIA ALFONSO GÓMEZ ESPINOSA, 2019).

1.3 Contribuições

A proposta deste trabalho vem auxiliar investidores e profissionais do mercado de *day trading*, suggestionando a compra e venda da criptomoeda Bitcoin de forma mais segura e rentável. Ademais, o reforço do uso de técnicas de AM no mercado financeiro para auxiliar na predição e classificação do cenário econômico.

1.4 Organização do Trabalho

O trabalho está organizado na sequência: o capítulo 2 apresenta resumidamente o embasamento teórico sobre o Bitcoin, as Redes Neurais Artificiais (RNAs) implementadas na proposta e o KNN; o capítulo 3 descreve todas as etapas do modelo computacional; o capítulo 4 mostra os experimentos realizados e os resultados alcançados pelo modelo computacional proposto. Por fim, o capítulo 5 destaca as conclusões.

Fundamentação Teórica

2.1 Bitcoin

Bitcoin é um protocolo de comunicação que facilita o uso de moedas virtuais (BÖHME et al., 2015) criado em 2008 por um grupo de desenvolvedores (NAKAMOTO, 2008). Estima-se que cerca de 25 milhões de pessoas ao redor do mundo possuam uma quantidade de Bitcoin, totalizando um valor aproximado de 186 bilhões de dólares (MATTHE et al., 2010). Isso leva a perceber a importância da moeda na sociedade atualmente, sendo em diversas situações uma forma de refúgio para países em situações de crises e hiperinflações, como por exemplo a Venezuela (FRANCO, 2017).

Devido ao fato de que não é necessário um órgão internacional que controle as transações da moeda, existe uma tecnologia responsável pela criação de uma base de registros, denominada de Blockchain que possui um histórico de todas as transações já realizadas (ULRICH, 2017). Esse histórico gera segurança e credibilidade no sentido de que não é possível fraudar dados nesta rede, pois cada usuário possui uma chave criptografada que é utilizada para assinar cada transação que este fizer dentro da rede. Então, todos os usuários pertencentes a rede vão receber uma cópia desta chave para assegurar que toda transação feita é segura (FRANCO, 2014).

2.2 Redes Neurais Artificiais

Uma Rede Neural Artificial (RNA) é uma máquina projetada e desenvolvida para ser capaz de simular e modelar a maneira que o cérebro humano realiza determinadas atividades (HAYKIN, 2007). Segundo Simon Haykin, um dos grandes cientistas da área, uma rede neural é um processador compacto que está distribuído paralelamente, constituído de unidades de processamento simples, os "neurônios". Estes são capazes de armazenar conhecimento experimental e disponibilizá-los para uso. A Figura 1 mostra o modelo de um neurônio artificial.

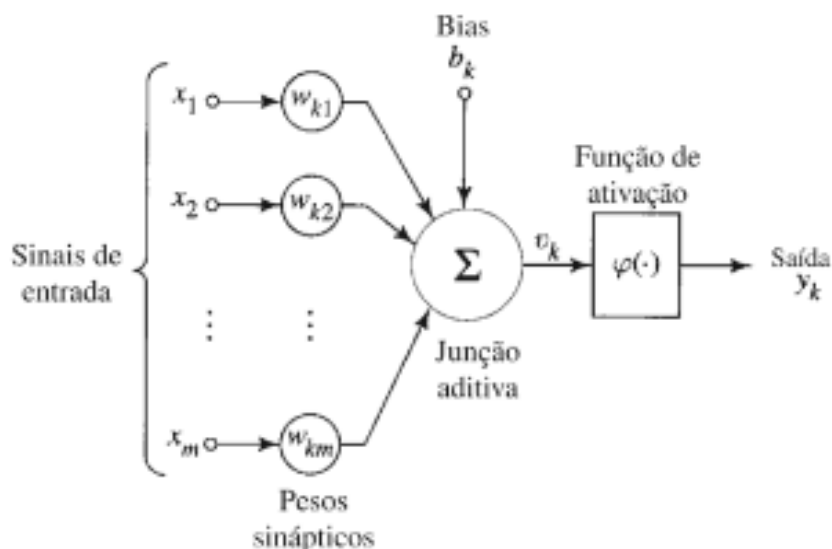


Figura 1 – Modelo não-linear de um neurônio.

Fonte: Obtida em Haykin (2007)

Esse processo é bastante semelhante ao que o cérebro humano faz, partindo da lógica que ele absorve conhecimento através da experiência e prática. Uma RNA no seu processo de treinamento e aprendizado possui duas características que se assemelham ao cérebro:

- Processo de aprendizagem: Por meio da experiência os neurônios serão capazes de absorver e coletar os dados dos resultados obtidos.
- Força de conexão entre os neurônios: Mais conhecida como os pesos sinápticos, eles são responsáveis por armazenar os conhecimentos obtidos.

Com o passar do tempo foi percebido que o cérebro humano possuía, por meio dos seus neurônios, a capacidade de aprendizado e de execução de atividades muito superior a qualquer tecnologia computacional.

No desenvolvimento deste trabalho será utilizada a arquitetura da rede neural denominada PMC, sendo este uma RNA que trabalha com uma camada de entrada, uma ou duas camadas ocultas e uma camada de saída. A Figura 2 exhibe a arquitetura de uma rede neural PMC.

Basicamente, o PMC trabalha com um algoritmo de retropropagação, ou seja, os dados de entrada se propagam pelas camadas neurais ocultas até a camada neural de saída, ativando os neurônios presentes nestas. Após obter o resultado pela camada neural de saída, a rede irá compará-lo com o resultado esperado. Esse cálculo informa o quanto a rede neural errou. Assim, para obter um melhor aprendizado tendendo a taxa de erro ao zero, recalcula-se os pesos sinápticos entre os neurônios, indo da camada neural de saída, passando pelas camadas neurais ocultas até a camada de entrada (HAYKIN, 2007).

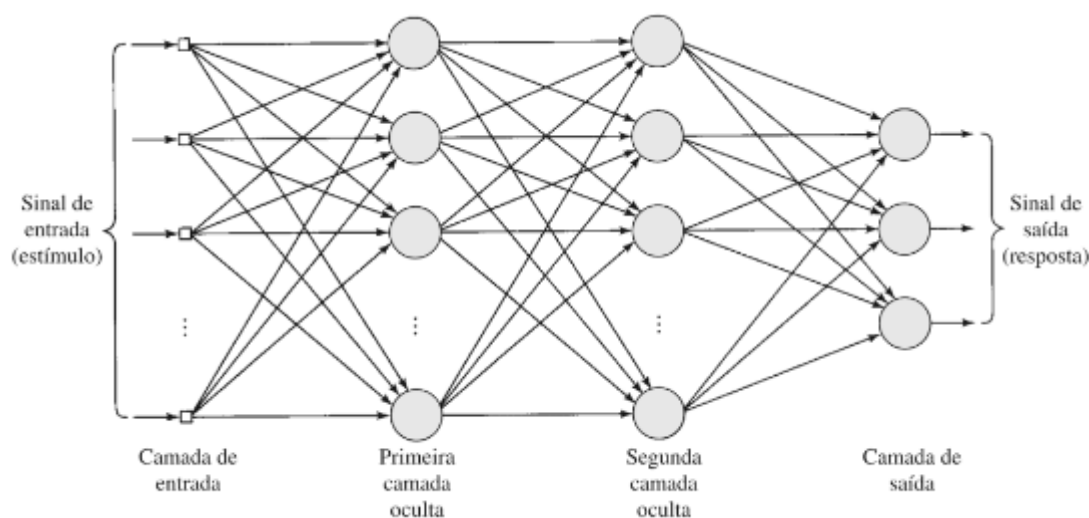


Figura 2 – Arquitetura de uma rede neural PMC.

Fonte: Obtida em Haykin (2007).

2.3 Redes Neurais Recorrentes

Uma rede neural recorrente é uma das arquiteturas de RNA que caracteriza-se pelas conexões ponderadas dentro de uma mesma camada. Em outras palavras, ela possibilita que uma rede possa exibir um comportamento dinâmico temporal, sendo isso de grande valia para trabalhos onde temos dados temporais, como por exemplo, o mercado de criptomedas e dados climáticos.

Como podemos visualizar na Figura 3, a rede neural recorrente permite que a camada de saída da rede possa voltar para uma camada anterior permitindo um maior reforço de aprendizagem. Segundo (HEWAMALAGE; BERGMEIR; BANDARA, 2021) as RNRs estão se tornando um método competitivo. Segundo esse estudo, as RNRs são bastante eficientes na modelagem de dados e séries temporais, porém reforçam que as mesmas possuem um longo caminho a percorrer ainda.

No trabalho de (KURUMATANI, 2020) podemos ver a aplicação de Redes Neurais Recorrentes para prever os preços futuros dos produtos agrícolas, sendo que o embasamento técnico para a utilização das RNRs, é o fato das mesmas serem uma tecnologia que possa se desempenhar melhor em problemas de séries temporais.

2.3.1 LSTM

A LSTM é uma Rede Neural Recorrente (RNR) muito utilizada em problemas com cenários de processamento de linguagem natural. Como já sabemos, as RNRs permitem uma maior *backpropagation* das informações, conseqüentemente ela é mais utilizada para problemas de séries temporais.

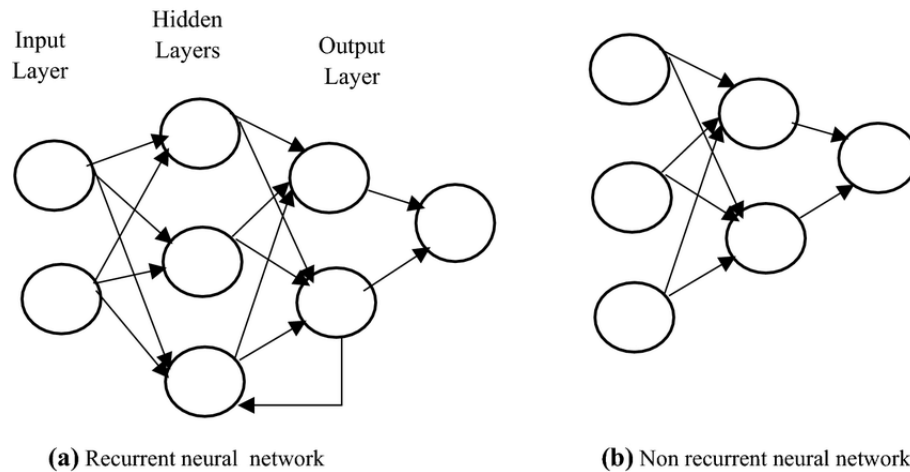


Figura 3 – Comparação Rede Recorrente e Não Recorrente.

Fonte: Obtida em Haykin (2007).

A rede LSTM possui uma arquitetura em forma de cadeia que contém quatro tipos de camadas neurais diferentes, sendo elas o *Input Gate*, *Input Modulation Gate*, *Forget Gate* e *Output Gate*. Além disso, ela contém blocos de memória que denominamos de células, como pode ser visto na Figura 4.

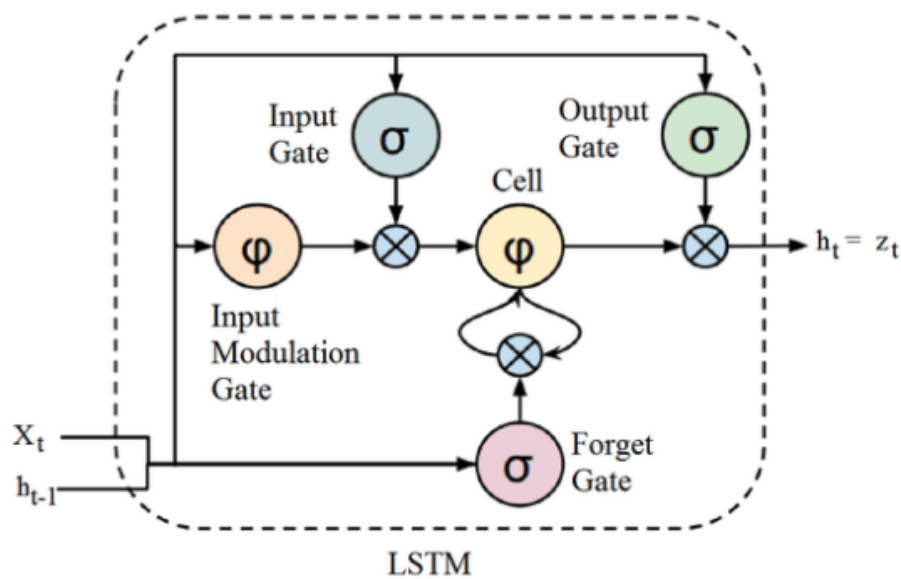


Figura 4 – Estrutura da LSTM.

Fonte: Academy. (2022)

Segundo o livro Academy. (2022), o *Input Gate* é responsável pela adição das informações úteis ao estado da célula, sendo que X_t é a entrada atual, enquanto que o h_{t-1} é a entrada anterior da célula. O *Forget Gate* é responsável por remover as informações que não são úteis para o estado da célula, que vão passar por uma função de ativação responsável por classificar a importância das informações para as futuras próximas entradas.

Por fim, o *Output Gate* é responsável por extrair informações úteis do estado da cédula atual processada pela rede, sendo essas repassadas para a próxima cédula.

O estado da cédula refere-se a questão da memória de longo prazo que é implementada na LSTM, sendo este modificado pelo *Forget Gate* quando o mesmo identifica informações que não são mais úteis.

2.4 Times Series Forecasting

Times Series Forecasting utiliza-se de dados históricos com o objetivo de fazer previsões futuras. Em outras palavras, trata-se de utilizar os dados já passados para construir modelos que possam fazer previsões futuras com base nas primeiras informações. A Figura 5 sugere a dedução de novos dados a partir de pontos iniciais. Essa técnica é bastante utilizada para modelos que trabalham com previsões de clima, tempo e mercado financeiro.

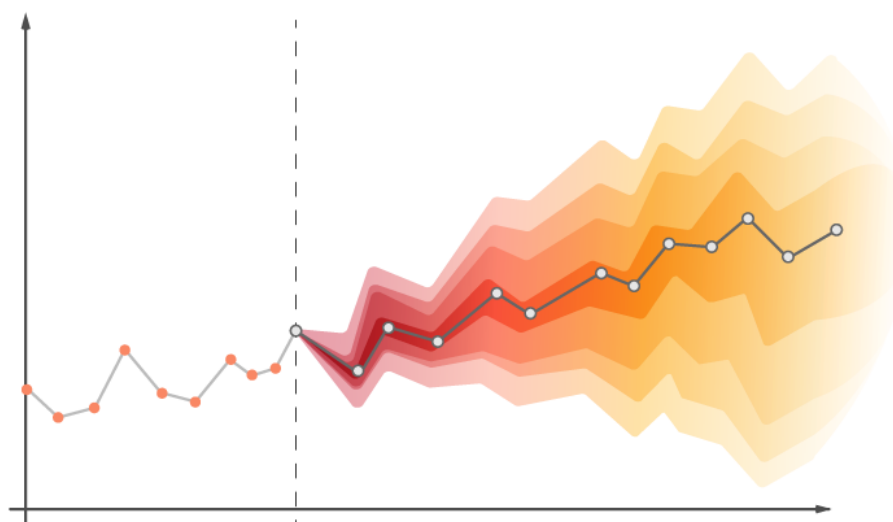


Figura 5 – Time Series Forecasting.

Fonte: Time Series Forecasting

A utilização de redes neurais recorrentes juntamente com a aplicação do *times series forecasting*, é algo que está crescendo bastante dentro das pesquisas acadêmicas, trabalhos como o de Livieris, Pintelas e Pintelas (2020), utilizaram-se da rede LSTM aplicadas diretamente em dados de séries temporais com o objetivo de criar um modelo de previsão da movimentação do preço do ouro.

A visão das séries temporais é utilizada em várias áreas de estudos, e na economia a mesma é utilizada a um longo tempo, principalmente após o trabalho de Trygve Haavelmo, que foi um economista norueguês, que começou a interpretar as séries temporais como realizações de processos aleatórios. Essa mesma interpretação pode ser dada à variação do preço do Bitcoin que pode variar no decorrer do dia em uma forma não padronizada.

2.5 KNN

O algoritmo do KNN, ou em português, K vizinhos mais próximos, é um algoritmo de aprendizado de máquina utilizado para classificação, onde o mesmo a partir de um dado procura encontrar os k-vizinhos mais próximos que ele tem, sendo que sua classe será determinada pela classe majoritária dos k-vizinhos próximos a ele.

Suponha o cenário da Figura 6. O ponto verde refere-se a classe que estamos classificando, enquanto que o círculo verde refere-se aos $k=3$ vizinhos mais próximos a ela.

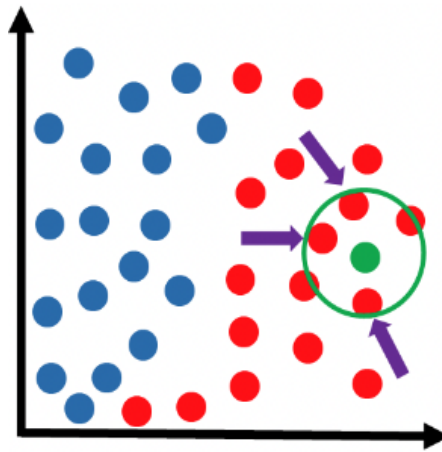


Figura 6 – Representação do algoritmo KNN.

Fonte: O que é e como funciona o algoritmo KNN?

Como pode-se verificar na Figura 7, a classe predominante é a vermelho, logo o ponto que está sendo classificado será vermelho. O valor de k possui seu valor na decisão dos resultados das classificações do modelo, pois quanto maior o valor, maior será o raio de observação e, conseqüentemente, ele vai sempre escolher a classe mais predominante neste raio.

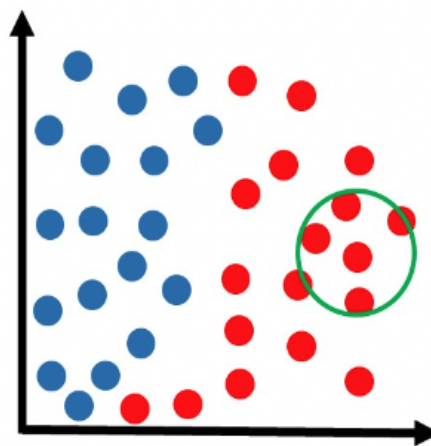


Figura 7 – KNN Classificado

Fonte: O que é e como funciona o algoritmo KNN?

Quanto menor for o valor k , mais sensível o algoritmo irá ficar para distâncias menores, ao contrário de quando o valor de k é muito alto, que nestes cenários o mesmo fica menos sujeito a ruídos.

2.6 Trabalhos Relacionados

No trabalho realizado por (FILHO, 2019), propõem a utilização de redes neurais juntamente com análise de sentimentos para prever a oscilação do preço de criptomoedas. Partiu-se da hipótese de que as emoções passadas pelas pessoas em mensagens de Twitter poderiam ter grande importância para a previsão do valor futuro da moeda, sendo isso de grande valia para profissionais que trabalham com a venda e compra de criptomoedas, nomeado por *traders*. A metodologia aplicada nesse trabalho foi o treinamento de modelos composto por redes neurais que atuaram em três diferentes bases de dados. São elas: dados do Twitter, outra com dados do mercado financeiro e a terceira formada pela junção das duas bases de dados anteriores. Assim, foram utilizadas Redes Neurais Perceptron Multicamadas, *Support Vector Machines* e *Random Forests* aplicadas diretamente em cada base de dados, obtendo 84% de precisão pelas Redes Neurais Perceptron na base de dados formada por dados do Twitter.

O estudo de Filho (2019) foi baseado no artigo desenvolvido por Valencia Alfonso Gómez Espinosa (2019), e que também alicercou nossa pesquisa. Os resultados de Valencia Alfonso Gómez Espinosa (2019) obtiveram 76% de precisão numa Rede Neural Perceptron Multicamadas com a base de dados composta por dados do Twitter e pelos dados do mercado de Bitcoin. O estudo desenvolvido por eles foi inovador, pois abordaram o problema de identificação do preço de criptomoedas pelo processo de classificação, ou seja, dado um intervalo de tempo classificaram se o preço iria cair ou subir.

Essa nova abordagem teve sucesso e apresentou resultados animadores, levando em consideração que se encaixa perfeitamente no problema deste estudo. Para um profissional como o *trader*, não importa o valor exato que a moeda vai estar, mas sim se ele vai perder ou ganhar dinheiro em uma janela de tempo próxima. Em outras palavras, não é necessário identificar o preço exato em 24 horas, mas sim se em 24 horas o mercado vai estar para venda ou compra.

Metodologia

A Figura 8 exibe as etapas do modelo computacional proposto para prever o valor do Bitcoin. Após a extração da base de dados, ela passará pela fase de pré-processamento, que consiste em tornar os dados mais apropriados para a fase subsequente. Assim que os dados são divididos entre os conjuntos de teste e de treinamento, os algoritmos de AM serão empregados. Os valores resultantes do modelo proposto serão analisados e avaliados na última etapa do modelo.

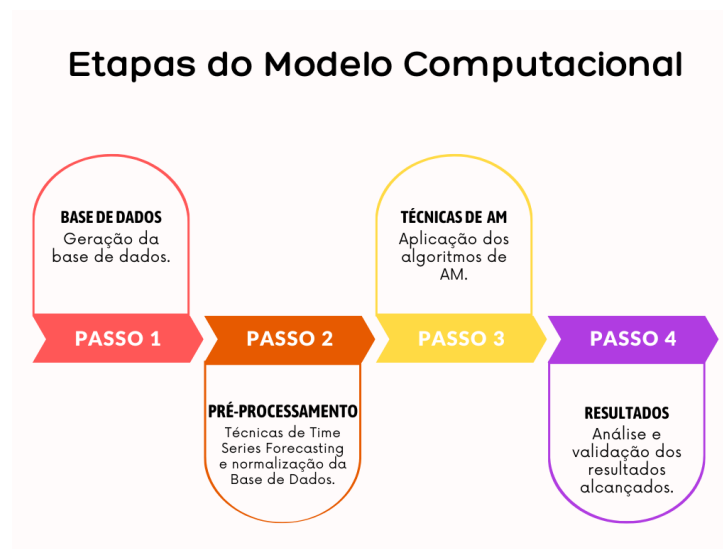


Figura 8 – Etapas do modelo computacional para predição dos valores da criptomoeda Bitcoin no mercado de ações.

As propriedades escolhidas e empregadas na criação do modelo computacional para predição estão descritas nas próximas seções deste capítulo.

3.1 Base de Dados

A base de dados foi extraída do site min-api.cryptocompare.com. Ela foi formada por 1968 amostras, constituída pelos atributos: *high*, *low*, *open*, *volumeFrom*, *close*. A

Tabela 2 apresenta as descrições desses atributos.

Tabela 1 – Descrição dos atributos da base de dados.

Tipo	Descrição
High	Preço mais alto do intervalo
Low	Preço mais baixo do intervalo
Open	Preço de abertura do intervalo
VolumeFrom	Volume movimentado no intervalo anterior
Close	Preço de fechamento do intervalo

Cada amostra da base de dados representa um período de intervalo de 1 hora, sendo que a cada hora teremos como informação todos os dados apresentados na Tabela 2. Os dados foram coletados no período de 27/05/2021 a 16/08/2021. Os targets de cada amostra da base leva em consideração o preço de abertura do dia e o preço de fechamento do dia, logo para todo o intervalo de tempo(t) = 24 horas, teremos todas as amostras classificadas igualmente. Isto quer dizer, que em um intervalo de t horas, todas as amostras dentro de um mesmo intervalo terão a mesma classificação.

Tabela 2 – Descrição dos targets de cada amostra.

Valor	Descrição
1	Compra
0	Venda

3.2 Técnicas de Pré-Processamento

Os dados coletados da API foram os dados brutos. A partir deles será efetuada o pré-processamento com o objetivo de eliminar alguns atributos que não vão irão contribuir para o aprendizado da rede neural como por exemplo os atributos `conversionSymbol`, `conversionType`, `timestamp` e `volumeTo`.

Dentro os atributos que não foram utilizados, podemos destacar o `timestamp` pelo fato do mesmo não agregar valor nas amostras, pois as mesmas já estão agrupadas corretamente de acordo com as suas respectivas datas. O `volumeTo` não será utilizado pois foi preferido utilizar somente o `volumeFrom`, e os atributos `conversionSymbol` e `conversionType` apresentam o mesmo valor para todas as amostras, logo não foi utilizado devido a isso.

As técnicas de pré-processamento tem como objetivo procurar melhorar uma formação dos dados mais adequada, tornando-os aptos para os algoritmos de predição e classificação. Dessa forma, utilizou-se no modelo proposto as técnicas de dados de séries temporais e normalização.

A classificação das amostras por intervalo de tempo caracteriza-se e utiliza da implementação do Time Series Forecasting, o que justifica os diferentes intervalos de tempo 24,

12, 6 e 3 horas, apresentados nas seções subsequentes. A intenção de fazer esta análise é verificar com qual série de tempo a rede se encaixa melhor para fazer a previsão e classificação de cada amostra, e conseqüentemente poder ter uma melhor acurácia ao final do treinamento e teste.

No pré-processamento dos dados, cada atributo das amostras está sendo normalizado utilizando o método `preprocessing.scale()` da biblioteca `sklearn`, da linguagem Python. Este método normaliza os valores em uma escala que varia de -1 a 1, após a separação da base de dados em conjunto de teste e de treinamento, como a própria biblioteca recomenda em sua documentação. Após a normalização, o próximo passo a ser executado é a montagem do array que vai servir como entrada para a nossa rede. O array é denominado de `sequential_data[]`, pois ele é o responsável por guardar a seqüência das amostras por período de tempo que a rede irá analisar nas etapas de aprendizagem e de treinamento.

Por fim, após separar o array temporal em intervalos de tempo, será executado o balanceamento da base, ou seja, será separada a quantidade de casos onde o preço será para venda ou compra. Caso alguma classe tenha mais amostras que outra, a mesma será balanceada, visando garantir uma distribuição mais homogênea dos dados e que não haja mais informações tendendo para uma única classificação. A Figura 9 exemplifica o conceito de balanceamento.

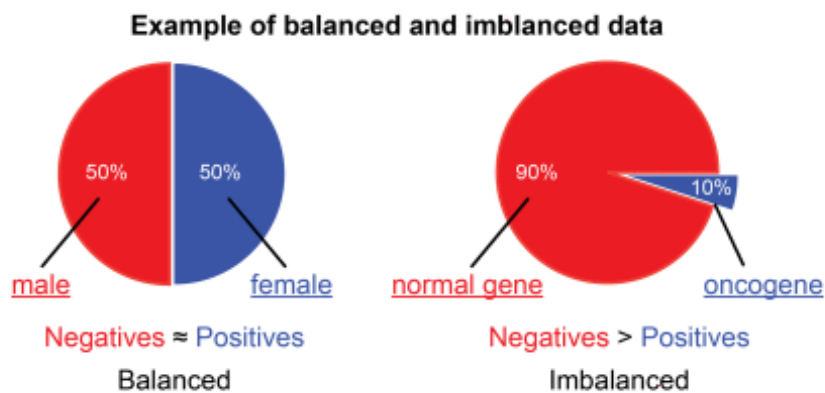


Figura 9 – Exemplo de balanceamento de dados.

3.3 Redes Neurais

O próximo passo é a execução da rede neural recorrente, neste caso utilizando-se da rede LSTM. Para a modelagem da rede foram utilizadas três camadas de entrada com duas camadas ocultas, sendo que na primeira camada oculta está sendo utilizada a função `relu` para ativação.

A estrutura proposta da rede LSTM foi definida a partir de uma seqüência de testes onde foi testado diferentes estruturas, logo a que apresentou os melhores resultados foi

utilizada como a final.

A função relu é uma função não linear que retorna 0 para valores negativos, e para casos positivos retorna o próprio valor, como pode ser visto nos gráficos da Figura 10.

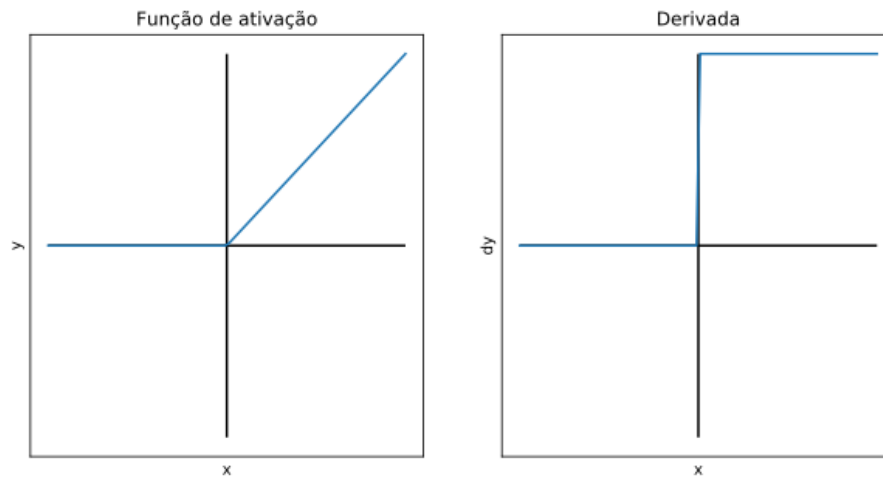


Figura 10 – Função de ativação Relu.

Fonte: Expert Academy: Funções de ativação: definição, características, e quando usar cada uma.

A segunda camada oculta temos como função de ativação a softmax, que é uma generalização da função sigmoide. Basicamente ela produz valores diferentes para cada classe de saída, e cada valor produzido representa a probabilidade daquela instância pertencer a uma classe específica, que no nosso caso, é de venda ou compra.

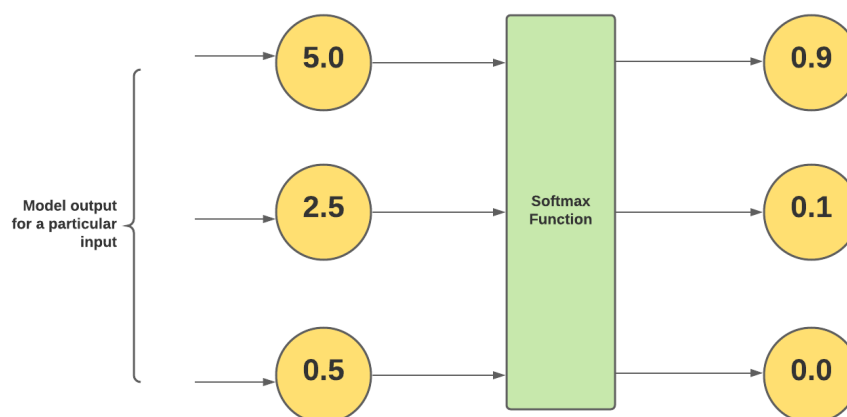


Figura 11 – Função de ativação Softmax.

Fonte: Data Analytics: Softmax Regression Explained with Python Example

A utilização da softmax como função de ativação na última camada de saída tem como motivação o fato do poder decisão que ela tem no decorrer das épocas, devido ao parâmetro de temperatura da sua equação, ela consegue controlar o quanto os dados serão

impactados de acordo com o decorrer do tempo, sendo que quando menor a temperatura mais rígida ela será na decisão.

Como podemos ver na Figura 11, dependendo do valor de saída do modelo, ela propõem um valor entre 0 e 1 que corresponde a probabilidade de uma determinada classe, neste caso apresentado, o valor 0.5 representa 0% de chance, enquanto que 5.0 representa 90%.

A função utilizada na compilação do modelo é a função Sparse Categorical Crossentropy, que é indicada para casos onde a saída são números inteiros, que neste caso, são 1 para a classe de venda e 0 para classe de compra. Como otimizador do modelo, está sendo utilizado o Adam, que é um otimizador lançado recentemente e que propõe uma otimização considerando a média exponencial, acarretando em uma maior aceleração da descida do gradiente.

Todo o código desenvolvido foi feito em Python, sendo utilizado as bibliotecas do TensorFlow e scikit-learn para as técnicas envolvendo aprendizado de máquina. Após todo o processo de treinamento da rede, o modelo será aplicado no conjunto de testes para verificar a predição destes novos valores.

3.4 Medida de Avaliação

A principal medida de avaliação que será utilizada durante o experimento é a acurácia, pois a partir dela poderemos ter uma maior compreensão sobre o quanto os resultados obtidos na rede são aproximados com relação aos dados reais.

A acurácia corresponde ao percentual de acertos entre os valores preditos e o resultado real. Antes de definir a equação da acurácia, é necessário entender os valores que compõem a matriz de confusão, que está representada na Figura 12. A matriz de confusão contém os valores que indicam os erros e acertos do modelo ao comparar com os resultados esperados.

Os valores da matriz de confusão são: VP (Verdadeiro Positivo), VN (Verdadeiro Negativo), FP (Falso positivo) e FN (Falso Negativo). Assim, a acurácia pode ser calculada pela equação:

$$Acurácia = \frac{VP + VN}{VP + VN + FP + FN} \quad (1)$$

Verdadeiro positivo são os casos onde a amostra foi classificada como preço de venda corretamente, ao contrário do Verdadeiro Negativo onde a mesma foi classificada como venda porém pertence a classe de compra. Falsos positivos seguem a mesma linha, são casos onde a amostra é classificada corretamente como venda, ao contrário do Falso negativo.

		Verdade	
		Condição Positiva	Condição Negativa
Predito	Condição positiva prevista	Verdadeiro Positivo (VP)	Falso Positivo (FP)
	Condição negativa prevista	Falso Negativo (FN)	Verdadeiro Negativo (VN)

Figura 12 – Representação da Matriz de Confusão.

Fonte: Autor.

Experimentos e Análise dos Resultados

Neste capítulo serão abordados os experimentos realizados durante o trabalho, desde o começo da estruturação das RNAs, todo o pré-processamento de dados e, na sequência, os resultados obtidos em diferentes testes. De forma resumida, durante a realização dos experimentos foram aplicadas técnica de normalização dos dados e também um balanceamento na base, com objetivo de facilitar a aprendizagem para ambos os casos de teste.

4.1 Experimentos

Como mencionado anteriormente, nesta seção será exibido os resultados obtidos durante o treinamento da Rede Neural Recorrente LSTM, em diferentes situações de testes. Ou seja, diferentes períodos de horas(t) foi utilizados como entrada na rede, além de diferentes tamanhos de base de treinamento e teste. Após exibir os resultados obtidos pela LSTM, também será avaliada a rede neural MLP e KNN com o mesmo conjunto de amostras.

Os resultados obtidos durante a fase de treinamento e de teste da base de dados para a rede LSTM, no intervalo de tempo de 24 horas, estão exibidas nas Figura 13, Figura 14 e Figura 15 .

É importante ressaltar que não foi realizada a validação cruzada com todos os resultados obtidos, logo os resultados exibidos a seguir são os melhores resultados obtidos durante o trabalho.

Com 5% da base de dados para teste aplicados a um intervalo de série temporal de 24 horas, o modelo conseguiu aproximadamente cerca de 62% de acurácia . A Figura 13 mostra esses resultados.

As quatro linhas exibidas no gráfico referem-se aos valores: perda no conjunto de treinamento($loss$), perda na validação do modelo pelo conjunto de teste (val_loss), acurácia obtida no conjunto de dados para treinamento (acc) e acurácia na validação do modelo pelo conjunto de dados para teste (val_acc).

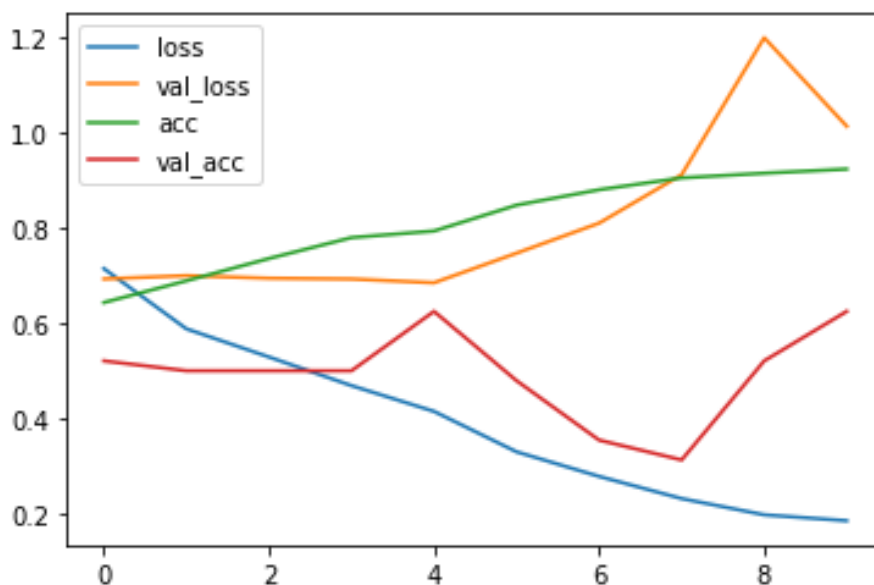


Figura 13 – Conjunto de teste com 5% dos dados para o período de 24 horas em uma rede LSTM.

A Figura 14 refere-se ao teste realizado com 20% da base de dados para o conjunto de teste. Aqui o modelo conseguiu aproximadamente 94% de acurácia no conjunto de treinamento, enquanto que para o conjunto de teste gerou 49% de acurácia. Observa que teve um leve aumento no valor da acurácia no treinamento, porém ainda não conseguiu atingir valores significativos para o conjunto de testes.

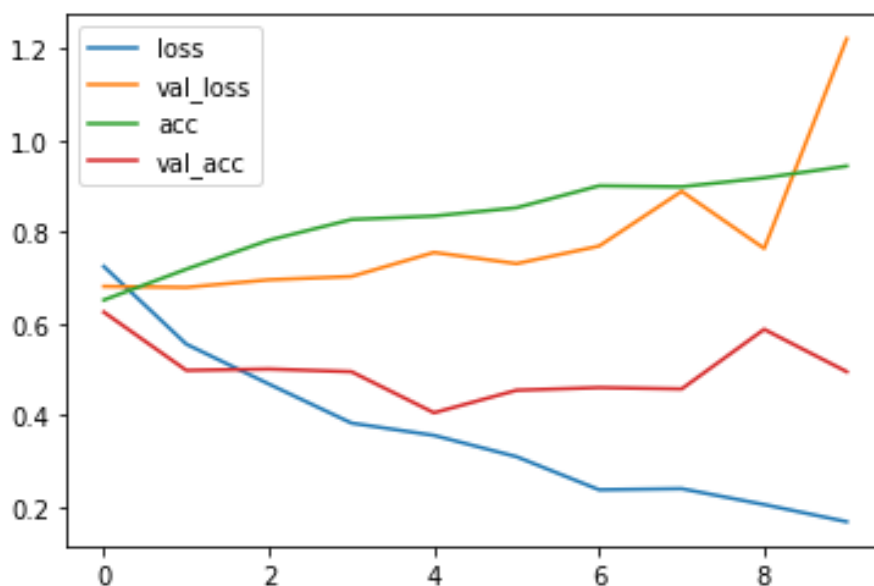


Figura 14 – Conjunto de teste com 20% dos dados para o período de 24 horas em uma rede LSTM.

Para a configuração de 30% da base de dados para teste, o modelo conseguiu aproximadamente 93% de acurácia no treinamento, e 63% de acurácia no conjunto de teste,

como pode ser visto na Figura 15. Nota-se ao comparar os valores atingidos nos três últimos gráficos que a acurácia teve uma pequena queda no treinamento e uma melhora gradativa no conjunto de dados para teste.

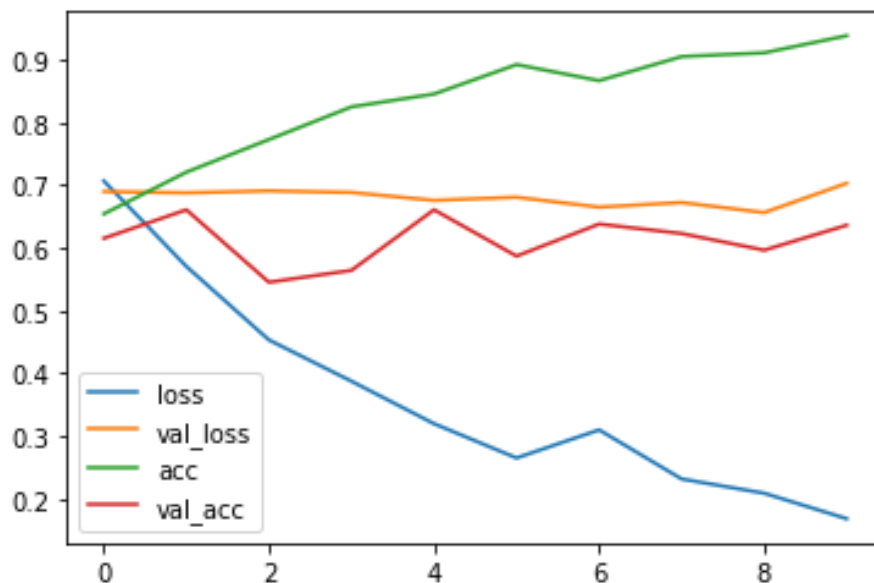


Figura 15 – Conjunto de teste com 30% dos dados para o período de 24 horas em uma rede LSTM.

O próximo cenário a ser avaliado é referente aos dados de entrada da rede estarem subdivididos no período de tempo de 12 horas para 5%, 20% e 30% da base de dados para teste. As Figura 16, Figura 17 e Figura 18 representam este cenário.

Ao analisar a Figura 16, houve uma ligeira queda no valor da acurácia para o conjunto de treinamento, porém mantendo o percentual de acertos em 61% no conjunto de testes ao comparar com o melhor resultado para o período de tempo de 24 horas (Figura 15).

Com 20% dos dados para teste, houve uma pequena queda da acurácia no treinamento, caindo de 80% para 79% aproximadamente, ao comparar com os resultados de Figura 16. Isso também foi refletido no conjunto de teste, reduzindo de 61% para 58% aproximadamente. Esses resultados podem ser vistos na Figura 17.

A Figura 18 mostra que de 30% dos dados para teste, obteve-se acurácia de 84% para o conjunto de treinamento e de 58% de acertos para o conjunto de teste.

Logo, conclui-se que não houve melhoras nos resultados diminuindo o período de 24 para 12 horas. Pelo contrário, houve uma ligeira queda na acurácia do modelo de treinamento, e manteve-se regular no conjunto de teste.

Na sequência, os resultados obtidos durante a fase de treinamento e de teste da base de dados para a rede LSTM, no intervalo de tempo de 6 horas nas Figura 19, Figura 20 e Figura 21.

O primeiro resultado obtido para o período de 6 horas resultou em uma acurácia de 71% no treinamento, e 63% de acertos para o conjunto de teste, como pode ser visto na

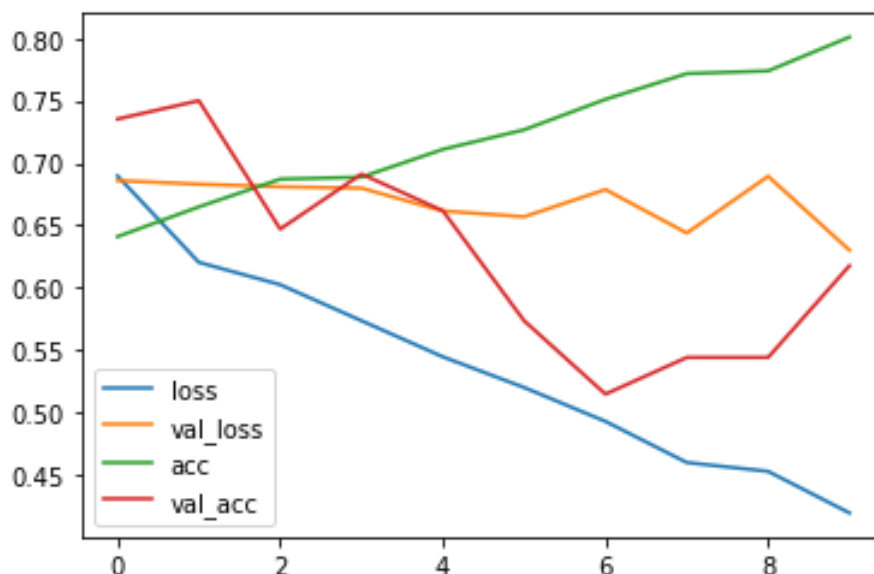


Figura 16 – Conjunto de teste com 5% dos dados para o período de 12 horas em uma rede LSTM.

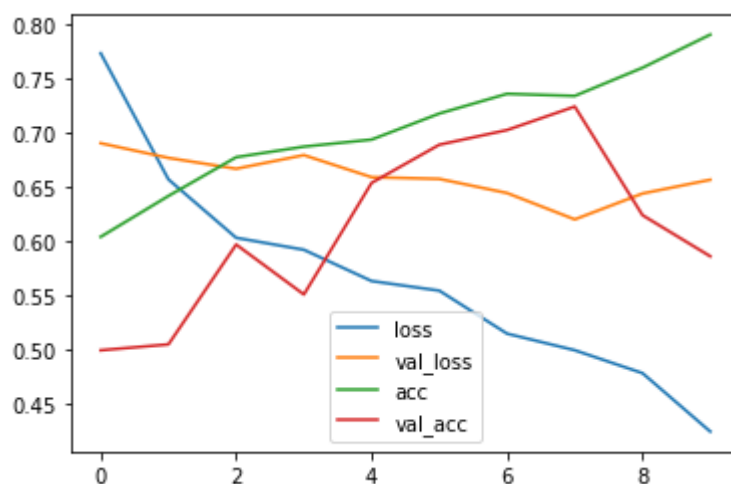


Figura 17 – Conjunto de teste com 20% dos dados para o período de 12 horas em uma rede LSTM.

Figura 19.

A Figura 20 mostra que com 20% da base de dados para o conjunto de teste, os resultados se mantiveram constantes, e 54% de acerto para o conjunto de teste.

Já a Figura 21 mostra que com 30% dos dados, a acurácia para o conjunto de treinamento foi 69,9% de acurácia, enquanto que o conjunto de teste ficou em 50% de acurácia.

Verifica-se que a medida que o período de tempo de agrupamento dos dados vai diminuindo, os resultados obtidos no treinamento pioram, mas os resultados obtidos na fase de teste se mantêm.

Por último, exibi-se os resultados gerados pela rede LSTM com a base de dados agrupada em períodos de tempo de 3 horas e subdividida em conjuntos de teste em 5%, 20%

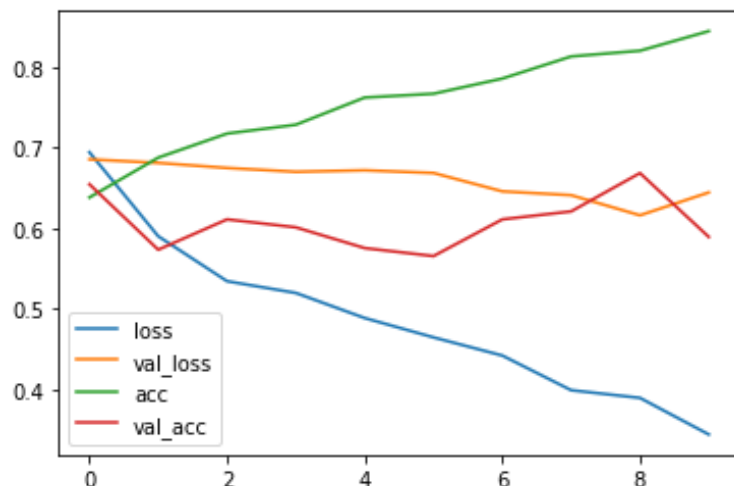


Figura 18 – Conjunto de teste com 30% dos dados para o período de 12 horas em uma rede LSTM.

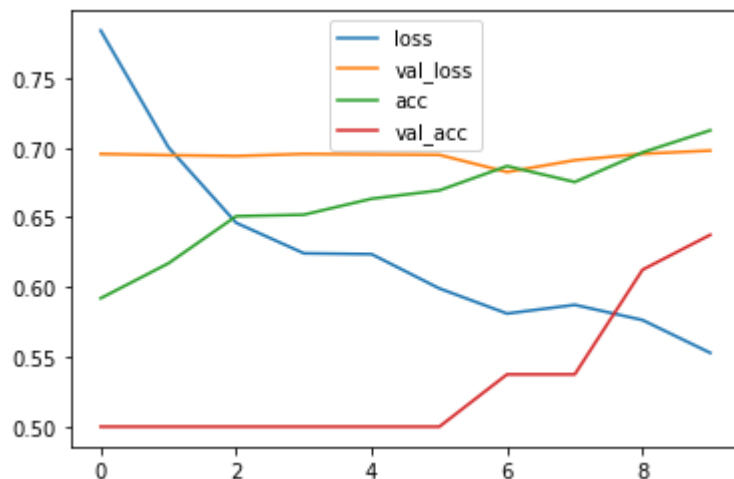


Figura 19 – Conjunto de teste com 5% dos dados para o período de 6 horas em uma rede LSTM.

e 30% nas respectivas Figura 22, Figura 23 e Figura 24.

Neste cenário, manteve-se o valor da acurácia na fase de treinamento, ficando em 74% de acertos. O que chama a atenção foi o resultado da acurácia na fase de teste, alcançando 70% de acertos (22). Esse foi o melhor resultado dentre os já apresentados.

A Figura 23, com 20% dos dados para teste, obteve 75% de acurácia em treinamento, e 71% de acertos para o conjunto de testes. Esse resultado é muito interessante, pois mostra que a rede se adaptou muito bem a este intervalo de 3 horas, pois a acurácia em teste e em treinamento estão constantes e similares.

Por fim, a Figura 24 com 30% dos dados para teste obteve 77% de acurácia em treinamento e 70% de acurácia em testes, concluindo que de todas as configurações testadas na LSTM, os melhores resultados foram para o agrupamento num período de 3 horas.

A Figura 25 exhibe os resultados alcançados pela rede MLP com a mesma base de

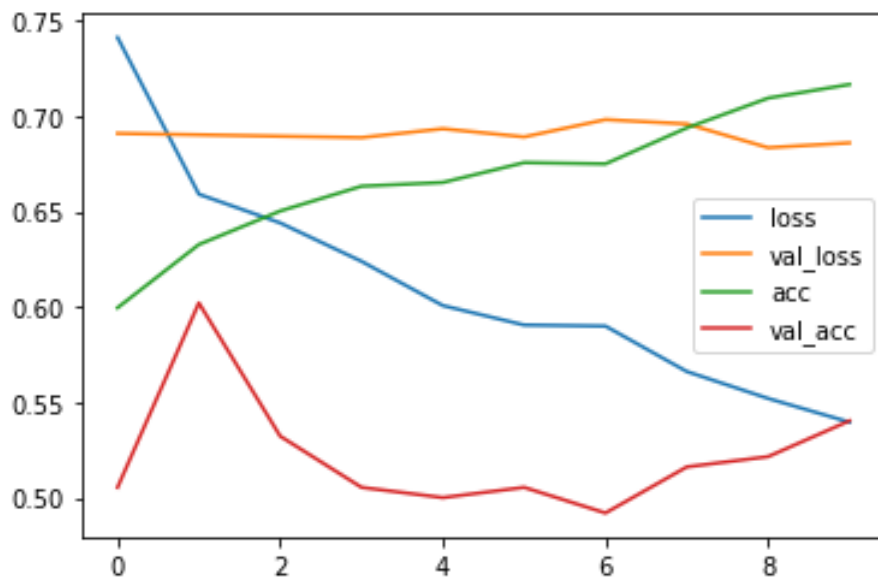


Figura 20 – Conjunto de teste com 20% dos dados para o período de 6 horas em uma rede LSTM.

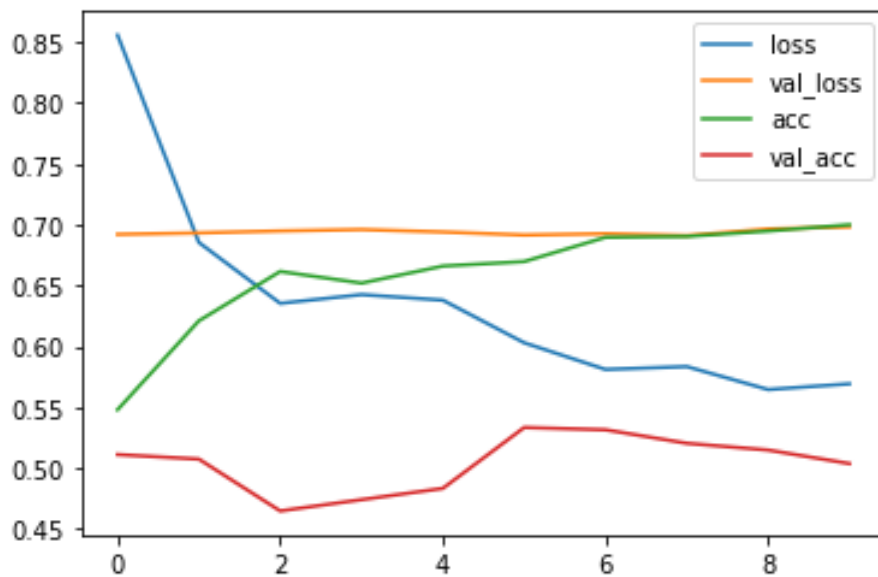


Figura 21 – Conjunto de teste com 30% dos dados para o período de 6 horas em uma rede LSTM.

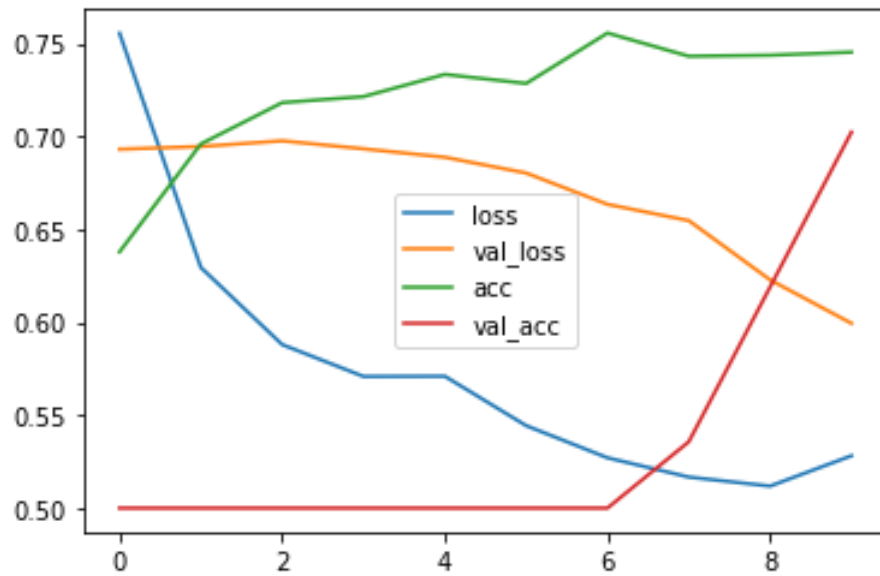


Figura 22 – Conjunto de teste com 5% dos dados para o período de 3 horas em uma rede LSTM.

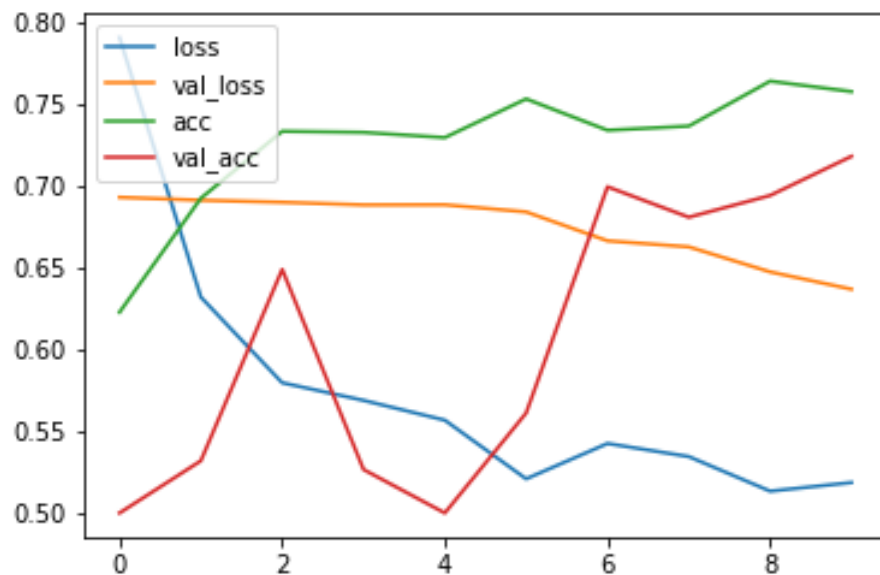


Figura 23 – Conjunto de teste com 20% dos dados para o período de 3 horas em uma rede LSTM.

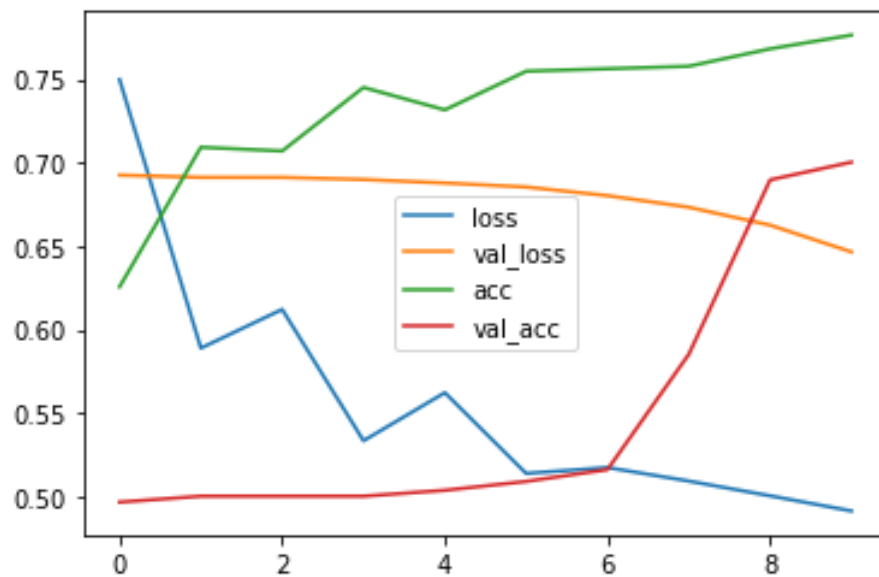


Figura 24 – Conjunto de teste 30% dos dados para o período de 3 horas em uma rede LSTM.

dados normalizada e balanceada. A rede foi montada com uma camada de entrada, uma camada oculta e por fim uma camada de saída com a função de ativação softmax.

A acurácia permaneceu em 50% de acertos tanto para o conjunto de treinamento como para o de teste. Logo o modelo não foi capaz de ter um aprendizado mínimo que fosse capaz de desempenhar uma melhor predição.

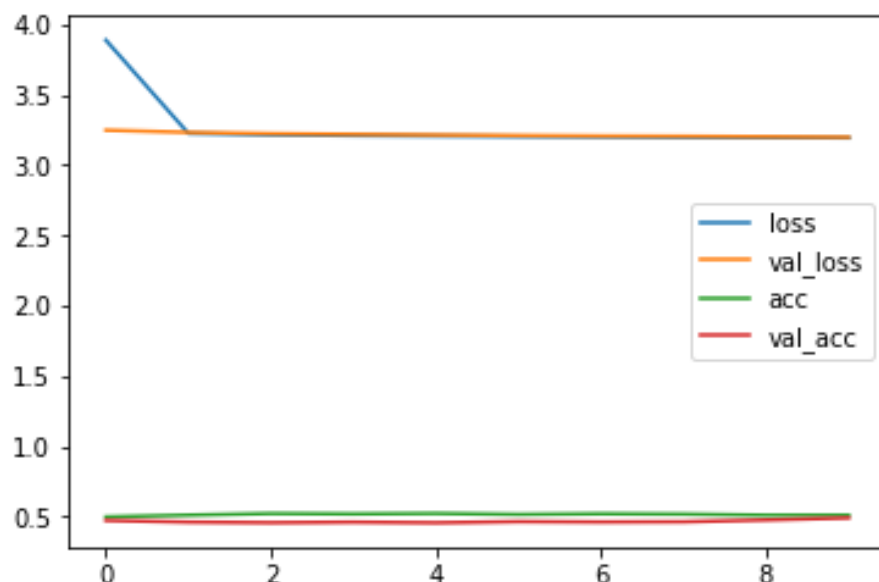


Figura 25 – Conjunto de teste 30% no período de 24 horas em uma rede MLP.

A técnica KNN também foi aplicada sob a mesma base de dados normalizada e balanceada. Foi-se utilizado $k=3$ vizinhos mais próximos, sendo tal valor escolhido de forma aleatória na rede. As Figura 26 e Figura 27 mostram os resultados alcançados por essa

técnica quando dividida a base de dados de teste em 5% e 20%, respectivamente. Em ambos resultados, a acurácia variou próximo a 75% de acertos na fase de treinamento e 53% de acertos na fase de testes. Os casos mostrados referem-se aos melhores resultados obtidos em cada rede.

```

train data: 1766 validation: 90
Dont buys: 883 Buys: 883
Validation Dont buys: 45 validation buys: 45
Acurácia de treinamento: 0.7508493771234428
Acurácia de teste: 0.5555555555555556
Relatório de classificação:
      precision    recall  f1-score   support

   compra         0.55         0.60         0.57         45
   venda         0.56         0.51         0.53         45

 accuracy
macro avg         0.56         0.56         0.55         90
weighted avg         0.56         0.56         0.55         90

```

Figura 26 – Resultado de medidas de avaliação do KNN sob o conjunto de teste de 5% da base de dados.

```

train data: 1478 validation: 384
Dont buys: 739 Buys: 739
Validation Dont buys: 192 validation buys: 192
Acurácia de treinamento: 0.7510148849797023
Acurácia de teste: 0.5338541666666666
Relatório de classificação:
      precision    recall  f1-score   support

   compra         0.53         0.53         0.53        192
   venda         0.53         0.54         0.54        192

 accuracy
macro avg         0.53         0.53         0.53        384
weighted avg         0.53         0.53         0.53        384

```

Figura 27 – Resultado de medidas de avaliação do KNN sob o conjunto de teste de 20% da base de dados

4.2 Avaliação dos Resultados

Os resultados alcançados, principalmente pela rede neural LSTM foram significativos, pois alcançou valores próximos à 71% de acurácia para o conjunto de dados para teste. A rede neural LSTM é bastante promissora, devido a sua capacidade de processamento de dados de séries temporais.

Outro ponto a observar, é referente ao desempenho da rede PMC e KNN. Ambas conseguiram ter um desempenho semelhante mas inferior à LSTM, o que pode ser justificado por: não adaptou ao Times Series Forecasting, ou pode ter sido modelada incorretamente.

Conclusão

A aplicação da rede neural recorrente LSTM numa base de dados real da criptomoeda Bitcoin atuou como um modelo computacional capaz de prever a movimentação desta moeda no mercado financeiro, dentro do período de tempo analisado. Assim, o percentual de 71% de acurácia sob os dados de teste é um excelente indicativo que a rede LSTM é uma técnica que deve ser explorada na predição de valores do Bitcoin.

Durante o trabalho conseguimos criar diferentes modelos computacionais aplicando as redes neurais LSTM e PMC e o KNN juntamente com as séries de dados temporais, variando o período de tempos em 3, 6, 12 e 24 horas. Conclui que a rede conseguiu se adaptar as informações para predizer-se os valores da moeda Bitcoin estavam em alta ou em queda. Também é importante ressaltar que o melhor resultado obtido pela rede LSTM aproximaram-se do melhor resultado obtido por (VALENCIA ALFONSO GÓMEZ ESPINOSA, 2019), com a diferença que o mesmo aplicou a análise de sentimentos no modelo, enquanto que neste trabalho foi-se utilizado somente os dados do mercado de Bitcoin.

O grande desafio enfrentado no decorrer do desenvolvimento do trabalho foi na montagem da resolução do problema. A origem do trabalho era continuar a pesquisa realizada por (VALENCIA ALFONSO GÓMEZ ESPINOSA, 2019) e aplicar a mesma técnicas em *trading* reais. Achar técnicas e metodologias para aplicar em si não é o problema, mas sim aplicar as mesmas dentro do cenário que queríamos atuar. Acredito que para trabalhos futuros este é cenário que pode ser bastante aprofundado e estudado, mas tendo em vista a situação trabalho, optamos pela aplicação da Rede Neural Recorrente para conseguir gerar um bom estudo sobre a mesma, tendo em vista sua característica de previsibilidade de dados em relação a outras técnicas.

5.1 Principais Contribuições

A principal contribuição produzida é a confirmação de que as redes neurais recorrentes tem uma crescente evolução em trabalhos com aplicação de dados de séries temporais para predição de subida ou descida de variáveis em estudo. O trabalho de (VALENCIA

ALFONSO GÓMEZ ESPINOSA, 2019) conseguiu resultados de 76% de acurácia aplicando análise de sentimentos juntamente com dados reais em redes neurais MLP, o que caracteriza uma boa comparação com os dados alcançados neste estudo.

Outra contribuição muito importante é o estudo de como o tamanho das séries temporais afetam a aprendizagem da rede LSTM, pois os melhores resultados foram obtidos com intervalos de tempos menores. Logo, conclui-se que tempos menores são melhores pois oferecem dados mais precisos em curtos intervalos de tempo, ou também, tempos maiores tem resultados inferiores devido a aleatoriedade já mencionada pelos dados temporais.

Dito isso, o trabalho conseguiu contribuir para novos estudos da aplicação de dados de séries temporais juntamente com RNRs. Por fim, ressalta-se o reconhecimento por todo o trabalho feito por (VALENCIA ALFONSO GÓMEZ ESPINOSA, 2019) e (FILHO, 2019), que alicerçou a pesquisa aqui desenvolvida.

5.2 Trabalhos Futuros

Para trabalhos futuros, a continuação dos estudos do *time series forecasting* aplicado a redes neurais recorrentes seria muito promissor e teria um grande leque de possibilidades, como por exemplo, estudar como intervalos de tempos diferentes influenciam diretamente no aprendizado do modelo.

Outro trabalho muito interessante, é aplicação da análise de sentimentos que foi realizada por (VALENCIA ALFONSO GÓMEZ ESPINOSA, 2019) com o objetivo de verificar se a mesma pode melhorar ainda mais os resultados da rede LSTM. Os 71% de acurácia obtidos na rede mostram que é bastante promissor os estudos, mas ainda tem-se muito a melhorar, estudar a aplicação da análise de sentimentos seria interessante para validar se é possível conseguir melhores percentuais para serem aplicados no mercado real.

Referências

ACADEMY., D. S. **Deep Learning Book**. Addison-Wesley, 2022. Disponível em: <<https://www.deeplearningbook.com.br/>>. Citado na página 18.

AGGARWAL, D.; CHANDRASEKARAN, S.; ANNAMALAI, B. A complete empirical ensemble mode decomposition and support vector machine-based approach to predict bitcoin prices. **Journal of Behavioral and Experimental Finance**, v. 27, p. 100335, 2020. ISSN 2214-6350. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2214635019302266>>. Citado na página 12.

BASHER, S. A.; SADORSKY, P. Forecasting bitcoin price direction with random forests: How important are interest rates, inflation, and market volatility? **Machine Learning with Applications**, v. 9, p. 100355, 2022. ISSN 2666-8270. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S266682702200055X>>. Citado na página 12.

BÖHME, R. et al. Bitcoin: Economics, technology, and governance. **Journal of Economic Perspectives**, 2015. ISSN 213-238. Disponível em: <<https://pubs.aeaweb.org/doi/pdfplus/10.1257/jep.29.2.213>>. Citado na página 15.

CHEN, Z.; LI, C.; SUN, W. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. **Journal of Computational and Applied Mathematics**, v. 365, p. 112395, 2020. ISSN 0377-0427. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S037704271930398X>>. Citado na página 12.

CORBET, S. et al. Cryptocurrencies as a financial asset: A systematic analysis. **International Review of Financial Analysis**, v. 62, n. C, p. 182–199, 2019. Citado na página 12.

FERREIRA, M. et al. Blockchain: A tale of two applications. **Applied Sciences**, v. 8, p. 1506, 09 2018. Citado na página 12.

FILHO, W. B. S. **Prevento o preço do Bitcoin com redes neurais usando dados do Twitter e de mercado**. Dissertação (Dissertação de Mestrado) — Universidade Federal de Uberlândia, 2019. Disponível em: <<http://repositorio.ufu.br/handle/123456789/28394>>. Citado 2 vezes nas páginas 21 e 39.

- FRANCO, M. Venezuelanos investem em bitcoin para encarar desemprego, hiperinflação e falta de notas. 2017. Disponível em: <<https://g1.globo.com/mundo/noticia/venezuelanos-investem-em-bitcoin-para-encarar-desemprego-hiperinflacao-e-falta-de-notas.ghtml>>. Citado na página 15.
- FRANCO, P. Understanding bitcoin: Cryptography, engineering and economics. 2014. Citado na página 15.
- HAYKIN, S. **Redes Neurais Princípios e prática**. [S.l.]: Artmed Editora S.A., 2007. ISBN 0-13-273350-1. Citado 4 vezes nas páginas 15, 16, 17 e 18.
- HEWAMALAGE, H.; BERGMEIR, C.; BANDARA, K. Recurrent neural networks for time series forecasting: Current status and future directions. **International Journal of Forecasting**, v. 37, n. 1, p. 388–427, 2021. ISSN 0169-2070. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0169207020300996>>. Citado na página 17.
- KURUMATANI, K. Time series forecasting of agricultural product prices based on recurrent neural networks and its evaluation method. 2020. Disponível em: <<https://doi.org/10.1007/s42452-020-03225-9>>. Citado na página 17.
- LAHMIRI, S.; BEKIROS, S. Chaos, randomness and multi-fractality in bitcoin market. **Chaos, Solitons Fractals**, v. 106, p. 28–34, 2018. ISSN 0960-0779. Citado na página 12.
- LIVIERIS, I. E.; PINTELAS, E.; PINTELAS, P. A cnn-lstm model for gold price time-series forecasting. **Neural Computing and Applications**, 2020. Disponível em: <<https://link.springer.com/article/10.1007/s00521-020-04867-x#citeas>>. Citado na página 19.
- MATTKE, J. et al. Bitcoin investment: a mixed methods study of investment motivations. **European Journal of Information Systems**, 2010. Disponível em: <<https://www.tandfonline.com/doi/full/10.1080/0960085X.2020.1787109>>. Citado na página 15.
- NAKAMOTO, S. A peer-to-peer electronic cash system. <http://www.bitcoin.org/>, 2008. ISSN 09254560. Disponível em: <<https://bitcoin.org/bitcoin.pdf>>. Citado 2 vezes nas páginas 13 e 15.
- ULRICH, F. **Bitcoin: A moeda na era digital**. [S.l.]: LVM Editora, 2017. Citado 2 vezes nas páginas 12 e 15.
- VALENCIA ALFONSO GÓMEZ ESPINOSA, B. V. F. Price movement prediction of cryptocurrencies using sentiment analysis and machine learning. **entropy**, 2019. Citado 6 vezes nas páginas 12, 13, 14, 21, 38 e 39.
- WUERGES ARTUR FILIPE EWALD E BORBA, J. A. Redes neurais, lógica nebulosa e algoritmos genéticos: aplicações e possibilidades em finanças e contabilidade. **JISTEM**, 2010. ISSN 1807-1775. Disponível em: <<http://dx.doi.org/10.4301/S1807-17752010000100007>>. Citado na página 13.