UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
FACULDADE DE ADMINISTRAÇÃO E CIÊNCIAS CONTABEIS
CURSO DE GRADUAÇÃO EM ADMINISTRAÇÃO

REDES NEURAIS PARA PREDIÇÃO DE SERIES TEMPORAIS

YÁ-SIN BARCELOS MGHAZLI

ORIENTADOR(A): Prof. Camila Avosani Zago

Rio de Janeiro, setembro de 2021

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
FACULDADE DE ADMINISTRAÇÃO E CIÊNCIAS CONTABEIS
CURSO DE GRADUAÇÃO EM ADMINISTRAÇÃO

REDES NEURAIS PARA PREDIÇÃO DE SERIES TEMPORAIS

YÁ-SIN BARCELOS MGHAZLI
DRE: 118108627
ORIENTADOR(A): Prof. Camila Avosani Zago

Monografia apresentada ao Departamento de Administração para obtenção do grau de Bacharel em Administração.

Rio de Janeiro, setembro de 2021

RESUMO

Diversas classes de modelos têm sido propostas como solução do dilema do passeio aleatório para previsão de séries temporais financeiras. Embora não haja nenhuma prova formal sobre sua previsibilidade, alguns trabalhos argumentam que, na prática, este fenômeno temporal é, de alguma forma, previsível. Portanto, este trabalho analisa qual a eficiência dos modelos (ARIMA, *MultiLayer Perceptrons* e *Long Short-Term Memory* para séries temporais propostos para a predição do índice Bovespa? como solução do dilema do passeio aleatório no problema de previsão de séries temporais financeiras. Uma análise experimental é conduzida com os modelos investigados utilizando uma série temporal relacionada ao Índice da Bolsa de Valores de São Paulo (IBOVESPA) utilizando normalização e diferenciação dos dados, janelamento temporal e otimização dos hiperparâmetros dos modelos. Os resultados alcançados demonstraram efetividade, em desempenho preditivo, dos modelos investigados.

Palavras-Chave: Redes Neurais, Séries Temporais, Mercado de Ações, Previsão, Dilema do Passeio Aleatório.

LISTA DE FIGURAS

SUMÁRIO

1. INTRODUÇÃO

Uma série temporal pode ser descrita como um conjunto de observações feitas de uma variável aleatória, sequencialmente temporal e com a mesma frequência de observação. A característica predominante para as séries temporais é que um conjunto de observações vizinhas são correlatas. Desta forma, pode-se supor que há um sistema causal que exerce influência sobre o conjunto de dados, sendo, assim, possível modelar o sistema de tal forma que se possa determinar o valor futuro da série temporal com base nos valores passados (SOUZA; CAMARGO, 2004).

A partir da modelagem do sistema causal são apresentadas as observações dos valores passados da série com uma ou mais variáveis ao modelo, no qual tentará fazer a predição dos valores futuros para a série, a qual as observações pertencem.

A análise e previsão de séries temporais são importantes instrumentos que se estende por diversas áreas de atuação, estudos, aplicações, formulação de planos de ação e estratégias para investidores.

Sendo assim, escolheu-se a utilização de diferentes modelos lineares estatísticos devido a sua consolidação na literatura e os modelos da área de inteligência artificial devido a sua capacidade de generalização.

A Hipótese do Mercado Eficiente forte diz que mesmo com informações ocultas ou "privilegiadas" o preço de mercado se reajusta para refletir esse conhecimento. Dessa forma, tem-se um paradoxo que, se existe uma informação ou um modelo eficiente para estimar o valor futuro de uma ação, ao utilizá-lo o mercado se reajusta.

Em Teoria Econômica tem-se o que é chamado de jogo de soma zero onde um jogador deve necessariamente perder para que outro jogador possa ganhar, considerando a Hipótese dos Mercados Eficientes no melhor caso, onde a média de seus ganhos é igual à média de mercado. Assim, é possível inferir que a média de todos os seus ganhos será igual a zero, considerando os custos operacionais e financeiros os investidores sempre teriam prejuízos a longo prazo.

Diante dos argumentos anteriores, pode-se inferir que a previsão de séries temporais financeiras é um problema difícil de ser solucionado, uma vez que o fator gerador de uma série temporal depende de padrões complexos que são difíceis de serem previstos. Um exemplo dessa complexidade pode ser notado no próprio IBOVESPA que diferentes informações de diversas empresas que o compõe pode interferir no valor do índice.

Apesar da complexidade das séries temporais financeiras, o presente estudo busca verificar a eficiência de modelos de previsão para séries temporais. Para este trabalho, foram

utilizados os valores Ibovespa dos últimos vinte anos para a predição de sua série temporal e comparados com os resultados dos diferentes modelos propostos.

Dessa forma, define-se como problema de pesquisa o seguinte questionamento: *Qual a eficiência dos modelos de séries temporais propostos para a predição do índice Bovespa?*

1.1.Relevância do trabalho

Existem aplicações para as séries temporais em diversas áreas de conhecimento como por exemplo: Economia (preços de ações, taxa de desemprego, taxa de juros), Medicina (Níveis de eletrocardiograma, eletroencefalograma), Epidemiologia (casos de sarampo, AIDS e dengue), Meteorologia (temperatura diária, precipitação de chuvas, velocidade dos ventos) e Energia (consumo de energia elétrica, vazão de reservatórios de usinas hidroelétricas) (MENESES JÚNIOR, 2012). Desta forma, a determinação de padrões de tendencia, periodicidade ou alterações estruturais no comportamento da série temporal pode contribuir para tomada de decisões em diversas áreas de pesquisa.

A necessidade de conhecimento sobre o comportamento futuro do mercado de ações tem despertado interesse de diversos pesquisadores visando encontrar meios para construir modelos capazes de estimar este fenômeno temporal. Desta forma, a previsão de séries temporais financeiras pode ser utilizada no processo de tomada de decisão para compra e venda de ações, permitindo aos investidores maximizar o lucro e minimizar o risco de suas operações (ARAÚJO, 2007).

Entretanto, ainda não existe nenhuma prova matemática que sustente a hipótese da previsibilidade (eficiência) do mercado de ações, embora diversos estudos tenham demonstrado, na prática, que o fenômeno gerador de séries financeiras é, de alguma forma, previsível. Assim, considerando o fato de as previsões geradas possuírem um atraso de uma unidade de tempo em relação aos valores reais da série, surge um problema nos modelos de previsão, conhecido como dilema do passeio aleatório para prever séries temporais financeiras (ARAÚJO, 2007).

1.2.Objetivos do trabalho

O presente trabalho tem como objetivo geral verificar a eficiência dos modelos de séries temporais (ARIMA, MLP e LSTM) para a predição do índice Bovespa a partir de um estudo sobre séries temporais financeiras. Para alcançar esse objetivo se fez uso dos objetivos específicos abaixo.

- Determinar a melhor janela temporal (*Time-Lag*) da série;
- Averiguar os melhores parâmetros de configuração para cada modelo proposto;
- Predizer os valores futuros de abertura do IBOVESPA com os modelos propostos;
- Analisar o desempenho para predição das séries temporais de diferentes modelos.

1.3.Estrutura do trabalho

A estrutura deste trabalho é composta por oito capítulos descritos a seguir:

Capítulo 1 – Introdução - apresenta uma introdução ao problema de previsão de series temporais financeiras, assim como a relevância e objetivos do trabalho.

Capítulo 2 – Referencial Teórico: nesse capítulo são apresentadas as técnicas descritivas, redes neurais utilizadas, modelos de previsão e as medidas para avaliação de desempenho.

Capítulo 2.1 – Técnicas Descritivas - define formalmente as séries temporais financeiras e apresenta os problemas relacionados com o tratamento dos dados de uma série temporal.
Capítulo 2.2 – Redes Neurais Artificiais - apresenta os conceitos de redes neurais artificiais, assim como suas funções matemáticas e seu algoritmo de aprendizagem.

Capítulo 2.3 – Modelos de Previsão - são apresentados os modelos de previsão que serão utilizados no trabalho, nos quais se diferem em modelo de regressão linear, modelo de rede neural artificial, e por fim, um modelo de rede neural artificial profunda.

Capítulo 2.4 – Medidas para Avaliação de Desempenho - são apresentadas as medidas para avaliação do desempenho das previsões de séries temporais financeiras, assim como suas respectivas interpretabilidade dos resultados.

Capítulo 3 – Metodologia - descreve o procedimento empregado para a análise da série temporal, escolha dos parâmetros dos modelos propostos, realização das simulações.

Capítulo 4 – Resultados e Discussão - são apresentados os resultados que foram analisados através das medidas de desempenho e gráficos, demonstrando o desempenho individual dos modelos propostos e uma discussão sobre os desempenhos e suas limitações.

Capítulo 5 – Considerações finais - neste capítulo final são apresentadas as considerações finais do trabalho, assim como sugestões para trabalhos futuros, seguidos das referências utilizadas.

2. Referencial Teórico

2.1. Técnicas Descritivas

2.1.1. Série Temporal

De acordo com Box, Jenkins e Reinsel (1994) e Morettin e Toloi (2006), uma série temporal é um conjunto de observações de dados sequencialmente ordenados no tempo, uma série temporal pode ser contínua ou discreta com intervalos constantes entre cada observação. Apesar de se referir ao parâmetro t como sendo o tempo, uma série Z(t) poderá ser função de outros parâmetros, como, por exemplo, latitude, longitude ou volume.

Dessa forma, uma série temporal pode ser definida por Morettin e Toloi (2006) um vetor Z(t), de ordem r × 1, e t sendo um vetor p × 1

$$Z(t) = \{z_t \in R | \, t = 1,2,3 \ldots N\}$$

Se diz que uma série é univariada quando $(r = 1)$ e multivariada quando $(r > 1)$, da mesma forma, tem-se que uma série é unidimensional quando $(t = 1)$ e multidimensional quando$(t > 1)$.

Os dados que constituem uma série temporal podem ser descritos como a soma de suas componentes(características) não observáveis. De acordo com Morettin e Toloi (2006) Z_t pode ser descrita em função de suas componentes como mostra a fórmula abaixo.

$$Z_t = T_t + S_t + a_t \tag{2.1}$$

Na qual T_t e $S_t$representa respectivamente a tendência e a sazonalidade que segundo Araújo (2016) pode ser descrita como tendência(incremento ou decremento gradual sistemático das observações do fenômeno temporal) e a sazonalidade (movimentações cíclicas na observação do fenômeno temporal com flutuações de caráter sazonal, isto é, com frequência bem definida da série temporal)para o vetor t, ao passo que a_té uma componente aleatória ou ruído, que apresenta media zero e variância constante.

Determinados modelos de predições pressupõem que a série temporal possui características específicas para que o modelo obtenha resultados satisfatórios. Nos tópicos seguintes são abordadas algumas condições que tais modelos tem como requisito.

2.1.2. Estacionariedade

Segundo Morettin e Toloi (2006), para a utilização de modelos constantemente se supõem que a série temporal seja estacionária e, de acordo com os autores, uma série é dita estacionária se suas características estatísticas (média, variância e autocorrelação), progridem sobre o vetor t de forma constante ou em um equilíbrio estável, sendo assim, as características

de Z_{t+r} são as mesmas para Z_t. No entanto, é comum que as séries sejam não-estacionárias como, por exemplo, séries temporais financeiras que em geral apresentam componentes de tendência, sazonalidade ou até mesmo explosivos.

Uma série temporal pode ser estacionária por um determinado período, mas pode desenvolver componentes de tendência e/ou sazonalidade durante outro período. Determinados modelos podem lidar com séries estacionárias e não-estacionárias como explica Morettin e Toloi (2006) para o modelo ARIMA, no entanto, têm problemas com séries temporais explosivas.

Visto que a maioria dos modelos utilizados pressupõe que a série temporal seja estacionária, é necessário testar Z_t para averiguar sua estacionariedade. Caso se verifique que Z_t seja uma série temporal não-estacionária, é crucial tratar Z_t a fim de torná-la estacionária. A transformação mais simples e comum para esse processo é feita através da diferenciação sucessiva de Z_t dada pela seguinte equação.

$$\Delta Z_t = [Z_t - Z_{(t-1)}] \tag{2.2}$$

De acordo com Morettin e Toloi (2006) normalmente a diferenciação de primeira ou segunda ordem é suficiente para tornar a série temporal estacionária. Sendo assim, pode-se definir a enésima diferenciação de Z_t na fórmula abaixo.

$$\Delta^n Z_t = \Delta[\Delta^{n-1} Z_t] \tag{2.3}$$

Existem diferentes metodologias para tornar uma série temporal em uma série estacionária no tempo, sendo que no próximo tópico têm-se outros métodos capazes de tornar uma série estacionária.

2.2. Decomposição Clássica

2.2.1. Séries com Tendência

Dada uma série temporal Z_t com a ausência de componente sazonal S_t, pode-se descrever Z_t em função exclusivamente da tendência T_t e sua componente aleatória a_t. Como pode-se ver na equação abaixo.

$$Z_t = T_t + a_t \tag{2.4}$$

De acordo com Morettin e Toloi (2006) existem diversos métodos que são capazes de estimar a tendência T_t, sendo que os mais comuns segundo os autores são através do ajuste de uma função do vetor t(por exemplo, função exponencial e logística), suavização dos valores da série em torno de um ponto para estimar a tendência t_t no ponto (por exemplo, médias móveis), suavização dos valores das séries através do método dos mínimos quadrados ponderados e

também é possível a eliminação da tendência com o método da diferenciação explicitado no tópico anterior.

2.2.2. Séries com Sazonalidade

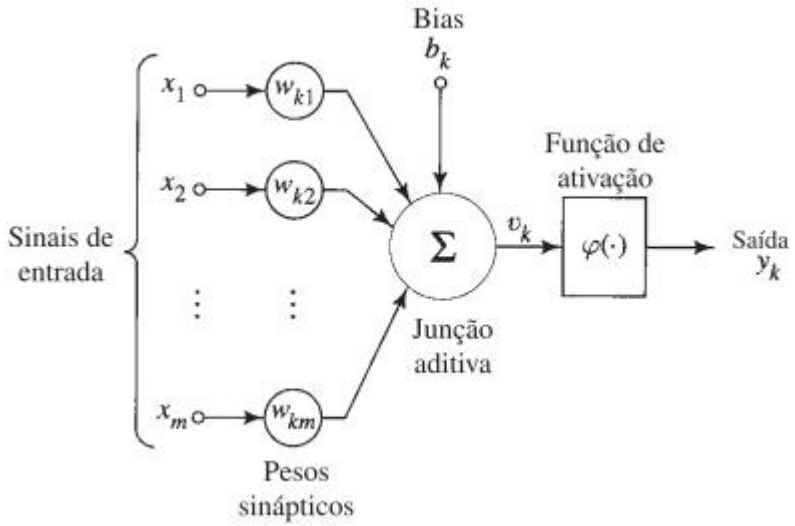
Para Morettin e Toloi (2006) o principal motivo de considerar o modelo de decomposição das componentes de uma série temporal Z_t e para estimar a componente sazonal da série temporal é ser capaz de reconstruir a série livre de sazonalidade. As componentes T_t e S_t frequentemente estão relacionadas e tem influência sobre a outra. Para sazonalidades determinísticas (série temporal cuja componente sazonal não varia com o tempo) os autores sugerem o método de regressão para estimar S_t, enquanto que para sazonalidades estocásticas (série temporal cuja componente sazonal varia com o tempo) propõe o método de médias móveis.

2.3.REDES NEURAIS ARTIFICIAIS

2.3.1. O neurônio artificial

Segundo Haykin (2007) um neurônio artificial e uma unidade de processamento de informação essencial para operação de uma rede neural. No diagrama em blocos da Figura 1 tem-se um modelo não linear de um neurônio onde é possível identificar três características básicas para o modelo neuronal.

Figura 1- Modelo não-linear de um neurônio



Fonte: Haykin (2007)

A primeira característica descrita por Haykin (2007) é o conjunto de sinapses ou elos de conexão que interliga os sinais de entrada com a junção aditiva, desta forma para cada respectivo sinal $x_j$tem-se um peso sináptico w_{kj} associado ao sinal do neurônio k.

A segunda característica descrita pelo autor é o somador que tem como objetivo ponderar os sinais de entrada com suas respectivas sinapses.

A função de ativação ou função restritiva é a última característica citada por Haykin (2007) que tem como objetivo restringir o intervalo permissível de amplitude de um neurônio. Normalmente esse intervalo permissível é escrito como um intervalo unitário fechado [0,1] ou alternativamente [-1, 1].

Matematicamente pode-se descrever um neurônio k com a Equação 2.5 e Equação 2.6.

$$u_k = \sum_{j=1}^{m} w_{kj} x_j + b_k \tag{2.5}$$

$$y_k = \emptyset(u_k) \tag{2.6}$$

Sendo bias b_k um parâmetro externo de um neurônio artificial k, onde de acordo com Haykin tem o efeito de aplicar uma transformação na saída do combinador linear.

Por fim, tem-se a função de ativação que tem o objetivo de reescalar o campo induzido u_k para liminar o intervalo permissível para a saída y_k. A qual é definida no próximo tópico.

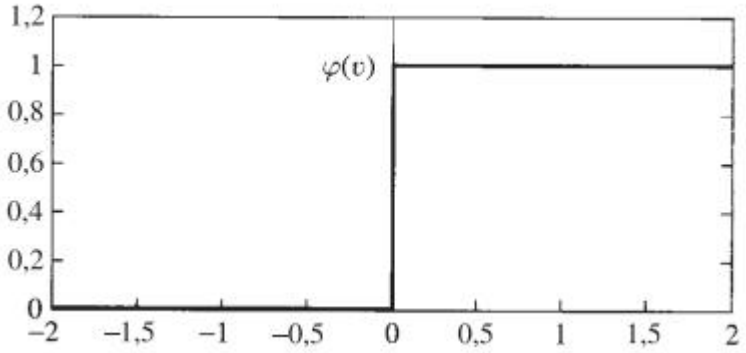
2.3.2. Função de Ativação

De acordo com Haykin (2007) existem três tipos básicos de funções de ativação que definem a saída de um neurônio. A função limiar ou *Heaviside* expressa na Equação 2.7

$$y_k = \begin{cases} 1, & u_k \geq 0, \\ 0, & u_k < 0. \end{cases} \tag{2.7}$$

Neste modelo o neurônio recebe o valor 1 se o campo local induzido pelo neurônio não é negativo, e 0 caso contrário. Essa função faz uso da propriedade tudo-ou-nada. Na Figura 3 pode-se ver o comportamento do campo induzido e a saída y_k.

Figura 2 - Função limiar



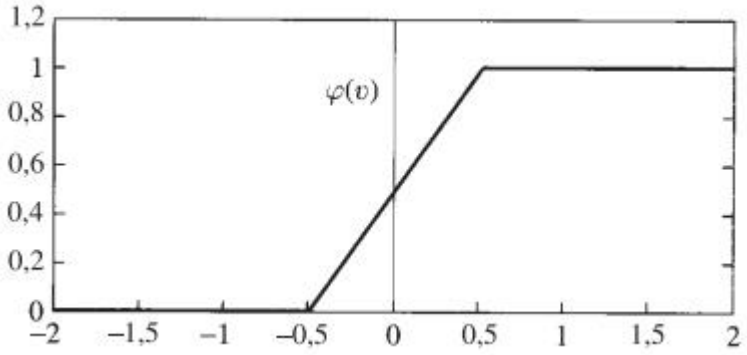
Fonte: Haykin (2007)

A Função Linear por Partes assume um fator de amplificação dentro da região linear de operação que segundo Haykin (2007) pode ser vista como uma aproximação de um amplificador não linear e com as características do modelo por partes tem-se um combinador linear com função de limiar reduzida.

$$\emptyset(u_k) = \begin{cases} 1, & u_k \geq +\frac{1}{2}, \\ u_k, & +\frac{1}{2} > u_k > -\frac{1}{2}, \\ 0, & u_k < -\frac{1}{2}. \end{cases} \tag{2.8}$$

Pode-se ver na Equação 2.8 que o campo induzido para o neurônio k, assume valor fixo e variáveis com o uso da função linear por partes. Como pode-se ver na Figura 3 o comportamento do campo induzido.

Figura 3 - Função limiar por partes



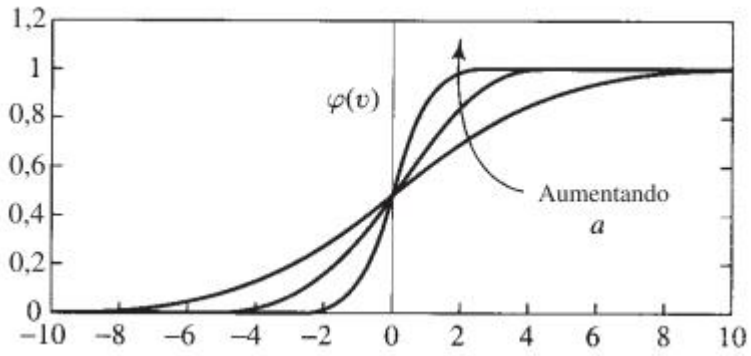
Fonte: Haykin (2009)

De acordo com Haykin (2007) a função sigmóide é a função de ativação mais utilizada na construção de redes neurais artificiais na qual possui um balanceamento adequado entre um comportamento linear e não linear, além de ser estritamente crescente. Pode-se ver sua definição na Equação 2.9.

$$\emptyset(u_k) = \frac{1}{1 + \exp(-\alpha\emptyset(u_k))} \tag{2.9}$$

O parâmetro de inclinação α na função sigmóide permite que a função tenha diferentes inclinações como se pode ver na Figura 4. Se o parâmetro de inclinação $\alpha \rightarrow \infty$ tem-se uma função limiar. Sendo a principal característica da função sigmóide a diferenciabilidade que de acordo com Haykin (2007) é uma característica importante para redes neurais.

Figura 4 - Função sigmóide com parâmetro de inclinação α



Fonte: Haykin (2009)

Como citado anteriormente, em determinadas ocasiões é desejável que a função de ativação assuma valores entre [-1, 1]. Para a função limiar da pode-se obter esse intervalo modificando-a como na Equação 2.10. Enquanto que para a função sigmóide pode-se utilizar a função tangente hiperbólica definida na Equação 2.11.

$$\emptyset(u_k) = \begin{cases} +1, & u_k > 0, \\ 0\,, & u_k = 0\,, \\ -1, & u_k < 0. \end{cases} \tag{2.10}$$

$$\emptyset(u_k) = \tanh(u_k) \tag{2.11}$$

Pode ser visto que a melhor função de ativação a ser usada tem relação direta com o tipo de problema a ser resolvido, assim como o modelo de treinamento e o processo de aprendizagem da rede neural que são apresentados nos próximos tópicos.

2.3.3. Algoritmo de aprendizagem

De acordo com Haykin (2009, p79) no contexto de redes neurais "aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre."

Para Haykin (2007) o algoritmo de aprendizagem de um modelo de rede neural tem suma importância para melhor extrair conhecimento dos dados a partir de seu ambiente e obter melhor desempenho para a rede neural, além de contribuir para o processo interativo para obter o melhor ajuste dos pesos sinápticos da rede neural. Para o autor o processo de aprendizagem implica em eventos subsequentes no qual a rede neural é estimulada por um ambiente, seus pesos sinápticos são modificações como resultado do estimulo, pôr fim a rede neural responde de uma maneira nova ao ambiente devido a modificações ocorridas em sua estrutura.

Um conjunto de regras bem definidas para a solução de problemas de aprendizagem pode ser considerado um algoritmo de aprendizagem de acordo com Haykin (2007). Uma vez que os problemas de aprendizagem dependem de um ponto de vista surge diferentes abordagens para um único projeto de redes neurais, tais abordagens de forma geral se diferem entre si pelas regras de ajuste dos pesos sinápticos de um neurônio ou pela topologia da rede neural.

O processo de aprendizagem por correção de erro é uma regra de aprendizagem que utiliza o sinal de erro da saída de cada iteração do modelo de rede neural para ajustar seus pesos sinápticos e bias Haykin (2007). O sinal de erro é obtido através da diferença entre o sinal de saída do modelo com a resposta desejada ou saída-alvo e aciona um mecanismo de controle

com objetivo de ajuste corretivo dos pesos sinápticos de um neurônio k para aproximar a cada iteração o sinal de saída y_k com a resposta desejada d_k. De acordo com Haykin (2007) esse objetivo é alcançado pela minimização de uma função de custo definida em termos do sinal de erro. A minimização da função de custo resulta, de acordo com Haykin (2007), na regra de aprendizagem conhecida como regra delta, o ajuste dos pesos sinápticos do neurônio será feito através da retropropagação do sinal de erro do neurônio que será aplicado um ajuste proporcional ao produto do sinal de erro pelo sinal de entrada da sinapse. Diante da natureza da regra delta em que se pressupõem o conhecimento da resposta desejável por alguma fonte externa esta deve ser mensurável e de acordo com o autor se ter cuidado com a taxa de aprendizagem definida na regra delta para se obter uma estabilidade ou convergência no processo de aprendizagem iterativo.

O processo de aprendizagem baseado em memória é outra regra de aprendizagem que se utiliza do armazenamento explícito dos exemplos de entrada-saída classificados para cada amostra (HAYKIN, 2007). De acordo com Haykin (2007), todos os algoritmos de aprendizagem baseada em memória possuem duas características essenciais que os diferem, o critério utilizado para definir a vizinhança do local do vetor de teste e a regra de aprendizagem aplicada nos exemplos de treinamento na vizinhança local do teste.

A forma mais simples e efetiva de aprendizagem baseada em memória, de acordo com Haykin (2007), é conhecida como regra do vizinho mais próximo em que a vizinhança local é exemplo de treinamento para o vetor de teste mais próximo, que contém metade da informação necessária para a classificação. Uma variante do classificador pelo vizinho mais próximo é o classificado pelos k vizinho mais próximo que identifica os k padrões que se encontram mais próximos do vetor de teste e atribui classe que está representada nos k vizinhos mais próximos.

Por fim, têm-se diversas outras abordagens a respeito de algoritmos de aprendizagem como por exemplo a aprendizagem hebbiana, aprendizagem competitiva, aprendizagem boltmann, aprendizagem com ou sem um professor. No entanto, foram utilizados para este trabalho exclusivamente os algoritmos de aprendizagem por correção do erro e baseados em memória.

2.4.MODELOS DE PREVISÃO

Nesse capitulo são apresentados os modelos de previsão que serão utilizados no trabalho, nos quais se diferem em modelo de regressão linear, modelo de rede neural artificial, e por fim, um modelo de rede neural artificial profunda.

2.4.1. Autorregressivo Integrado de Médias Móveis (ARIMA)

De acordo com Morettin e Toloi (2006) a construção do modelo ARIMA (p, d, q) é feita através de um ciclo iterativo que consiste em ajustar modelos autorregressivos integrados de médias móveis com base nos próprios dados. Os estágios iterativos são:

(a) Uma classe geral de modelos é considerada para análise (especificação);
(b) Há identificação de um modelo, com base na análise de Função de Autocorrelação (FAC) e Função Autocorrelação Parcial (FACP) e outros critérios;
(c) Nesta fase os parâmetros do modelo identificado são estimados;
(d) Há verificação ou diagnóstico do modelo ajustado, através de uma análise de resíduos para saber se este é adequado para os fins de previsão.

Por fim, caso o modelo não seja adequado (fornecer o menor erro quadrático médio de previsão) o ciclo interativo é repetido voltando para a fase de identificação. De acordo com Morettin e Toloi (2006) a fase de identificação é crítica para o problema de séries temporais devido ser possível identificar modelos diferentes para uma mesma série temporal.

Segundo Morettin e Toloi (2006) o procedimento de identificação consiste em três partes, verifica se existe a necessidade de transformação da série original para estabilizar sua variância, tomar diferenciações até a obtenção de uma série temporal estacionária e o processo ARIMA (p, d, q) seja reduzido a um modelo ARMA(p,q). E por fim, identificar o processo resultante através da análise da autocorrelação e autocorrelação parcial.

Para Morettin e Toloi (2006) se a série temporal for estacionária, esta pode ser representada pelo modelo ARMA. Enquanto que se a série temporal for estacionária por um processo de diferenciação, esta segue um modelo autorregressivo, integrado, de médias móveis ou ARIMA, que é definida de acordo com a Equação 2.12.

$$\emptyset(B)\Delta^d Z_t = \theta(B)a_t \tag{2.12}$$

Onde Δ^d é o operador de diferenciação calculado pela expansão binomial, desta forma tem-se $\Delta^d Z_t = \alpha_0 + (-1)^{(d-1)}\alpha_1 z(t-1) + (-1)^{(d-2)}\alpha_2 z(t-2) \ldots + \alpha_d z(t-d)$, sendo $\emptyset(B)$ o operador autorregressivo estacionário de ordem p que pode ser denotado por $\emptyset(B) = 1 - \emptyset_1 B - \cdots - \emptyset_p B$. E por fim, $\theta(B)$operador de medias moveis de ordem q denotado por $\theta(B) = 1 - \theta_1(B)^1 - \cdots - \theta_q(B)^q$ (MORETTIN e TOLOI, 2006).

No próximo tópico tem-se uma prévia da descrição do funcionamento de uma MLP através de sua topologia e o fluxo de sinais com um grafo arquitetural.

1.4.Rede *Multilayer Perceptron (MLP)*

2.4.1.1. Aspectos gerais

As redes de múltiplas camadas alimentadas adiante são consideradas por Haykin (2007) uma importante classe de redes neurais. Elas consistem em unidades sensoriais que se propagam a informação da camada de entrada para uma ou mais camadas ocultas de neurônios e por fim interligando a camada de saída, essas redes são conhecidas usualmente como perceptrons de múltiplas camadas (*Multilayer Perceptron* - MLP). De acordo com Haykin (2007) a MLP é uma generalização do modelo de neurônio visto no capítulo 3.

Segundo Haykin (2007) os perceptrons de múltiplas camadas com o treinado supervisionado têm sido capazes de solucionar diversos problemas complexos através de seu algoritmo de retropropagação do erro que faz os ajustes dos pesos sinápticos da rede neural. O algoritmo de retropropagação do erro pode ser visto como uma generalização do algoritmo do mínimo quadrado médio visto no capítulo 3.

O processo de aprendizagem por retropropagação do erro consiste, de acordo com Haykin (2007) em dois passos. A propagação dos sinais de entrada para a frente na qual os valores dos pesos sinápticos da rede neural se mantêm fixos, resultando em vetor de saída e um erro associado. Na retropropagação os pesos sinápticos da rede neural são ajustados de acordo com uma regra de correção de erro com objetivo de minimizar o erro associado à saída para a próxima iteração da rede.

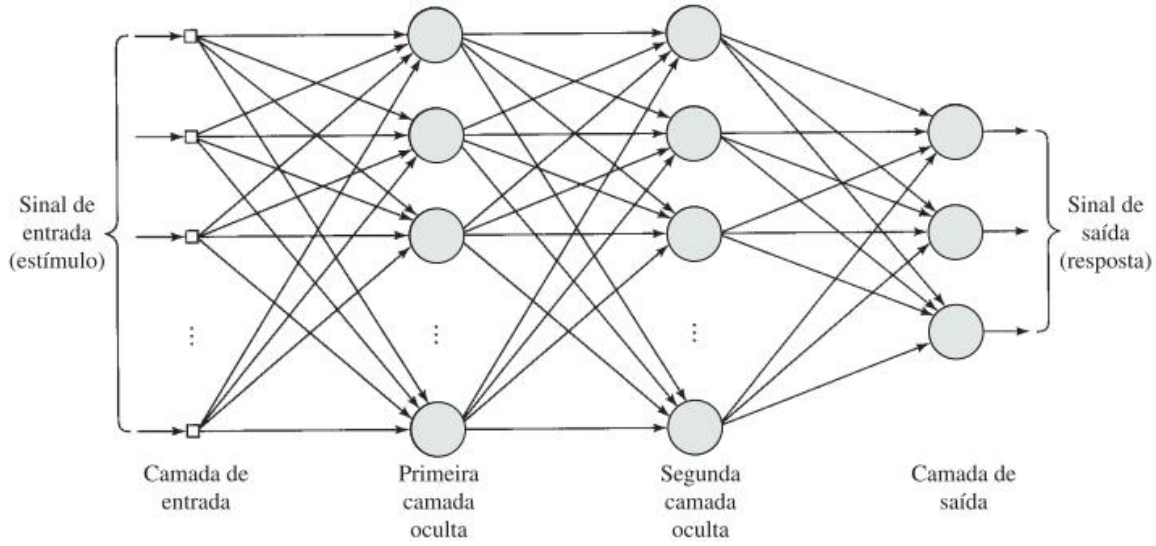
De acordo com Haykin (2007) uma MLP possui três características distintivas. (1) O modelo de cada neurônio da rede inclui uma função de ativação não-linear diferençável em qualquer ponto que, como foi visto anteriormente, é comum a utilização da função de ativação sigmóide. (2) A rede contém uma ou mais camadas de neurônios ocultos que não são partes do vetor de entrada e que possibilita a rede aprender tarefas complexas. Por fim, (3) a rede apresenta um alto grau de conectividade devido a conexão das sinapses da rede, que segundo Haykin (2007) é através dessas características que as redes neurais derivam seu poder computacional e o mesmo que gera a deficiência no estado de conhecimento a respeito do comportamento da rede.

No próximo tópico tem-se uma prévia da descrição do funcionamento de uma MLP através de sua topologia e o fluxo de sinais com um grafo arquitetural.

2.4.1.2. Topologia da rede

Na Figura 5 pode-se ver uma das etapas para treinamento de uma rede MLP descrita no tópico anterior, tem-se a progressão do fluxo de sinal para frente (da esquerda para direita) com uma rede totalmente conectada que significa que todos os nós ou neurônios estão interligados com os demais (HAYKIN, 2007).

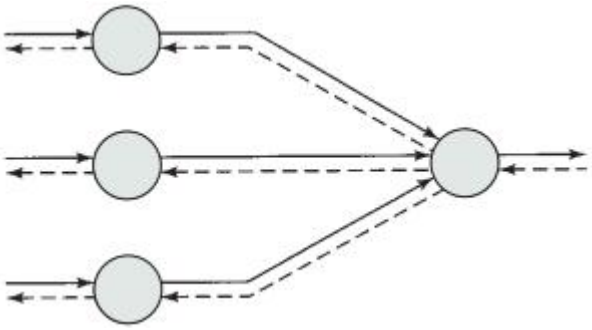
Figura 5 - Grafo Arquitetural de um perceptron de múltiplas camadas



Fonte: Haykin (2007)

Na Figura 6 tem-se uma representação do fluxo de sinais de uma MLP que Haykin (2007) separa em sinais funcionais e sinais de erro. De acordo com autor, os sinais funcionais são os estímulos que incidem no terminal de entrada da rede e se propagam para frente até a saída. E o sinal de erro sendo o que se origina a partir da saída da rede neural em comparação com a saída desejada e se retropropaga pela rede para ajuste dos pesos sinápticos.

Figura 6 - Fluxo de sinais básicos de um perceptron de múltiplas camadas



Fonte: Haykin (2007)

De acordo com Haykin (2007), cada neurônio na camada oculta tem a função de calcular o sinal que será expresso como uma função não-linear do sinal de entrada e pesos sinápticos

associados à unidade e calcular uma estimativa do vetor gradiente que será necessário para a retropropagação do erro.

No próximo tópico é aprofundado como o erro se retropropaga através da rede MLP e como é feito o ajuste dos pesos sinápticos.

2.4.1.3. Algoritmo de aprendizagem

O sinal de erro em um nó de saída de um neurônio j pode ser visto na Equação 2.13 como discutido nos tópicos anteriores.

$$\varepsilon_j(n) = d_j(n) - y_j(n) \tag{2.13}$$

De acordo com Haykin (2007) o valor instantâneo da energia total do erro pode ser obtido somando-se todos os sinais de erro das saídas como mostra a Equação 2.14.

$$E(n) = \frac{1}{2}\sum_{j \in C} {\varepsilon_j}^2(n) \tag{2.14}$$

Onde n representa o número total de padrões apresentados para a rede que estão contidos no conjunto de treinamento e C o conjunto de neurônios que estão na camada de saída da rede neural (HAYKIN, 2007). A energia *média do erro quadrado* é então obtida através da normalização do valor instantâneo da energia total do erro para todas as entradas apresentadas, como se pode ver na Equação 2.15.

$$E_{med} = \frac{1}{N}\sum_{n=1}^{N} E(n) \tag{2.15}$$

A energia instantânea média do erro, segundo Haykin (2007), é uma função de todos os parâmetros livres (pesos sinápticos e níveis de bias) da rede. Desta forma, para um conjunto de treinamento E_{med} representa a função custo como uma medida do comportamento de aprendizagem da rede neural que tem como objetivo ajustar os parâmetros livres da rede para minimizar a função custo.

O algoritmo de retropropagação aplica uma correção $\Delta w_{kj}(n)$ ao peso sináptico, de acordo com Haykin (2007), que é proporcional à derivada parcial $\partial E(n)/w_{ji}(n)$ que de acordo com a regra da cadeia do cálculo pode-se expressar o gradiente local como mostra na Equação 2.16.

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)}\frac{\partial e_j(n)}{\partial y_i(n)}\frac{\partial y_i(n)}{\partial u_{ji}(n)}\frac{\partial u_{ji}(n)}{\partial w_{ji}(n)} \tag{2.16}$$

A correção do peso $\Delta w_{kj}(n)$ aplicada ao peso sináptico pode ser feita através da regra delta definida na Equação 2.17.

$$\Delta w_{kj}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \tag{2.17}$$

Onde η é o parâmetro da taxa de aprendizagem do algoritmo de retropropagação na qual indica o tamanho do passo que será dado em direção ao gradiente e o sinal negativo indica a descida do gradiente que tem relação com o objetivo de minimizar a função de custo. Por fim, de acordo com Haykin (2007) pode-se resumir o algoritmo de retropropagação para correção de $w_{kj}(n)$ como o produto entre a taxa de aprendizagem, o gradiente local e o sinal de entrada do neurônio j.

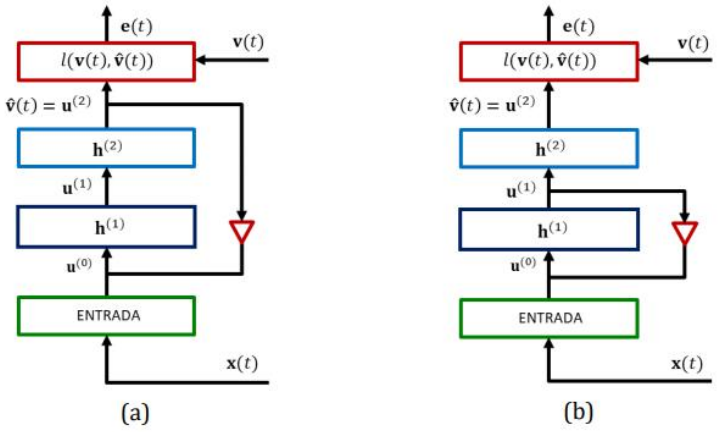
2.4.2. Long Short-Term Memory (LSTM)

O modelo de Jordan foi a primeira rede neural recorrente de acordo com Evsukoff (2020), na qual não utilizada explicitamente variáveis de estado e utilizava um vetor de ativação da última camada como um vetor de estados que é reinjetado na entrada do modelo. Esse ciclo de retroalimentação pode ser representado por um operador de defasagem (shift) como se pode ver na Figura 7(a) o triângulo vermelho.

Nas redes recorrentes existe pelo menos um operado de defasagem que faz a retroalimentação, em que o sinal retorna para uma camada anterior. De acordo com Evsukoff (2020), o modelo de Elman é uma rede neural de duas camadas onde o operador de defasagem acontece na primeira camada e reinjeta sua saída no instante seguinte como se pode ver na Figura 7(b)

Figura 7 - Modelos de redes neurais recorrentes:

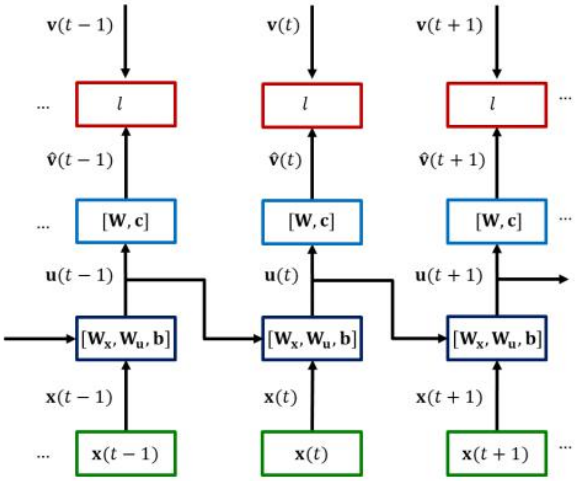
(a) modelo erro de saída não linear de Jordan; (b) modelo de espaço de estados não linear de Elman



Fonte: Evsukoff (2020)

As redes neurais recorrentes podem ser desdobradas no tempo em camadas, permitindo que a construção de diferentes topologias de redes e solucionando os problemas relacionados com o operador de defasagem e os métodos de treinamento para ajustes dos parâmetros da rede (EVSUKOFF, 2020). Na Figura 8 pode-se ver uma rede de Elman desdobrada no tempo em camadas com parâmetros compartilhados na qual a camada recorrente é conectada com o instante de tempo seguinte, tornando o processo sequencial no tempo.

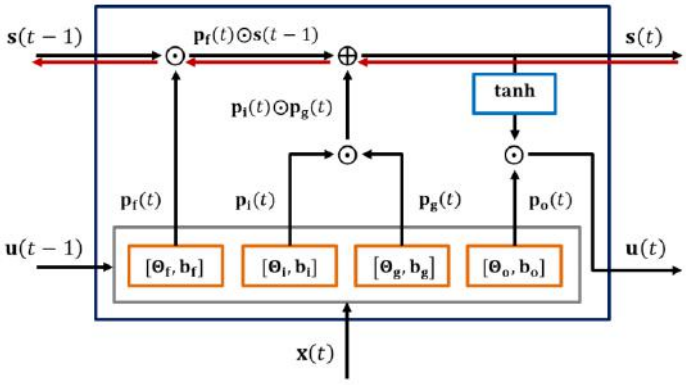
Figura 8 - Rede neural recorrente desdobra no tempo



Fonte: Evsukoff (2020)

De acordo com Evsukoff (2020), o modelo de rede neural com memória longa de curto prazo (LSTM) foi introduzido por Hochreiter e Schmidhuber. O modelo proposto utiliza da camada recorrente do modelo como uma célula de memória que contém o vetor de estados internos da rede, e um conjunto de portas que controlam o fluxo de informação dentro do neurônio. Na Figura 9 pode-se ver as portas que controlam o fluxo de informação, a porta $P_f(t)$do "esquecimento" tem o objetivo de limitar a proporção da ativação do vetor de estados interno na retropropagação do erro para o item anterior, as portas $P_g(t)$ e $P_i(t)$ controlam o fluxo de informação do vetor de entrada evitando ruído e informações irrelevantes e porta $P_o(t)$ controla a proporção do estado interno que será utilizada para atualização dos pesos no estado recorrente da rede através do algoritmo (*Backpropagation Through Time* – BPTT) apresentada por Graves.

Figura 9 - Visão Esquemática da Camada LSTM



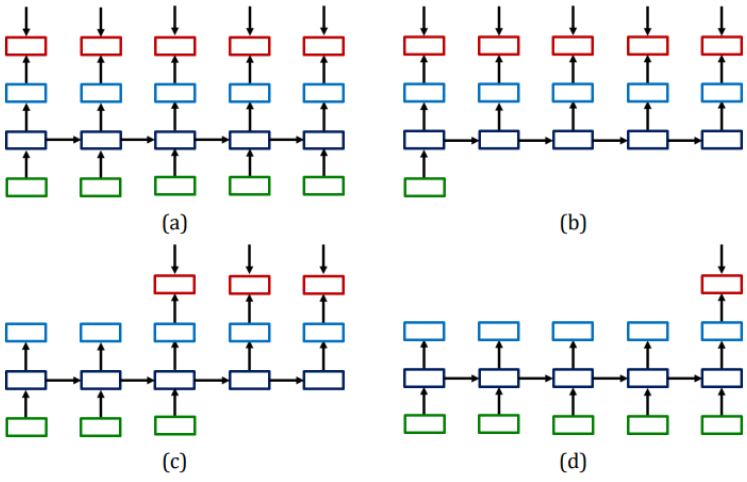
Fonte: Evsukoff (2020)

Segundo Evsukoff (2020), os diferentes modelos de LSTM têm sido utilizados com sucesso em diferentes aplicações como reconhecimento de fala, tradução automática, legendagem de imagens e previsão de séries temporais no mercado financeiro. No próximo tópico é aprofundado o funcionamento da rede LSTM proposta por Gers , Schmidhuber e Cummins (2000.)

2.4.2.1. Topologia da rede

As redes neurais recorrentes desdobradas no tempo são redes em camadas com parâmetros compartilhados que, de acordo com Evsukoff (2020), permitem a construção de diferentes topologias conforme a necessidade dos problemas. As camadas de entrada das redes desdobradas no tempo (Figura 11) permitem a combinação de uma ou mais camadas de modelos diferentes.

Figura 10 - Topologias de RNN desdobradas no tempo



Fonte: Evsukoff (2020)

Diferentes topologias são adequadas a problemas específicos Evsukoff (2020). Na Figura 10(a) tem-se uma topologia completa com múltiplas entradas e múltiplas saídas onde são indicadas para problemas de sistemas dinâmicos ou sequências de imagens. Na Figura 10(b) se tem uma topologia com única entrada e múltiplas saídas na qual são indicadas para problemas como no caso que uma imagem é apresentada como entrada e uma sequência de palavras na saída com a descrição da imagem. Na Figura 10(c) tem-se uma topologia com múltiplas entradas e múltiplas saídas que pode ser usada em aplicações de tradução. Por fim, na Figura 10(d) tem-se uma topologia com múltiplas entradas e única saída, um exemplo para seu uso e no caso da análise de sentimentos em que a saída do modelo e um conceito em função da sequência de palavras de entrada.

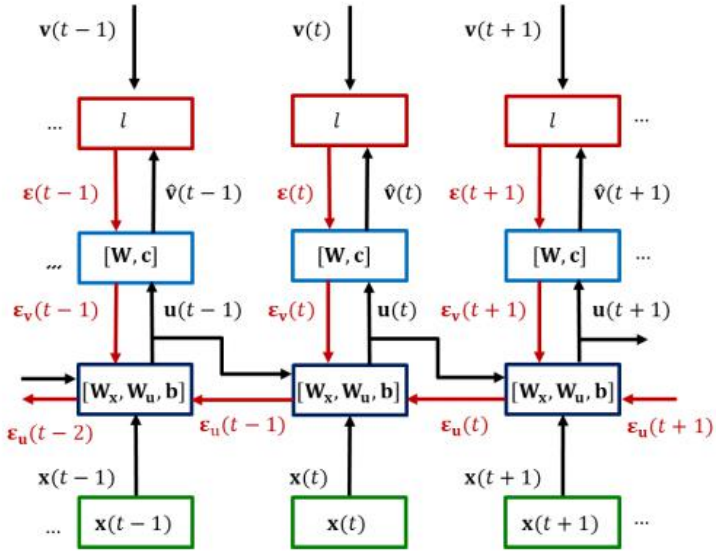
O ajuste dos parâmetros da LSTM é realizado pelo algoritmo de retropropagação através do tempo que será visto no tópico seguinte.

2.4.2.2. Algoritmo de aprendizagem

De acordo com Evsukoff (2020) a camada da rede neural recorrente (RNN) padrão representa a dinâmica temporal do sistema em $u(t)$ e o vetor de estados $u(t-1)$ armazena toda a informação necessária para a previsão de $u(t)$. Exigindo um processamento sequência no tempo no qual o próximo vetor de estados possa ser processado apenas com a finalização do anterior. Na direção da saída da rede neural seu processamento se assemelha ao de uma rede neural MLP com uma entrada adicional de $u(t-1)$ para a camada recorrente.

O algoritmo BPTT é aplicado de forma similar ao que acontece na rede MLP supervisionada, sendo na direção reversa do processamento da saída da RNN e chegando para ajuste dos pesos sinápticos de acordo com a topologia da rede até o vetor de estado por meio da regra da cadeia como pode-se ver na Figura 11.

Figura 11- Retropropagação do erro horizontal e vertical na BPTT



Fonte: Evsukoff (2020)

Nos modelos de redes recorrentes o gradiente local pode ser um problema em dois casos (EVSUKOFF, 2020): (i) quando o gradiente local diverge ($\delta_{1i} \rightarrow \infty$) que pode ser resolvido com a técnica de corte do gradiente; (ii) ou quando o gradiente local se anula ($\delta_{1i} \rightarrow 0$) que impede a modelagem de longas sequências. De acordo com o autor esse problema pode ser resolvido utilizando modelos com portas como por exemplo as LSTM que permite um fluxo direto da retropropagação do erro como pode-se ver na Figura 11 na linha em vermelho com o fluxo de retropropagação do erro nos estados internos do modelo sem a necessidade de uma multiplicação matricial como nas redes RNN.

2.5.MEDIDAS PARA AVALIAÇÃO DE DESEMPENHO

O erro médio quadrático (*Mean Squared Error* - MSE) é a principal e mais utilizada medida para avaliação da previsão de séries temporais definida por Clements e Hendry (1993) como pode-se ver na Equação 2.18.

$$\mathrm{MSE} = \frac{1}{\mathrm{N}}\sum_{\mathrm{j}=1}^{\mathrm{N}}(\mathrm{e_j})^2 \tag{2.18}$$

onde N representa a quantidade de predições, $\mathrm{e_j}$ o erro instantâneo para a predição no tempo j, que pode ser definido pela Equação 2.19.

$$\mathrm{e_j} = (\mathrm{x_j} - \hat{\mathrm{x}}_\mathrm{j}) \tag{2.19}$$

em que o valor previsto da série temporal no tempo j e representado por $\hat{x}_j$ é o valor real para a série temporal no tempo j e dado por x_j. Nota-se que, o modelo ideal de predição tem o $MSE \rightarrow 0$. Tal medida, frequentemente é utilizada em modelos de redes neurais em seu processo de aprendizagem, no entanto, outras medidas de desempenho podem ser consideradas para comparar os diferentes modelos.

O erro médio percentual absoluto (*Mean Absolute Percentage Error* - MAPE) é uma medida que fornecendo o tamanho médio do erro percentual permite identificar precisamente os desvios percentuais do modelo de previsão, o modelo de previsão ideal tem $MAPE = 0$, sua definição e dado por Clements e Hendry (1993) .Como pode-se ver na Equação 2.20.

$$\mathrm{MAPE} = \frac{1}{N}\sum_{j=1}^{N}\left|\frac{e_j}{x_j}\right| \tag{2.20}$$

A estatística U de theil (u *of theil statistic*, THEIL) definida por Hann e Steurer (1996) é outra forma de avaliar um modelo de previsão em séries temporais em relação a perspectiva da existência de um passeio aleatório, que pode ser descrito Sitte e Sitte (2002) como a adição de um ruído com distribuição gaussiana ao valor anterior da série temporal como fator gerador do instante seguinte. Sua definição matemática é dada na Equação 2.21.

$$\mathrm{THEIL} = \frac{\frac{1}{N}\sum_{j=1}^{N-1}(e_j)^2}{\frac{1}{N}\sum_{j=1}^{N}(\hat{x}_j - \hat{x}_{j-1})^2} \tag{2.21}$$

A interpretação para a estatística U de theil é feita em comparação ao desempenho de um passeio aleatório, THEIL ≥ 1 indica que o modelo de previsão utilizado tem desempenho igual ou pior que um passeio aleatório. Em contrapartida, tem-se que se THEIL ≤ 1indica que o modelo de previsão utilizado tem desempenho igual ou melhor que um passeio aleatório, desta forma se tem que para o modelo ideal de previsão $\mathrm{THEIL} = 0$.

A medida previsão de mudança de direção (*Prediction Of Change In Direction* - POCID) registra o desempenho da previsão em relação à direção futura da série temporal, sendo possível saber o percentual de vezes que o modelo foi capaz de acertar que o valor do futuro da série temporal subiu ou desceu Yao e Tan (2000).Sua definição matemática é dada na Equação 2.22 e Equação 2.23.

$$\mathrm{POCID} = \frac{100}{N}\sum_{j=1}^{N}(D_j)^2 \tag{2.22}$$

$$D_j = \begin{cases} 1, & se\ (x_j - x_{j-1})(\hat{x}_j - \hat{x}_{j-1}) > 0, \\ 0, & caso\ contrário. \end{cases} \tag{2.23}$$

O coeficiente de determinação ou R-quadrado é uma medida estatística que indica o quão bem os resultados do modelo são ajustados aos dados reais. O modelo ideal de previsão tem R2 = 1. Sua definição formal está na Equação 2.24.

$$R^2 = 1 - \frac{\sum(e_j)^2}{\sum(y_j - y)^2} \tag{2.24}$$

No capítulo seguinte tem-se a discussão da metodologia adotada no trabalho assim como a decisão por trás da escolha de determinados parâmetros e ajustes para os modelos utilizados.

3. METODOLOGIA

Nesse capítulo do trabalho é apresentado a caracterização da pesquisa e os devidos procedimentos metodológicos implementados. De acordo com Vergara (2000) pode classificar quanto a utilização dos resultados como pesquisa aplicada, por ênfase na solução do problema. Quanto ao método pode-se classificar como sendo uma pesquisa quantitativa, visto que essa usa métodos estatísticos, procedimentos sistemáticos e quantifica os dados. Quanto aos fins pode-se classificar como descritiva por expor as características da série temporal. Por fim, quanto aos meios pode-se classificar como bibliográfica por utilizar os materiais como livros e revistas como referência e *Ex Post Facto* por apresentar experimentos com acontecimentos posteriores aos fatores gerados.

O ponto de partida do presente trabalho deu-se pela pesquisa bibliográfica onde se recuperou os modelos estatístico e econométrico, bem como os estudos aplicados em redes neurais artificiais e aprendizado profundo para a predição de séries temporais.

A metodologia adotada apresenta limitações quanto a fonte de coleta dos dados e seu tratamento. No que se refere a coleta de dados, pode-se dizer que a dificuldade está na obtenção de informações fidedignas a respeito da realidade da série temporal estudada. No tratamento dos dados a limitação é observada principalmente nos possíveis dados faltantes/feriados e nos métodos utilizados para sua correção. Quanto aos fins a pesquisa limita-se por não existir controle sobre o fenômeno de interesse e utilizar-se de conhecimento passado que leva a possíveis perdas de informações. Por fim, nos resultados apesar da tentativa de generalização dos modelos com métodos estatísticos, têm-se limitações devido ao grupo de teste representar apenas uma amostra da população.

O presente trabalho contou com os seguintes procedimentos metodológicos para a consecução de seus objetivos que serão apresentados em detalhes nos tópicos seguintes.

3.1. Coleta e Tratamento dos dados

O Ibovespa (Índice Bovespa) é o principal índice de ações da Bolsa de Valores, Mercadorias e Futuros de São Paulo (BM&FBOVESPA). Seu objetivo é refletir a média de desempenho das ações com maior representatividade do mercado acionário brasileiro e, para isso, o Ibovespa é composto pelas ações com maior volume de negociação, apesar de não ser uma ação é possível o investimento no mesmo por outros métodos.

A coleta dos dados foi efetuada através do *'website' Yahoo! Finance* por sua biblioteca *Yahooquery* do *Python* que faz a importação dos dados relativos ao IBOVESPA no período de 01-01-2000 a 01-06-2021.

Os dados coletados foram armazenados localmente e delimitados aos valores diários de abertura do índice para obtenção de uma série temporal univariada. Todos os dados faltantes do conjunto de dados eram feriados, para estes foi incluída a média do valor anterior e posterior ao valor ausente.

Antes da realização da modelagem preditiva da série temporal, foi feita a transformação e normalização da série para analisar seu comportamento (estacionariedade) estatístico e verificar a presença ou não de raízes unitárias que é discutido com mais detalhes no tópico seguinte.

3.2. Análise da Série Temporal

De acordo com Zhang, Patuwo e Hu (1998) todas as séries temporais devem passar por um processo de normalização com o objetivo de prover conformidade entre os valores da série temporal e os valores preditos pelo modelo. É possível encontrar diferentes maneiras de realizar a normalização dos dados que dependerá do problema em questão. Nos tópicos seguintes, tem-se as normalizações utilizadas no presente trabalho.

3.2.1. Normalização linear da série de retorno

As séries originais foram transformadas em série de retorno, onde foi aplicada a primeira diferença logarítmica dada pela Equação 3.1, onde $R_{z(t)}$ é o valor de retorno da série no tempo t e z_t é o valor da série original no tempo t. Essa transformação torna a série mais estacionária removendo sua tendência conforme discutido nos capítulos anteriores.

$$R_{z(t)} = \log\left(\frac{z_t}{z_{t-1}}\right) \tag{3.1}$$

Para prover conformidade e um bom condicionamento de processos numéricos de otimização a série de retornos $R_{z(t)}$ foi submetida a uma normalização linear dos dados para o intervalo [0,1].

No tópico seguinte, se tem o teste de estacionariedade da série temporal através de um teste estatístico.

3.2.2. Augmented Dickey-Fuller (ADF)

Conforme discutido nos capítulos anteriores, a modelagem preditiva dos modelos pressupõe um comportamento de estacionariedade da série temporal. Uma das estatísticas que auxilia na detecção da presença da estacionariedade e o Teste ADF que testa duas hipóteses: H_0: tem raiz unitária (série não é estacionária) ou H_1: não tem raiz unitária (série é estacionária). Confirmada H_1, tem-se uma série com os pré-requisitos satisfeitos para submete-la aos modelos

preditivos. Em casos de confirmação de H_0, serão necessárias novas rodadas de diferenciações da série temporal até torná-la estacionária.

A escolha da janela temporal tem impacto direto no desempenho do modelo preditivo. No tópico seguinte tem-se a metodologia adotada para a escolha da melhor janela temporal para a série utilizada no trabalho.

3.3. Janela Temporal

Neste trabalho a janela temporal foi construída de acordo com o teorema de Imersão de Takens (1980) que fornece a base teórica para a predição de séries temporais não-lineares, onde o valor futuro da série temporal pode ser descrito através de uma função de mapeamento da janela temporal de z_t. Desta forma, sendo necessária a escolha do melhor tamanho de janela temporal para um bom mapeamento da função que aproxima z_{t+1}.

Nos capítulos seguintes, tem-se os métodos utilizados para a determinação do melhor tamanho de janela temporal para a predição do dia seguinte.

3.3.1. Função de Autocorrelação e Informação Mútua Média

A função de autocorreção é uma medida quantitativa da dependência temporal entre amostras consecutivas e uma das principais ferramentas para estimação de independência entre termos. Uma escolha comum para o tamanho da janela temporal é quando a FAC atinge seu primeiro valor nulo, na qual os termos passam a ser linearmente não-correlacionados. Outra regra utilizada para escolha do tamanho da janela temporal é quando a FAC decai para $\frac{1}{e}$ (KANTZ e SCHREIBER, 2003) ou até mesmo o primeiro mínimo da FAC.

Uma objeção aos procedimentos mencionados anteriormente é que o tamanho da janela temporal determinada através da FAC é baseado em estatísticas lineares e não leva em considerações as correlações não-lineares (KANTZ e SCHREIBER, 2003). Para lidar com correlações não-lineares foi utilizado um critério que se baseia numa medida de independência geral. Desta forma, para a escolha do tamanho da janela temporal foi utilizado em conjunto com a FAC o cálculo da informação mútua média que utiliza a entropia de Shannon para se criar um histograma dos dados com a probabilidade de que o sinal assuma um valor do histograma.

Após a normalização da série temporal e determinação do melhor tamanho de janela temporal, faz-se necessária a determinação dos parâmetros dos modelos utilizados e a divisão do conjunto de dados, os quais são feitos nos tópicos seguintes.

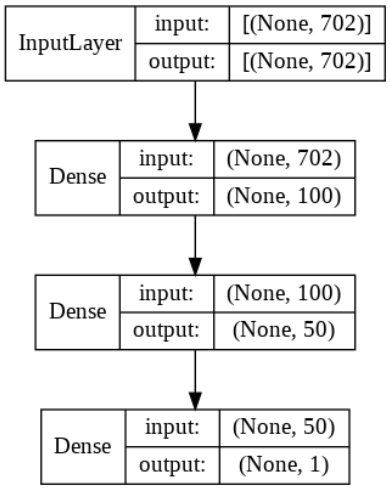
3.4. Seleção dos Parâmetros

Foi utilizada a biblioteca “Keras” do Python para construção dos modelos aplicados neste trabalho, com vinte rodadas de treinamento para cada modelo.

A série temporal diferenciada foi dividida em conjunto de treinamento (70%) para o processo de aprendizagem dos modelos, conjunto de validação (20%) para possível parada prematura para evitar sobre-treino e/ou sobre-ajuste, e bloco de teste (10%) para confirmar o desempenho dos modelos de predição.

Para taxa de aprendizagem dos modelos MLP, CNN e LSTM foi utilizado neste trabalho o algoritmo “Adam” com os valores padrões disponíveis na biblioteca do “keras” e para função perda do modelo foi utilizado o erro médio quadrático (Mean Squared Error, MSE).

Figura 12- Arquitetura MLP



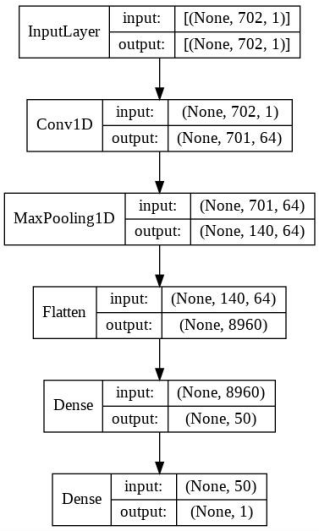
Fonte: Elaborado pelos Autores (2021)

Para o modelo MLP, foram utilizadas três camadas totalmente conectadas com função de unidade linear retificada (ReLU) e função de ativação sigmóide na camada de saída conforme a Figura 12.

Na Figura 13 tem-se a arquitetura do modelo CNN utilizada neste trabalho, que consiste em uma camada de convolução com 64 filtros com função de unidade linear retificada (ReLU) para extrair o mapa de características dos dados de entrada com um ‘*kernel*’. Em seguida, a camada de subamostra (*Maxpooling*), para reduzir o tamanho dos mapas de características e uma camada para achatar o vetor anterior em um vetor unidimensional. Imediatamente duas camadas totalmente conectadas como função de ativação a função de unidade linear retificada (ReLU) para evitar o desaparecimento do gradiente. Por fim, a camada de saída totalmente conectada

com função de ativação sigmóide para predizer os valores de fechamento do dia seguinte (LIVIERIS, 2020).

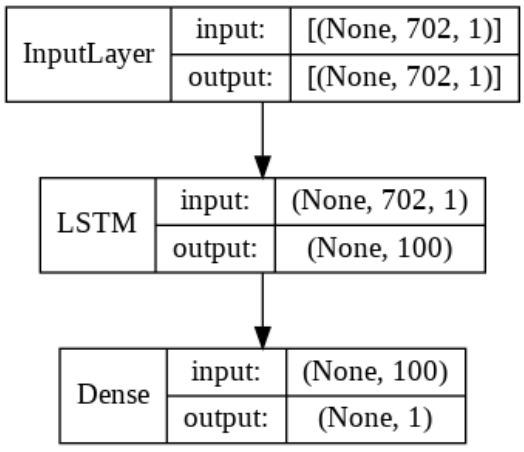
Figura 13 - Arquitetura CNN



Fonte: Elaborado pelos Autores (2021)

Para o modelo recorrente LSTM, foi utilizado uma camada com unidades LSTM e função de unidade linear retificada (ReLU), e uma camada totalmente conectada na saída com função de ativação sigmóide conforme ilustrado na Figura 14.

Figura 14 - - Arquitetura LSTM



Fonte: Elaborado pelos Autores (2021)

No capítulo seguinte, serão discutidos os resultados obtidos com a execução do trabalho e as implicações

4. RESULTADOS E DISCURSÃO

Conforme as métricas médias de avaliação (Quadro 1), todos os modelos obtiveram resultados positivos para POCID > 50% onde indica que os modelos utilizados tem desempenho melhor que a média de mercado. Todos os modelos obtiveram resultados para UTS < 1 onde indica que os modelos superaram o desempenho de um passeio aleatório. ARIMA apresentou o menor valor para R^2, enquanto os demais modelos expuseram melhores resultados próximos.

QUADRO 1 - MÉTRICAS DE AVALIAÇÃO

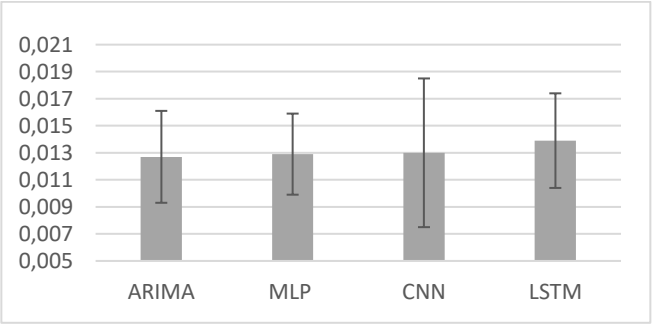
Métricas de Avaliação	**Modelos**			
	ARIMA	MLP	CNN	LSTM
MAPE	0.0127	0.0129	0.0130	0.0139
POCID	51.41	52.29	52.16	52.63
UTS	0.999	0.998	0.999	0.986
R^2	0.863	0.942	0.941	0.931

Fonte: Elaborado pelos Autores (2021)

Os modelos de redes neurais fazem uso de inicialização aleatória de seus pesos sinápticos. Desta forma, sendo possível obter diversos resultados de acordo com os valores iniciais, rodadas de treinamento dos algoritmos para se obter estatística descritiva (medidas de centralidade e medidas de dispersão) dos resultados.

Ao considerar a raiz do valor quadrático médio (*Root Mean Square*, RMS) do desempenho dos modelos para a métrica MAPE é possível constatar estatisticamente que todos os modelos apresentam resultados similares, não sendo possível determinar o que obteve melhor resultado como pode ser constatado no Gráfico 1.

Gráfico 1 – Mean Absolute Percentage Error (MAPE)

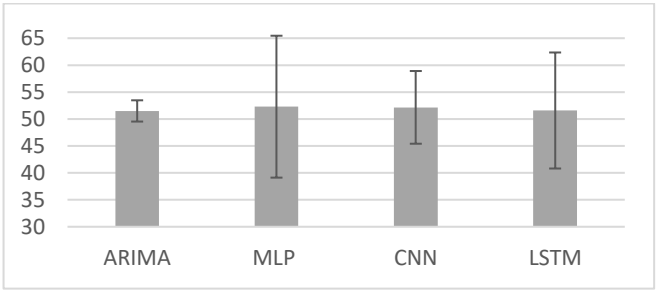


Fonte: Elaborado pelos Autores (2021)

De acordo com o Gráfico 2, todos os modelos apresentaram valor médio para POCID > 50%. No entanto, MLP, CNN e LSTM apresentaram alta variância, onde indica desempenhos eventuais inferiores à média de mercado, que pode causar perdas financeiras ao investidor de acordo com a HME. Apenas o modelo ARIMA apresentou resultados consistentes para a

medida de desempenho POCID, onde apresenta um menor intervalo de RMS. Desta forma, se tem maior confiança e estabilidade na acertabilidade da tendência da série temporal.

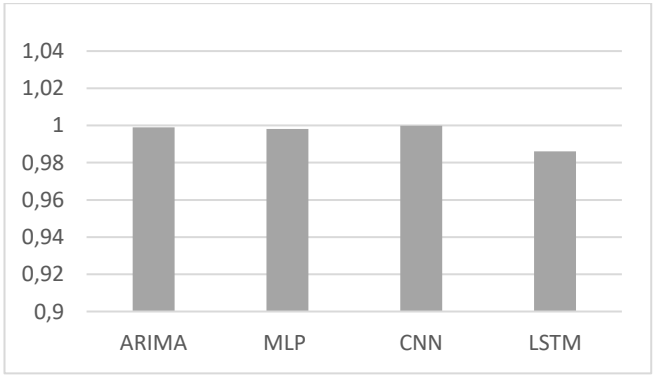
Gráfico 2 – Prediction Of Change In Direction (POCID)



Fonte: Elaborado pelos Autores (2021)

Todos os modelos obtiveram UTS < 1 indicando que seus resultados foram superiores a um processo estocástico conforme mostra a Gráfico 3. Dentre as arquiteturas usadas, LSTM apresentou o melhor desempenho para UTS. Enquanto que os demais modelos apresentam valores próximos ao limiar de decisão para seu desempenho ser considerado igual ao de um processo estocástico.

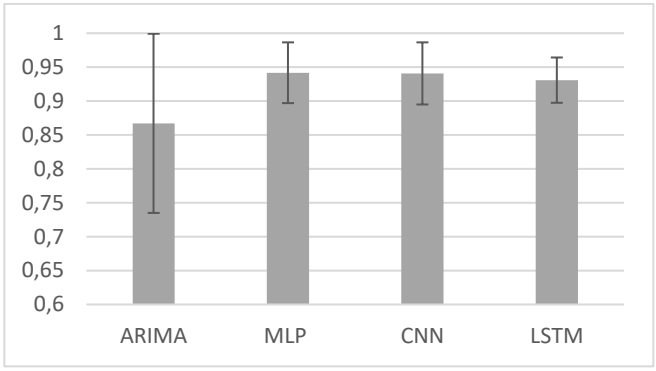
Gráfico 3 – U de Theils (UTS)



Fonte: Elaborado pelos Autores (2021)

Ao considerar o RMS no coeficiente de determinação (R2) todos os modelos em um intervalo de confiança, embora MLP, CNN e LSTM apresentam os valores médios superiores que o ARIMA, como pode ser observado na Gráfico 4.

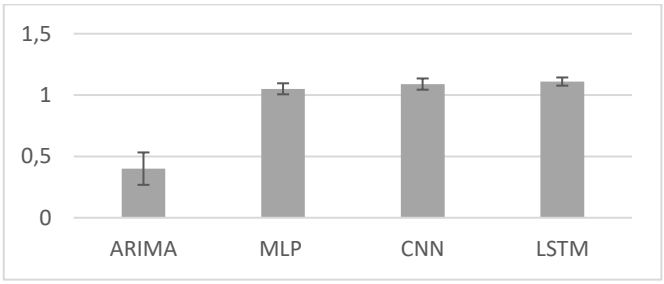
Gráfico 4 – Coefficient of Determination-R2



Fonte: Elaborado pelos Autores (2021)

Devido à quantidade de parâmetros os modelos de redes neurais, teve um processo computacionalmente mais custoso, em relação ao modelo estatístico. As redes neurais apresentaram custos computacionais próximos. Desta forma, esse indicador pode desempenhar um importante papel na decisão da utilização de um modelo estatístico ou de redes neurais como solução dos problemas de séries temporais em se utilizar em perspectiva aos seus possíveis resultados alcançados e o tempo de processamento dos modelos como pode ser observado no Gráfico 5.

Gráfico 5 - Custo Computacional - log(tempo)



Fonte: Elaborado pelos Autores (2021)

No capítulo seguinte tem-se as considerações finais do trabalho assim como as conclusões alcançadas de acordo com os objetivos estabelecidos.

5. CONSIDERAÇÕES FINAIS

A determinação da melhor janela temporal para a série do temporal IBOVESPA foi possível através da informação mutual e da autocorrelação dos dados que foram capazes de estabelecer um conjunto de valores possíveis.

Os resultados mostram através das métricas MAPE e POCID que todos os modelos propostos foram capazes de fazer predição da série temporal do IBOVESPA. Apresentando valores médios dos modelos, para MAPE = 1.3% e POCID > 52%. Ao considerar as métricas e o valor esperado de uma aplicação, é possível afirmar que a longo prazo o investidor teria lucro superior ao valor médio de mercado. Para a métrica R2, as redes neurais obtiveram em média desempenho superior a 93% com os menores valores de RMS em comparação ao modelo estatístico com 86.3% e maior valor para o RMS. Desta forma, pode-se concluir que as redes neurais obtiveram um melhor ajuste dos dados dos valores preditos.

Através das métricas utilizadas foi possível analisar o desempenho das predições do IBOVESPA para os diferentes modelos utilizados. Sendo, o custo computacional dos modelos de redes neurais superiores ao modelo estatístico utilizado, sendo esse critério um possível limiar de decisão para a escolha da arquitetura a ser utilizada para problemas de séries temporais de acordo com a necessidade do problema e o objetivo a ser alcançado.

Devido à sensibilidade dos hiperparâmetros e sua alta complexidade, é possível que a configuração ótima dos parâmetros não tenha sido alcançada, sendo provável assim a obtenção de melhores resultados para a predição de séries temporais.

Por fim, sugere-se para trabalhos futuros métodos capazes de reduzir o valor de RMS das métricas utilizadas com o objetivo de trazer confiabilidade nas predições.

REFERÊNCIAS

ARAÚJO, R. A. Swarm-Based Hybrid Intelligent Forecasting Method for Financial Time. **Learning and Nonlinear Models – Revista da Sociedade Brasileira de Redes Neurais**, Campinas, v. Vol. 5, No. 2, p. pp. 137–154, 2007.

ARAÚJO, R. D. A. **MERCADO DE AÇÕES BRASILEIRO EM ALTA-FREQUÊNCIA: EVIDÊNCIAS DE SUA PREVISIBILIDADE COM MODELAGEM MORFOLÓGICA-LINEAR**. RECIFE. 2016.

BOX, G. E. P.; JENKINS, G. M.; REINSEL, G. C. **Time Series Analysis:** forecasting and control. New Jersey: Prentice Hall, 1994.

CLEMENTS, M. P.; HENDRY, D. F. On the Limitations of Comparing Mean Square Forecast. **Journal of Forecasting**, p. p.617–637, 1993. Disponivel em: <10.1002/for.3980120802. >.

EVSUKOFF, A. G. **Inteligência computacional:** Fundamentos e aplicações. 1. ed. ed. Rio de Janeiro: E-papers, 2020.

FAMA, E. Efficient Capital Markets: A Review of Theory and Empirical Work. **The Journal of Finance**, 1970.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS. F. Learning to Forget: Continual. **Neural Computation**, p. v. 12, n. 10, p. 2451-2471, 2000.

HANN, T. H.; STEURER, E. Much ado about nothing? Exchange rate forecasting: neural. **Neurocomputing**, p. p.323–339, 1996.

HANN, T. H.; STEURER, E. Much ado about nothing? Exchange rate forecasting: neural. **Neurocomputing**, S.L, v. v.10, p. p.323–339, 1996.

HAYKIN, S. **Neural Networks and Learning Machines**. Canada: McMaster University, 2007.

KANTZ, H.; SCHREIBER, T. Nonlinear Time Series analysis. **Cambridge University Press**, New York, n. 2.ed., 2003.

LIVIERIS, I. E. . P. E. . P. P. . A CNN-LSTM model for gold price time series forecasting. **Neural Comput**, 2020.

MENESES JÚNIOR, J. M. P. D. **CONTRIBUIÇÕES AO PROBLEMA DE PREDIÇÃO RECURSIVA DE SÉRIES TEMPORAIS UNIVARIADAS USANDO REDES NEURAIS RECORRENTES**. Universidade Federal do Ceará. Fortaleza. 2012.

MORETTIN, P. A.; TOLOI, C. M. C. **Analise de Series Temporais**. São Paulo: Editora Edgard Blucher, 2006.

SITTE, R.; SITTE, J. Neural Networks Approach to the Random Walk Dilemma of Financial. **Applied Intelligence**, [S.l.], v. v.16, n.3, p. p.163–171, May 2002.

TAKENS, F. Detecting Strange Attractor in Turbulence. In: DYNAMICAL SYSTEMS AND TURBULENCE. **Springer-Verlag**, New York, p. p.366–381, 1980.

VERGARA, S. C. **Projetos e Relatórios de Pesquisa em**. 3 ed. ed. São Paulo: Atlas, 2000.

YAO, J.; TAN, C. L. A case study on using neural networks to perform technical forecasting of. **Neurocomputing**, S.L, v. v.34, p. p.79–98, 2000.

YAO, J.; TAN, C. L. A. A case study on using neural networks to perform technical forecasting of. **Neurocomputing**, [S.l.], v. v.34, n. n.1-4, p. p.79–98, 2000.

ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with Artificial Neural Networks: the state of the art. **International Journal of Forecasting**, S.l., v. v.14, p. p.35–62, 1998.