

# Previsão do Mercado Financeiro com Redes Neurais

Ya-Sin B. Mghazli\*, Ricardo de A. Araujo†, Jose M. de Seixas\*

\*Laboratorio de Processamento de Sinais, COPPE/POLI, Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, Brasil

†Laboratorio de Inteligência Computacional do Araripe, Instituto Federal do Sertão Pernambucano, Ouricuri, PE, Brasil  
yasinbarcelos@hotmail.com, ricardo.araujo@ifsertao-pe.edu.br, seixas@lps.ufrj.br

**Abstract**—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, esse trabalho analisa o desempenho de modelos de redes neurais artificiais (multilayer perceptrons, convolutional neural networks, long short-term memory e temporal fusion transformers) 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) e os resultados alcançados demonstraram efetividade, em desempenho preditivo, dos modelos investigados.

**Index Terms**—Redes Neurais, Séries Temporais, Mercado de Ações, Previsão, Dilema do Passeio Aleatório.

## I. INTRODUÇÃO

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 [1]. Desta forma, a previsão 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 [2].

Na literatura, diversas classes de modelos têm sido propostas para prever séries temporais financeiras [1]- [12]. Neste contexto, o surgimento da hipótese do mercado eficiente (efficient market hypothesis, EMH) [13] possibilita um consenso sobre a existência de algum tipo relacionamento não-linear presente no fenômeno gerador de séries temporais relacionadas ao mercado de ações [1].

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 [1], [7]- [12]. Assim, considerando o fato das previsões geradas possuírem um atraso de uma unidade de tempo em relação aos valores reais da série,

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior — Brasil (CAPES) — Código de Financiamento 001, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ)

surge um problema nos modelos de previsão, conhecido como dilema do passeio aleatório (random walk dilemma, RWD) [14], para prever séries temporais financeiras [1].

Neste sentido, vale mencionar que o atraso da previsão em uma unidade de tempo ainda vem sendo reportado recentemente na literatura de séries financeiras [10]- [12], bem como é possível verificar que o dilema permanece sem uma solução formal (devido à inexistência de quaisquer provas matemáticas para justificar tal comportamento). No entanto, os resultados apresentados por Araújo et al. [1] têm se revelado promissores, mostrando novos caminhos a serem traçados para modelagem de fenômenos temporais relacionados ao mercado de ações.

Desta forma, neste trabalho apresentamos uma análise de modelos de redes neurais artificiais (multilayer perceptrons, convolutional neural networks, long short-term memory e temporal fusion transformers) como solução do dilema do passeio aleatório no problema de previsão de séries temporais financeiras. De modo a avaliar o desempenho preditivo, foi utilizada uma série temporal relacionada ao Índice da Bolsa de Valores de São Paulo (IBOVESPA), e os resultados alcançados demonstraram efetividade dos modelos investigados.

## II. FUNDAMENTOS

### A. O Problema de Previsão de Séries Temporais

De acordo com Box et al. [15], uma série temporal representa uma sequência de observações, discretas ou contínuas, sobre um fenômeno que evolui com o tempo. Portanto, uma série temporal ( $\mathbf{X}$ ) pode ser formalmente definida por [15].

$$\mathbf{X} = \{x_t \in \mathbb{R} \mid t = 1, 2, 3 \dots N\}, \quad (1)$$

Onde o termo  $t$  representa um índice temporal (tempo), que determina a granularidade das observações, e o termo  $N$  representa o número de observações ou pontos da série temporal.

O principal objetivo de uma técnica de predição temporal é a construção de um mapeamento capaz de estimar um dado fenômeno temporal [15], isto é, criar um mecanismo que permita, com certa precisão, a previsão dos valores futuros da série, dados por  $x_{t+h}$ , em que  $h$  é o horizonte de previsão de  $h$  passos a frente [16]. Portanto, a ideia básica de um dado problema de previsão é definir uma janela temporal, que contém as observações do passado da série, para construção posterior de um modelo com o objetivo de determinar as suas observações futuras. Note que a janela temporal é formada

por um conjunto de observações (pontos), conhecidos como retardos temporais (Time Lags, TL) [15]. Assim, a escolha correta dos retardos temporais (um retardo temporal é relevante quando gera alguma categoria de relacionamento entre os dados históricos) possibilita caracterizar as leis que governam o fenômeno gerador da série, para incrementar o desempenho do preditor [16].

### B. O Dilema do Passeio Aleatório

Uma observação no tempo  $t$  de uma série temporal, dada por  $x_t$ , pode ser representada por [14].

$$x_t = x_{t-1} + r_t \quad (2)$$

em que  $x_{t-1}$  é uma observação da série temporal no tempo  $x_{t-1}$  e  $r_t$ , é um ruído branco (white noise).

Conforme a Equação 2, as observações futuras dependem das observações passadas da série temporal adicionado a um termo de ruído branco, o futuro seria imprevisível. Este comportamento é característico em séries temporais financeiras, é conhecido como dilema do passeio aleatório (Random Walk Dilemma, RWD) [14]. Nesse contexto, percebe-se que as leis que governam o fenômeno gerador de séries temporais financeiras são bastante similares às leis que governam o fenômeno gerador de um passeio aleatório e, tal fato implica que a previsão gerada por qualquer modelo terá, obrigatoriamente, um atraso de um-passo-adiante [14].

## III. MODELOS DE PREVISÃO

### A. Multilayer Perceptron (MLP)

O perceptron multicamadas (multilayer perceptron, MLP) [17], é uma rede neural com arquitetura em camadas sequenciais, isto é, os neurônios são organizados em uma ou mais camadas de processamento, sendo a rede mais frequentemente encontrada na literatura de previsão de séries temporais financeiras [2]. Nesse contexto, o modelo MLP com melhor desempenho para previsão de séries temporais financeiras utiliza função de ativação sigmoide logística para todas as unidades de processamento escondidas. A unidade de processamento de saída utiliza função de ativação linear com seu bias passando por função sigmoide logística, dada por.

$$y_k(t) = \sum_{j=1}^{n_h} W_{jk} \text{Sig} \left[ \sum_{i=1}^{n_{in}} W_{ij} x_i(t) + b_j^1 \right] + \text{Sig} (b_k^2), \quad (3)$$

em que  $x_i(t)$  ( $i = 1, 2, \dots, n_{in}$ ) São os valores de entrada (retardos temporais),  $n_{in}$  e  $n_h$  representam, respectivamente, a dimensionalidade das entradas da rede e a quantidade de unidades de processamento na camada escondida. Note que, como a previsão pretendida é de um-passo-adiante, utiliza-se apenas uma unidade de processamento na camada de saída ( $k = 1$ ). O termo  $\text{Sig}(\cdot)$  é uma função sigmoide logística definida por

$$\text{Sig}(x) = \frac{1}{1 + \exp(-x)} \quad (4)$$

De acordo com Haykin [17], a propriedade mais relevante desta rede é sua capacidade de aprendizagem através de um

processo supervisionado, que é caracterizado pela presença de um agente externo que induz a rede a uma resposta desejada a um determinado estímulo apresentado pelo ambiente, de maneira a construir um mapeamento entre a entrada e a saída, através da minimização de uma função de custo  $f$ , de modo que a resposta observada se aproxime do alvo desejado a cada iteração, definida como época, no processo de aprendizagem [17].

Note que uma função de custo  $f$  define uma superfície de erro sobre o espaço de pesos [17]. Observe que, se  $P$  representa a dimensionalidade dos vetores de pesos ajustáveis na rede neural e  $N$  representa a dimensionalidade do padrão de saída do problema, então  $f : \mathbb{R}^P \rightarrow \mathbb{R}^N$ . Nesta superfície, tem-se a presença de mínimos locais e globais, e os métodos de otimização utilizados para minimizar a função  $f$  utilizam informações do gradiente descendente do erro para ajustar os parâmetros da rede [17]. O método clássico utilizado no processo de aprendizagem destas redes é o algoritmo de retropropagação do erro (back-propagation, BP) [17].

### B. Convolutional Neural Networks (CNN)

As redes neurais convolucionais são úteis para o pré-processamento de dados que através de filtros de convolução se extrai características do vetor de entrada por meio do operador de convolução que é semelhante à função de correlação cruzada [18]. Diferentes filtros de convolução produzem diversas características resultando em múltiplos recursos convolucionais que são geralmente mais úteis que os dados originais [21]. Por exemplo, o resultado de uma convolução 1D se equivale ao modelo Auto Regressivo (AR) [18].

Uma camada de “pooling” recorre a um processo semelhante às camadas convolucionais, onde é utilizado uma técnica de subamostragem onde através de uma janela deslizante sob os valores de entrada se produz novos valores conforme a operação de pooling para construir um modelo convolucional mais robusto [21].

Os filtros convolucionais, são conectados a camadas intermediárias e usualmente conecta-se sua saída a uma rede neural totalmente conectada, e os parâmetros do modelo são ajustados pelo algoritmo de retropropagação. No entanto, as redes convolucionais são flexíveis e computacionalmente eficientes, podem ser combinadas com diferentes modelos e para diversas aplicações [18] - [21].

### C. Long Short-Term Memory (LSTM)

A arquitetura da LSTM é uma rede neural recorrente, onde se utiliza da saída do neurônio como parâmetro de entrada, 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 [18].

A porta  $f_t$  do “esquecimento” tem o objetivo de limitar a proporção da ativação do vetor de estados internos na retropropagação do erro do vetor recorrente, definida formalmente na Equação 5 [19].

$$f_t = \sigma (W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (5)$$

As portas  $i_t$  e  $g_t$  controlam o fluxo de informação do vetor de entrada evitando ruído e informações irrelevantes [18]. Suas definições matemáticas são dadas nas Equações 6 e 7 [19].

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (6)$$

$$g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + W_{cg}c_{t-1} + b_g) \quad (7)$$

A porta  $o_t$  definida na Equação 8, 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) [18].

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (8)$$

Por fim,  $c_t$  é utilizado para ajuste dos pesos com o algoritmo BPTT e  $h_t$  entrada do neurônio seguinte ou saída do neurônio. Suas definições matemáticas são dadas nas Equações 9 e 10 [19].

$$c_t = f_t c_{t-1} + i_t g_t \quad (9)$$

$$h_t = o_t \tanh(c_t) \quad (10)$$

Os diferentes modelos de LSTM têm sido utilizados com sucesso em diversas aplicações como reconhecimento de fala, tradução automática, dublagem de imagens e previsão de séries temporais no mercado financeiro [18].

#### D. Temporal Fusion Transformers (TFT)

O modelo foi projetado para a utilização de componentes canônicos para representar com eficiência diferentes entradas e ter um alto desempenho para diversas categorias de problemas. Sua construção tem como base os principais constituintes seguintes: [23].

(1) O Mecanismo de portão é utilizado com *Gated Residual Network* (GRN) para ignorar quaisquer componentes não relevantes para a arquitetura por meio do contexto externo fornecido pelas estatísticas dos dados, desta forma, tem-se profundidade adaptativa aos dados e complexidade a rede. Acomodando um amplo conjunto de categorias de dados, cenários e permite explorar a relação entre as entradas e variáveis exógenas.

(2) O módulo de Seleção de Variável (Variable Selection) é utilizado com GRNs e o contexto estatístico para extrair conforme as estatísticas dos dados, as variáveis não relevantes e entradas ruidosas que embora estejam disponíveis, o modelo não as considera necessárias ou podem prejudicar seu desempenho.

(3) As Covariáveis estatísticas são extraídas a partir do uso do módulo de seleção de variáveis, usadas em diversos pontos do modelo conforme se tem na Fig. 1. São usadas para relacionar os recursos estáticos com a codificação de vetores de contexto para a dinâmica temporal [23].

(4) O Processamento Temporal tem o objetivo de aprender as relações temporais de longo e curto prazo das variáveis de entrada e o vetor de contexto no tempo observado para aumentar a explicabilidade da predição. As dependências de curto prazo são capturadas com o processamento local através

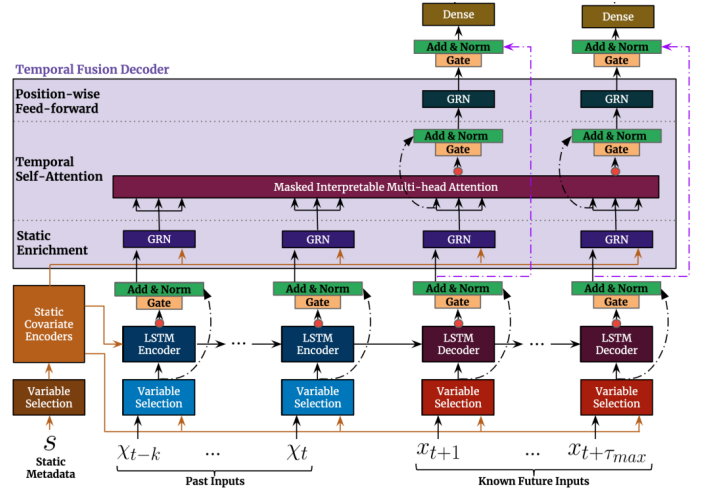


Fig. 1. Arquitetura TFT. Tem como entrada, metadados, janela temporal passada e informações futuras da série. Seu fluxo segue pelo módulo de Seleção de Variáveis, Encoder-Decoder-LSTM, Enriquecimento Estático, Auto-Atenção Temporal e tem como saída uma MLP.

de módulos Encoder-Decoder-LSTM, e as de longo prazo por blocos de auto-atenção temporal [23].

(5) O Intervalo de previsão é feito através de previsões quantílica que se determina um intervalo de valores para a predição. Desta forma, se faz necessário treinar o modelo para minimizar a perda conjunta quantílica [23].

Na Figura 1 tem-se a arquitetura do modelo TFT em que se pode ver todos os principais constituintes da rede.

#### IV. MEDIDAS PARA AVALIAÇÃO DE DESEMPENHO

O erro médio percentual absoluto (*Mean Absolute Percentage Error* — MAPE) é uma medida que fornece o tamanho médio do erro percentual e permite identificar precisamente os desvios percentuais do modelo de previsão [20]. O modelo de previsão ideal tem MAPE = 0. Sua definição matemática é dada na Equação 11

$$MAPE = \frac{1}{T} \sum_{t=1}^T 100 \left| \frac{e_t}{y_t} \right| \quad (11)$$

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 determinar o percentual de vezes que o modelo conseguiu prever corretamente a tendência futura da série temporal [24]. Sua definição matemática é dada nas Equações 12 e 13.

$$POCID = \frac{\sum_{t=1}^m D_t}{m} \times 100, \quad (12)$$

$$D_t = \begin{cases} 1, & \text{se } (y_t^a - y_{t-1}^a) \times (y_t^p - y_{t-1}^p) > 0 \\ 0, & \text{caso contrário} \end{cases} \quad (13)$$

Onde  $y_t^a$  e  $y_t^p$  respectivamente são os valores alvos e os valores preditos, o modelo de previsão ideal tem  $POCID = 100$ .

A estatística U de Theil (u of theil statistic, UTS) é outra forma de avaliar um modelo de previsão em séries temporais

em relação à perspectiva da existência de um passeio aleatório, sua interpretação é feita em comparação ao desempenho de um processo estocástico,  $UTS > 1$  indica que o modelo de previsão utilizado é pior que um passeio aleatório. Em contrapartida, tem-se que se  $UTS \leq 1$  indica que o modelo de previsão utilizado tem desempenho igual, ou melhor, que um processo estocástico, então dessa forma se tem para o modelo ideal de previsão  $UTS = 0$ . [14]. Sua definição pode ser observada na Equação 14.

$$UTS = \frac{\sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - f_t)^2}}{\sqrt{\frac{1}{n} \sum_{t=1}^n y_t^2} + \sqrt{\frac{1}{n} \sum_{t=1}^n f_t^2}} \quad (14)$$

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  $R^2 = 1$ . Sua definição formal está na Equação 15.

$$R^2 = 1 - \frac{\sum (e_i^2)}{\sum (y_i - \bar{y})^2} \quad (15)$$

## V. METODOLOGIA

### A. Tamanho da Janela Temporal

Dada uma série temporal ( $\mathbf{X}$ ), a reconstrução do espaço de estados pressupõe que, uma variável de estado  $x_i$  contém informações a respeito de outras variáveis de estados não-observáveis, e tais informações podem ser utilizadas no processo de predição de  $\mathbf{x}(n)$  da série temporal [25].

O Teorema de Imersão de Takens mostra que a partir de uma série temporal obtida através de um sistema dinâmico determinístico e livre de ruído  $r_i$ , será possível conforme a Equação 16 a reconstrução do espaço de estado do sistema por suas coordenadas de atraso [25].

$$x_{t+1} = f([x_t, x_{t-1}, x_{t-2}, \dots, x_{t-d_e}]) \quad (16)$$

Se as condições iniciais do teorema são satisfeitas, existe uma função que geralmente é desconhecida capaz de determinar o valor futuro de uma série temporal. Então, a dimensão de imersão  $d_e$  é um importante parâmetro a ser determinado para a construção da janela temporal [25].

No presente trabalho foi utilizado uma das principais ferramentas para determinação da dimensão de imersão que é a Função de Autocorrelação (FAC), foi empregado o valor seguinte ao seu decaimento para  $1/e$ , de modo a evitar um sistema sem memória ou ruído branco quando  $FAC \approx 0$  [25].

### B. Normalização dos dados

Os dados utilizados nesse trabalho são públicos, baixados da biblioteca “yahooquery” no Python. Foram utilizados os valores de fechamento diário entre o período de 01/01/2000 e 29/12/2020 da série temporal do Índice da Bolsa de Valores de São Paulo.

Foi aplicado um processo de normalização dos dados para o intervalo  $[0,1]$  com o objetivo de trazer conformidade entre os

dados de entrada e o valores preditos pelos modelos de redes neurais conforme a Equação 17 [26].

$$xn_i = \frac{x_i - \min(\mathbf{X})}{\max(\mathbf{X}) - \min(\mathbf{X})} \quad (17)$$

De acordo com Morettin [16], é comum que séries de preços de ativos financeiros contenham médias móveis e tendências, responsáveis por possíveis efeitos prejudiciais nos modelos de predição. Para contornar tal problema, foi utilizado uma diferenciação nesse trabalho de primeira ordem como mostra a Equação 18, que reduz os fatores de tendência e diminui a estacionariedade da série.

$$r_t = \log\left(\frac{xn_t}{xn_{t-1}}\right) \quad (18)$$

Como teste de estacionariedade foi executado o Augmented Dickey-Fuller (ADF), onde apresentou o  $p$ -value =  $1.35e^{-21}$  rejeitando a hipótese nula do teste e confirmando a estacionariedade da série temporal normalizada.

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 nesse 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) e comparado com o quantil 50% da função perda quantílica do modelo TFT. Para ajuste dos hiperparâmetros do TFT foi usado o algoritmo disponibilizado pelo autor.

### C. Arquiteturas dos Modelos

Foi utilizada a biblioteca “Keras” do Python para construção dos modelos aplicados nesse trabalho, com vinte rodadas de treinamento para cada modelo. Para o modelo MLP, foram utilizadas três camadas totalmente conectadas com função de unidade linear retificada (ReLU) e função de ativação sigmoide na camada de saída conforme a Figura 2.

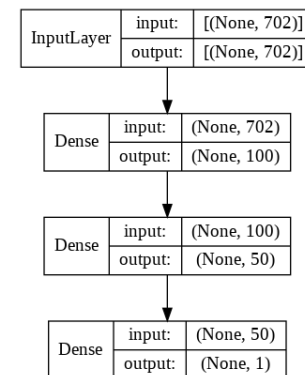


Fig. 2. Arquitetura MLP.

Na Figura 3 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 sub-amostra(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 sigmoide para prever os valores de fechamento do dia seguinte [22].

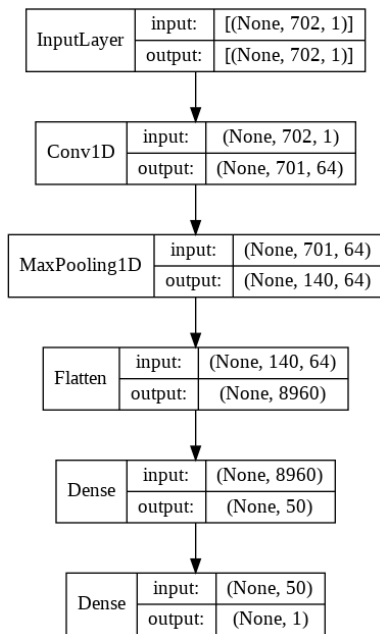


Fig. 3. Arquitetura CNN.

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 sigmoide conforme ilustrado na Figura 4.

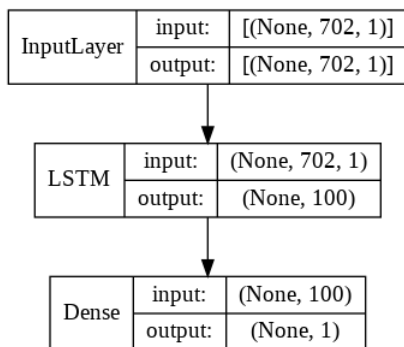


Fig. 4. Arquitetura LSTM.

Para o modelo TFT foi usado o conjunto de validação para ajuste de hiperparâmetros e um conjunto de teste para avaliação de desempenho. Foi mantida a camada de atenção interpretável única do modelo original, assim como, os tamanhos de 'kernel' para a camada de processamento convolucional [23].

## VI. RESULTADOS

Conforme as métricas médias de avaliação (Tabela I), 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. TFT apresentou o maior valor para  $MAPE$ , enquanto os demais modelos expôs melhores resultados, similarmente acontece para a métrica  $R2$  onde TFT apresentou o menor valor médio e os demais modelos obtiveram melhores resultados.

TABELA I  
MÉTRICAS DE AVALIAÇÃO

| Métricas de Avaliação | Modelos    |            |             |            |
|-----------------------|------------|------------|-------------|------------|
|                       | <i>MLP</i> | <i>CNN</i> | <i>LSTM</i> | <i>TFT</i> |
| MAPE                  | 0.0129     | 0.0130     | 0.0139      | 0.0176     |
| POCID                 | 52.29%     | 52.16%     | 51.58%      | 51.91%     |
| UTS                   | 0.998      | 0.999      | 0.986       | 0.996      |
| R2                    | 0.942      | 0.941      | 0.931       | 0.831      |

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 na Figura 5.

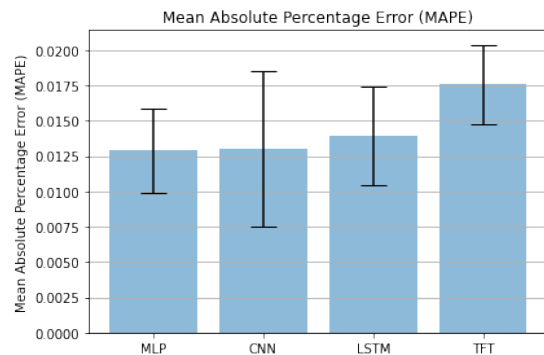


Fig. 5. Mean Absolute Percentage Error (MAPE)

De acordo a Figura 6, todos os modelos apresentaram valor médio para  $POCID > 50\%$ . No entanto, MLP, CNN e LSTM apresentaram alta variância, onde indica desempenhos eventuais inferior à média de mercado, que pode causar perdas financeiras ao investidor. Apenas o modelo TFT apresentou resultados consistentes para a medida de desempenho  $POCID$ , onde ao considerar o intervalo de RMS, o modelo apresenta

desempenho mínimo superior a 50%. Desta forma, se tem maior confiança e estabilidade na acertabilidade da tendência da série temporal do predito TFT.

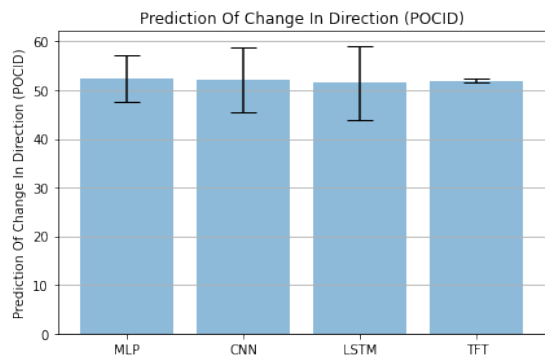


Fig. 6. Prediction Of Change In Direction (POCID)

Todos os modelos obtiveram  $UTS < 1$  indicando que seus resultados foram superiores a um processo estocástico conforme mostra a Figura 7. Dentre as arquiteturas usadas, LSTM apresentou o melhor desempenho para UTS.

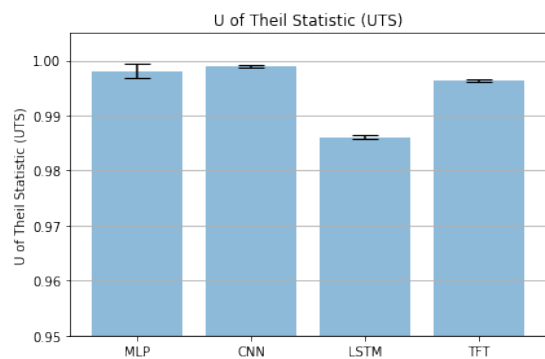


Fig. 7. U of Theil Statistic (UTS)

Ao considerar o RMS no coeficiente de determinação (R2) todos os modelos em um intervalo de confiança, embora MLP, CNN e LSTM apresentem os valores médios cerca de 13% melhores que o TFT, como pode ser observado na Figura 8.

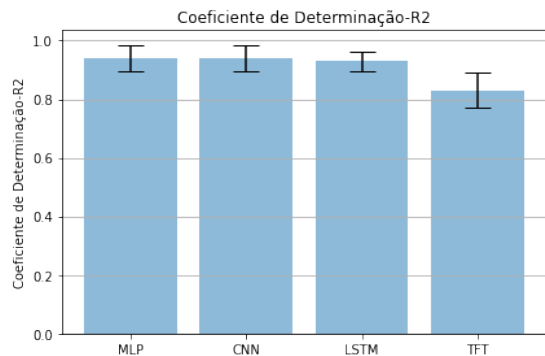


Fig. 8. Coeficiente de Determinação (R2)

Devido à quantidade de parâmetros do modelo TFT, este foi o processo computacionalmente mais custoso, em relação as demais arquiteturas, estas apresentaram custos computacionais próximos. Desta forma, esse indicador pode desempenha um importante papel na decisão do modelo a se utilizar em perspectiva aos seus possíveis benefícios como pode ser observado na Figura 9.

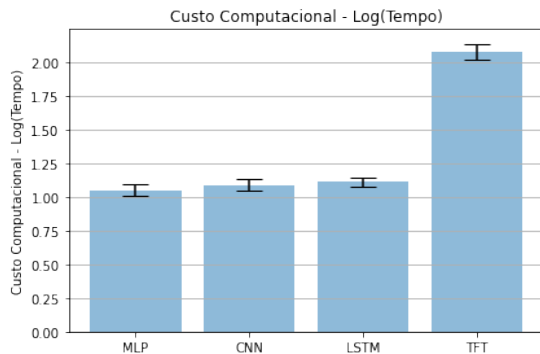


Fig. 9. Custo Computacional — Log(Tempo)

## VII. CONCLUSÕES

Nossos resultados mostram que todos os modelos utilizados são capazes de superar o dilema do passeio aleatório. Assim como, todas as arquiteturas foram capazes em média de ultrapassar o desempenho médio do mercado financeiro.

Todas as arquiteturas apresentaram resultados similares nas medidas de desempenho ao considerar o intervalo do RMS, sendo TFT o único modelo que apresentou resultados consistentes para predição da tendência da série temporal no longo prazo. Por fim, vale mencionar que devido à sensibilidade dos hiperparâmetros e sua alta complexidade, é possível que a configuração ótima não tenha sido alcançada, provável assim a obtenção de melhores resultados para a predição de séries temporais.

## REFERENCES

- [1] R. de A. Araujo, N. Nedjah, A. L. I. Oliveira, and S. R. de L. Meira, "A deep increasing decreasing-linear neural network for financial time series prediction," *Neurocomputing*, vol. 347, pp. 59–81, 2019.
- [2] R. de A. Araujo, A. L. I. Oliveira, and S. Meira, "A hybrid model for high-frequency stock market forecasting," *Expert Systems with Applications*, vol. 42, no. 8, pp. 4081 – 4096, 2015.
- [3] M. Podsiadlo and H. Rybinski, "Financial time series forecasting using rough sets with time-weighted rule voting," *Expert Systems with Applications*, vol. 66, pp. 219 – 233, 2016.
- [4] T. Zhou, S. Gao, J. Wang, C. Chu, Y. Todo, and Z. Tang, "Financial time series prediction using a dendritic neuron model," *Knowledge-Based Systems*, vol. 105, pp. 214 – 224, 2016.
- [5] A. J. Hussain, D. Al-Jumeily, H. Al-Askar, and N. Radi, "Regularized dynamic self-organized neural network inspired by the immune algorithm for financial time series prediction," *Neurocomputing*, vol. 188, pp. 23 – 30, 2016.
- [6] X. Sun, T. Li, Q. Li, Y. Huang, and Y. Li, "Deep belief echo-state network and its application to time series prediction," *Knowledge-Based Systems*, vol. 130, pp. 17 – 29, 2017.
- [7] J. Xue, S. Zhou, Q. Liu, X. Liu, and J. Yin, "Financial time series prediction using l2,l1rf-elm," *Neurocomputing*, vol. 277, pp. 176 – 186, 2018, hierarchical Extreme Learning Machines.

- [8] E. Horta and F. Ziegelmann, "Dynamics of financial returns densities: A functional approach applied to the bovespa intraday index," *International Journal of Forecasting*, vol. 34, no. 1, pp. 75 – 88, 2018.
- [9] B. Alhnaity and M. Abbod, "A new hybrid financial time series prediction model," *Engineering Applications of Artificial Intelligence*, vol. 95, p. 103873, 2020.
- [10] Z. Yang, J. Keung, M. A. Kabir, X. Yu, Y. Tang, M. Zhang, and S. Feng, "Acomnn: Attention enhanced compound neural network for financial time-series forecasting with cross-regional features," *Applied Soft Computing*, vol. 111, p. 107649, 2021.
- [11] Y. Shi, B. Li, G. Du, and W. Dai, "Clustering framework based on multiscale analysis of intraday financial time series", *Physica A: Statistical Mechanics and its Applications*, vol. 567, p. 125728, 2021.
- [12] D. Cheng, F. Yang, S. Xiang, and J. Liu, "Financial time series forecasting with multi-modality graph neural network," *Pattern Recognition*, vol. 121, p. 108218, 2022.
- [13] A. K. Tiwari, C. T. Albulescu, and S. M. Yoon, "A multifractal detrended fluctuation analysis of financial market efficiency: Comparison using dow jones sector etf indices," *Physica A: Statistical Mechanics and its Applications*, vol. 483, pp. 182 – 192, 2017.
- [14] R. Sitte and J. Sitte, "Neural networks approach to the random walk dilemma of financial time series," *Applied Intelligence*, vol. 16, no. 3, pp. 163 – 171, 2002.
- [15] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*, 3rd ed. New Jersey: Prentice Hall, 1994.
- [16] MORETTIN, P. A.; TOLOI, C. M. C. *Análise de Séries Temporais*. 2. ed. São Paulo: ABE - Projeto Fisher, Edgard Blücher, 2006
- [17] S. Haykin, *Neural networks: A comprehensive foundation*. New Jersey: Prentice Hall, 1998.
- [18] EVSUKOFF, Alexandre G. *Inteligência Computacional: fundamentos e aplicações*. Rio de Janeiro: E-Papers, 2020.
- [19] Gers, F. et al. "Learning to Forget: Continual Prediction with LSTM." *Neural Computation* 12 (2000): 2451-2471.
- [20] CLEMENTS, M. P., HENDRY, D. F. (1993). On the Limitations of Comparing Mean Square Forecast. *Journal of Forecasting*, p.617–637. Fonte: 10.1002/for.3980120802.
- [21] Rawat, Waseem; Wang, Zenghui. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*. 29. 1-98.
- [22] Livieris, I.E., Pintelas, E., Pintelas, P.: A CNN-LSTM model for gold price time series forecasting. *Neural Comput. Appl.* (2020)
- [23] Lim, B., Arik, S.Ö., Loeff, N., Pfister, T. (2019). Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. *ArXiv*, abs/1912.09363.
- [24] Albuquerque Filho, F. S. de, madeiro, F., Fernandes, S. M. M., Mattos Neto, P. S. G. de, Ferreira, T. A. E. (2013). Time-series forecasting of pollutant concentration levels using particle swarm optimization and artificial neural networks. *Química Nova*, 36(6), 783–789.
- [25] MENEZES, J. M. P.; BARRETO, G. A.; FREIRE, A. L. *Redes Neurais Recorrentes para Predição Recursiva de Séries Temporais Caóticas: um estudo comparativo*. Congresso Brasileiro de Redes Neurais Artificiais, [S.l.], 2009.
- [26] G. ZHANG, B. E. PATUWO, M. Y. H., 1998, "Forecasting with artificial neural networks: The state of the art", *International Journal of Forecasting*, v. 14, pp. 35–62.