LUIZ FELIPE ALAMINO DE LIMA
RHENAN SILVA NEHLSEN

ANÁLISE DE SÉRIES TEMPORAIS FINANCEIRAS E TREND-FOLLOWING UTILIZANDO O MODELO TEMPORAL FUSION TRANSFORMER

São Paulo
2023

LUIZ FELIPE ALAMINO DE LIMA
RHENAN SILVA NEHLSEN

ANÁLISE DE SÉRIES TEMPORAIS FINANCEIRAS E TREND-FOLLOWING UTILIZANDO O MODELO TEMPORAL FUSION TRANSFORMER

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro Eletricista com ênfase em Computação.

São Paulo
2023

LUIZ FELIPE ALAMINO DE LIMA
RHENAN SILVA NEHLSEN

ANÁLISE DE SÉRIES TEMPORAIS FINANCEIRAS E TREND-FOLLOWING UTILIZANDO O MODELO TEMPORAL FUSION TRANSFORMER

Trabalho apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro Eletricista com ênfase em Computação.

Área de Concentração:
Engenharia da Computação

Orientador:
Prof. Dr. Edson Satoshi Gomi

Co-orientador:
Fabio Katsumi Shinohara de Souza

São Paulo
2023

AGRADECIMENTOS

Agradecemos aos nossos pais que nos apoiam e acompanham em todas as empreitadas da vida, mostrando que, apesar de desafiador, o caminho do conhecimento deve ser seguido.

Agradecemos ao Professor Dr. Edson Gomi por nos aceitar como orientados e incentivar o grupo ao longo do projeto, bem como pelo interesse no tema e pela disponibilidade ao longo do ano.

Agradecimentos ao mestrando Fabio Katsumi Shinohara de Souza por co-orientar nosso projeto de TCC e por seu acompanhamento para as aplicações financeiras desenvolvidas com os modelos de machine learning. Sua orientação e expertise foram fundamentais para o sucesso do trabalho. Agradecemos pela disponibilidade e apoio ao longo do projeto.

Eu, Rhenan, agradeço também aos colegas de Morgan Stanley, Guilherme Brunetti, Yohan Garcia e Felipe Teti, por estarem sempre dispostos discutir e me ensinar sobre finanças quantitativas, o que me motivou a realizar este trabalho.

Eu, Luiz, agradeço a meus colegas da Giant Steps Capital, Iago Dantas e Rodrigo Amorim por acompanharem e auxiliarem o desenvolvimento de tanto os aspectos técnicos quanto financeiros do projeto.

RESUMO

Este estudo introduz uma metodologia para a previsão de uma variedade de instrumentos financeiros empregando o modelo Temporal Fusion Transformer (*Temporal Fusion Transformer* (TFT)). O TFT é uma arquitetura de rede neural avançada especialmente concebida para gerir séries temporais multivariadas com eficiência, captando dinâmicas complexas e interdependências temporais. A pesquisa visa a comparação do TFT com os modelos tradicionais de trend-following, que normalmente se concentram em seguir tendências de mercado já estabelecidas.

A metodologia utilizada neste trabalho inclui a análise de um conjunto de dados financeiros, processados para a extração de indicadores técnicos chave. A avaliação do desempenho do modelo se foca na taxa de Sharpe, um indicador reconhecido para medir o retorno ajustado ao risco, evidenciando assim a competência do TFT em previsões financeiras. O objetivo primordial deste estudo é comparar a eficácia do modelo TFT na previsão de séries temporais financeiras, fornecendo uma contribuição tanto para a pesquisa quanto para a prática no campo da previsão financeira.

Palavras-Chave - Previsão de índices financeiros, Temporal Fusion Transformer (TFT), Redes neurais avançadas, Séries temporais multivariadas, Dinâmicas complexas, Interdependências temporais, Modelos de trend-following, Padrões de dados sutis, Análise de dados financeiros, Taxa de Sharpe, Retorno ajustado ao risco, Previsão de séries temporais, Pesquisa em previsão financeira, Indicadores técnicos, Eficiência do modelo TFT.

ABSTRACT

This study introduces a methodology for forecasting a variety of financial instruments using the Temporal Fusion Transformer (TFT) model. The TFT, an advanced neural network architecture, is specifically designed for efficiently managing multivariate time series, capturing complex dynamics and temporal interdependencies. The research aims to compare the TFT with traditional trend-following models, which typically focus on following already established market trends.

The methodology includes analyzing a dataset of financial instruments, processed to extract key technical indicators. The model's performance is evaluated based on the Sharpe ratio, a recognized measure of risk-adjusted return, thus highlighting the TFT's competency in financial forecasting. The primary goal of this study is to compare the effectiveness of the TFT model in forecasting financial time series, providing contributions to both research and practice in the field of financial forecasting.

Keywords - Financial indices forecasting, Temporal Fusion Transformer (TFT), Advanced neural networks, Multivariate time series, Complex dynamics, Temporal interdependencies, Trend-following models, Subtle data patterns, Financial data analysis, Sharpe ratio, Risk-adjusted return, Time series forecasting, Financial forecasting research, Technical indicators, TFT model efficiency.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

BERT *Bidirectional Encoder Representations from Transformers*

CNN *Convolutional Neural Network*

ETFs *Exchange-Traded Funds*

EWMA *Exponential Weighted Moving Average*

GARCH *Generalized Autoregressive Conditional Heteroskedasticity*

GPT-3 *Generative Pre-trained Transformer 3*

LSTM *Long Short-Term Memory*

MACD *Moving Average Convergence Divergence*

MAE *Mean Absolute Error*

ReLU *Rectified Linear Units*

RNN *Recurrent Neural Network*

RMSE *Root Mean Square Error*

SMAE *Squared Mean Aggregate Error*

TFT *Temporal Fusion Transformer*

TSMOM *Time Series Momentum*

SUMÁRIO

1 INTRODUÇÃO

No atual cenário dos mercados financeiros, caracterizado pela sua dinâmica intrínseca e evolução constante, a análise meticulosa das séries temporais torna-se essencial. Isso é especialmente verdadeiro considerando a variedade de produtos disponíves no mercado e suas caractísticas únicas. A abordagem tradicional para análise dessas séries temporais está cada vez mais se mostrando ultrapassada. Modelos convencionais, como ARCH e *Generalized Autoregressive Conditional Heteroskedasticity* (GARCH), frequentemente baseiam-se em estruturas estatísticas rígidas e premissas simplistas, revelando-se deficientes, especialmente em períodos de crise financeira [7]. Por exemplo, a crise do "Quant Quake"de 2007 demonstrou as falhas das estratégias quantitativas, majoritariamente baseadas em modelos estatísticos históricos, resultando em perdas significativas. A queda abrupta e não linear de vários índices, incluindo o S&P 500 durante a crise de 2020, ressalta as limitações desses modelos.

Nesse sentido, as estratégias de investimento, tais como o Trend Following, que são baseadas nos métodos tradicionais de análise de séries temporais tem seu potencial retorno limitado às capacidades destes modelos e as premissas simplistas e por vezes falsas.

Desta forma, o Temporal Fusion Transformer (TFT) surge como uma alternativa promissora, superando as limitações dos métodos tradicionais de análise de séries temporais. O TFT, capaz de manipular e sintetizar múltiplas séries temporais através de mecanismos de atenção e codificação de variáveis exógenas, representa um avanço significativo na análise de dados. Ao contrário dos modelos tradicionais, o TFT considera um leque mais amplo de informações, desde dados macroeconômicos até métricas específicas de ativos,

para criar previsões mais informadas e precisas.

A inovação do TFT não reside apenas na sua capacidade de processar grandes volumes de dados, mas também na sua habilidade de decifrar interdependências temporais complexas através de mecanismos de atenção multi-cabeça. Este recurso permite ao TFT atribuir pesos diferentes a várias entradas temporais, focando em pontos de dados específicos quando mais relevantes, fornecendo assim uma representação mais rica e detalhada dos fenômenos estudados.

Visando um avanço significativo na análise financeira, este projeto busca explorar o potencial do TFT para desenvolver uma estratégia de investimentos. Nosso objetivo é comparar o desempenho desta estratégia com outras abordagens ao Trend Following.

1.1 Organização do Trabalho

Este trabalho está organizado em seis capítulos, cada um abordando aspectos distintos e essenciais do projeto.

No Capítulo 2, discutimos os conceitos fundamentais de finanças, fornecendo um alicerce teórico para a compreensão das estratégias de Trend Following e suas métricas de avaliação. Este conhecimento é crucial para o entendimento do contexto em que o Temporal Fusion Transformer será aplicado.

O Capítulo 3 se aprofunda na arquitetura do Temporal Fusion Transformer, explorando suas capacidades e aplicabilidade no processamento de séries temporais financeiras. Aqui, descrevemos as inovações e os desafios técnicos associados a esta arquitetura de aprendizado profundo.

No Capítulo 4, detalhamos a implementação prática do projeto. Este capítulo abrange desde a aquisição de dados até a configuração específica do modelo de aprendizado profundo, incluindo escolhas de arquitetura, parâmetros e técnicas de otimização.

O Capítulo 5 é dedicado à discussão dos resultados obtidos. Aqui, analisamos o desem-

penho do modelo em comparação com estratégias clássicas e atuais de Trend Following, fornecendo insights sobre a eficácia e a inovação do nosso modelo.

Finalmente, o Capítulo 6 apresenta as conclusões do estudo. Neste capítulo, resumimos os achados chave, discutimos as implicações práticas e teóricas do trabalho, e sugerimos direções para pesquisas futuras.

1.2 Metodologia de Trabalho

A metodologia adotada neste trabalho é meticulosamente planejada para abordar todos os aspectos cruciais do projeto, desde a concepção inicial até a análise final dos resultados. Os passos detalhados são os seguintes:

1. **Escolha e Aquisição de Dados Financeiros:** Nesta etapa inicial, realizamos uma seleção criteriosa de ativos financeiros com base em critérios de liquidez, volatilidade e representatividade no mercado. Após a seleção, procedemos com a aquisição de séries temporais de preços dos ativos escolhidos, garantindo a coleta de dados históricos abrangentes e confiáveis.

2. **Análise e Comparação de Estratégias de Trend-Following:** Aqui, conduzimos uma análise comparativa entre diferentes estratégias de Trend Following. Este estudo nos permite entender as práticas atuais no mercado e estabelecer uma base de referência para o desempenho do nosso modelo.

3. **Estudo Aprofundado das Arquiteturas de Deep Learning:** Focamos no estudo detalhado de modelos de aprendizado profundo, principalmente dos Transformers e TFT. Investigamos suas características, funcionalidades e mecanismos de otimização para entender como podem ser aplicados de maneira eficiente em nosso contexto.

4. **Seleção de Bibliotecas e Ferramentas:** Avaliamos e selecionamos as bibliotecas e ferramentas mais adequadas para a implementação dos modelos. Esta seleção é

baseada em critérios como suporte à comunidade, documentação, eficiência e facilidade de uso.

5. **Análise Exploratória e Pré-processamento dos Dados:** Esta etapa envolve a análise exploratória para identificar padrões, anomalias e características dos dados. Seguida pelo pré-processamento, que inclui a limpeza, normalização e transformação dos dados para adequá-los ao modelo.

6. **Construção e Configuração da Arquitetura do Modelo:** Procedemos com a construção da arquitetura da rede neural, definindo a configuração dos parâmetros e as técnicas de otimização. Esta fase é crítica para garantir a eficácia e a eficiência do modelo.

7. **Tuning e Otimização do Modelo:** Após a construção inicial, realizamos o tuning do modelo, ajustando parâmetros e estratégias de treinamento para melhorar seu desempenho.

8. **Validação e Backtesting do Modelo:** Nesta fase, o modelo é submetido a um backtesting rigoroso para avaliar sua performance. Os resultados são comparados com estratégias de Trend Following e estudos anteriores para validar a eficácia do modelo.

9. **Análise de Resultados e Identificação de Insights:** Após o backtesting, realizamos uma análise detalhada dos resultados, identificando insights valiosos e compreendendo as implicações dos achados.

10. **Documentação e Preparação da Apresentação:** Por fim, documentamos todo o processo e os resultados em um formato claro e compreensível, preparando-nos para a apresentação final do projeto.

2 TREND-FOLLOWING DE ATIVOS FINANCEIROS

Este capítulo versa sobre os conceitos fundamentais para o desenvolvimento deste projeto. Nele, introduziremos brevemente o mercado financeiro, as estratégias de Trend-Following e a aprendizagem de máquina, destacando os modelos de Transformers e do Temporal Fusion Transformer.

2.1 Mercado Financeiro e seus Produtos

Em uma economia, o mercado financeiro é o ambiente onde se negociam ativos. Os quais podem ser ações, moedas, títulos, derivativos, commodities, etc. Como o objeto de estudo do presente trabalho está inserido no contexto do mercado financeiro, faz-se necessário introduzir tópicos que facilitarão o entendimento das decisões tomadas no projeto.

Há basicamente dois tipos de mercados onde a negociação de ativos é possível, mercados de Bolsa ou de Balcão.

Os Mercados de Bolsa funcionam como ponto de encontro entre compradores e vendedores. Suas principal característica é que as bolsas possuem câmaras de compensação que ajudam a mitigar o risco de contraparte. Além disso, os contratos de derivativos são padronizados, em mercados de bolsa.

Nos Mercados de Balcão, ou OTC (*Over-the-Counter*), bancos, fundos e empresas negociam diretamente entre si, ou seja, fora da bolsa de valores. As negociações podem ser bilaterais ou por meio de uma contraparte central. Os contratos de derivativos são

customizáveis nos mercados OTC.

Neste projeto, desenvolveremos uma estratégia que visa obter lucros na negociação de contratos futuros. Para entendê-los, é necessário explicar também o que são derivativos e contratos à termo.

- Derivativos

  Como descrito por [8], um derivativo é um instrumento financeiro cujo valor deriva de variáveis subjacentes, que normalmente são preços de outros ativos. Exemplos de derivativos são opções, swaps, contratos à termo e contratos futuros.

- Contrato à Termo

  O contrato à termo é um dos tipos de derivativos mais básicos. Nele, as partes acordam em comprar ou vender um ativo em predeterminados preço e momento de entrega no futuro. Este tipo de contrato é importante não apenas para especuladores ou *traders* do mercado, mas também para empresas, pois permite travar o preço de um ativo no futuro, mitigando a sua incerteza.

  Sejam T a data de maturidade do contrato, S_T o preço à vista do ativo subjacente em T e K o preço de entrega acordado no contrato, também chamado de *strike*. O resultado da parte compradora do contrato é dado por:

$$S_T - K \tag{2.1}$$

  A parte vendedora tem resultado oposto, $K - S_T$. Ou seja, o comprador(vendedor) obtêm lucro(perda) caso o preço à vista seja maior que o preço de entrega.

- Contrato Futuro

  Similarmente ao contrato à termo, as partes de um contrato futuro acordam em comprar ou vender um ativo a predeterminados preço e momento de entrega no futuro. A diferença entre os contratos é que enquanto o primeiro é negociado no

mercado de balcão, o segundo é um produto de bolsa. Desta forma, para gerenciar o risco de contraparte dos contratos futuros, a bolsa exige postagem de margens.

O uso de contratos futuros na estratégia a ser construída nesse projeto se justifica por três motivos:

- **Liquidez:** Para algumas classes de ativos, por exemplo *Exchange-Traded Funds* (ETFs), o contrato futuro é mais líquido (é mais fácil encontrar compradores e vendedores dispostos a negociar) do que o próprio ativo.
- **Alavancagem:** Em geral, os derivativos permitem que o especulador obtenha a mesma exposição ao ativo subjacente, como mostra a equação 2.1, a partir de um investimento inicial menor.
- **Possibilidade de Rolagem:** Uma estratégia utilizada por investidores de contratos futuros é a rolagem. Nela, o investidor desfaz-se de sua posição no contrato de vencimento mais próximo, para posicionar-se no contrato de vencimento subsequente. Desta maneira, evita-se a entrega do ativo subjacente. O que é extremamente importante no caso especulação em commodities, as quais possuem custos de armazenamento envolvidos.

Por fim, vale mencionar as formas em que é possível posicionar-se de três formas. Um participante do mercado pode comprar um ativo e vendê-lo em um posterior momento. Assim, obtêm lucro caso o preço de venda for maior que o de compra.

Há também a possibilidade da venda descoberta, na qual o participante vende o ativo sem o possuir. Para tanto, é necessário alugar o ativo de um detentor. Nesta estratégia, o especulador vende o ativo alugado e obtêm lucro caso consiga recomprá-lo a um preço menor. A estratégia desenvolvida neste projeto adota as três formas de posicionamento mencionadas.

2.2 Trend-Following

Estratégias de *Momentum* de séries temporais [9], também conhecidas como Trend-following, são baseadas na heurística de comprar ativos onde o retorno tende a ser positivo e, por outro lado, vender os que tendem a ter retorno negativo.

A premissa básica do trend-following é que, uma vez estabelecida uma tendência, seja de alta ou de baixa, é mais provável que ela continue do que reverta. Portanto, a estratégia envolve dimensionar a posição operada de acordo com a intensidade da tendência identificada. Os *trend-followers* não tentam prever ou antecipar reversões de tendência; em vez disso, eles simplesmente seguem a tendência atual até que ela se inverta.

Durante a 'Crise do Monday Black' de 1987, os mercados financeiros sofreram uma queda abrupta, com o Dow Jones Industrial Average (DJIA) perdendo 22,6% em um único dia, 19 de outubro. Estratégias de trend-following, no entanto, mostraram resiliência. Por exemplo, o famoso Quantum Fund, co-gerenciado por George Soros, conseguiu evitar grandes perdas e até obteve ganhos significativos nesse período. Isso foi possível porque o trend-following não se baseia em previsões de mercado, mas em reagir a movimentos já ocorridos. Este evento demonstra como o trend-following pode atuar como um mecanismo de proteção em períodos de alta volatilidade, sendo capaz de adaptar-se rapidamente a mudanças drásticas no mercado. A análise detalhada desses eventos históricos fornece insights valiosos sobre o funcionamento prático do trend-following, sublinhando sua relevância em períodos de alta volatilidade de mercado.

2.2.1 Métodos Tradicionais de Trend-Following

Os métodos tradicionais de trend-following geralmente se baseiam em indicadores técnicos para identificar e seguir tendências. Esses indicadores podem incluir:

Médias Móveis: Uma média móvel é uma média de dados de preços ao longo de um determinado período de tempo que se move ao longo do tempo. Quando o preço atual de um ativo está acima de sua média móvel, isso pode indicar uma tendência de alta. Da

mesma forma, quando o preço atual está abaixo da média móvel, isso pode indicar uma tendência de baixa.

Osciladores de Momentum: Os osciladores de momentum medem a taxa de variação dos preços, operando com a premissa de que a velocidade de movimento dos preços de um ativo é um indicativo de sua força de mercado. Esses indicadores oscilam em torno de uma linha central ou entre valores máximos e mínimos definidos, proporcionando uma perspectiva sobre a direção e a força da tendência atual.

Entre os osciladores de momentum mais conhecidos estão o Índice de Força Relativa (RSI), o Estocástico e o *Moving Average Convergence Divergence* (MACD) (Convergência e Divergência de Médias Móveis). O RSI, por exemplo, mede a velocidade e a mudança dos movimentos de preços, oscilando entre 0 e 100. Ele é frequentemente usado para identificar condições de sobrecompra (acima de 70) ou sobrevenda (abaixo de 30). Já o Estocástico compara o preço de fechamento de um ativo com sua faixa de preço ao longo de um período específico, enquanto o MACD é usado para identificar mudanças na direção, força, momentum e duração de uma tendência.

A aplicação dos osciladores de momentum na análise técnica é amplamente reconhecida por sua capacidade de fornecer sinais de negociação em tempo hábil. Eles são particularmente úteis em mercados que não seguem tendências claras, onde podem sinalizar oportunidades de compra ou venda em cenários onde outros indicadores podem falhar.

É de nota, contudo, que os osciladores podem produzir sinais falsos, especialmente em mercados voláteis, e sua eficácia pode variar com base nas condições de mercado específicas e na configuração dos parâmetros do indicador.

Canais de Donchian: Os Canais de Donchian, um indicador técnico de significativa relevância no campo da análise financeira, foram criados em 1936 por Richard Donchian, frequentemente referido como o pai do trend trading. A concepção deste indicador surgiu da necessidade de Donchian, um trader de futuros, de possuir uma ferramenta simplificada

para medir a volatilidade do mercado, uma tarefa que, naquela época, apresentava-se como extremamente desafiadora:

Este indicador é composto por três linhas principais: a banda superior, a banda inferior e, opcionalmente, a banda média. A banda superior é definida pelo preço mais alto atingido durante um período específico, enquanto a banda inferior é estabelecida pelo preço mais baixo no mesmo intervalo. Embora menos comum, a banda média pode ser incluída como a média das duas bandas anteriores. O principal propósito dos Canais de Donchian é destacar a volatilidade e as tendências dos preços, servindo como um instrumento crucial para identificar rompimentos de tendência, reversões e potenciais condições de sobrecompra ou sobrevenda.

Na prática, os Canais de Donchian são amplamente utilizados por traders para identificar pontos estratégicos de entrada e saída no mercado. Um exemplo típico de uso é a geração de um sinal de compra quando o preço do ativo ultrapassa a banda superior, indicando uma possível tendência de alta. De forma inversa, um sinal de venda pode ser considerado quando o preço cai abaixo da banda inferior, sugerindo uma tendência de baixa.

A eficácia dos Canais de Donchian está intimamente ligada à escolha do período de análise. Períodos mais longos tendem a fornecer sinais mais estáveis, enquanto períodos mais curtos podem ser mais adequados para captar movimentos rápidos do mercado. Contudo, é importante notar que, como qualquer indicador técnico, os Canais de Donchian não são infalíveis e podem gerar sinais falsos, especialmente em mercados laterais ou altamente voláteis, onde o preço flutua frequentemente entre as bandas superior e inferior sem estabelecer uma tendência definida.

Time Series Momentum(*Time Series Momentum* (TSMOM)): O trabalho de *Moskowitz et al.* [9], desenvolve uma estratégia de Trend-Following que leva em consideração a volatilidade do ativo no dimensionamento da posição. O chamado Times Series Momentum, foca apenas na série temporal de preços do ativo, em oposição ao Cross-

Sectional Momentum, que procura extrair sinais da perfomance relativa dos ativos. A estratégia será utilizada para validar nosso projeto.

A essência da estratégia TSMOM é avaliar o retorno total de um ativo ao longo de uma janela de tempo no passado, geralmente o último ano, e tomar uma posição comprada, se este retorno foi positivo. Ou uma posição vendida, se o retorno foi negativo. Essa abordagem se fundamenta na hipótese de que os ativos que apresentaram um bom desempenho no passado tendem a continuar performando bem no futuro próximo, e vice-versa.

Matematicamente, sejam (σ_{tgt}) a volatilidade-alvo e (σ_t) a volatilidade realizada do ativo. A volatilidade-alvo é um parâmetro definido pelo investidor, representando sua tolerância ao risco. O retorno da estratégia TSMOM para um ativo em um intervalo de tempo específico é calculado pela seguinte fórmula:

$$r_{t,t+1}^{TSMOM} = \text{sign}(r_{t-252,t})\frac{\sigma_{tgt}}{\sigma_t} r_{t,t+1} \tag{2.2}$$

Onde, $r_{t-252,t}$ é o retorno total do ativo no último ano e $r_{t,t+1}$ é o retorno do ativo entre os períodos t e $t+1$. A função $\text{sign}(r_{t-252,t})$ determina a direção da posição (comprado ou vendido) baseada no sinal do retorno passado e $\frac{\sigma_{tgt}}{\sigma_t}$ o tamanho desta posição.

Quando aplicada a uma carteira contendo N ativos diferentes, o retorno total da carteira usando a estratégia TSMOM é obtido pela média dos retornos de cada ativo, conforme a seguinte equação:

$$rp_{t,t+1}^{TSMOM} = \frac{1}{N}\sum_{i=1}^{N} \text{sign}(r_{t-252,t}^{(i)})\frac{\sigma_{tgt}}{\sigma_t^{(i)}} r_{t,t+1}^{(i)} \tag{2.3}$$

Nesta equação, $r_{t-252,t}^{(i)}$ é o retorno total do i-ésimo ativo no último ano, $\sigma_t^{(i)}$ é a volatilidade realizada desse ativo, e $r_{t,t+1}^{(i)}$ é o retorno desse ativo entre os períodos t e $t+1$. A estratégia, portanto, combina os retornos ponderados de todos os ativos da carteira, ajustando cada posição com base na volatilidade-alvo e na volatilidade realizada

de cada ativo.

Convergência e Divergência de Médias Móveis (MACD) *Baz et al.* [10] desenvoleram uma estratégia de Trend Following baseada no cruzamento de médias móveis exponencias (*Exponential Weighted Moving Average* (EWMA)). A estratégia consiste em escolher três conjuntos de escalas temporais, cada um contendo uma escala curta e uma longa, que se traduzem na meia vida da EWMA.

Desta forma, define-se o indicador MACD como:

$$MACD(t, S, L) = EWMA(p_{-\infty:t}, S) - EWMA(p_{-\infty:t}, L) \tag{2.4}$$

Onde, $p_{-\infty:t}$ é a série de preços de fechamento até o tempo t e a meia vida da EWMA é dada por $HL(x) = log(0.5)/log(1 - \frac{1}{x})$.

O indicador MACD é normalizado com a volatilidade realizada dos últimos três meses:

$$q(t, S, L) = \frac{MACD(t, S, L)}{\sigma_{p_{[t-63:t]}}} \tag{2.5}$$

O sinal de tendência é então construído normalizado a série $q(t)$ com sua própria volatilidade realizada no último ano:

$$z(t, S, L) = \frac{q(t, S, L)}{\sigma_{p_{[t-252:t]}}} \tag{2.6}$$

A estratégia ainda agrega três combinações de escalas de tempo para definir o sinal de tendência final:

$$Z(t) = \frac{1}{3} \sum_{k=1}^{3} z(t, S_k, L_k) \tag{2.7}$$

Onde $S_k \in \{8, 16, 32\}$ e $L_k \in \{24, 48, 96\}$. Por fim, o dimensionamento da posição é modelado como uma função de resposta ao sinal Z:

$$x(z) = \frac{z \exp(\frac{-z^2}{4})}{0.89} \tag{2.8}$$

A função $x(z)$ tende a zero quando $|z| > \sqrt{2}$, tal propriedade permite que a estratégia reduza sua posição em situações em que está muito comprada ou muito vendida - definidas quando $|q(t)|$ é maior que $\sqrt{2}$ vezes o seu desvio padrão do último ano.

Os métodos tradicionais de trend-following são simples e diretos, mas também têm suas limitações. A principal delas é a sensibilidade a parâmetros arbitrários definidos pelo investidor, como o tamanho da janela de médias móveis, por exemplo. Além disso, os sinais gerados podem ser enganos e as mudanças na dinâmica do mercado podem não ser capturadas.

2.2.2 Aprendizagem de Máquina em Trend Following

No artigo [3], *Katsumi e Gomi* propuseram uma estratégia de Trend Following baseada em Redes Bayesianas. A rede é um grafo de dependências que codifica relações probabilísticas entre as variáveis discretas. Desta forma, a probabilidade de ocorrência de um estado modelado pela rede é dada por:

$$p(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} p(X_i | pa(X_i)) \tag{2.9}$$

Onde $pa(X_i)$ são os nós pais de X_i.

A estratégia consiste em estimar a probabilidade do retorno do ser positivo no próximo dia. Caso a probabilidade estimada seja maior que 50%, assume-se a posição comprada, e caso contrário, vende-se o ativo.

O modelo desenvolvido em [3] estima a probabilidade de retorno positivo do índice *SPX500* a partir de dados de fechamento do próprio *SPX500* em conjunto com o título de 10 anos do governo dos Estados Unidos e do índice *VIX*. Além desses dados, o modelo também incorpora um atributo de tendência, $s_n(t) = \frac{p(t-1) - EWMA(P_{[-\infty:t-1]}, n)}{\sigma_n(t-1)}$.

A arquitetura da rede é exibida a seguir:

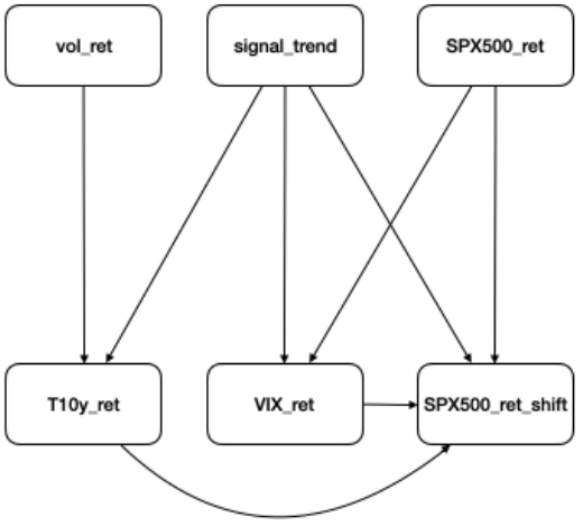

Figura 1: Arquitetura da Rede Bayesiana
Fonte: *Katsumi et al.* [3]

Variável	**Descrição**	**Valores**
SPX500_ret	Retorno do SPX500 > 0	{0,1}
signal_trend	Sinal de $s_n(t)$	{-1,1}
vol_ret	$\sigma_{t-1} - \sigma_{t-2} > 0$	{0,1}
T10y_ret	Retorno da Treasury 10 years > 0	{0,1}
VIX_ret	Retorno do Vix > 0	{0,1}
SPX500_ret_shift	Retorno do SPX500 no próximo dia > 0	{0,1}

Tabela 1: Descrição das variáveis da Rede Bayesiana

2.3 Backtesting

Em finanças, o Backtesting é uma forma de avaliar uma estratégia de negociação [11]. O método se baseia na simulação da utilização da estratégia desenvolvida em dados do passado, observando o que teria acontecido. Ela assume que se caso a estratégia tenha funcionado no passado, há chances de que ela continue funcionando no futuro e, do contrário, caso a estratégia não tenha performado bem no passado, provavelmente também não performará bem no futuro e pode ser descartada.

A avaliação da estratégia é realizada por meio de indicadores de desempenho, descritos a seguir:

2.3.1 Métricas de Desempenho

Retorno

O retorno nada mais é que a variação percentual do valor das posições tomadas nos ativos. Ele pode ser medido considerando vários horizontes temporais, por exemplo, o retorno diário é dado por:

$$r_d = \frac{v_d - v_{d-1}}{v_{d-1}} \tag{2.10}$$

Onde v_i representa o valor da carteira no dia i.

Dado uma série de retornos diários $(r_1, r_2, ..., r_n)$, obtém-se as seguintes medidas.

Retorno total (R)

Mostra o quanto a estratégia teria rendido no período de backtesting.

$$R = \prod_{d=1}^{n}(1 + r_d) - 1 \tag{2.11}$$

Retorno médio anualizado ($\overline{R}$)

Anualiza o retorno, o tornando uma medida comparável com outras taxas do mercado.

$$\overline{R} = \frac{252}{N}\sum_{d=1}^{N} r_d \tag{2.12}$$

Volatilidade anualizada (σ)

Indica o nível de risco da estratégia.

$$\sigma = \sqrt{Var(r_1, r_2, ..., r_n)} \times \sqrt{252} \tag{2.13}$$

Sharpe Ratio

Pondera o retorno sobre o risco, levando em consideração que espera-se que investimentos mais arriscados gerem maior retorno.

$$Sharpe = \frac{\overline{R}}{\sigma} \tag{2.14}$$

Sortino Ratio

É uma variação do Sharpe Ratio que ajusta o retorno de uma estratégia a uma meta de returno (r_T). Deste modo, apenas retornos abaixo da meta são penalizados.

$$Sortino = \frac{\overline{R}}{\sigma_{ds}} \tag{2.15}$$

Onde $\sigma_{ds} = stdev[min(r_1 - r_T, 0), ..., min(r_n - r_T, 0)] \times \sqrt{252}$

Drawdown Máximo

Medida de risco que mensura a maior queda no valor da carteira durante o período de testes.

$$MDD = \frac{min(r_1, ..., r_n) - max(r_1, ..., r_n)}{max(r_1, ..., r_n)} \tag{2.16}$$

Win rate (WR)

Calcula a razão dos dias em que a estratégia gerou retorno positivo.

$$WR = \frac{1}{N} \sum_{d=1}^{N} u(r_d) \tag{2.17}$$

Onde $u(x)$ é a função de Heaviside.

3 TEMPORAL FUSION TRANSFORMER

3.1 Motivação

O progresso contínuo no campo da inteligência artificial tem sido impulsionado pela complexidade crescente dos problemas enfrentados no mundo real, destacando a necessidade de métodos de análise de dados mais avançados e abrangentes. Os modelos iniciais de aprendizado de máquina, apesar de eficazes em contextos específicos, enfrentavam limitações significativas relacionadas à simplicidade na representação dos dados. Essas limitações manifestavam-se particularmente em cenários que exigiam a compreensão de padrões complexos, relações não lineares, ou quando era necessário discernir nuances sutis em grandes conjuntos de dados, conforme documentado em estudos de ciência de dados [12]. Por exemplo, modelos mais simples, como as Shallow Neural Networks, eram inadequados para tarefas como processamento de linguagem natural ou análise de séries temporais complexas, onde a interação entre diferentes variáveis ao longo do tempo é crucial.

Nesse contexto, o aprendizado profundo (deep learning) emergiu como uma solução inovadora para essas crescentes demandas analíticas. Esta abordagem se destaca pela sua capacidade de construir e manipular representações de dados mais complexas e abstratas, superando as restrições dos modelos anteriores de aprendizado de máquina. O aprendizado profundo baseia-se na premissa de que conceitos complexos podem ser decompostos e compreendidos através de representações mais simples e fundamentais. Esta ideia é exemplificada no design das redes neurais profundas, onde cada camada aprende a identificar padrões progressivamente mais complexos a partir de informações básicas [5].

Nas redes neurais profundas, as camadas iniciais geralmente se concentram em detectar características básicas, como bordas em imagens, enquanto as camadas subsequentes combinam essas informações iniciais para identificar formas e padrões mais complexos. Esta metodologia hierárquica é fundamental para o processamento eficaz de dados de alta dimensionalidade e complexidade, como observado em aplicações de processamento de imagem, áudio e linguagem natural [6]. Por exemplo, em redes neurais convolucionais (CNNs), as camadas iniciais podem identificar texturas e padrões, enquanto as camadas mais profundas podem reconhecer objetos específicos ou cenas inteiras em imagens.

A expansão do aprendizado profundo também promoveu o desenvolvimento de diversas arquiteturas inovadoras, cada uma adaptada a necessidades específicas. As CNNs, por exemplo, revolucionaram o campo do reconhecimento de imagens e vídeo, mostrando-se particularmente eficazes na identificação de padrões visuais complexos. Já as Redes Neurais Recorrentes (*Recurrent Neural Network* (RNN)s) e suas variantes, como as Long Short-Term Memory (LSTM), demonstraram ser fundamentais no processamento de sequências temporais, essenciais em tarefas como análise de texto e previsão de séries temporais [4]. Essas arquiteturas especializadas permitem que o aprendizado profundo aborde eficientemente uma gama diversificada de desafios, desde a análise detalhada de dados temporais até o reconhecimento e a classificação de imagens complexas.

Esses avanços marcaram uma nova era na capacidade dos sistemas de aprendizado de máquina de enfrentar e solucionar problemas complexos e variados. O aprendizado profundo não apenas melhorou a precisão e eficiência das soluções de aprendizado de máquina, mas também possibilitou inovações em diversas áreas, abrindo caminhos para avanços significativos em campos como reconhecimento de fala, tradução automática e diagnóstico médico. Este progresso demonstra a versatilidade e o poder do aprendizado profundo em uma variedade de domínios aplicados, evidenciando seu papel central na condução de inovações em diversas áreas da ciência e tecnologia [13].

3.2 Transformers

Dentro da arquitetura dos Transformers [14], dois mecanismos de atenção se destacam: a atenção de produto escalar escalonado (Scaled Dot-Product Attention) e a atenção multi-cabeça (Multi-Head Attention). Ambos são cruciais para o funcionamento avançado dos Transformers e permitem que esses modelos processem dados com uma eficiência e uma precisão notáveis.

A atenção por produto escalar é um mecanismo que avalia a importância relativa de diferentes partes de uma sequência de dados.Este mecanismo é projetado para avaliar e atribuir importância relativa às diferentes partes de uma sequência de dados, tornando-o fundamental para compreender relações complexas e interdependências, especialmente em contextos como processamento de linguagem natural [14].

Sua importância reside na capacidade de permitir que o modelo foque em partes específicas da sequência de dados enquanto processa e gera informações. Isso é particularmente valioso em tarefas como tradução automática, compreensão de texto e geração de linguagem, onde entender o contexto e a relevância de cada palavra ou frase em relação às outras é essencial para uma interpretação precisa e significativa.

O funcionamento da atenção por produto escalar é descrito pelas seguintes equações matemáticas:

$$Q = W_Q^T \cdot X_Q + B_Q \tag{3.1}$$

$$K = W_K^T \cdot X_K + B_K \tag{3.2}$$

$$V = W_V^T \cdot X_V + B_V \tag{3.3}$$

Nestas equações, Q, K, e V representam as matrizes de consulta, chave e valor, respectivamente. Cada uma é formada pela multiplicação dos dados de entrada X_Q, X_K, e X_V por matrizes de peso treináveis W_Q, W_K, e W_V, seguidas pela adição dos vetores de

viés B_Q, B_K, e B_V. Estes pesos e viés são ajustados durante o treinamento para capturar as informações mais relevantes dos dados.

A atenção é calculada usando o produto escalar entre os vetores de consulta e chave, seguido por um escalonamento:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3.4}$$

Aqui, d_k é a dimensão dos vetores chave. O escalonamento é realizado dividindo o produto escalar pela raiz quadrada de d_k para evitar gradientes excessivamente grandes durante o treinamento. A função softmax é aplicada para obter um conjunto de pesos normalizados, que são então usados para criar uma combinação ponderada dos vetores de valor, resultando na saída final da atenção; a qual reflete a informação de uma posição específica na sequência e como essa posição se relaciona com todas as outras, permitindo previsões ou geração de texto com um contexto detalhado .

A atenção multi-cabeça expande essa ideia ao dividir a consulta, a chave e o valor em múltiplas "cabeças", permitindo que o modelo execute a atenção de produto escalar escalonado em paralelo [14].

A mecânica da atenção multi-cabeça culmina na concatenação das saídas individuais de cada cabeça de atenção, seguida por uma transformação linear final. A concatenação das saídas é uma etapa crítica que agrega as representações atencionais paralelas em uma única representação composta. Matematicamente, a concatenação das saídas de todas as cabeças de atenção é denotada pela função *Concat*:

$$\text{ConcatenatedOutput} = \text{Concat}(head_1, head_2, \ldots, head_h) \tag{3.5}$$

onde cada vetor $head_i$ é o resultado da aplicação da atenção por uma cabeça individual, capturando aspectos distintos da entrada.

Após a concatenação, a representação combinada é refinada por uma transformação

linear final. Esta transformação é realizada por uma matriz de pesos treináveis W^O, que integra as informações de todas as cabeças em uma representação coesa:

$$\text{Output} = \text{ConcatenatedOutput} \cdot W^O \tag{3.6}$$

Esta transformação linear final não apenas redimensiona a saída para as dimensões subsequentes requeridas pelo modelo, mas também promove a integração das características atencionais diversas. A saída resultante é uma representação rica que encapsula informações contextuais detalhadas e abrangentes, possibilitando uma compreensão profunda da sequência de entrada.

Através da atenção multi-cabeça, os Transformers são capazes de capturar uma variedade mais ampla de dependências contextuais em dados sequenciais. Esta funcionalidade é particularmente valiosa em tarefas complexas de processamento de linguagem natural, onde a significância das interações entre elementos distantes pode ser tão impactante quanto as conexões locais [14]. A abordagem paralela e multifacetada dos Transformers proporciona uma compreensão contextual avançada, essencial para realizar previsões e inferências precisas em uma gama extensa de aplicações de aprendizado de máquina.

3.2.1 Codificador e Decodificador

A arquitetura dos Transformers é notavelmente composta por duas entidades distintas: o codificador e o decodificador. Cada uma dessas entidades desempenha um papel vital no processamento da sequência de entrada e na geração da sequência de saída, respectivamente.

3.2.1.1 Codificador

O codificador é constituído por uma pilha de N camadas idênticas, cada uma compreendendo duas sub-camadas principais:

- Uma sub-camada de atenção multi-cabeça, que executa a atenção simultaneamente

em diferentes posições da sequência de entrada, capturando assim as interdependências contextuais.

- Uma sub-camada de feed-forward neural network, consistindo de duas camadas lineares com uma ativação *Rectified Linear Units* (ReLU) intermediária.

Após a atenção multi-cabeça, cada elemento da sequência passa por uma rede neural feed-forward (FFNN), que desempenha um papel crucial no processamento subsequente dos dados. A FFNN é constituída por duas camadas lineares e uma função de ativação ReLU (Rectified Linear Unit) entre elas. Esta sub-camada é projetada para aplicar transformações não lineares aos dados, permitindo que o modelo capture relações complexas e sutis que não são possíveis de serem identificadas apenas com transformações lineares.

$$\text{FFNN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2 \tag{3.7}$$

Nesta equação, x representa o vetor de entrada, W_1 e W_2 são matrizes de pesos, e b_1 e b_2 são vetores de viés. Esses parâmetros são aprendidos durante o treinamento do modelo. A função ReLU, definida como $\text{ReLU}(z) = \max(0, z)$, é aplicada elemento a elemento e introduz não-linearidade no modelo, permitindo que ele aprenda funções mais complexas.

A ReLU é uma escolha popular para a função de ativação em muitas arquiteturas de redes neurais devido à sua simplicidade e eficácia. Ela ajuda a mitigar o problema do desvanecimento do gradiente, permitindo que modelos profundos sejam treinados de forma mais eficiente. Além disso, ao zerar valores negativos, a ReLU contribui para a criação de representações esparsas, o que pode ser benéfico em termos de eficiência computacional e capacidade do modelo de generalizar a partir dos dados de treinamento.

A presença da FFNN no codificador do Transformer reforça sua capacidade de processar e transformar a sequência de entrada, preparando-a eficientemente para as etapas subsequentes de processamento no modelo.

Além disso, cada sub-camada é seguida por uma operação de conexão residual e

normalização de camada, melhorando a estabilidade do treinamento em redes profundas.

3.2.1.2 Decodificador

O decodificador dos Transformers é fundamental para a geração de sequências de saída e possui uma arquitetura que ecoa o codificador, mas com algumas modificações chave para suportar a tarefa de geração de sequência. As principais sub-camadas do decodificador incluem:

1. **Sub-camada de atenção multi-cabeça**: Similar à do codificador, esta sub-camada permite que o decodificador processe a sequência de entrada. No entanto, no decodificador, esta atenção é aplicada às saídas intermediárias do próprio decodificador, permitindo que ele considere o que já foi gerado ao produzir a próxima parte da sequência.

2. **Sub-camada de atenção multi-cabeça modificada**: Esta camada é exclusiva do decodificador e é crucial para a auto-regressividade do modelo. Ela permite que o decodificador foque nas posições da sequência que já foram decodificadas, evitando a utilização de informações futuras na geração da sequência atual. Este mecanismo de atenção multi-cabeça modificada é implementado utilizando uma abordagem de mascaramento, que bloqueia efetivamente a atenção para as posições futuras na sequência de saída. Este mascaramento é realizado aplicando uma máscara à matriz de atenção antes da aplicação da função softmax. A máscara é configurada de tal forma que as posições futuras na sequência recebem valores negativos extremamente grandes (usualmente, $-\infty$) após a operação de produto escalar, resultando em valores próximos a zero após a aplicação da função softmax.

   A equação para a atenção com mascaramento pode ser expressa da seguinte maneira:

$$\text{Attention\_masked}(Q, K, V, \text{mask}) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \text{mask}\right)V \tag{3.8}$$

Nesta equação, mask é a matriz de mascaramento que bloqueia as posições futuras. As matrizes Q, K e V representam, respectivamente, as matrizes de consulta, chave e valor, semelhantes às usadas na atenção multi-cabeça padrão do codificador. O processo de mascaramento assegura que, durante a geração da sequência, cada posição no decodificador só pode atender às posições até e incluindo sua própria posição na sequência.

Essa abordagem de mascaramento é essencial para preservar a natureza auto-regressiva do Transformer no decodificador. Ela garante que cada elemento na sequência de saída seja condicionado apenas pelas informações disponíveis até aquele ponto, respeitando a ordem sequencial dos dados e permitindo que o modelo gere sequências coerentes.

3. **Sub-camada de feed-forward neural network**: Idêntica à encontrada no codificador, esta sub-camada aplica transformações não lineares às saídas das camadas de atenção para processar e refinar ainda mais as representações da sequência.

Assim como no codificador, cada sub-camada no decodificador é acompanhada por uma conexão residual e normalização de camada, técnicas que melhoram a estabilidade do treinamento e a capacidade do modelo de aprender eficazmente em arquiteturas profundas.

3.2.1.3 Processo de Codificação e Decodificação

No processo de codificação, a sequência de entrada percorre as camadas do codificador, com cada camada aplicando atenção e transformações neurais. A saída de cada camada do codificador serve como entrada para a próxima, culminando em uma representação rica da entrada.

O decodificador, por sua vez, utiliza a saída do codificador em cada passo de decodificação. As sub-camadas dentro do decodificador trabalham de forma integrada para considerar tanto a saída do codificador quanto as saídas previamente decodificadas, garantindo que cada elemento decodificado seja condicionado pela entrada completa e pelas

saídas anteriores.

Essa arquitetura interconectada permite que os Transformers processem sequências de maneira eficiente, sem depender de recorrência ou convoluções, e é essencial para a paralelização e a desempenho em tarefas complexas de sequência como tradução automática e análise de texto.

3.2.2 Temporal Fusion Transformer (TFT)

O Temporal Fusion Transformer (TFT), uma inovação significativa na arquitetura de redes neurais, representa uma evolução importante na análise de séries temporais. Introduzido por Bryan Lim e colaboradores em 2020 no artigo "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting"[5], o TFT foi projetado especificamente para lidar com a complexidade das séries temporais multivariadas. Sua introdução no campo da análise de dados marca um avanço notável, especialmente considerando as demandas cada vez maiores por modelos que não apenas processam grandes volumes de dados, mas que também capturam as interdependências e dinâmicas temporais complexas.

A distinção do Temporal Fusion Transformer (TFT) em relação aos Transformers padrão reside fundamentalmente na sua abordagem inovadora quanto à temporalidade dos dados. Os Transformers tradicionais, conhecidos por sua eficácia no processamento de sequências de dados, tratam cada elemento dentro de uma série temporal de forma igualitária, sem dar ênfase especial à ordem em que os dados ocorrem. No entanto, o TFT altera essa dinâmica ao incorporar uma sensibilidade aguçada à temporalidade, um aspecto crucial para a análise de séries temporais [5].

Essa sensibilidade é alcançada principalmente através da camada de atenção temporal do TFT, uma inovação que permite ao modelo avaliar e ponderar a importância relativa dos diferentes pontos no tempo [5]. Esta camada não apenas considera cada ponto de dados individualmente, mas também avalia sua relevância em relação ao horizonte de

previsão. Por exemplo, em uma série temporal financeira, dados mais recentes podem ser mais indicativos das tendências futuras do mercado do que dados mais antigos. O TFT, através de sua camada de atenção temporal, identifica e dá mais peso a esses dados recentes, possibilitando assim previsões mais precisas.

Além disso, a camada de atenção temporal do TFT é capaz de identificar padrões e dependências temporais que podem não ser imediatamente aparentes [5]. Por exemplo, pode reconhecer como certos eventos ou condições em momentos específicos influenciam os desenvolvimentos futuros, uma capacidade essencial para prever tendências em mercados voláteis. Esta abordagem melhora significativamente a capacidade do modelo de fazer previsões informadas, permitindo-lhe reconhecer padrões complexos e sutis que poderiam ser perdidos em modelos mais tradicionais.

A seguir, apresenta-se um diagrama do modelo e, subsequentemente, seus componentes:

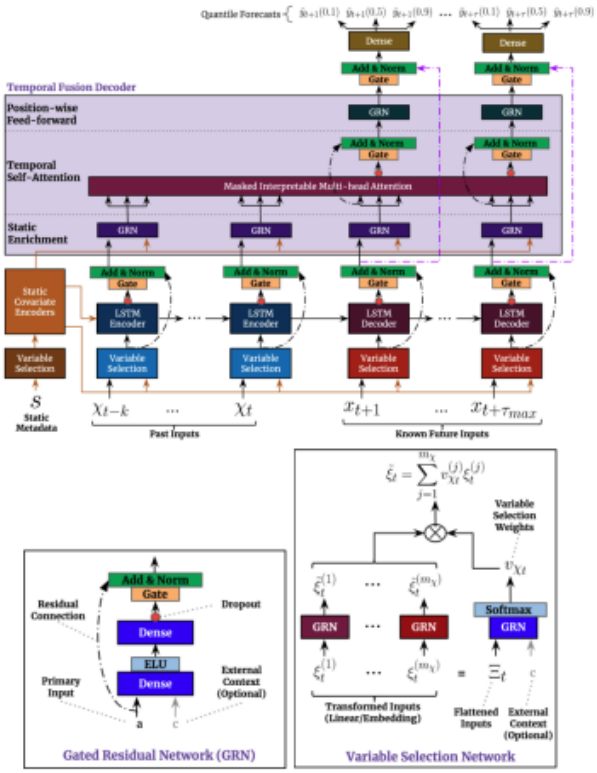

Figura 2: Arquitetura do Temporal Fusion Transformer.
Fonte: *Lim et al.* [5]

3.2.2.1 Long Short-Term Memory Units (LSTM)

As Long Short-Term Memory Units (LSTMs) são uma especialização das redes neurais recorrentes, concebidas para capturar dependências temporais de longo alcance em sequências de dados. Estas unidades superam as limitações das redes recorrentes tradicionais, que frequentemente enfrentam dificuldades para manter dependências temporais extensas devido ao problema do desvanecimento do gradiente.

As LSTMs são caracterizadas por uma arquitetura única que inclui estruturas de portões, responsáveis por regular o fluxo de informações ao longo da sequência. Estes portões incluem:

- **Portão de Entrada:** Determina a adição de novas informações ao estado da célula. É definido pelas equações:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i), \quad \% \text{ onde } \sigma \text{ é a função sigmoide} \tag{3.9}$$

$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C), \quad \% \text{ tanh é a tangente hiperbólica} \tag{3.10}$$

  Aqui, i_t é a ativação do portão de entrada, e as matrizes W e U são os pesos, enquanto b representa o vetor de viés.

- **Portão de Esquecimento:** Gerencia a retenção de informações do estado anterior da célula:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f). \tag{3.11}$$

- **Atualização do Estado da Célula:** Combina o estado anterior da célula com as novas informações:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t. \tag{3.12}$$

- **Portão de Saída:** Regula as informações do estado da célula que contribuirão para

a saída:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \tag{3.13}$$

$$h_t = o_t \cdot \tanh(C_t). \tag{3.14}$$

3.2.2.2 Gated Residual Network (GRN)

A inclusão de portões nas GRNs é inspirada nas células das LSTMs, que são conhecidas por sua eficácia em capturar dependências temporais em dados sequenciais [1]. As GRNs refinam essa abordagem ao combinar a capacidade de regulação dos portões com a habilidade de conexões residuais em mitigar o problema do desaparecimento do gradiente, um desafio comum em redes profundas [5].

A função primária dos portões nas GRNs é determinar a relevância das informações processadas por cada camada. Isto é crucial para manter a relevância e a precisão dos dados processados. Uma inovação das GRNs é a mitigação do problema do desvanecimento do gradiente em redes profundas. A combinação dos portões com as conexões residuais facilita a manutenção dos gradientes essenciais ao longo do processo de aprendizado.

A operação de uma GRN pode ser matematicamente representada por um conjunto de equações que ilustram a transformação dos dados de entrada e a aplicação dos mecanismos de portões e de conexão residual:

$$Z = W_z X + b_z, \tag{3.15}$$

$$R = \sigma(W_r Z + U_r H + b_r), \tag{3.16}$$

$$H' = \mathrm{ReLU}(W_h Z + U_h(R \odot H) + b_h), \tag{3.17}$$

onde W_z, W_r, W_h são matrizes de pesos, b_z, b_r, b_h são vetores de viés, σ representa a função sigmóide, ReLU é a função de ativação Rectified Linear Unit, $\odot$ denota a operação de multiplicação elemento a elemento, e H é o estado anterior da rede.

A saída da GRN é então obtida pela combinação do estado atualizado H' com a entrada original X, utilizando uma conexão residual:

$$H_{\text{next}} = H' + X. \tag{3.18}$$

3.2.2.3 Gated Linear Units (GLU)

Os Gated Linear Units (GLUs) são componentes fundamentais nas modernas arquiteturas de redes neurais, desempenhando um papel crucial na modelagem de interações complexas e na captura de padrões não lineares em dados. Esses elementos arquiteturais são uma fusão inovadora entre as operações lineares tradicionais e um mecanismo de controle adaptativo, conhecido como portão. Este portão, geralmente implementado através de uma função sigmoide, regula o fluxo de informação processada de maneira linear, permitindo uma seleção mais precisa e contextual das características dos dados [5].

A operação dos GLUs pode ser matematicamente descrita como a multiplicação elementar da transformação linear dos dados de entrada por um vetor de portão. A transformação linear é tipicamente realizada por uma matriz de pesos seguida de um viés, enquanto o vetor de portão é calculado por uma função sigmoide aplicada a uma transformação semelhante dos dados. A função sigmoide, com sua característica de ativação não linear, modula a informação linear, permitindo ao GLU controlar a influência de cada característica na representação final. Esta combinação de linearidade com controle adaptativo confere aos GLUs a capacidade de capturar interações complexas entre características, o que é particularmente benéfico em contextos como séries temporais e processamento de linguagem natural.

A matemática subjacente aos GLUs pode ser expressa através das seguintes equações:

$$\text{Linear} = W \cdot X + b, \tag{3.19}$$

$$\text{Gate} = \sigma(W_g \cdot X + b_g), \tag{3.20}$$

$$\text{GLU} = \text{Linear} \odot \text{Gate}, \tag{3.21}$$

onde W e W_g representam as matrizes de pesos para a transformação linear e para o cálculo do portão, respectivamente; b e b_g são os vetores de viés correspondentes; σ denota a função sigmoide; e $\odot$ simboliza a multiplicação elementar.

Através do uso de GLUs, os modelos de aprendizado profundo podem realizar uma seleção dinâmica e adaptativa das informações relevantes. Isso é essencial para lidar com a dinâmica temporal em séries temporais, onde a relevância de características específicas pode variar significativamente ao longo do tempo ou em diferentes cenários. Os GLUs, portanto, não apenas aprimoram a capacidade do modelo de lidar com complexidade e não linearidade, mas também aumentam a interpretabilidade e a eficácia dos modelos.

3.2.2.4 Variable Selection Networks (VSNs)

As *Variable Selection Networks* (VSNs), integradas ao *Temporal Fusion Transformer* (TFT), representam um avanço significativo na modelagem de séries temporais multivariadas. Estas redes são essenciais para gerir eficazmente a complexidade decorrente da abundância de variáveis de entrada em dados temporais de múltiplas variáveis.

A VSN avalia cada variável individualmente e atribui pesos que refletem sua importância para a previsão utilizando técnicas avançadas de atenção. Este processo dinâmico de ponderação é ajustado continuamente, permitindo que a rede adapte sua estratégia de seleção de variáveis em resposta a mudanças nos dados.

A operação de uma VSN pode ser descrita por uma série de transformações e aplicações de atenção. Inicialmente, cada variável de entrada é processada por uma transformação linear, seguida pela aplicação de uma função de atenção. Este processo é representado

pelas seguintes equações:

$$Z_{\text{var}} = W_{\text{var}} X_{\text{var}} + b_{\text{var}}, \tag{3.22}$$

$$\alpha_{\text{var}} = \text{softmax}(W_{\text{atten}} Z_{\text{var}} + b_{\text{atten}}), \tag{3.23}$$

onde X_{var} são as variáveis de entrada, W_{var} e b_{var} são os pesos e vieses da transformação linear, W_{atten} e b_{atten} são os pesos e vieses da camada de atenção, e α_{var} são os pesos de atenção atribuídos a cada variável.

A saída da VSN é uma combinação ponderada das variáveis de entrada, refletindo sua importância relativa:

$$X_{\text{selected}} = \sum_{\text{var}} \alpha_{\text{var}} \cdot X_{\text{var}}. \tag{3.24}$$

Esta abordagem não apenas melhora a eficácia do modelo em termos de precisão preditiva, mas também aumenta sua transparência e interpretabilidade [5].

3.2.2.5 Convolutional Neural Networks (CNN)

As *Convolutional Neural Networks* (CNNs), embora tradicionalmente associadas ao processamento de imagens, encontram aplicabilidade no tratamento de dados sequenciais no *Temporal Fusion Transformer* (TFT). Nestes modelos, as CNNs são adaptadas para identificar padrões e características locais em séries temporais.

A capacidade das CNNs de aplicar filtros ou *kernels* sobre a entrada permite a detecção eficaz de padrões temporais recorrentes. Esta habilidade é fundamental para extrair características como tendências e ciclos sazonais, cruciais em séries temporais.

Além disso, as CNNs contribuem significativamente para a redução da dimensionalidade dos dados, condensando informações complexas em representações mais abstratas. Este processo é vital para a eficiência computacional e para a prevenção de sobreajuste,

aumentando a generalizabilidade e robustez do modelo [5].

As operações principais das CNNs em séries temporais podem ser descritas matematicamente:

1. **Aplicação de Kernels:**

$$F_{\text{conv}} = \text{ReLU}(W_{\text{kernel}} * X + b_{\text{kernel}}), \tag{3.25}$$

onde W_{kernel} e b_{kernel} são os pesos e vieses do kernel, X é a entrada, $*$ denota a operação de convolução, e ReLU é a função de ativação.

2. **Redução de Dimensionalidade:**

$$X_{\text{reduced}} = \text{Pooling}(F_{\text{conv}}), \tag{3.26}$$

onde Pooling refere-se a uma operação de agrupamento, como max pooling ou average pooling, que condensa as características extraídas.

3.2.2.6 Dense Layers

As *Dense Layers*, ou camadas totalmente conectadas, desempenham um papel crítico após as operações de atenção e convolução, sendo responsáveis por transformações não lineares e integração de características extraídas.

Nestas camadas, cada neurônio está conectado a todos os neurônios da camada anterior, facilitando a execução de transformações abrangentes e complexas. Esta conexão total é essencial para a aprendizagem de representações abstratas e integradas, cruciais para a análise de dados complexos, como os encontrados em séries temporais.

O papel das camadas densas no TFT é integrar as características processadas pelas camadas de atenção e convolução, transformando-as em uma representação mais abstrata e informativa. Isso é alcançado através da aplicação de ativações não lineares, como a função *ReLU* (Rectified Linear Unit), permitindo a modelagem de relações não lineares entre os dados.

As operações em uma camada densa incluem:

1. **Transformação Linear:**

$$Z = W \cdot X + b, \tag{3.27}$$

onde W e b representam os pesos e vieses da camada, respectivamente, e X é a entrada.

2. **Aplicação da Função de Ativação Não Linear:**

$$A = \text{ReLU}(Z), \tag{3.28}$$

onde ReLU é a função de ativação, adicionando não linearidades ao modelo.

A integração de camadas densas no TFT ilustra a necessidade de representações poderosas e abstratas na análise de séries temporais complexas, destacando a capacidade do modelo de aprender e prever padrões sofisticados [5].

3.2.2.7 Skip Connections

As *Skip Connections*, ou conexões residuais, são uma inovação arquitetural essencial nas redes neurais, especialmente no *Temporal Fusion Transformer* (TFT) [1]. Estas conexões permitem o fluxo direto de informações entre camadas não adjacentes, melhorando a propagação de gradientes durante o treinamento e aumentando a eficiência do aprendizado em redes profundas.

Um desafio comum em redes neurais profundas é o desvanecimento do gradiente, onde a magnitude do gradiente de erro se reduz à medida que atravessa as camadas. Isso pode prejudicar o aprendizado, principalmente nas camadas mais internas da rede. As *skip connections* abordam esse problema ao fornecer um caminho alternativo para o gradiente, preservando sua força ao longo da rede.

No TFT, a importância das *skip connections* reside em manter a integridade das informações ao longo do processo de aprendizado, assegurando que características essenciais não sejam perdidas e que camadas profundas continuem a aprender efetivamente [5].

A representação matemática das *skip connections* é dada por:

$$H_{\text{next}} = \text{Activation}(H_{\text{current}} + X), \tag{3.29}$$

onde H_{current} é a saída da camada atual, X é a entrada da camada, e Activation é a função de ativação aplicada. Esta estrutura permite a preservação da informação original e sua combinação com as transformações aprendidas, fortalecendo a robustez e eficácia do modelo [5].

3.2.2.8 Normalization and Dropout

A *Normalization* visa estabilizar e acelerar o treinamento ajustando as ativações de cada camada para manter uma distribuição consistente. Isso é vital para evitar problemas como a explosão do gradiente, que podem comprometer o aprendizado [5].

No TFT, técnicas como *Batch Normalization* ou *Layer Normalization* são aplicadas para normalizar as ativações, garantindo que a média das saídas de cada camada esteja próxima a zero e o desvio padrão próximo a um. Esta normalização facilita um treinamento mais rápido e estável:

- A *Batch Normalization* normaliza as ativações de uma camada ao longo de um batch de dados. Esta técnica ajusta e escala as ativações para que a média fique próxima de zero e o desvio padrão próximo de um. Isso é realizado através do cálculo da média e do desvio padrão das ativações para cada batch, seguido pela aplicação da normalização. A principal vantagem desta técnica é a redução do "internal covariate shift", facilitando o treinamento e melhorando a eficiência [5].

- A *Layer Normalization* difere da *Batch Normalization* ao normalizar os dados ao longo de todas as ativações de uma camada para cada amostra individualmente. Isso é especialmente benéfico em situações com tamanhos de batch menores ou em aplicações com dados sequenciais, comuns no uso do TFT [5].

Por outro lado, o *Dropout* é uma estratégia de regularização usada para reduzir o overfitting. Durante o treinamento, alguns neurônios são aleatoriamente desativados, o que ajuda o modelo a não se tornar excessivamente dependente de características específicas dos dados de treinamento, melhorando a generalização para dados novos. No TFT, o *Dropout* é implementado para aumentar a robustez e a capacidade de generalização do modelo, essencial para previsões precisas em dados desconhecidos [5].

4 IMPLEMENTAÇÃO E EXPERIMENTOS

4.1 Aquisição de dados

As séries históricas de dados financeiras foram obtidas a partir da API do *Yahoo Finance* [15]. Nela, a seleção dos dados é feita a partir do *ticker* do ativo, das datas de início e final da série e da frequência das observações.

Os dados retornados seguem o formato abaixo:

Variável	**Descrição**	**Tipo de dado**
Date	Data (Y-m-d) correspondente a observação.	String
Open	Preço de abertura do dia.	Float
High	Maior preço observado no período.	Float
Low	Menor preço observado no período	Float
Close	Preço de fechamento do dia.	Float
Adjclose	Preço de fechamento desconsiderando proventos.	Float
Volume	Volume de negociação obeservado no período.	Integer

Tabela 2: Documentação API Yahoo Finance

No presente trabalho, foi decidido utilizar uma série de dados diários iniciando em 01/01/2000 e finalizando em 13/11/2023 dos *tickers* da tabela (3):

Ticker	**Descrição**	**Setor**
ES=F	E-mini S&P 500	Equities
YM=F	Mini Dow Jones	Equities
NQ=F	Nasdaq 100	Equities
GC=F	Ouro	Commodities
SI=F	Prata	Commodities
ZC=F	Milho	Commodities
CL=F	Petroléo Cru	Commodities
SB=F	Açucar	Commodities
CT=F	Algodão	Commodities
ZB=F	Título do Tesouro Americano	Fixed Income
ZT=F	Nota de 2 anos do Tesouro Americano	Fixed Income
ZF=F	Nota de 5 anos do Tesouro Americano	Fixed Income
ZN=F	Nota de 10 anos do Tesouro Americano	Fixed Income
EUR=X	Euro	FX
JPY=X	Iene Japonês	FX
GBP=X	Libra Esterlina	FX
BRL=X	Real Brasileiro	FX
MXN=X	Peso Mexicano	FX
CAD=X	Dólar Canadense	FX

Tabela 3: Ativos utilizados no modelo

4.2 Limpeza dos Dados

A etapa de limpeza e preparação dos dados é fundamental para assegurar que o modelo de machine learning receba informações de qualidade e relevantes para o processo de tomada de decisões preditivas. Neste estudo, a atenção concentra-se exclusivamente nos dados de *Adjusted Close*, que refletem os movimentos dos preços ajustados por eventos corporativos como dividendos e splits [15] e são considerados uma representação mais precisa do valor de mercado. Uma etapa meticulosa de pré-processamento foi realizada para isolar esses dados, conforme ilustrado na Figura (3).

O pré-processamento incluiu o tratamento de dados faltantes e a remoção de outliers - identificados quando o retorno diário era maior que 50%. Para ambos os casos, os dados foram preenchidos repetindo a observação anterior. Assim, é possível simular que não ouve atividade no mercado nos dias de má qualidade dos dados.

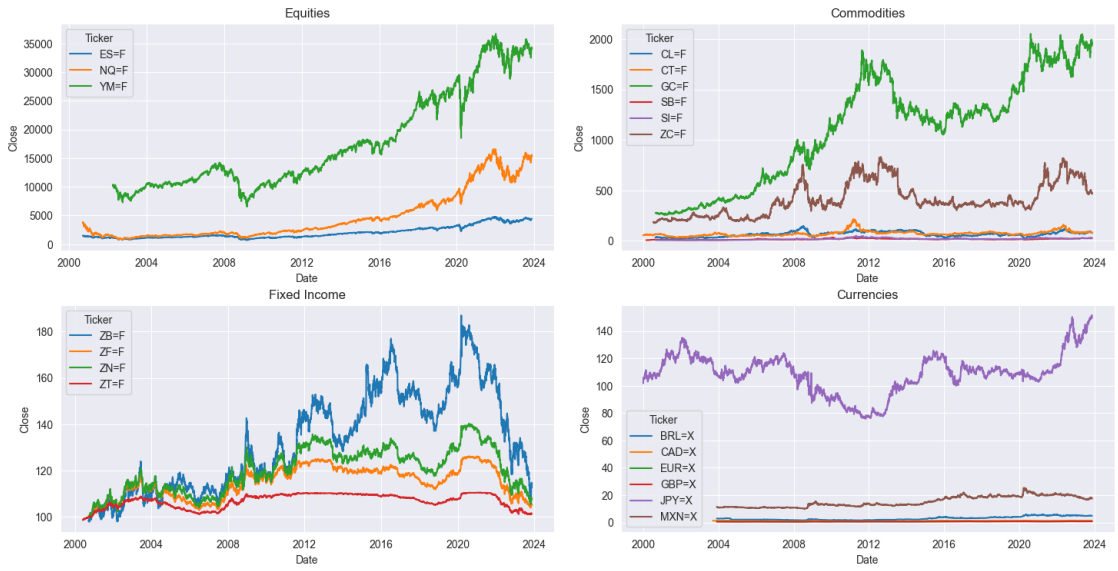

Figura 3: Adjusted Close dos tickers selecionados por setor

4.3 Implementação das estratégias de validação

A fim de conseguirmos validar se a estratégia construída neste projeto supera outras abordagens de Trend Following vamos compará-la com as estratégias de Moskowitz,

MACD e a Rede Bayesiana discutidas no capítulo 2. Incluiremos também a estratégia Long Only, a qual sempre assume posição comprada.

Na implementação das estratégia de Moskowitz foi utilizado σ_{tgt} de 5% para os futuros de renda fixa e 15% para os demais ativos. Adotamos as escalas de tempo da literatura no MACD, ou seja $S_k \in \{8, 16, 32\}$ e $L_k \in \{24, 48, 96\}$.

Ambas estratégias foram implementadas usando a biblioteca *Numpy* de computação numérica, nela é possível realizar os cálculos descritos no capítulo 2 de forma vetorizada.

Já para a Rede Bayesiana, adaptamos o código fonte do artigo [3] para considerar retornos proporcionais, conforme a equação (2.10). Além disto, utilizamos apenas os ativos do setor de Equities, pois a Rede foi projetada para operar sobre o SPX500. Desta forma, estendê-la para outras classes de ativos não faz sentido e tornaria a comparação injusta.

4.4 Implementação da estratégia baseada no TFT

A seguir, descreveremos e justificaremos os passos tomados para desenvolver a estratégia de Trend Following baseada no Temporal Fusion Transformer.

4.4.1 Engenharia de atributos

Conforme discutido em [16] e em [17], a etapa de engenharia de atributos é fundamental na construção de modelos de aprendizagem máquina. Por meio dela é possível adicionar informações implícitas no dado cru e assim facilitar a aprendizagem do modelo. Nesta seção iremos apresentar os atributos que incorporamos ao modelo.

A fim de aprimorar a análise pelo modelo e conferir maior estacionariedade à série temporal — característica crucial para a eficácia de modelos preditivos —, optou-se pela transformação logarítmica dos dados de retorno [18]. Esta técnica é reconhecida por sua capacidade de estabilizar flutuações nos retornos e por sua eficiência em linearizar relações

que originalmente apresentam um comportamento exponencial ou de potência. A transformação logarítmica, portanto, é uma intervenção estratégica nos dados que visa facilitar a identificação de padrões e tendências pelo modelo, contribuindo significativamente para a precisão das previsões.

Também adicionamos atributos que refletem as tendências, ciclos e volatilidade do mercado, que são elementos cruciais na previsão de séries temporais financeiras. São eles:

- **Retorno atrasado correspondente ao tamanho do encoder**: permite ao modelo acessar informações sobre o comportamento passado do preço ao longo de um período equivalente ao horizonte de codificação.
- **Retorno atrasado correspondente ao tamanho da previsão**: Oferece contexto temporal direto para o período que o modelo está tentando prever.
- **Média móvel do retorno**: Calculada em uma janela temporal de um mês, fornecendo uma visão suavizada da tendência de médio prazo.

  A média móvel é calculada seguindo este procedimento:

  1. Primeiramente, determina-se a média do retorno dos preços de fechamento ao longo de uma janela móvel de 22 dias úteis. Esta média fornece uma visão suavizada das tendências do mercado, atenuando as flutuações diárias e destacando tendências de médio prazo.
  2. A média móvel do retorno (MM_{22d}) é expressa como:

     $$\mathrm{MM}_{22d} = \frac{1}{22} \sum_{i=t-21}^{t} \mathrm{Retorno}_i, \tag{4.1}$$

     onde $\mathrm{Retorno}_i$ é o retorno no dia i, e t representa o dia atual.

- **Média móvel da volatilidade**: Em uma janela de um mês, que reflete as flutuações e a incerteza do mercado.

  Calculada da mesma forma que a média móvel do retorno, substituindo os valores diários do retorno pela volatilidade diária.

- **Média móvel exponencial ponderada de retorno**: Atribui maior peso aos dados mais recentes, capturando a inércia dos preços de uma maneira mais responsiva:

  A EWMA é calculada de acordo com os seguintes passos:

  1. A EWMA atribui maior peso aos preços mais recentes, tornando-a mais sensível às últimas mudanças do mercado comparada à média móvel simples. Isso é conseguido através de uma fórmula de ponderação exponencial.
  2. A EWMA para um período específico é calculada utilizando a seguinte fórmula:

$$\text{EWMA}_t = (\text{Preço}_t \times K) + (\text{EMA}_{t-1} \times (1 - K)), \tag{4.2}$$

     onde EWMA_t é a **EMWA!** (**EMWA!**) no tempo t, Preço_t é o preço de fechamento no tempo t, EWMA_{t-1} é a EWMA no tempo $t-1$, e K é o fator de suavização, dado por $K = \frac{2}{\text{Período}+1}$.

- **Índice de força relativa (RSI) (7, 14, 22, 30 e 60 dias)**: Mede a velocidade e a mudança dos movimentos de preço, um indicador amplamente utilizado para identificar condições de sobrecompra ou sobrevenda:

  O RSI é calculado seguindo estas etapas:

  1. Primeiramente, calcula-se a variação diária de preços (*delta*), que é a diferença entre o preço de fechamento consecutivo dos ativos:

$$\text{Delta} = \text{Preço de Fechamento}_t - \text{Preço de Fechamento}_{t-1}. \tag{4.3}$$

  2. Em seguida, essa variação é decomposta em componentes positivos (ganhos) e negativos (perdas). Os ganhos são definidos como zero para dias com variação negativa e as perdas como zero para dias com variação positiva.
  3. Para diferentes períodos , calculam-se as médias dos ganhos e perdas. Essas médias são usadas para calcular a força relativa (RS), que é a razão entre a

média de ganhos e a média de perdas:

$$\text{RS} = \frac{\text{Média de Ganhos}}{\text{Média de Perdas}}. \tag{4.4}$$

4. Finalmente, o RSI é calculado como:

$$\text{RSI} = 100 - \frac{100}{1 + \text{RS}}. \tag{4.5}$$

- **Convergência e Divergência de Médias Móveis (MACD)**: Indicador MACD, introduzidos no capítulo 2 e definidos pela equação 2.4, considerando $S_k = 12$ e $L_k = 26$.

- **Codificação do mês e dia do mês como variáveis cíclicas**: Capturar padrões sazonais; a transformação é realizada da seguinte maneira:

  1. Para cada coluna temporal (mês ou dia do mês), aplica-se uma transformação baseada em funções trigonométricas para converter os valores lineares em uma representação cíclica. A fórmula aplicada é:

$$\text{Coluna Codificada} = \cos\left(2\pi \frac{\text{Coluna Original}}{\text{Valor Máximo}}\right). \tag{4.6}$$

  2. Nesta fórmula, o Valor Máximo é o valor máximo da coluna original (por exemplo, 12 para meses e 31 para dias). Esta transformação resulta em valores que respeitam a natureza cíclica da característica temporal.

4.4.2 Otimização do Modelo

4.4.2.1 Função de Perda - Squared Mean Aggregated Error (SMAE)

A função de perda *Squared Mean Aggregated Error* (SMAE) é uma métrica desenvolvida para avaliar o desempenho de modelos de previsão em tarefas onde a previsão agregada é mais relevante do que previsões pontuais [19].

Considerando um conjunto de previsões $\hat{y}_t$ e valores reais y_t para um horizonte de previsão T, a SMAE é calculada da seguinte maneira:

1. Primeiro, as previsões são agregadas para cada série temporal:

$$S_{\hat{y}} = \sum_{t=1}^{T} \hat{y}_t \tag{4.7}$$

$$S_y = \sum_{t=1}^{T} y_t \tag{4.8}$$

2. O erro absoluto é calculado como a diferença absoluta entre as somas agregadas:

$$E_t = |S_{\hat{y}} - S_y| \tag{4.9}$$

3. Este erro é então elevado ao quadrado para penalizar mais intensamente desvios maiores:

$$E_t^2 = (E_t)^2 \tag{4.10}$$

4. A SMAE é a média dos erros quadráticos ao longo de todas as séries temporais:

$$\text{SMAE} = \frac{1}{N} \sum_{i=1}^{N} E_i^2$$

onde N representa o número total de séries temporais analisadas.

A utilização da SMAE como função de perda em modelos de previsão enfatiza a precisão em estimativas agregadas, tornando-a uma métrica adequada para aplicações onde a soma ou o valor total previsto é de interesse principal [19].

4.4.2.2 SMAE Comparada a MAE, MSE e RMSE

A seleção da função de perda *Squared Mean Aggregated Error* (SMAE) foi uma decisão estratégica, refletindo as necessidades específicas da tarefa de previsão em questão. Diferentemente de métricas tradicionais como Mean Absolute Error (MAE), Mean Squared Error (MSE), ou Root Mean Square Error (RMSE), que se concentram em avaliar erros em previsões pontuais, a SMAE enfoca a precisão das estimativas agregadas. Esta característica é de particular importância em análises onde o desempenho agregado ao longo do tempo é mais relevante do que a precisão em momentos isolados.

Além de seu enfoque nas estimativas agregadas, a SMAE incorpora uma penalização mais severa para desvios maiores, alinhando-se com cenários onde grandes erros são menos toleráveis. Esta penalização mais intensa para erros maiores é crucial em aplicações financeiras e de risco, onde desvios significativos podem ter consequências substanciais.

Outro fator chave na escolha da SMAE é a sua capacidade de simplificar o processo de otimização do modelo. Ao agregar as previsões antes do cálculo do erro e, em seguida, elevar esse erro ao quadrado, a SMAE facilita a identificação de padrões e tendências agregadas, tornando o processo de otimização mais direcionado e eficiente.

A SMAE também se destaca por sua interpretabilidade e relevância em contextos onde as decisões são baseadas no resultado agregado de previsões. Em tais cenários, a SMAE oferece uma visão clara e intuitiva do desempenho do modelo, facilitando a compreensão e a tomada de decisões informadas.

A abordagem de agregação empregada na SMAE ajuda a reduzir a variância nas estimativas de erro, proporcionando uma medida de desempenho mais estável e confiável. Esse aspecto é especialmente valioso em análises de séries temporais, onde a consistência nas estimativas de desempenho é fundamental.

4.4.2.3 Early Stopping

No processo de treinamento, o *Early Stopping* é uma técnica essencial para prevenir overfitting. Ela é configurada no PyTorch Lightning [**?**] com os seguintes parâmetros:

- **Monitor:** Define a métrica que o *Early Stopping* irá observar. A métrica selecionada é utilizada como referência para determinar se o treinamento está progredindo e se deve ser continuado.
- **Min_delta:** Estabelece a alteração mínima na métrica observada que é considerada uma melhoria. Este parâmetro evita que o treinamento seja interrompido por variações insignificantes, garantindo que apenas mudanças significativas na métrica influenciem a decisão de parar o treinamento.

- **Patience:** Determina o número de épocas sem melhoria após o qual o treinamento será interrompido. Este parâmetro é vital para proporcionar ao modelo uma chance adequada de melhorar antes de cessar o treinamento.

- **Mode:** Especifica se o treinamento deve ser interrompido com base na minimização ('min') ou maximização ('max') da métrica monitorada. A escolha depende se a métrica deve ser otimizada por valores menores ou maiores.

Esses parâmetros são ajustados para alinhar o treinamento do modelo com os objetivos específicos, permitindo uma interrupção oportuna do processo para evitar que o modelo se sobreajuste aos dados de treinamento.

4.4.2.4 Hiperparâmetros

O Temporal Fusion Transformer, assim como diversos outros modelos de machine learning, é sensível a alguns hiperparâmetros customizáveis, os quais afetam expressivamente sua desempenho. No caso de nosso modelo, são mais relevantes os seguintes:

- **gradient_clip_val**: Este hiperparâmetro está associado à técnica de "clipping"de gradientes, a qual é empregada para mitigar o problema de explosão de gradientes durante o processo de treinamento. Ao limitar o valor máximo do gradiente, assegura-se que os passos de atualização durante a otimização não se tornem demasiado grandes, garantindo assim uma convergência mais estável durante o treinamento.

- **hidden_size**: Corresponde à dimensão das camadas ocultas na arquitetura do Transformer. Esta dimensão está diretamente relacionada à capacidade do modelo de aprender padrões complexos nos dados, onde uma dimensão maior pode permitir a aprendizagem de padrões mais intrincados à custa de uma maior exigência computacional e possibilidade aumentada de overfitting.

- **dropout**: Este hiperparâmetro está ligado à taxa de dropout aplicada às camadas

do modelo, sendo uma estratégia de regularização amplamente utilizada para prevenir o overfitting durante o treinamento. A técnica de dropout envolve a desativação aleatória de uma fração dos neurônios durante o treinamento, forçando o modelo a aprender representações mais generalizadas dos dados.

- **hidden_continuous_size**: Define a dimensão do espaço em que as variáveis contínuas são mapeadas nas camadas internas do modelo. Esse mapeamento pode auxiliar na captura de relações mais complexas nas séries temporais contínuas que estão sendo modeladas.

- **attention_head_size**: Este hiperparâmetro designa o número de "cabeças"em cada camada de atenção multi-cabeça no modelo. A introdução de múltiplas cabeças de atenção permite que o modelo capture diferentes tipos de dependências nos dados de maneira paralela, potencialmente enriquecendo as representações aprendidas.

- **learning_rate**: Refere-se à taxa de aprendizado adotada pelo otimizador durante o treinamento. Uma taxa de aprendizado adequada é vital para a convergência eficaz durante o treinamento, onde uma taxa muito alta pode levar à instabilidade e uma muito baixa pode resultar em uma convergência muito lenta.

Os ajustes dos hiperparâmetros foram realizados através da utilização da biblioteca Optuna [20]. Definiu-se especificamente os valores de *"attention_heads"* em 4 e *"lstm_layers"* em 2. Um total de 50 testes foram conduzidos, selecionando-se o que apresentou o melhor valor médio na função de perda. As faixas consideradas para a otimização foram as seguintes:

Hiperparâmetro	**Faixa**	**Valor Final**
gradient_clip_val	[0.01; 5.00].	0.021007981936097743
hidden_size	[128; 512].	167
hidden_continuous_size	[128; 512]	155
dropout	[0.1; 0.5]	0.31205391907053703
learning_rate	[0.0001; 0.01]	0.0006025595860743578

Tabela 4: Hiperparâmetros selecionados

4.4.3 Treinamentos

Para cada ativo avaliado, foram realizados 25 treinamentos utilizando sementes diferentes, porém fixas. Em todos os casos, se mantiveram os mesmos hiperparâmetros, alterando apenas o ativo objetivado e as séries de entrada para que fosse usado apenas ativos da mesma classe do que está sendo previsto, da seguinte forma:

Setor	**Ativos Utilizados no Treinamento**
Equities	ES=F, YM=F, NQ=F
Commodities	GC=F, SI=F, ZC=F, CL=F, SB=F, CT=F
Moedas	EUR=X, JPY=X, GBP=X, BRL=X, MXN=X, CAD=X
Renda Fixa	ZF=F, ZT=F, ZB=F, ZN=F

Tabela 5: Ativos utilizados para cada setor

4.4.4 Cálculo da Desempenho

Após o treinamento do modelo TFT foi efetuada a análise utilizando um conjunto de ferramentas de visualização e métricas de desempenho para interpretar e comparar os resultados.

4.4.4.1 Análise de Desempenho

A análise de desempenho envolveu processar as previsões de retorno geradas pelo modelo para cada ativo financeiro e compará-las com os retornos reais. Para cada semente, os seguintes passos foram realizados:

1. A cada 10 dias, extrai-se uma previsão baseada nos dias anteriores
2. Para os dias em que não há previsão, copia-se a previsão anterior, de forma que, em termos de trading real, mantém-se a posição do dia anterior.
3. Foi realizada a conversão dos retornos logarítmicos em retornos simples para facilitar a análise.

4. Calculou-se a taxa de acerto (win rate), que indica a porcentagem de vezes que a direção da previsão estava alinhada com a direção do retorno real.

5. Os retornos foram ajustados com base na direção das previsões para calcular a série de retornos efetivos do modelo.

6. A razão de Sharpe para cada série de retornos foi calculada como uma métrica de desempenho, quantificando o retorno ajustado ao risco.

4.4.4.2 Seleção dos Melhores Modelos e Análise Agregada

Após a análise individual de cada semente, os modelos com as melhores razões de Sharpe foram identificados. Esses modelos foram então utilizados para uma análise agregada, visando combinar as previsões dos melhores modelos para obter um resultado coletivo.

1. Os retornos previstos pelos modelos selecionados foram agregados para formar uma previsão combinada.

2. Esta previsão combinada foi comparada novamente com os retornos reais, e uma nova série de retornos efetivos foi calculada.

3. A razão de Sharpe da série de retornos efetivos combinados foi calculada, fornecendo uma métrica de desempenho para a previsão agregada.

4. Ferramentas de visualização foram utilizadas para apresentar graficamente a desempenho agregada do modelo em comparação com os benchmarks do mercado.

4.4.5 Estratégia

Finalmente, a estratégia de Trend Following desenvolvida nesse projeto pode ser sintetizada pelo seguinte diagrama:

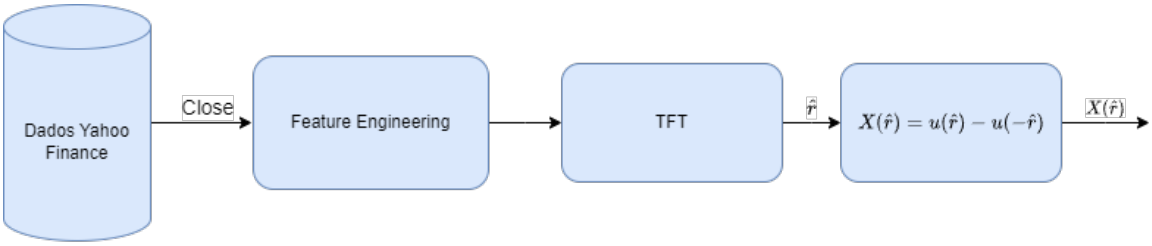

Figura 4: Diagrama da Estratégia baseada no TFT

Nele, os dados extraídos do Yahoo Finance passam por uma etapa de pré-processamento, na qual são gerados atributos para o modelo do TFT. O TFT então fornece uma previsão do retorno do ativo para os próximos dias. Se o retorno previsto for positivo, então compra-se o ativo e, caso contrário, for negativo, vende-se o ativo. Seja $X(\hat{r})$ a função que dimensiona a posição da estratégia a partir do retorno previsto, temos que:

$$X(\hat{r}) = \begin{cases} 1, \hat{r} \geq 0 \\ -1, \hat{r} < 0 \end{cases} \tag{4.11}$$

5 ANÁLISE DOS RESULTADOS

Para facilitar a análise dos resultados, vamos considerar quatro carteiras de ativos igualmente balanceadas, uma para cada setor. Assim compararemos o desempenho de cada estratégia em cada carteira. O retorno diário de uma carteira de ativos igualmente balanceada pode ser calculado da seguinte maneira:

$$r_t^c = \frac{1}{N} \sum_{i=1}^{N} r_t^i \tag{5.1}$$

Onde, r_t^i é o retorno do i-ésimo ativo no dia t, e r_t^c é o retorno da carteira no dia t. A partir da série de retornos diários da carteira, podemos calcular as métricas apresentadas na seção 2.3.1.

5.1 Equities

Figura 5: Retorno acumulado da Carteira de Equities

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,897	0,597	0,824	0,207	-0,333
MACD	0,367	0,451	0,628	**0,129**	**-0,202**
Moskowitz	0,013	0,089	0,115	0,149	-0,333
Bayes	0,481	0,433	0,578	0,201	-0,333
TFT	**1,204**	**0,820**	**1,160**	0,172	-0,273

Tabela 6: Métricas da Carteira de Equities

Conforme o gráfico e a tabela anterior mostram, a estratégia baseada no TFT superou todas as outras para carteira de Equities. Ela obteve o maior retorno, sem comprometer o risco. O que se evidencia pelas índices de Sharpe e de Sortino.

5.2 Moedas

Figura 6: Retorno Acumulado da Carteira de Moedas

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,170	0,440	0,645	0,059	-0,128
MACD	0,050	0,216	0,305	**0,038**	-0,066
Moskowitz	-0,021	-0,005	-0,007	0,075	-0,169
TFT	**0,467**	**1,366**	**2,051**	0,044	**-0,053**

Tabela 7: Métricas da Carteira Referente a Moedas

Na carteira de Moedas, a estratégia desenvolvida superou com facilidades as demais estratégias. Apesar do retorno total ser baixo, quando ajustamos esse retorno ao risco, os resultados obtidos são impressionantes.

5.3 Commodities

Figura 7: Retorno acumulado do Ouro futuro

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,592	0,579	0,828	0,147	**-0,209**
MACD	0,024	0,090	0,124	**0,109**	-0,219
Moskowitz	-0,139	-0,070	-0,096	0,160	-0,367
Bayes	**0,731**	**0,700**	**1,015**	0,149	-0,289
TFT	0,490	0,507	0,720	0,147	-0,305

Tabela 8: Métricas para o Ouro futuro

Devido a má qualidade dos dados, só foi possível construir estratégias para o contrato futuro do Ouro. Nesse caso, a estratégia vencedora foi a baseada na Rede Bayesiana. É possível notar, pelo gráfico, que o TFT não foi capaz de identificar a tendência de baixa no ano de 2022 e acumulou perdas nesse período.

5.4 Renda Fixa

Figura 8: Retorno Acumulado da Carteira de Renda Fixa

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	-0,153	-0,487	-0,683	0,051	-0,238
MACD	0,251	0,994	1,446	0,037	**-0,056**
Moskowitz	**0,403**	**1,157**	**1,704**	0,048	-0,069
TFT	0,084	0,405	0,595	**0,033**	-0,086

Tabela 9: Métrica da Carteira de Renda Fixa

Por fim, na carteira de renda fixa as estratégias clássicas superaram o TFT. Tal resultado pode ser justificado pelo ajuste de volatilidade incorporado nessas estratégias. Como a volatilidade da renda fixa é baixa, as estratégias clássicas assumem um posição mais elevada nos ativos. Como mostram as equações (2.2) e (2.6)

6 CONSIDERAÇÕES FINAIS

Neste capítulo iremos explicitar as conclusões obtidas e relatar as possibilidades de continuidade do projeto.

6.1 Conclusões do Projeto de Formatura

Este projeto de formatura propôs-se a estudar a viabilidade da implementação de uma estratégia de Trend Following baseada no Temporal Fusion Transformer. Para tanto comparou-se o desempenho da estratégia com outras estratégias da literatura.

A partir dos resultados, é possível concluir que o modelo é válido para as classes de ativos de Equities e Moedas, onde seu desempenho foi muito superior aos demais.

Por meio dos gráficos de importância das varáveis do TFT, disponíveis nos Anexos, é possível perceber que o modelo deu muita importância ao próprio preço de fechamento dos ativos para prevê-lo. Isso indica que possivelmente há espaço para acrescentar atributos que não são diretamente ligados ao preço de fechamento ao modelo.

Por fim, acreditamos que a solução baseada no TFT é promissora e apresentaremos a seguir possibilidades de aprimoramento, que podem estender a validade da estratégia para outras classes de ativos além de Equities e Moedas.

6.2 Perspectivas de Continuidade

O estudo realizado oferece um ponto de partida para várias investigações futuras na aplicação do Temporal Fusion Transformer (TFT) em séries temporais financeiras mas mostra claros pontos para desenvolvimentos futuros:

6.2.1 Refinamento do Modelo TFT

O aprimoramento do TFT pode ser uma área frutífera de pesquisa. Por exemplo, para permitir um dimensionamento mais refinado da posição o uso de funções de custo que maximizem métricas financeiras, como o Sharpe negativo, pode ser aplicado.

Além disto, é válido explorar a otimização de hiperparâmetros através de técnicas avançadas como *Bayesian Optimization* poderia potencialmente melhorar o desempenho do modelo [21]. A introdução de regularizações adaptativas, como *Dropout Variacional* [**?**] ou *L1/L2 Regularization* [22], pode ajudar a prevenir o sobreajuste e tornar o modelo mais robusto.

Adicionalmente, a integração de dados externos, como índices de volatilidade ou notícias de mercado, pode enriquecer a capacidade do modelo de capturar nuances do mercado financeiro.

6.2.2 Exploração de Outras Arquiteturas e Modelos

Comparar o TFT com modelos recentes como *Bidirectional Encoder Representations from Transformers (BERT)* [23] ou *Generative Pre-trained Transformer 3 (GPT-3)* [24] em tarefas financeiras pode fornecer insights valiosos sobre as forças e limitações dessas arquiteturas. Além disso, investigar modelos híbridos que combinam o TFT com técnicas de *Ensemble Learning* [25] ou *Deep Reinforcement Learning* [26] poderia abrir novas perspectivas na previsão financeira.

6.2.3 Estudos de Caso em Diferentes Mercados Financeiros

Aplicar o TFT a diferentes contextos de mercado, como mercados emergentes ou altamente voláteis, pode revelar sua adaptabilidade e eficácia sob diferentes condições econômicas. Por exemplo, a análise do comportamento do modelo em mercados de criptomoedas durante eventos de alta volatilidade pode fornecer insights sobre a robustez do modelo em ambientes extremos [27]. Estudos de caso específicos em ativos como *ETFs* (Exchange-Traded Funds) ou commodities também podem ser valiosos.

6.2.4 Aplicações Práticas e Implementação em Ambientes Reais

Uma área fundamental é a avaliação da aplicabilidade prática do TFT em ambientes de negociação ao vivo. Por exemplo, integrar o modelo em uma plataforma de negociação algorítmica e avaliar seu desempenho em tempo real durante diferentes ciclos de mercado [28]. Além disso, desenvolver interfaces de usuário amigáveis e sistemas de notificação para traders, baseados nas previsões do modelo, poderia ser uma forma de tornar a tecnologia acessível e prática para usuários finais.

7 ANEXOS

7.1 Resultados por Ativo

7.1.1 Equity

Figura 9: Resultados Mini S&P 500

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,715	0,529	0,727	0,201	-0,344
MACD	0,306	0,372	0,528	0,141	-0,243
Moskowitz	0,011	0,092	0,121	0,163	-0,363
Bayes	0,682	0,532	0,729	0,205	-0,344
TFT	0,747	0,544	0,752	0,201	-0,344

Tabela 10: Resultados para ES

Figura 10: Resultados Dow Jones

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,550	0,453	0,620	0,198	-0,373
MACD	0,124	0,201	0,284	0,145	-0,275
Moskowitz	-0,144	-0,070	-0,093	0,162	-0,494
Bayes	-0,213	-0,097	-0,123	0,203	-0,482
TFT	0,827	0,585	0,849	0,198	-0,226

Tabela 11: Resultados para YM

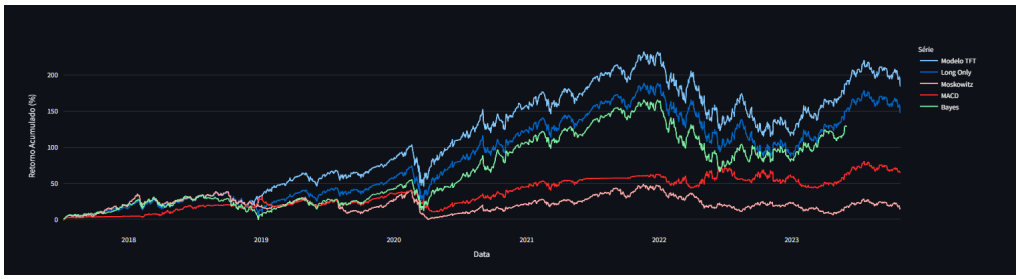

Figura 11: Resultados Nasdaq

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	1,484	0,715	1,003	0,245	-0,353
MACD	0,649	0,572	0,801	0,163	-0,208
Moskowitz	0,157	0,226	0,300	0,159	-0,288
Bayes	1,282	0,686	0,965	0,249	-0,375
TFT	1,858	0,807	1,138	0,245	-0,353

Tabela 12: Resultados para NQ

7.1.2 Currency

Figura 12: Resultados Euro

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,058	0,156	0,226	0,072	-0,146
MACD	0,222	0,619	0,911	0,052	-0,074
Moskowitz	0,122	0,192	0,277	0,154	-0,264
TFT	0,249	0,509	0,737	0,072	-0,167

Tabela 13: Resultados para EUR

Figura 13: Resultados Libra

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,035	0,103	0,148	0,092	-0,204
MACD	0,121	0,297	0,423	0,067	-0,100
Moskowitz	0,045	0,121	0,174	0,155	-0,359
TFT	0,191	0,339	0,473	0,092	-0,275

Tabela 14: Resultados para GBP

Figura 14: Resultados Yen

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,369	0,634	0,903	0,081	-0,147
MACD	0,080	0,257	0,362	0,051	-0,115
Moskowitz	0,679	0,594	0,875	0,154	-0,278
TFT	0,714	1,060	1,608	0,081	-0,105

Tabela 15: Resultados para JPY

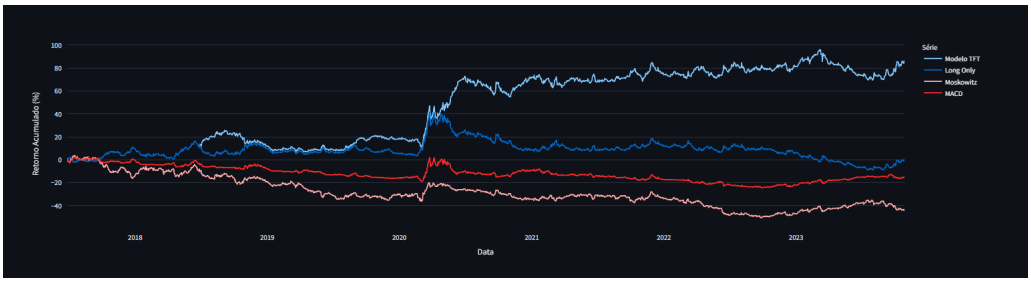

Figura 15: Resultados Peso Mexicano

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	-0,010	0,049	0,076	0,123	-0,356
MACD	-0,157	-0,243	-0,348	0,091	-0,261
Moskowitz	-0,432	-0,481	-0,666	0,155	-0,523
TFT	0,842	0,824	1,275	0,123	-0,149

Tabela 16: Resultados para MXN

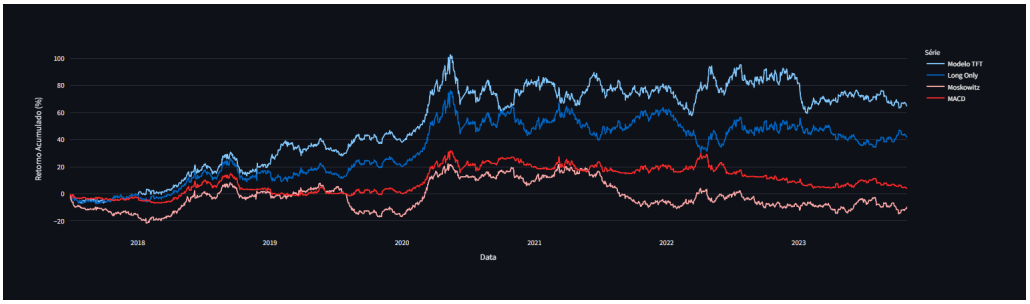

Figura 16: Resultados Real

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,419	0,403	0,586	0,169	-0,258
MACD	0,041	0,111	0,154	0,113	-0,210
Moskowitz	-0,096	-0,025	-0,035	0,152	-0,301
TFT	0,653	0,542	0,788	0,169	-0,221

Tabela 17: Resultados para BRL

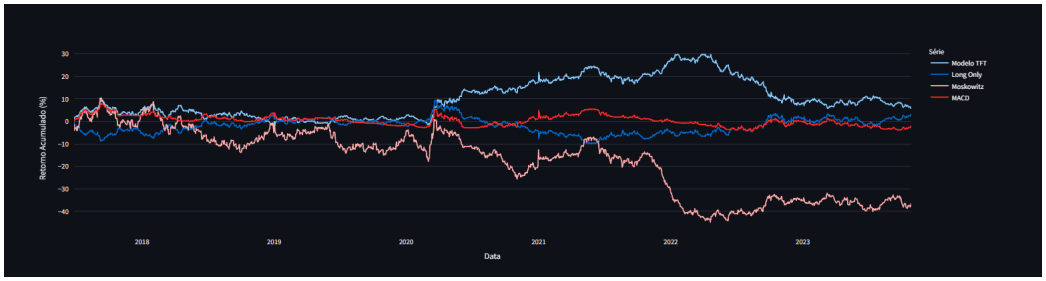

Figura 17: Resultados Dólar Canadense

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,032	0,104	0,148	0,071	-0,177
MACD	-0,020	-0,029	-0,041	0,055	-0,114
Moskowitz	-0,365	-0,372	-0,527	0,155	-0,500
TFT	0,056	0,153	0,224	0,071	-0,187

Tabela 18: Resultados para CAD

7.1.3 Commodities

Figura 18: Resultados Ouro

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	0,592	0,579	0,828	0,147	**-0,209**
MACD	0,024	0,090	0,124	**0,109**	-0,219
Moskowitz	-0,139	-0,070	-0,096	0,160	-0,367
Bayes	**0,731**	**0,700**	**1,015**	0,149	-0,289
TFT	0,490	0,507	0,720	0,147	-0,305

Tabela 19: Resultados para Ouro

7.1.4 Fixed Income

Figura 19: Resultados ZB

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	-0,286	-0,451	-0,626	0,106	-0,424
MACD	0,485	0,832	1,212	0,080	-0,123
Moskowitz	0,239	0,667	0,971	0,053	-0,065
TFT	-0,183	-0,249	-0,352	0,106	-0,412

Tabela 20: Resultados para ZB

Figura 20: Resultados ZT

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	-0,063	-0,636	-0,894	0,016	-0,086
MACD	0,075	1,094	1,603	0,011	-0,019
Moskowitz	0,613	1,410	2,174	0,055	-0,078
TFT	0,107	1,013	1,483	0,016	-0,042

Tabela 21: Resultados para ZT

Figura 21: Resultados ZF

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	-0,110	-0,470	-0,660	0,038	-0,177
MACD	0,160	0,952	1,372	0,025	-0,042
Moskowitz	0,426	1,082	1,581	0,054	-0,085
TFT	0,121	0,501	0,729	0,038	-0,078

Tabela 22: Resultados para ZF

Figura 22: Resultados ZN

Estratégia	Retorno Total	Sharpe	Sortino	Volatilidade Anual	Drawdown Máximo
Long Only	-0,149	-0,430	-0,609	0,056	-0,249
MACD	0,309	1,052	1,552	0,042	-0,056
Moskowitz	0,351	0,924	1,360	0,053	-0,082
TFT	0,310	0,795	1,154	0,056	-0,092

Tabela 23: Resultados para ZN

7.2 Interpretabilidade

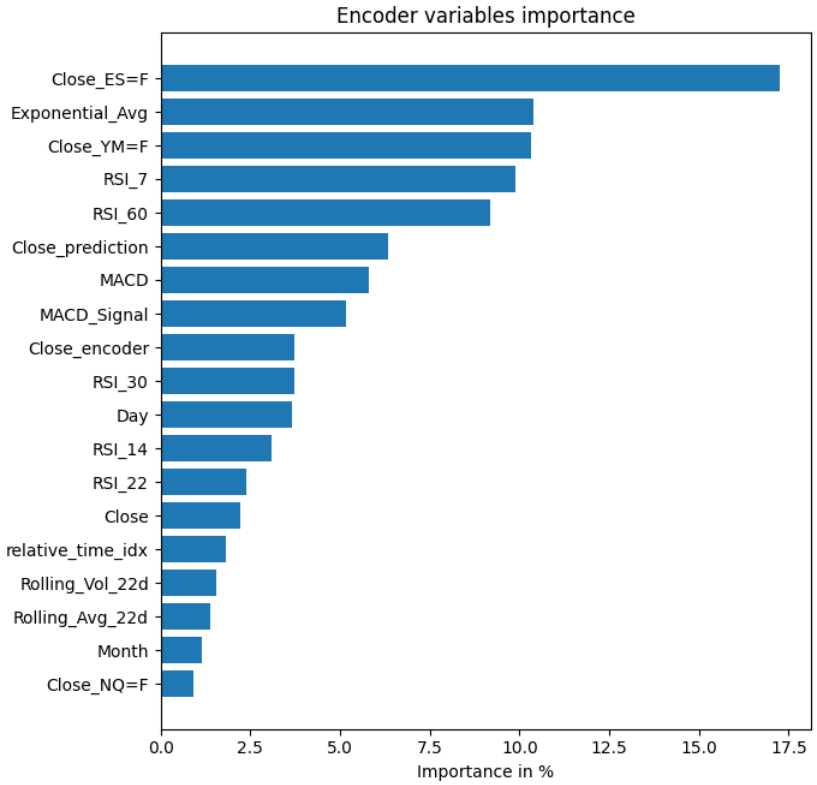

Figura 23: Importância de Variáveis para a Melhor Carteira de Equities

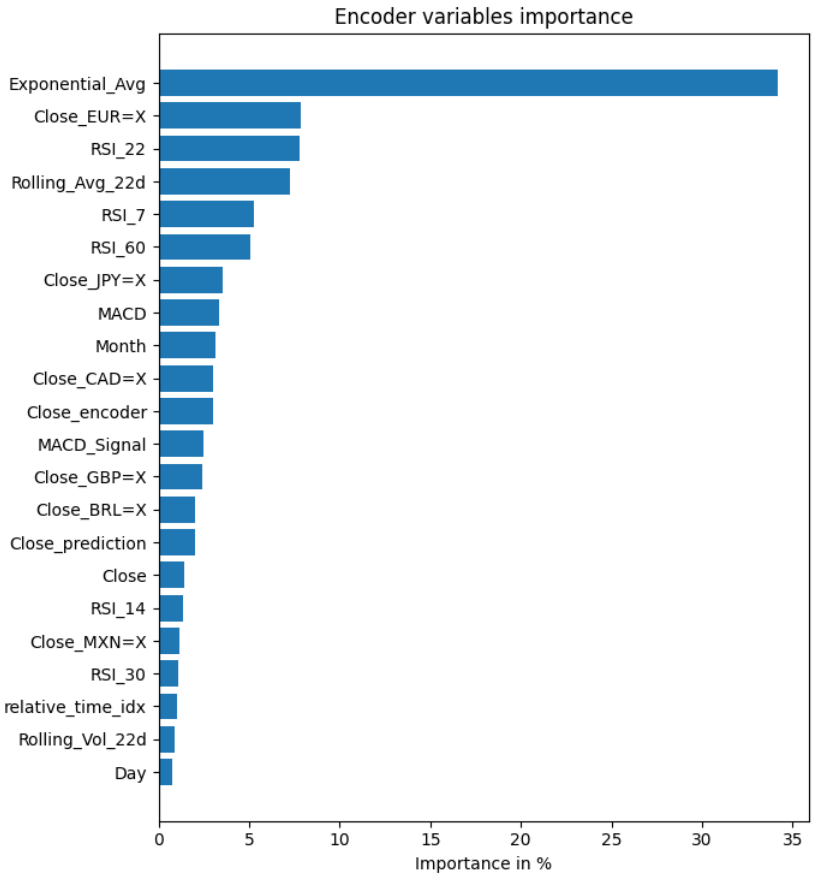

Figura 24: Importância de Variáveis para a Melhor Carteira de Moedas

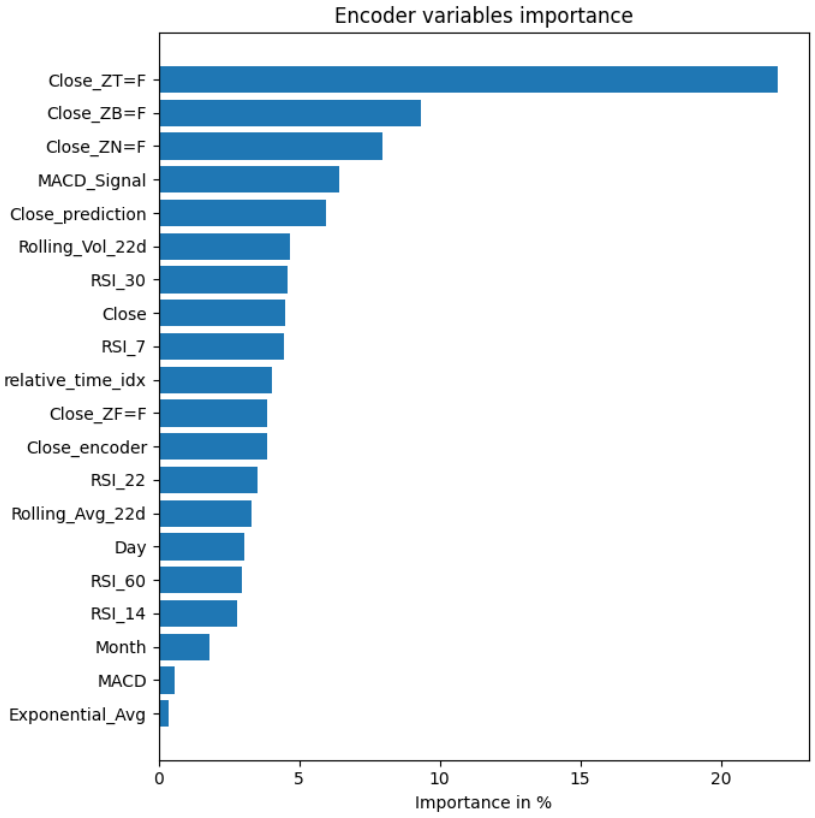

Figura 25: Importância de Variáveis para a Melhor Carteira de Renda Fixa

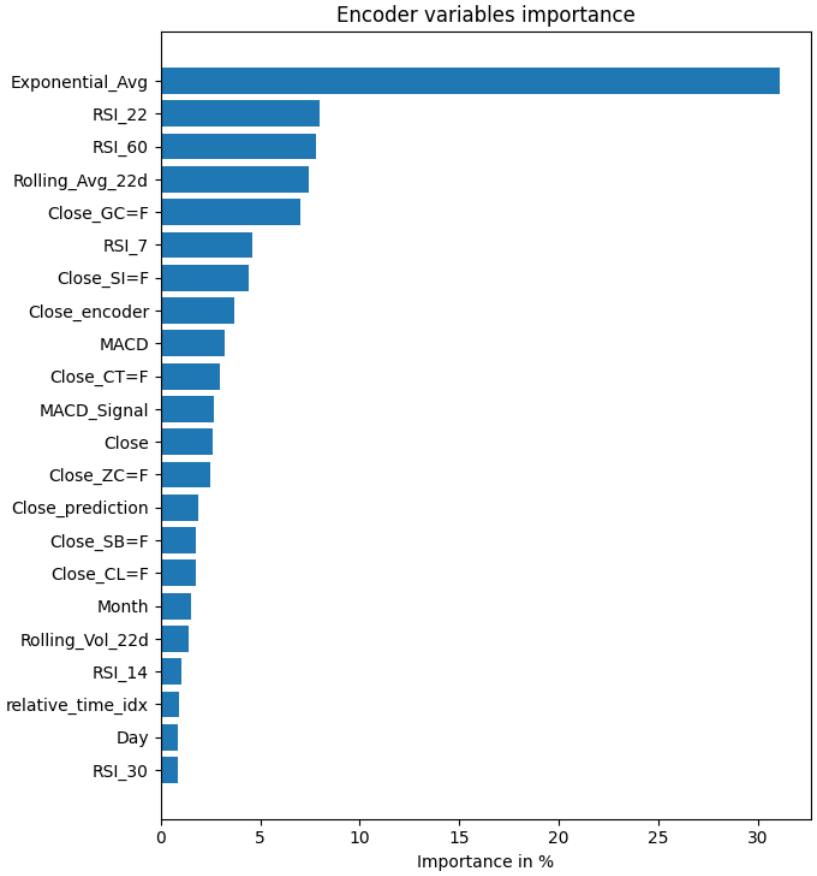

Figura 26: Importância de Variáveis as Melhores Previsões para Ouro

REFERÊNCIAS

1 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

2 ISICHENKO, M. *Quantitative portfolio management: The art and science of statistical arbitrage*. [S.l.]: John Wiley & Sons, 2021.

3 KATSUMI, F.; GOMI, E. S. A bayesian network model to improve stock market trend following strategies. In: SBC. *Anais do I Brazilian Workshop on Artificial Intelligence in Finance*. [S.l.], 2022. p. 81–92.

4 LEZMI, E.; XU, J. Time series forecasting with transformer models and application to asset management. *Available at SSRN 4375798*, 2023.

5 LIM, B. et al. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, Elsevier, v. 37, n. 4, p. 1748–1764, 2021.

6 WOOD, K. et al. Trading with the momentum transformer: An intelligent and interpretable architecture. *arXiv preprint arXiv:2112.08534*, 2021.

7 MCCAULEY, J. L. Arch and garch models vs. martingale volatility of finance market returns. *International Review of Financial Analysis*, v. 18, n. 4, p. 151–153, 2009. ISSN 1057-5219. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1057521909000362>.

8 HULL, J. C. *Options, Futures, and Other Derivatives, 9th edition*. [S.l.]: Prentice Hall, 2014.

9 MOSKOWITZ, T. J.; OOI, Y. H.; PEDERSEN, L. H. Time series momentum. *Journal of financial economics*, Elsevier, v. 104, n. 2, p. 228–250, 2012.

10 BAZ, J. et al. Dissecting investment strategies in the cross section and time series. *Available at SSRN 2695101*, 2015.

11 HARVEY, C. R.; LIU, Y. Backtesting. *The Journal of Portfolio Management*, Institutional Investor Journals Umbrella, v. 42, n. 1, p. 13–28, 2015.

12 MCKINNEY, W. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. 2nd. ed. [S.l.]: O'Reilly Media, Inc., 2017. ISBN 1491957662.

13 JIANG, W. Applications of deep learning in stock market prediction: recent progress. *Expert Systems with Applications*, Elsevier, v. 184, p. 115537, 2021.

14 VASWANI, A. et al. Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017.

15 YFINANCE. 2022. <https://pypi.org/project/yfinance/>. Acesso em 2023-11-21.

16 DONG, G.; LIU, H. *Feature engineering for machine learning and data analytics*. [S.l.]: CRC press, 2018.

17 DIXON, M. F.; HALPERIN, I.; BILOKON, P. *Machine learning in finance*. [S.l.]: Springer, 2020. v. 1170.

18 BROCKWELL, P. J.; DAVIS, R. A. *Introduction to time series and forecasting*. [S.l.]: Springer, 2002.

19 CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, Copernicus Publications Göttingen, Germany, v. 7, n. 3, p. 1247–1250, 2014.

20 AKIBA, T. et al. *Optuna: A Next-generation Hyperparameter Optimization Framework*. [S.l.]: GitHub, 2019. <https://github.com/optuna/optuna>. Acesso em: 21-11-2023.

21 SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. Practical bayesian optimization of machine learning algorithms. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 2951–2959.

22 NG, A. Y. Feature selection, l1 vs. l2 regularization, and rotational invariance. In: ACM. *Twenty-first international conference on Machine learning - ICML '04*. [S.l.], 2004.

23 DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

24 BROWN, T. B. et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

25 DIETTERICH, T. G. Ensemble methods in machine learning. p. 1–15, 2000.

26 MNIH, V. et al. Human-level control through deep reinforcement learning. *Nature*, Nature Publishing Group, v. 518, n. 7540, p. 529–533, 2015.

27 MAKAROV, I.; SCHOAR, A. Trading and arbitrage in cryptocurrency markets. *Journal of Financial Economics*, Elsevier, v. 135, n. 2, p. 293–319, 2020.

28 TRELEAVEN, P.; GALAS, M.; LALCHAND, V. Algorithmic trading review. *Communications of the ACM*, ACM New York, NY, USA, v. 56, n. 11, p. 76–85, 2013.