

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

**Estudo comparativo entre *transformers* para
predição de séries temporais financeiras**

Rodrigo de Almeida Sodré

Monografia - MBA em Inteligência Artificial e Big Data

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Rodrigo de Almeida Sodré

Estudo comparativo entre *transformers* para predição de séries temporais financeiras

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Antônio Rafael Sabino Parmezan

Versão original

São Carlos
2024

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

S679p	Sodré, Rodrigo de Almeida Estudo comparativo entre <i>transformers</i> para predição de séries temporais financeiras / Rodrigo de Almeida Sodré ; orientador Antonio Rafael Sabino Parmezan. – São Carlos, 2024. 84 p. : il. (algumas color.) ; 30 cm.
	Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2024. 1. Predição 2. Séries Temporais. 3. Finanças 4. Transformers. I. Parmezan, Antonio Rafael Sabino, orient. II. Título.

Rodrigo de Almeida Sodré

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Antônio Rafael Sabino
Parmezan

Original version

São Carlos
2024

Dedico esse trabalho à minha família, a quem devo tudo o que sou. Meus queridos pais que não mediram sacrifícios para que tivesse as melhores condições de me preparar para minha jornada. A minha filha, meu orgulho, a que um dia ainda farei entender que a única coisa que ela não é capaz é de me decepcionar. À minha amada Nazanin, que desejo com todo meu coração que se junte à minha família. À minha querida irmã Milene, que partiu durante essa caminhada, minha primeira melhor amiga.

AGRADECIMENTOS

Meus maiores agradecimentos à toda equipe do ICMC USP SC por ter preparado esse curso, nos capacitado e apoiado durante o desafio de adentrar nessa fascinante área da ciência e da computação.

Especialmente agradecido pelo meu orientador Antônio Parmezan pelos tempo dedicado a aperfeiçoar este trabalho e meus conhecimentos.

O mais profundo agradecimento por ter conhecido e ter sido apoiado pela Profa. Solange, carinhosamente chamada de "*tia Sô*", pelo tratamento humanizado que tanto precisamos além dos conhecimentos científicos que também nos propiciou.

*“O estudo, a busca da verdade e da beleza são domínios
em que nos é consentido sermos crianças por toda a vida.”*

Albert Einstein

RESUMO

SODRÉ, R. A. **Estudo comparativo entre *transformers* para predição de séries temporais financeiras**. 2024. 84 p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Desde sua proposta em 2019, os modelos de *Transformers* vem revolucionando as aplicações em várias tarefas, de Grandes Modelos de Linguagens à Inteligência Artificial Gerativa. No âmbito de séries temporais não foi diferente, desde o início variações do modelo original vêm sendo propostas para avaliar se tal tecnologia também poderia beneficiar tarefas de predição. Por outro lado, estudos recentes questionam a real praticidade desses modelos, em virtude de seu alto custo e qualidade de predição competitiva ou inferior a modelos mais simples. Este estudo visa avaliar os principais modelos conhecidos de *Transformers* adaptados para a predição de séries temporais, comparando-os com modelos clássicos e amplamente utilizados para esta mesma tarefa, especificamente no domínio de observações financeiras. Os resultados demonstram que mesmo para um conjunto de dados com poucas observações, *Transformers* conseguem obter resultados similares aos de modelos tradicionalmente utilizados para esta tarefa, porém a um maior custo computacional. O uso de conjuntos de dados de benchmark e o entendimento do comportamento de modelos bem estabelecidos permitirá a construção de um relevante elo de referência para comparação com outros modelos e estudos.

Palavras-chave: análise univariada, predição de séries temporais financeiras, *transformers*.

ABSTRACT

SODRÉ, R. A. **Comparative study between transformers for financial time series prediction.** 2024. 84 p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2024.

Since its proposal in 2019, *Transformers* models have been revolutionizing applications in various tasks, from Large Language Models to Generative Artificial Intelligence. In the context of time series it was no different, since the beginning variations of the original model have been proposed to assess whether such technology could also benefit prediction tasks. On the other hand, recent studies question the real applicability of these models, due to their high cost but inferior prediction quality compared to simpler and more efficient models. This study aims to evaluate the main known *Transformers* models adapted for the prediction of time series, comparing them with widely used classical models for this same task, specifically in the domain of financial observations. The results demonstrate that even for a dataset with few observations, *Transformers* can obtain results similar to those of models traditionally used for this task, but at a higher computational cost. The use of benchmark datasets and the understanding of the behavior of well-established models will allow the construction of a relevant reference link for comparison with other models and studies.

Keywords: univariate analysis, financial time series forecast, transformers.

Keywords: financial time series prediction transformer

LISTA DE FIGURAS

Figura 1 – <i>Pipeline</i> da metodologia	41
Figura 2 – Conjunto de dados M6 completo	42
Figura 3 – <i>MAE</i> , <i>RMSE</i> e <i>POCID</i> de todos os modelos. Os modelos de referência estão representados em azul escuro e os modelos <i>Transformers</i> avaliados foram destacados em azul claro.	56
Figura 4 – <i>MAE</i> , <i>RMSE</i> e <i>POCID</i> de todos os modelos exceto <i>FEDFormer</i> . Os modelos de referência estão representados em azul escuro e os modelos <i>Transformers</i> avaliados foram destacados em azul claro.	57
Figura 5 – Processamento máximo de GPU durante o treinamento	58
Figura 6 – Quantidade total de memória (do computador e GPU) consumida durante o treinamento	58
Figura 7 – Tempo de execução do treinamento de ajuste ao conjunto de dados . .	59
Figura 8 – Consumo total de energia da GPU durante o treinamento	59
Figura 9 – Predições com menores medidas de erro – <i>ARIMA</i>	60
Figura 10 – Predições com maiores medidas de erro – <i>ARIMA</i>	60
Figura 11 – Predições com menores medidas de erro – <i>LSTM</i> sem atualização . .	61
Figura 12 – Predições com maiores medidas de erro – <i>LSTM</i> sem atualização . .	61
Figura 13 – Predições com menores medidas de erro – <i>LSTM</i> com atualização . .	62
Figura 14 – Predições com maiores medidas de erro – <i>LSTM</i> com atualização . .	62
Figura 15 – Predições com menores medidas de erro – <i>MLP</i> sem atualização . . .	63
Figura 16 – Predições com maiores medidas de erro – <i>MLP</i> sem atualização . . .	63
Figura 17 – Predições com menores medidas de erro – <i>MLP</i> com atualização . . .	64
Figura 18 – Predições com maiores medidas de erro – <i>MLP</i> com atualização . . .	64
Figura 19 – Predições com menores medidas de erro – <i>DLinear</i> sem atualização . .	65
Figura 20 – Predições com maiores medidas de erro – <i>DLinear</i> sem atualização . .	65
Figura 21 – Predições com menores medidas de erro – <i>DLinear</i> com atualização . .	66
Figura 22 – Predições com maiores medidas de erro – <i>DLinear</i> com atualização . .	66
Figura 23 – Predições com menores medidas de erro – <i>NLinear</i> sem atualização . .	67
Figura 24 – Predições com maiores medidas de erro – <i>NLinear</i> sem atualização . .	67
Figura 25 – Predições com menores medidas de erro – <i>NLinear</i> com atualização . .	68
Figura 26 – Predições com maiores medidas de erro – <i>NLinear</i> com atualização . .	68
Figura 27 – Predições com menores medidas de erro – <i>Informer</i> sem atualização . .	69
Figura 28 – Predições com maiores medidas de erro – <i>Informer</i> sem atualização . .	69
Figura 29 – Predições com menores medidas de erro – <i>Informer</i> com atualização . .	70
Figura 30 – Predições com maiores medidas de erro – <i>Informer</i> com atualização . .	70
Figura 31 – Predições com menores medidas de erro – <i>Autoformer</i> sem atualização .	71

Figura 32 – Predições com maiores medidas de erro – <i>Autoformer</i> sem atualização .	72
Figura 33 – Predições com menores medidas de erro – <i>Autoformer</i> com atualização	73
Figura 34 – Predições com maiores medidas de erro – <i>Autoformer</i> com atualização	74
Figura 35 – Predições com menores medidas de erro – <i>FEDFormer</i> sem atualização	74
Figura 36 – Predições com maiores medidas de erro – <i>FEDFormer</i> sem atualização	75
Figura 37 – Predições com menores medidas de erro – <i>FEDFormer</i> com atualização	75
Figura 38 – Predições com maiores medidas de erro – <i>FEDFormer</i> com atualização	76

LISTA DE TABELAS

Tabela 1 – Parâmetros <i>AutoARIMA</i>	44
Tabela 2 – <i>MAE</i> , <i>RMSE</i> e <i>POCID</i> de todos os modelos. Os valores são as médias das medidas de avaliação e os desvios padrão estão entre parênteses. . .	52
Tabela 3 – Recursos consumidos pelos algoritmos dos modelos durante o treinamento	55
Tabela 4 – Relação de gráficos com menores e maiores valores de <i>MAE</i> , <i>RMSE</i> e <i>POCID</i>	77
Tabela 5 – Relação de gráficos com menores e maiores valores de <i>MAE</i> , <i>RMSE</i> e <i>POCID</i> (cont.)	78

LISTA DE ABREVIATURAS E SIGLAS

<i>AI</i>	<i>Artificial Intelligence</i>
<i>AM</i>	<i>Autoregressive Model</i>
<i>AT</i>	<i>Algorithmic Trading</i>
<i>ARIMA</i>	<i>Autoregressive Integrated Moving Average</i>
<i>CNN</i>	<i>Convolutional Neural Networks</i>
<i>DL</i>	<i>Deep Learning</i>
<i>EMA</i>	<i>Exponential Moving Average</i>
<i>FA</i>	<i>Fundamental Analysis</i>
<i>FTS</i>	<i>Financial Time Series</i>
<i>GAN</i>	<i>Generative Adversarial Network</i>
<i>GNN</i>	<i>Graph Neural Networks</i>
<i>GARCH</i>	<i>Generalized Autoregressive Conditional Heteroskedasticity</i>
<i>GRU</i>	<i>Gated Recurrent Unit</i>
<i>HFT</i>	<i>High Frequency Trading</i>
<i>LLM</i>	<i>Large Language Model</i>
<i>LRA</i>	<i>Learning to Rotate Attention</i>
<i>LSTM</i>	<i>Long Short Term Memory</i>
<i>MA</i>	<i>Moving Average</i>
<i>MAE</i>	<i>Mean Average Error</i>
<i>ML</i>	<i>Machine Learning</i>
<i>MLP</i>	<i>Multi Layer Perceptron</i>
<i>NLP</i>	<i>Natural Language Processing</i>
<i>OHLC</i>	<i>Open High Low Close</i>
<i>POCID</i>	<i>Prediction of Change In Direction</i>

<i>RMSE</i>	<i>Root Mean Squared Error</i>
<i>RNN</i>	<i>Recurrent Neural Network</i>
<i>SA</i>	<i>Sentiment Analysis</i>
<i>SAR</i>	<i>Seasoned Autoregressive</i>
<i>SARIMA</i>	<i>Seasoned Autoregressive Integrated Moving Average</i>
<i>SMA</i>	<i>Seasonal Moving Averages</i>
<i>SMA</i>	<i>Simple Moving Averages</i>
<i>TA</i>	<i>Technical Analysis</i>
<i>TS</i>	<i>Time Series</i>
<i>WMA</i>	<i>Weighted Moving Average</i>

SUMÁRIO

1	INTRODUÇÃO	25
2	FUNDAMENTAÇÃO TEÓRICA	29
2.1	Séries Temporais	29
2.1.1	Divisões das Séries Temporais	30
2.2	Métodos de Aprendizado de Máquina	31
2.2.1	Métodos de Aprendizado Profundo	31
2.2.2	Tarefas envolvendo Séries Temporais	32
2.2.3	O Processo de Predição de Séries Temporais	33
2.2.4	Séries Financeiras	34
2.3	Predições em Séries Temporais Financeiras Aplicadas à Operações de <i>Trading</i>	35
2.3.1	Conceitos em Investimentos Financeiros	35
2.3.2	Tipos de Análise Tradicionais	35
2.3.3	Conceitos de <i>Trading</i>	36
2.3.4	<i>Output</i> da Série Temporal para Operações de <i>Trading</i>	37
2.4	Estado da Arte em Séries Temporais Financeiras	38
3	METODOLOGIA	41
3.1	Pipeline da Proposta	41
3.2	Dados Utilizados	41
3.3	Tecnologias Utilizadas	43
3.4	Parâmetros	44
3.5	Procedimentos da Avaliação	47
3.6	Métricas de Avaliação Experimental	48
3.7	Configurações	49
4	AVALIAÇÃO EXPERIMENTAL	51
4.1	Desempenho Geral dos Modelos	51
4.2	Consumo de Recursos	52
4.3	Amostras de Gráficos de Predições	54
4.3.1	Tabela de Detalhamento das Predições Individuais	54
5	CONCLUSÕES	79
5.1	Limitações do Estudo	80
5.2	Trabalhos Futuros	80

1 INTRODUÇÃO

Os mercados de investimentos desempenham um papel essencial nas economias globais e nacionais. Por intermédio deles, investidores viabilizam o financiamento de projetos de empresas e governos, contribuindo para um ciclo positivo de geração de novas riquezas. Os poupadões, por sua vez, buscam retornos financeiros em troca de seus empréstimos. Nesse contexto, diversos fatores são considerados pelos investidores ao decidir quais investimentos atenderão melhor as suas necessidades de retorno. Dentre esses fatores, o retorno e o risco associados ao investimento são comumente os mais analisados.

Se a tolerância à perda for baixa, é aconselhável que o investidor opte por ativos de menor risco, o que implica em um período mais extenso para o aumento de seu patrimônio. Por outro lado, um investidor com maior tolerância à perda pode escolher ativos mais arriscados, proporcionando a possibilidade de ganhos significativos em um período mais curto.

Do lado do tomador do empréstimo, se uma empresa ou governo apresenta fundamentos financeiros sólidos, seu risco inerente é reduzido, tornando mais acessível a obtenção de recursos. Em contrapartida, é possível oferecer dividendos mais baixos. Em mercados mais voláteis, a potencial perda parcial ou mesmo total do capital investido precisa ser compensada por expectativas de retornos mais elevados para atrair os recursos necessários.

Empresas e governos representam apenas alguns dos ativos elegíveis para investimentos. De modo mais amplo, incluem-se: ações, títulos de dívida pública ou privada, *commodities*, moedas e criptomoedas. Tradicionalmente, diversas abordagens têm sido utilizadas para selecionar os melhores ativos dentro de um determinado risco. As mais reconhecidas abrangem as análises fundamentalista e técnica, respaldadas por avaliações contábeis e estatísticas, tanto em escala micro quanto macroeconômica.

A informatização dos sistemas de investimentos tornou ativos em nível global acessíveis a uma grande parcela da população. Segundo a *World Federation of Exchanges*, atualmente, há mais de U\$ 100 trilhões listados em pelo menos 20 bolsas ao redor do mundo (WFE, 2024). Hoje, um investidor no Brasil pode adquirir ações de uma empresa asiática com a mesma comodidade que uma empresa local. Não há dúvida de que essa nova realidade proporciona oportunidades únicas, porém, ao mesmo tempo, exige um esforço considerável para gerenciar e analisar o volume de dados disponíveis num espaço de tempo tão curto. Isso é crucial para selecionar as melhores opções em meio a cenários tão voláteis e dinâmicos.

Com a evolução do poder computacional e da Inteligência Artificial (*Artificial Intel-*

ligence – AI), em particular dos algoritmos de Aprendizado de Máquina (*Machine learning – ML*), a atenção de investidores se voltou para abordagens de identificação automática dos ativos mais apropriados. Algumas dessas abordagens permitem o acompanhamento quase em tempo real de todos os ativos globais, analisando até mesmo correlações entre ativos e mercados que anteriormente não eram passíveis de detecção pela revisão humana. Entre outros benefícios oferecidos, destacam-se estimativas mais precisas, o que maximiza a segurança e a rentabilidade. Além disso, há a possibilidade de realizar Negociação em Alta Frequência (*High Frequency Trading – HFT*), a qual envolve a análise e execução de negociações em questão de segundos (Yoshinaga; Castro, 2023).

Técnicas de *ML* aplicadas às previsões¹ financeiras têm despertado o interesse de investidores, impulsionando um aumento significativo nas pesquisas nessa área. Nas últimas duas décadas, observa-se um comportamento exponencial na quantidade de publicações, sendo que a maior parte das pesquisas na área ocorreu nos últimos cinco anos (Sezer; Gudelek; Ozbayoglu, 2020).

Em termos gerais, os algoritmos de predição de ativos podem ser discriminados em duas categorias principais: (i) métodos paramétricos, que compreendem modelos estatísticos lineares (*ARIMA*, *SARIMA*, *VAR*) e não lineares (*GARCH*); e (ii) métodos não-paramétricos, que incluem modelos de Aprendizado de Máquina (*SVM*, *Random Forest*, *k-NN*) e aqueles baseados em lógica *Fuzzy* (Patel; Jariwala; Chattopadhyay, 2023). Devido ao seu considerável desempenho no tratamento de grandes volumes de dados, com alta complexidade, dimensionalidade e dinamismo, além da capacidade de identificar relacionamentos ocultos nos dados, os métodos de *ML* aplicados a finanças quantitativas vêm se sobressaindo às abordagens tradicionais (Rundo *et al.*, 2019).

Por sua vez, os métodos mais recentes de *ML*, particularmente um subconjunto concebido em conformidade com o paradigma de Aprendizado Profundo (*Deep Learning – DL*), têm demonstrado alcançar maior desempenho preditivo em comparação com métodos econômétricos e de *ML* convencionais. Sua habilidade de extrair automaticamente relacionamentos complexos ocultos nos dados, os torna especialmente apropriados para a predição de dados temporais financeiros caóticos e altamente voláteis (Patel; Jariwala; Chattopadhyay, 2023).

Vários algoritmos de *DL* vêm sendo testados para predição financeira, seja de forma individual ou em associação homogênea ou heterogênea com outros modelos em contextos específicos. Devido à sua aplicação relativamente recente nessa área, embora algumas abordagens já apontem desempenho superior, ainda não há predominância ou consenso estabelecido. Dentre esses modelos, *Transformers* têm se mostrado eficiente ao apresentar erros menores em comparação com outros métodos (Patel; Jariwala; Chattopadhyay, 2023).

¹ Neste trabalho, os termos "predição" e "previsão" são usados indistintamente.

Outra técnica que tem sido empregada em conjunto com diferentes algoritmos de *DL*, elevando a qualidade da predição, é a Análise de Sentimento (*Sentiment Analysis – SA*). Mercados financeiros sofrem grande influência especulativa, mesmo quando números contábeis e econômicos são sólidos. A Economia Comportamental estuda os motivos que levam as pessoas a não seguir as previsões dos modelos econômicos (KENTON, 2023). As influências de efeitos de manada (HAYES, 2022) no mercado financeiro precisam ser identificadas para melhorar as decisões financeiras. As informações não apenas têm um papel como consequência, mas também atuam como causa e influência nos resultados da predição, fenômeno conhecido por “reflexividade” (ESTEVEZ, 2020). Muitas propostas recentes vêm obtendo desempenhos significativos utilizando técnicas diversas de mineração de texto para extrair, quantificar e aprender a influência dessas informações; entre elas, ressaltam-se: *embeddings*, como Word2Vec (Prosksy *et al.*, 2017) e (Liu *et al.*, 2018) e Stock2Vec – especialização de Word2Vec para o contexto de ações (Minh *et al.*, 2018) – e ferramentas de Processamento de Linguagem Natural (*Natural Language Processing – NLP*), como *BERT* (Sonkiya; Bajpai; Bansal, 2021) e *FinBERT – BERT* pré-treinado para o contexto financeiro (Araci, 2019).

Em 2023, os *Transformers* ganharam grande atenção no mercado em geral com a popularização de ferramentas de *NLP*, como o ChatGPT. A aplicação dessa mesma tecnologia para um domínio específico é bastante promissora no respeito à obtenção de resultados melhores para predição financeira. Alguns desafios associados a essa abordagem já são conhecidos. Em primeiro lugar, modelos de *DL* precisam de uma maior quantidade de dados para seu treinamento, e não há garantia de que os dados brutos históricos dos valores dos ativos sejam suficientes para superar o desempenho de abordagens tradicionais.

Outra questão em aberto diz respeito à abrangência dos testes executados. Os resultados obtidos até então foram baseados em dados de mercados mais estáveis, como o Americano ou o Asiático. Seria necessário testar os modelos em mercados mais voláteis de economias emergentes ou de menor capitalização para verificar sua eficiência. Por fim, pelo próprio caráter recente da tecnologia, variações de combinações de modelos de *DL*, como *Dynamic GNN*, ou impacto do sentimento a longo prazo, podem ser avaliados em termos de melhoria da acurácia (Patel; Jariwala; Chattopadhyay, 2023).

Este trabalho tem como objetivo explorar modelos de *Transformers* para a predição de séries temporais financeiras, a fim de validar o desempenho de suas predições em conjunto de dados reais. Para quantificar a eficiência dessa proposta, seu desempenho preditivo é confrontado com preditores amplamente difundidos, tais como *ARIMA* e *LSTM*. Os desempenhos dos modelos preditivos são analisados tanto em termos de *MAE* e *RMSE* (Patel; Jariwala; Chattopadhyay, 2023) quanto *POCID* (Parmezan; Souza; Batista, 2022).

Com o desenvolvimento de um modelo mais eficiente na identificação de oportunidades financeiras, seja de maior rentabilidade ou de menor risco, os investidores terão à

disposição recursos mais robustos para escolher investimentos mais alinhados ao seu perfil, possibilitando um aumento nos retornos ou a minimização de perdas. Ao mesmo tempo, a análise do custo-benefício de um modelo baseado em *Transformers* para predição de séries temporais financeiras serão importantes para a pesquisa em *AI*, contribuindo para evidenciar a promissora natureza do referido tópico de pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Séries Temporais

Séries Temporais (*Time Series – TS*) são coleções de observações feitas sequencialmente através do tempo. Uma série temporal S pode ser definida como um conjunto de valores ordenados S , tal que $S = (s_1, s_2, \dots, s_n)$ onde $s_t \in \mathbb{R}$, $\forall t \in [1, \dots, n]$ e s_t consiste em uma observação s no momento t .

Os principais componentes de uma TS são tendência, sazonalidade, ciclos e resíduo (Chatfield; Xing, 2019):

Tendência movimento do dado num sentido geral de aumentar, diminuir ou permanecer estável ao longo do tempo;

Sazonalidade padrões de variação que se repetem em intervalos regulares, como diário, quinzenal, trimestral ou bianual;

Ciclo variações causadas por uma influência externa durante um longo período, sem um tempo determinado como os padrões sazonais porém predizíveis de alguma forma;

Resíduo movimento dos dados de curto prazo, imprevisível e aleatório, não sendo clara a identificação de um padrão particular, seja esta uma tendência, sazonalidade ou ciclo

Podem haver quatro grandes objetivos ao se analisar TS (Chatfield; Xing, 2019):

Descrição tem como objetivo identificar os padrões da TS , os componentes mais característicos em sua composição (tendência, sazonalidade, ciclo e resíduos) que regem sua variação ou mesmo a presença, e devido tratamento, de *outliers*. Alguns modelos podem ser mais simples, por exemplo, explicados por apenas tendência e sazonalidade, enquanto outros mais complexos podem exigir técnicas mais sofisticadas como processos estocásticos;

Explicação procura entender o comportamento da variação de uma TS a partir de outra, pela identificação de relações de causa e efeito entre elas, levando a uma compreensão mais completa dos mecanismos que formam a TS . Modelos de Regressão Múltiplas ou Sistemas Lineares podem ser empregados nesta análise;

Predição consiste em tentar aproximar os valores das observações futuras da TS em questão a partir de um modelo proposto com base em descrições e explicações identificadas;

Controle nesta abordagem, a análise de *TS* tem como objetivo final manter determinadas condições de um processo ou meta específica. Por exemplo, assegurar a margem mínima de qualidade dos produtos em um processo de fabricação ou manter a meta inflacionária macroeconômica.

2.1.1 Divisões das Séries Temporais

Métodos de análise de *TS* se dividem em estatísticos e de *ML*. Métodos estatísticos são fórmulas, modelos ou técnicas matemáticas utilizadas na análise estatística de dados brutos de pesquisa. A aplicação de métodos estatísticos extrai informação de dados de investigação e proporciona diferentes formas de avaliar a robustez de resultados de investigação (Nature, 2024). Os principais modelos estatísticos utilizados em análise de *TS* são (Parmezan, 2016):

Médias Móveis (*Moving Average – MA*) realiza uma média dos valores das últimas n observações para predizer o próximo valor. Existem diferentes variações desse modelo. No de Médias Móveis Simples (*Simple Moving Averages – SMA*) o cálculo envolve média aritmética simples dos últimos n valores. Outras variações são Médias Móveis Ponderadas (*Weighted Moving Average – WMA*), no qual as observações mais recentes recebem pesos maiores, e Médias Móveis Exponenciais (*Exponential Moving Averages – EMA*) onde os pesos das observações mais recentes crescem exponencialmente. Consistem numa forma eficaz e simples de suavizam a *TS* de ruídos e revelam sua tendência;

Modelo Autoregressivo (*Autoregressive Model – AM*) analisa o relacionamento entre uma observação e outras anteriores da mesma variável, assumindo um relação de combinação linear entre o valor atual e seus valores passados, acrescido de algum ruído;

Autoregressivos Integrados de Médias Móveis (*Autoregressive Integrated Moving Average – ARIMA*) combina *MA* e *AM* com um processo de Integração, de diferenciação dos dados para atingir estacionariedade, ou seja, tornar suas propriedades estatísticas (como a média e a variância) imutáveis ao longo do tempo. Desta forma o *ARIMA* captura tendências ao mesmo tempo que reduz influência de erros aleatórios;

Autoregressivos Integrados de Médias Móveis Sazonal (*Seasoned Autoregressive Integrated Moving Average – SARIMA*) incorpora sazonalidade ao modelo *ARIMA* para *TS* com padrões recorrentes ao utilizar modelo Autoregressivo Sazonal (*Seasonal Autoregressive – SAR*) e Médias Móveis Sazonais (*Seasonal Moving Average – SMA*);

Heterocedasticidade Condicional Autoregressiva Generalizada (*Generalized Autoregressive Conditional Heteroskedasticity – GARCH*) modelo mais comumente

utilizado em *TS* financeiras. Ao contrário de modelos de *TS* tradicionais, pressupõe que a volatilidade muda com o tempo, em resultado da influência de eventos externos. Um componente *AR* é utilizado para capturar a persistência dos choques de volatilidade e o componente *MA* captura o impacto de choques passados na volatilidade presente. Modelos *GARCH* são eficientes para capturar agrupamento da volatilidade e a persistência observadas em *TS* financeiras.

Métodos estatísticos foram os primeiros a serem empregados em análises de *TS* e até hoje são utilizados produzindo resultados com desempenho e acurácia bastante satisfatórios. Por outro lado, é necessário assumir uma distribuição dos dados para se obter o melhor desempenho. Métodos de *ML* não exigem o conhecimento prévio da distribuição, podendo trabalhar tanto com a linearidade quanto a não-linearidade dos dados, além de poderem ser aplicados em cenários de tempo real. Esta flexibilidade faz *ML* ser mais atrativo para análise de *TS*.

2.2 Métodos de Aprendizado de Máquina

Inteligência é a habilidade de aprender e executar técnicas apropriadas para resolver problemas e atingir metas apropriadas ao contexto em um mundo incerto e variável. *AI* foi definida em 1955 pelo Prof. John McCarthy como “a ciência e engenharia de criar máquinas inteligentes”. *ML* foi definida pela primeira vez em 1969 pelo Prof. Arthur Samuel com “o campo de estudo que confere aos computadores a habilidade de aprender se serem explicitamente programados” e tem como base ciência da computação, estatística, psicologia, neurociência e teoria do controle. *DL* é o uso de grandes redes neurais artificiais que calculam utilizando representações contínuas, em uma organização similar aos neurônios biológicos em um cérebro humano. Atualmente *DL* é a abordagem mais bem sucedida de *ML*, com melhores generalizações e melhor ganho de escala a maiores quantidades de dados (Manning, 2020).

2.2.1 Métodos de Aprendizado Profundo

Modelos de *ML* tradicionalmente mais utilizados, assim como os que recentemente vêm obtendo melhores resultados em predição de séries temporais financeiras são (Patel; Jariwala; Chattopadhyay, 2023):

Redes Neurais Recorrentes (*Recurrent Neural Network – RNN*) diferentemente de outras redes neurais artificiais, levam em consideração contexto e transição de tempo, ideais para análise de séries temporais financeiras. As *RNN* mais utilizadas são:

Long Short Term Memory – LSTM um tipo de *RNN* onde a rede pode se lembrar tanto de valores de termo curto (recentes) quanto longos (mais antigos). Resolvem problemas comuns às demais *RNN* do aumento da complexidade à

medida que o período de tempo é mais longo e do desaparecimento ou explosão do gradiente descendente;

Gated Recurrent Unit – GRU variação de *LSTM*, mais simples e mais rápido de ser testado, porém com desempenho equivalente;

Redes Neurais Convolucionais (*Convolutional Neural Networks – CNN*) redes neurais baseadas em convoluções, utilizadas principalmente para tratamento de imagens. Uma variação, **Redes Convolucionais Temporais** (*Temporal Convolution Network - TCN*), foi projetada para lidar com *TS*, utilizando convolução causal e dilatado para capturar dependências temporais;

Redes Adversariais Generativas (*Generative Adversarial Network – GAN*) consiste no arranjo de dois modelos, um Gerador e um Discriminador. O Gerador tenta produzir informação mais realista possível. O Discriminador recebe dados reais junto com os produzidos pelo Gerador e tenta classificar qual é verdadeiro e qual é falso. À medida que o treinamento evolui, tanto o Gerador quanto o Discriminador aumentam respectivamente a qualidade tanto das informações produzidas quanto da classificação;

Redes Neurais em Grafo (*Graph Neural Networks - GNNs*) comprovadamente poderosas para aprender os relacionamentos complexos e interdependências entre os dados que, no caso do mercado financeiro, podem tanto ser ações entre empresas de um mesmo setor quanto de informações em redes sociais sobre investimentos que influenciam os mercados;

Processamento de Linguagem Natural (*Natural Language Processing – NLP*) área da *AI* que estuda a capacitação dos computadores a analisar e sintetizar a linguagem humana de forma significativa e relevante. Uma de suas tarefas, **Análise de Sentimentos** (*Sentiment Analysis – SA*) é essencial para uma melhor compreensão de *TS* financeiras, visto que rumores de notícias econômicas exercem forte influência no comportamento de seus componentes;

Transformers atualmente o estado da arte em *IA*, são construídas sobre o mecanismo de *Multi-head Self Attention*, que captura dependências de longo alcance presentes em dados de séries temporais e podem aprender padrões complexos não lineares. **Transformers** tem sido grande objeto de pesquisa recente em várias áreas de *AI*, inclusive em *TS*.

2.2.2 Tarefas envolvendo Séries Temporais

Dentre as tarefas que podem ser desempenhadas em *TS* temos detecção de anomalias, detecção de *motifs*, classificação e previsão (Fu, 2011):

Detecção de anomalias Em conjuntos de dados, observações que se desviam significativamente das demais, seja por uma falha na medida ou por ocorrência natural, são conhecidos por *outliers*. Em *TS*, *outliers* podem indicar anomalias significativas no comportamento de seus componentes. A detecção destas anomalias consiste no processo de descoberta de itens ou comportamentos não esperados ou comuns em um conjunto de dados (Musa; Bouras, 2022). A distinção de uma anomalia entre genuína ou erro é tão complexa quanto importante, uma vez que se genuína, é de interesse o estudo de suas ocorrências, enquanto que se causada por erros ou ruídos ela precisa ser removida para evitar conclusões incorretas;

Detecção de padrões morfológicos (*motifs*) tarefa não-trivial de descoberta e agrupamento de padrões de interesse, incluindo padrões frequentes e imprevistos (*surprising*) em *TS*;

Classificação categorização de uma base de dados em classes ou categorias pré-definidas. Desta forma, consiste em uma tarefa supervisionada, ou seja, cada observação é associada a um rótulo de classe ao qual pertence e o algoritmo irá aprender. O objetivo da classificação é a construção de um modelo que possa prever a classe de novas instâncias, com base em um conjunto de atributos;

Previsão tem como objetivo construir um modelo a partir de uma base de observações passadas e extrapolar para um horizonte à frente, para predizer valores futuros.

2.2.3 O Processo de Predição de Séries Temporais

O processo de construção de modelos para predição de *TS* é, em geral, definido por seis etapas (Parmezan; Batista, 2016):

Preparação dos dados inicia pelo pré-processamento das bases de dados, tais como normalização e tratamento de *outliers*, assim como outros tratamentos específicos para *TS*, como transformação de amostragem irregular para uma representação equidistante e remoção de resíduo. Após isso, a série é segmentada em sequência de treinamento, para construção e ajuste do modelo, e teste, para medir a adequação do modelo quanto à qualidade dos dados preditos;

Estruturação do modelo com base nas características dos dados de treinamento, analisadas por um algoritmo de entrada, o conjunto de parâmetros do modelo é estimado. Sucessivas iterações são executadas para minimizar o erro do desempenho do modelo e ajustar os parâmetros;

Geração do modelo com base na sequência de treinamento e nos parâmetros identificados na segunda etapa, o modelo é construído e ajustado;

Teste do modelo os parâmetros definidos e ajustados no passo anterior são validados com as sequências de teste, extrapolando o modelo para um período de predição h , o qual corresponde a um horizonte de predição constituído por um ou mais valores. Após a extração, o valor predito (passo aproximado) ou o real (passo atualizado) podem retroalimentar o modelo para o cálculo a previsão seguinte;

Avaliação mesmo utilizando ajustes semelhantes, modelos distintos podem produzir valores de predição diferentes. Dessa forma, é importante avaliar a acurácia, ou seja, o quanto próximos os valores preditos da sequência estão dos dados de teste, a fim de assegurar a qualidade do modelo;

Predição por fim, as previsões do modelo são realizadas para horizontes futuros da TS . A partir desse ponto, a base de observações é complementada com valores reais observados ao mesmo tempo que se monitora o erro, a fim de reajustar o modelo e manter sua acurácia.

2.2.4 Séries Financeiras

Séries Temporais Financeiras (*Financial Time Series – FTS*) são um subconjunto de TS , porém, geralmente não obedecem aos requisitos comuns de análise de TS (Taylor, 2008). As principais características que as diferem são (Arlt; Arltová, 2001) e (Sewell, 2011):

Alta frequência observações podem ser de informações diárias, minuto a minuto ou mesmo ordens de compra ou venda individuais (*ticks*), o que pode levar à grandes bases de dados, demandando técnicas específicas de análise;

Volatilidade dados financeiros são mais propensos a oscilação comparado a dados de outras TS . Também é característico de FTS o agrupamento de volatilidade, onde períodos de volatilidade tendem a ficarem próximos;

Não normalidade FTS não seguem distribuição normal em geral, exibindo assimetrias e curtos acentuadas. Modelos estatísticos que assumem normalidade podem não ser os mais adequados;

Não estacionariedade FTS são em sua maioria não estacionárias, ou seja, tem suas propriedades estatísticas, tais como média, variância ou autocorrelação, variando conforme o tempo, apresentando tendências ou sazonalidades;

Sazonalidade enquanto TS podem apresentar sazonalidade diária, semanal ou mensal, FTS podem exibi-la tanto em períodos curtos, repetidamente ao longo do dia, quanto em muito longos, tal como por meses ou anos;

Dependência de fatores externos indicadores macroeconômicos, notícias políticas, políticas bancárias e sentimento de mercado em geral, oriundos de fontes oficiais ou mesmo rumores, são alguns dos fatores externos que influenciam diretamente o comportamento de *FTS*. Identificar e considerar esses elementos é um fator importante para a predição de valores futuros;

Correlação serial o valor de um determinado ponto em uma *FTS* pode estar correlacionado a valores próximos no tempo.

O modelo *GARCH* é uma evolução do *ARIMA* que considera estes fatores específicos de *FTS* e em geral melhoram a predição para estas séries. O *GARCH* é o principal modelo estatístico utilizado em *FTS* (Fryzlewicz, 2007). Entretanto, ambos modelos seguem sendo bastante empregados. *ARIMA* possui menos parâmetros de configuração e ajuste e em mercados mais estáveis consegue obter resultados superiores ao *GARCH* (Miswan *et al.*, 2014).

2.3 Predições em Séries Temporais Financeiras Aplicadas à Operações de *Trading*

2.3.1 Conceitos em Investimentos Financeiros

Apesar do caráter interdisciplinar deste trabalho, o objetivo principal aqui é a avaliação do desempenho de modelos baseados na arquitetura de *Transformers* para predição de *FTS* em ambientes *HFT*. O estudo de *FTS* é feito há décadas utilizando análises tradicionais sofisticadas, contábeis e estatísticas, que vão além do escopo desta obra. Para esse contexto será apenas caracterizado quais são as abordagens tradicionalmente mais utilizadas, a análise proposta, o conceito da influência do sentimento do mercado, *trading* e suas operações básicas que serão preditas pelo modelo proposto.

2.3.2 Tipos de Análise Tradicionais

A Análise Fundamentalista (*Fundamental Analysis – FA*) é um dos métodos clássicos mais utilizados para determinar o valor intrínseco de um título, através da avaliação de fatores financeiros. O valor intrínseco de um investimento é baseado na situação financeira da empresa que o emitiu e condições econômicas e de mercado. São considerados todos os fatores que podem afetar o valor do título, sejam tanto macroeconômicos, como a situação da economia ou condições da indústria, quanto microeconômicos, tal como a eficiência da gestão da empresa. O objetivo final é estimar um valor adequado para o título e compará-lo com o valor atualmente negociado, se está sub ou sobrevalorizado por outros investidores (Segal, 2023a).

Enquanto a *FA* procura avaliar o valor de um título com base no desempenho do negócio como vendas e ganhos, a Análise Técnica (*Technical Analysis – TA*) é uma disciplina que analisa tendências estatísticas observada em padrões de atividades comerciais,

como movimentos de preço e volume de negociação. *TA* considera que o histórico de preços de um título pode ser um indicador importante do movimento futuro. O estudo de *FTS* é uma subárea da *TA* (Hayes, 2024).

Operações de negociação (*trading*) são transações financeiras que ocorrem em corretoras especializadas em processar ordens de compra e venda de diversos tipos de ativos negociáveis: ações, títulos, opções, *commodities*, derivativos, moedas, dentre outros. Negociação Algorítmica (*Algorithmic Trading – AT*) é um método automatizado de execução de ordens via instruções pré-programadas, com base em parâmetros como preço, volume ou horário (Lin, 2012). Operações de Negociação em Alta Frequência (*High Frequency Trading – HFT*) são caracterizadas por tecnologias de baixa latência de comunicação e alto volume de mensagens, sejam de ordens, cotações ou cancelamentos. A velocidade e complexidade necessárias para analisar o volume de informações em *HFT* ultrapassa a capacidade humana e, nestas situações, *AT* não apenas é mais apropriada mas a única alternativa que viabiliza a tomada de decisão (Aldridge, 2013).

2.3.3 Conceitos de *Trading*

Para compreender as saídas esperadas da análise de uma *FTS*, é necessário conhecer as possíveis operações de *trading*. Neste trabalho, é abordado apenas um subconjunto dessas operações, suficientes para fazer atividades básicas mas completas de investimento. As duas mensagens mais comuns são a ordem à mercado e ordem à limite. Uma ordem à mercado (*market order*) é um comando de compra ou venda de um ativo ao melhor preço disponível. Devido à velocidade das transações e das diversas ofertas de um ativo, o preço que a ordem será executada não é garantido. Por exemplo, um ativo está com um valor de R\$ 10 por ação num momento específico e é enviada ordem à mercado para compra de 1000 ações (*shares*). Dependendo das possibilidades de oferta, as 500 primeiras ações podem ser adquiridas a R\$ 10 e as 500 seguintes podem ser adquiridas a R\$ 11 (Securities; Commission, 2011).

Um investidor que acredita que determinado título irá valorizar, pode emitir uma ordem à mercado de compra. Neste caso, é dito que ele assumiu uma posição de compra ou longa (*long position*). Caso acredite que o título poderá desvalorizar, este investidor pode emitir uma ordem à mercado de venda, assumindo a chamada posição de venda ou curta (*short position*). É possível que o investidor venda um ativo que não possui através de operações de aluguel do ativo ou por opções (Kramer, 2024).

Outro recurso utilizado em *trading* é a alavancagem (*leveraging*). Segundo Securities e Commission (2021), estratégias de investimento alavancado são tentativas de aumentar os retornos através de: (1) empréstimo de dinheiro (margem), (2) uso de opções (*options*), (3) investimento em títulos alavancados. Negociação de margem (*margin trading*) usa dinheiro emprestado para adquirir ou vender títulos. Alavancagens de até 50 vezes o valor

investido são usualmente disponibilizadas a investidores. Esta operação eleva o risco pelo aumento inerente da possibilidade de perda. Negociação de opções (*option trading*) se referem a contratos de direito de compra ou venda futura de um ativo. Assim como a negociação de margem, a negociação permite investir recursos que o investidor não tem ao mesmo tempo que eleva o risco de perda total ou superior dos recursos que possui. Títulos alavancados (*leveraged securities*) não serão tratados no escopo deste trabalho.

Em um determinado momento numa mesma corretora, o valor que um título tem para ser comprado estará sempre acima do valor ofertado para ser vendido. No contexto de *trading* o *spread* é a diferença entre o valor de compra e o de venda de um título. Outra característica do *spread* é variar dinamicamente de acordo com alguns parâmetros como, liquidez ou a volatilidade do mercado desse título. É importante considerar o valor do *spread* de um título em determinado momento uma vez que a expectativa de retorno precisa ser no mínimo igual ao *spread*, caso contrário, a operação de *trading* acarretará em perda (Segal, 2023b).

O retorno (*return*) de um investimento é a diferença entre o valor de compra e o de venda. Se o valor de venda for superior, o retorno é positivo, caso o valor de venda for inferior ao de compra, o retorno é negativo. A maior parte dos estudos financeiros envolve retornos em vez de preços. O retorno é um resumo completo e livre de escala da oportunidade de investimento. Séries de retorno são também mais simples de se analisar do que de preços porque possuem propriedades estatísticas mais interessantes (Tsay, 2005).

Com respeito ao tempo em que o investidor pode permanecer com o ativo na posição adquirida até fechá-la (ou seja, reverter a operação realizando retorno positivo ou negativo), o acompanhamento pode exigir uma participação mais ou menos ativa do investidor. Na modalidade *buy-and-hold* o investidor espera que seu retorno venha a longo prazo podendo manter o ativo por vários meses ou anos, não mudando sua estratégia com base em variações de curto prazo. À medida que o prazo de expectativa de retorno diminui, passa a ser mais importante o acompanhamento e análise da *FTS* para que a decisão seja reavaliada. Dentre as modalidades mais conhecidas de *trading ativo*, um título pode ser adquirido por semanas ou meses (*swing trading*), apenas com relação ao valor de abertura e fechamento em um mesmo dia (*day trading*) ou mesmo em pequenas oscilações durante o dia (*intraday*) onde o tempo entre o início e o fim da transação pode levar segundos (Zucchi, 2023).

2.3.4 *Output* da Série Temporal para Operações de *Trading*

O tipo mais comum de predição em *FTS* é considerar o preço do ativo. Entretanto, existem outras variações possíveis. A previsão de preço pode ser utilizada para os seguintes elementos: ações, títulos, volatilidade e moedas. Quanto à predição, ela pode ser aplicada a índices e preços de moedas ou *commodities*. Em geral, os tipos de saída de uma *FTS*

podem ser divididos em dois grandes grupos: predição de preço e predição de tendência (*trend*). Previsão do preço é um problema de regressão, porém em termos de *FTS*, a predição do valor mais preciso possível do preço não é um fator tão importante quanto a correta identificação do sentido do movimento. Dessa maneira, a predição da tendência passa a ser um problema de classificação. É possível predizer o preço por dois movimentos, tendência de alta ou baixa, ou em três movimentos, incluindo o neutro (Sezer; Gudelek; Ozbayoglu, 2020).

Como resultado da análise do modelo de predição da *FTS* são esperadas, seja diretamente por uma tarefa de classificação, ou indiretamente através de uma regressão de pela análise dos valores previstos, as seguintes informações:

Sentido da tendência (1) Quando for identificada tendência de alta, para dar suporte à uma ordem de mercado de compra; (2) Quando for identificada tendência de baixa, para dar suporte à uma ordem de mercado de venda;

Magnitude da tendência No caso de análise de *FTS* múltiplas, comparar a expectativa de retorno de cada título, possibilitando uma realocação de recursos. Um objetivo importante com relação à magnitude é verificar se o retorno irá superar o *spread* previsto;

Intervalos fixos a *FTS* será usada no treinamento de um modelo dentro de um intervalo pré-determinado. As saídas geradas a partir desse modelo serão interpretadas como o retorno esperado de cada ativo ao final desse período – segundos, minutos ou horas para a modalidade de *intraday*; dia, para *day trading*; dias, semanas ou meses para *swing-trading*; meses ou anos para *buy-and-hold*;

Intervalos variados a saída do modelo gerado a partir da *FTS* incluirá a predição do período de realização do retorno;

Probabilidade da predição se concretizar ou não Além de informar a tendência, magnitude e tempo da predição, o modelo gerado a partir da *FTS* também poderá retornar a probabilidade de realização da predição. Esta informação é relevante para avaliar se o risco da operação é tolerado.

2.4 Estado da Arte em Séries Temporais Financeiras

Desde o crescimento do uso de *Transformers*, visto os resultados obtidos especialmente em Processamento de Linguagens Naturais (*Natural Language Processing – NLP*), a aplicação dessa tecnologia tem sido testada em outras áreas e tarefas de *AI*. O modelo original de *Transformers* proposto por (Vaswani *et al.*, 2017) foi adaptado em trabalhos recentes para a tarefa de predição de *TS*, os principais sendo:

LogTrans (Li *et al.*, 2019) utiliza mecanismo de auto-atenção convolucional para gerar chaves e consultas e procura reduzir a complexidade computacional de $O(N^2)$ para $O(N \log N)$ pela introdução de uma máscara esparsa de viés;

Informer (Zhou *et al.*, 2021) reduz a complexidade quadrática selecionando consultas dominantes baseada em similaridades de consultas e chaves. Também projeta um decodificador gerativo de estilos para produzir previsão de longo-prazo, evitando acumular e propagar erro;

Autoformer (Wu *et al.*, 2020) usa uma estrutura codificador-decodificador gerativa adversarial para treinar um modelo esparso de *Transformer* para previsão de *TS*, evitando acumulação do erro pela inferência um passo à frente;

Pyraformer (Liu *et al.*, 2021): constroi um módulo de atenção com base em uma hierarquia piramidal, com uma árvore binária para capturar dependências temporais de diferentes interlagos com tempo linear e complexidade de memória;

Quatformer (Chen *et al.*, 2022): baseada em quaternions, propõe um mecanismo de atenção aprender-para-rotacionar (*learning-to-rotate attention – LRA*) que introduz informações em fase e períodos de aprendizado para identificador padrões complexos de periodicidade;

FEDformer (Zhou *et al.*, 2022): utiliza transformada de Fourier para operação de atenção no domínio de frequência. Alcança complexidade linear pela seleção de um subconjunto de frequências de tamanho fixo;

TimesFM (Das *et al.*, 2023): um modelo *LLM* adaptado para *TS* proposto pelo Google-AI e pré-treinado num corpus de 100 bilhões de pontos reais. Seu desempenho *zero-shot*, sem treinamento em domínio específico, supera o de modelos estatísticos e de *ML* de *TS*.

Dentre os principais objetivos desta propostas estão a redução da complexidade quadrática do algoritmo do modelo de atenção original de *Transformers*, acúmulo e propagação do erro em previsões de longo-prazo e da inclusão de atributos relativos à ordem em *TS*. (Zeng *et al.*, 2023) faz uma análise destes modelos, contrastando com um modelo linear básico de redes neurais, chamado *LTSM-Linear*, proposto como referência. Através de uma série de experimentos, o trabalho constata que esse modelo supera com margem considerável o desempenho de todos os demais modelos por e conclui que não conseguem aprender os componentes das *TS* nem predizer adequadamente os valores futuros.

As estratégias de uso de *Transformers* que apresentaram resultados mais significativos, particularmente em *FTS*, foram as que consideraram sua aplicação para as tarefas

comprovadamente eficientes de *SA* e *NLP* e sua combinação com modelos mais consagrados de *TS* como *ARIMA*, *GARCH* e *LSTM*. Grandes Modelos de Linguagem (*Large Language Models – LLM*) também são eficientes ao orquestrar essas ferramentas, ajustando seus hiperparâmetros de forma a otimizar os resultados (Li *et al.*, 2023).

As bases de dados de *FTS* de *trading* são amplamente disponíveis. Histórico de dados financeiros são encontradas em grandes repositórios como Yahoo Finance (Yahoo, 2024) ou Google Finance (Google, 2024). Para as tarefas de *SA* são utilizadas informações públicas extraídas de sites de notícias financeiras e redes sociais.

Quanto a métricas de avaliação dos métodos, elas podem ser categorizadas com relação às tarefas. Para classificação, Acurácia e o Coeficiente de Determinação (*R-Squared* – R^2) são as mais utilizadas. Desempenho de regressão é medido pelo Erro Quadrático Médio (*Root Mean Squared Error – RMSE*) e Erro Médio Absoluto (*Mean Average Error – MAE*). Para fins de avaliação financeira, são igualmente importantes métricas específicas como o Índice de Sharpe (*Sharpe Ratio*), para estimar a relação entre o retorno excedente ao ativo livre de risco e sua volatilidade, assim como simulação de resultados em *backtest* (Xing; Cambria; Welsch, 2018).

Em muitas aplicações envolvendo *FTS*, a predição do preço correto do ativo não é considerado tão importante quanto identificar corretamente a direção do movimento. Dessa forma, pesquisadores consideram a predição da tendência, ou seja, prever em qual sentido o preço vai evoluir, consiste em um estudo mais crítico que comparado à predição do valor exato (Sezer; Gudelek; Ozbayoglu, 2020). Para medir o desempenho dos modelos com relação à tendência, a métrica de Predição da Mudança na Direção (*Prediction Of Change In Direction – POCID*) pode ser empregada (Parmezan; Souza; Batista, 2022).

Neste trabalho, é proposta a avaliação do desempenho de modelos baseados em *Transformers* para a predição de *FTS*. Seu desempenho será comparado com modelos estatísticos e de *ML* tradicionalmente utilizados para essas tarefas, como *ARIMA*, *LSTM* ou *MLP*, utilizando métricas padrão de *MAE*, *RMSE* e *POCID*. O intervalo de estudo abrange, a priori, de dois anos, considerando os quatro tipos de estratégias de trading: *intraday*, *day*, *swing* e *buy-and-hold*. Serão consideradas bases de ações, índices e preços, sem limitação geográfica do mercado. Para ampliar a possibilidade de comparação dos resultados desse estudo com de outros modelos, será utilizada a mesma base de dados da competição M6 de previsão de *FTS* (UNIC, 2024).

Dentre algumas limitações das propostas supracitadas, um número maior de validações precisam ser feitas para confirmar seu desempenho em *FTS* com diferentes ambientes, tal como em mercados mais voláteis ou de menor capitalização. Desta maneira, é esperado confirmar a adequação dessas tecnologias na previsão de *FTS*, explorando o impacto da influência de fontes textuais para elevar a acurácia da tarefa de predição de *FTS*.

3 METODOLOGIA

Neste capítulo é apresentada a metodologia de análise para avaliar a eficácia de modelos de *Transformers* para predição de *FTS* em comparação aos tradicionalmente utilizados.

3.1 Pipeline da Proposta

Na Figura 1 é ilustrado a *pipeline*, ou seja, a sequência de etapas da metodologia adotada neste trabalho.

3.2 Dados Utilizados

A primeira etapa consiste no levantamento do conjunto de dados utilizado para análise. Serão utilizados apenas dados reais de *FTS*. Como este estudo tem como um de seus propósitos a geração de dados para comparações de modelos para predição de *FTS*, além de modelos *baseline* tradicionais para serem utilizados como referência, também é interessante utilizar um mesmo conjunto escolhido em outros trabalhos. Desta forma, foi selecionado o conjunto de dados da competição M6 de previsão de *FTS* (UNIC, 2024).

Para extrair o conjunto de dados M6, é utilizado o *Yahoo Finance* (Yahoo, 2024), um repositório mundial centralizado de décadas de histórico financeiro de dezenas de milhares de ativos. Possui acesso livre e seus dados podem ser extraídos em tempo real por *webscraping* através da biblioteca Python yFinance (Aroussi, 2019). Algumas das principais limitações do yFinance são que os valores são disponibilizados apenas no formato *OHLC* (*Open High Low Close*) (Mitchell, 2024), o menor período é de 1 minuto por no máximo de 7 dias, limites de requisições diárias, falta de garantia de estabilidade e

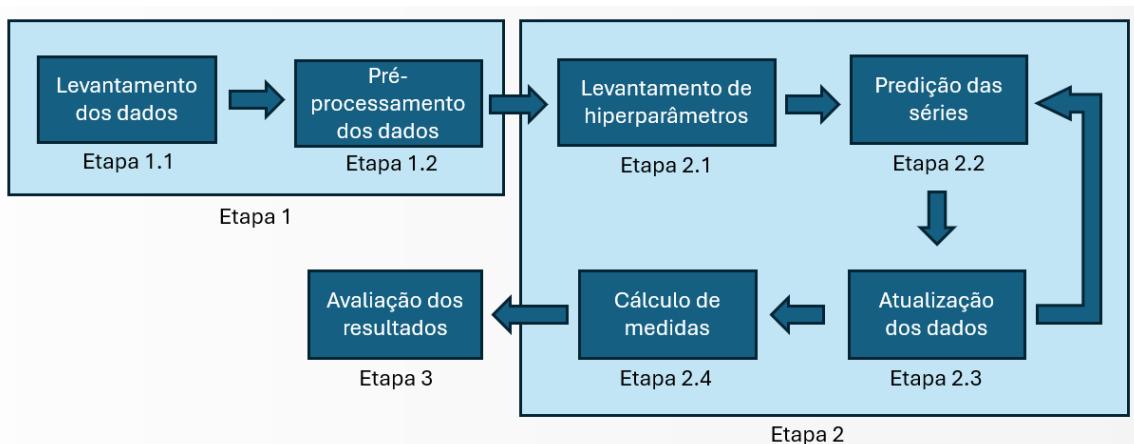


Figura 1 – *Pipeline* da metodologia

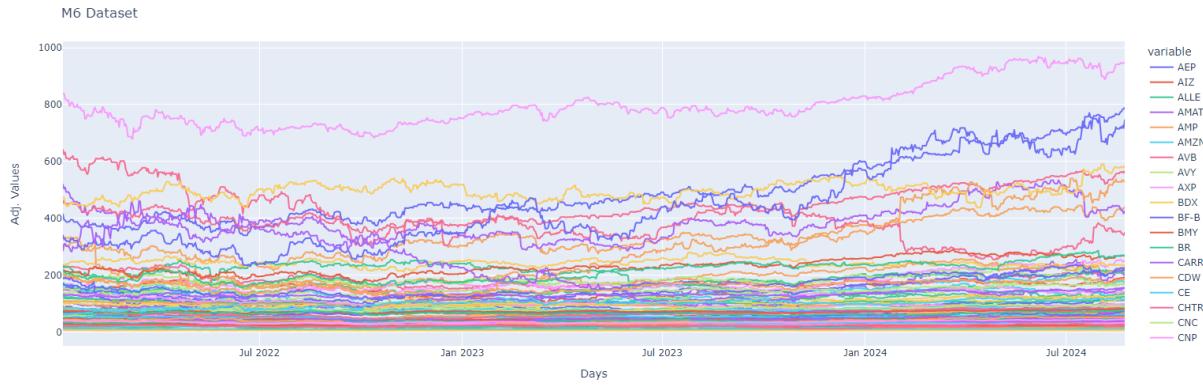


Figura 2 – Conjunto de dados M6 completo

possibilidade de quebras de funcionamento no caso de alteração do *layout* das páginas. Em geral, a biblioteca yFinance não é recomendada para ambientes de produção, porém é suficientemente adequado para fins de análise deste estudo. Um código Python baseado na biblioteca yFinance para baixar as séries da competição M6 está disponível no respectivo repositório Github¹.

O conjunto de dados M6 consiste de observações de 100 ativos, 50 ações e 50 ETF, cada um identificado pelo seu identificador padrão de mercado, conhecido por *ticker*. Dessa lista, não foi possível acessar os ativos “DRE” e “RE” que, segundo o retorno da biblioteca yFinance, possivelmente pela sua exclusão do repositório do *Yahoo Finance*, deixando o conjunto analisado com 98 ativos. Para cada um, foram recuperadas observações diárias de 01/01/2022 até 23/08/2024, quando a análise foi iniciada, totalizando 681 observações. Para cada ativo são recuperadas cinco colunas *OHLC*, correspondentes ao valor do ativo de abertura do dia (*Open*), maior valor do dia (*High*), menor valor do dia (*Low*), valor de fechamento (*Close*) e fechamento ajustado (*Adj Close*). Segundo o conjunto de dados M6 apenas a coluna *Adj Close* será considerada para cada ativo. Considerando a coluna adicional referente à data da observação e à de título contendo os *tickers*, identificadores dos ativos, o conjunto de dados M6 avaliado tem dimensão de 681 linhas por 99 colunas com valores contínuos. A Figura 2 exibe o conjunto de dados M6 completo utilizado neste estudo.

Um primeiro ajuste realizado no conjunto de dados é com relação a observações ausentes que ocorrem principalmente por dois motivos. Primeiramente, é possível que a biblioteca yFinance não retorne uma observação por falha de comunicação, *timeout* (tempo de resposta excedido) ou mesmo por limitação do *Yahoo Finance* imposta para agentes automatizados que recuperam dados em grandes volumes. Um segundo fator que causa valores omissos nas observações está relacionado aos dias de operação de negociações

¹ <https://github.com/Mcompetitions/M6-methods/blob/main/Download%20historical%20data%20using%20Yahoo%20API.py>

desses ativos, relativo a cada país aos quais pertencem. Desta maneira, não há observação em dias não comerciais, como finais de semana e feriados.

Como os algoritmos de construção dos modelos de predição de séries temporais exigem que o período de aquisição seja fixo, é necessário preencher tais valores omissos (Etapa 1.2). Neste estudo, é utilizada a abordagem de adicionar uma linha para cada dia faltante e preencher as observações com a última observação válida. Essa abordagem corresponde à prática de mercado para o preço dos ativos, pois quando não há negociação o valor permanece fixo até próxima abertura de sua negociação. Caso a primeira linha esteja vazia ela é eliminada, apesar disso não ter sido observado no conjunto de dados obtido. Após esse tratamento, o conjunto de dados passou a ter dimensão de 964 linhas por 99 colunas.

Uma última consideração sobre o o conjunto de dados M6 é com relação à data inicial de aquisição. A pandemia mundial da COVID causou forte impacto no preço dos ativos em todo mundo, iniciado ao final de 2019, com quedas de 23 a 25% em determinados mercados, cuja recuperação aos níveis normais pós pandêmicos ocorreu em 2021 (Uddin *et al.*, 2021). Esse fator exógeno poderia ter impactado na qualidade dos resultados da análise devido à inconstância de eventos dessa magnitude, entretanto, como o próprio conjunto de dados M6 determina o período inicial para 01/01/2022, parte-se da premissa que essa influência foi eliminada nos valores observados.

3.3 Tecnologias Utilizadas

A etapa 2 consiste na execução do procedimento para a predição dos valores futuros das *FTS*. Para os experimentos serão utilizados um modelo estatístico e sete de *ML/DL*, onde três são modelos específicos da arquitetura *Transformers*. Os modelos escolhidos são:

- Estatístico: *ARIMA/SARIMA*;
- *ML/DL*: *LSTM*, *MLP*, *DLinear* e *NLinear*;
- *Transformers*: *Autoformer*, *Informer*, *FEDFormer*.

Cada um desses modelos possui parâmetros específicos e possuem sua curva própria de aprendizado para ser otimizada durante o treinamento nos experimentos. Como a quantidade de modelos é elevada, faz-se necessária a escolha de implementações que possuam otimização automática dos hiperparâmetros de cada um durante o treinamento. O conjunto de bibliotecas NIXTLA, *statsforecast* e *neuralforecast*, possuem versões de otimização automática de hiperparâmetros para cada um destes modelos (Nixtlaverse...).

Parâmetro	Tipo	Valor padrão	Descrição
max_p	int	5	Valor máximo de p, autorregressivo
max_q	int	5	Valor máximo de q, média móvel
max_P	int	2	Valor máximo de P, autorregressivo sazonal
max_Q	int	2	Valor máximo de Q, média móvel sazonal
max_order	int	5	Valor máximo de p+q+P+Q, se stepwise = False
max_d	int	2	Valor máximo de diferenciações não sazonais
max_D	int	1	Valor máximo de diferenciações sazonais
start_p	int	2	Valor inicial de p, se stepwise = True
start_q	int	2	Valor inicial de q, se stepwise = True
start_P	int	1	Valor inicial de P, se stepwise = True
start_Q	int	1	Valor inicial de Q, se stepwise = True
ic	str	aicc	<i>Information criterion</i> , utilizado na seleção do modelo
stepwise	bool	True	Se True, faz seleção <i>stepwise</i> , em etapas (mais rápido)
nmodels	int	94	Número de modelos considerados na busca stepwise
seasonal_test	str	seas	Seleção dos métodos para diferenças sazonais
season_length	int	1	Número de observações por unidade de tempo

Tabela 1 – Parâmetros *AutoARIMA*

3.4 Parâmetros

A Tabela 1 exibe os parâmetros ativos da implementação *AutoARIMA* da biblioteca *statsforecast*², utilizados para a otimização automática dos hiperparâmetros do modelo:

Todos os demais modelos utilizados neste estudo baseados em redes neurais possuem implementação com versão de otimização automática dos hiperparâmetros na biblioteca *neuralforecast*³. As configurações padrão de seus parâmetros não estão documentadas porém são facilmente encontradas no código-fonte aberto da biblioteca⁴:

```
AutoLSTM.default_config = {
    "input_size_multiplier": [-1, 4, 16, 64],
    "inference_input_size_multiplier": [-1],
    "encoder_hidden_size": tune.choice([50, 100, 200, 300]),
    "encoder_n_layers": tune.randint(1, 4),
    "context_size": tune.choice([5, 10, 50]),
    "decoder_hidden_size": tune.choice([64, 128, 256, 512]),
    "learning_rate": tune.loguniform(1e-4, 1e-1),
    "max_steps": tune.choice([500, 1000]),
    "batch_size": tune.choice([16, 32]),
    "random_seed": tune.randint(1, 20),
```

² <https://nixtlaverse.nixtla.io/statsforecast/src/core/models.htmlautoarima-2>

³ <https://nixtlaverse.nixtla.io/neuralforecast/models.html>

⁴ <https://github.com/Nixtla/neuralforecast/blob/main/neuralforecast/auto.py>

```
}

AutoMLP.default_config = {
    "input_size_multiplier": [1, 2, 3, 4, 5],
    "hidden_size": tune.choice([256, 512, 1024]),
    "num_layers": tune.randint(2, 6),
    "learning_rate": tune.loguniform(1e-4, 1e-1),
    "scaler_type": tune.choice([None, "robust", "standard"]),
    "max_steps": tune.choice([500, 1000]),
    "batch_size": tune.choice([32, 64, 128, 256]),
    "windows_batch_size": tune.choice([128, 256, 512, 1024]),
    "random_seed": tune.randint(1, 20),
}

AutoDLinear.default_config = {
    "input_size_multiplier": [1, 2, 3, 4, 5],
    "moving_avg_window": tune.choice([11, 25, 51]),
    "learning_rate": tune.loguniform(1e-4, 1e-1),
    "scaler_type": tune.choice([None, "robust", "standard"]),
    "max_steps": tune.quniform(lower=500, upper=1500, q=100),
    "batch_size": tune.choice([32, 64, 128, 256]),
    "windows_batch_size": tune.choice([128, 256, 512, 1024]),
    "random_seed": tune.randint(lower=1, upper=20),
}

AutoNLinear.default_config = {
    "input_size_multiplier": [1, 2, 3, 4, 5],
    "learning_rate": tune.loguniform(1e-4, 1e-1),
    "scaler_type": tune.choice([None, "robust", "standard"]),
    "max_steps": tune.quniform(lower=500, upper=1500, q=100),
    "batch_size": tune.choice([32, 64, 128, 256]),
    "windows_batch_size": tune.choice([128, 256, 512, 1024]),
    "random_seed": tune.randint(lower=1, upper=20),
}

AutoInformer.default_config = {
    "input_size_multiplier": [1, 2, 3, 4, 5],
    "hidden_size": tune.choice([64, 128, 256]),
    "n_head": tune.choice([4, 8]),
```

```
    "learning_rate": tune.loguniform(1e-4, 1e-1),
    "scaler_type": tune.choice([None, "robust", "standard"]),
    "max_steps": tune.choice([500, 1000, 2000]),
    "batch_size": tune.choice([32, 64, 128, 256]),
    "windows_batch_size": tune.choice([128, 256, 512, 1024]),
    "random_seed": tune.randint(1, 20),
}

AutoAutoformer.default_config = {
    "input_size_multiplier": [1, 2, 3, 4, 5],
    "hidden_size": tune.choice([64, 128, 256]),
    "n_head": tune.choice([4, 8]),
    "learning_rate": tune.loguniform(1e-4, 1e-1),
    "scaler_type": tune.choice([None, "robust", "standard"]),
    "max_steps": tune.choice([500, 1000, 2000]),
    "batch_size": tune.choice([32, 64, 128, 256]),
    "windows_batch_size": tune.choice([128, 256, 512, 1024]),
    "random_seed": tune.randint(1, 20),
}

AutoFEDformer.default_config = {
    "input_size_multiplier": [1, 2, 3, 4, 5],
    "hidden_size": tune.choice([64, 128, 256]),
    "learning_rate": tune.loguniform(1e-4, 1e-1),
    "scaler_type": tune.choice([None, "robust", "standard"]),
    "max_steps": tune.choice([500, 1000, 2000]),
    "batch_size": tune.choice([32, 64, 128, 256]),
    "windows_batch_size": tune.choice([128, 256, 512, 1024]),
    "random_seed": tune.randint(1, 20),
}
```

Especificamente no caso do *AutoARIMA*, é solicitado o hiperparâmetro de sazonalidade. Para estimar esse valor, será utilizado um procedimento externo baseado na técnica do *scatter plot* (Parmezan; Batista, 2016). Essa abordagem pode melhorar os resultados do *ARIMA* com relação aos demais modelos e deve ser considerado no momento de conclusões.

Para todos os modelos será definido um horizonte de predição de sete observações, correspondente aos próximos sete dias de cada série.

3.5 Procedimentos da Avaliação

O procedimento padrão a ser adotado na etapa 2.2 para a geração dos modelos e predição dos valores é:

1. A sequência de treinamento é fornecida para o algoritmo;
2. O algoritmo ajusta o modelo aos dados fazendo o processo de otimização dos hiperparâmetros;
3. O algoritmo retorna o modelo ajustado;
4. O modelo extrapola eventos futuros com o horizonte solicitado;
5. Os valores preditos pelo modelo são comparados com uma sequência de teste.

Também será executada uma variação desse procedimento, consistindo na interação das etapas 2.2 com a etapa 2.3, onde o horizonte de predição será fixado em apenas um passo à frente e em cada passo o conjunto de dados será atualizado, isto é, o valor real será adicionado ao conjunto para uma nova iteração de predição do passo seguinte. Essa variação é chamada de predição com atualização e tem a vantagem de evitar a acumulação e propagação do erro em valores futuros:

1. A sequência de treinamento é fornecida para o algoritmo;
2. O algoritmo ajusta o modelo aos dados fazendo o processo de otimização dos hiperparâmetros;
3. O algoritmo retorna o modelo ajustado;
4. Repetir h vezes (onde h = horizonte de predição):
 - a) O modelo extrapola apenas 1 evento futuro;
 - b) O valor previsto é salvo em um dataframe separado de dados de predição;
 - c) O valor real é adicionado ao conjunto de dados;
 - d) O modelo é retreinado com os mesmos hiperparâmetros do passo 2, sem nova otimização.
5. Os valores preditos pelo modelo são comparados com uma sequência de teste.

Pelo fato do *ARIMA* ser um dos modelos mais tradicionais e amplamente estudados, iremos medir apenas a versão com atualização para este modelo. Os demais serão medidos para ambas as versões, com e sem atualização.

3.6 Métricas de Avaliação Experimental

A última etapa do processo de predição será o cálculo de medidas para avaliar o desempenho dos modelos no conjunto de dados M6. Neste estudo serão consideradas três métricas: *MAE*, *RMSE* e *POCID*. As medidas *MAE* e *RMSE* estão entre as mais utilizadas em pesquisas para avaliar o desempenho de modelos de predição de ST, podendo servir de referência de comparação com outros estudos.

O *MAE*, Erro Quadrático Médio (Equação 3.1), é utilizado para medir a magnitude da diferença entre os valores reais (y_t) e preditos (\hat{y}_t). Valores menores do *MAE* significam uma melhor acurácia na predição, significando que o modelo consegue predizer valores mais próximos dos reais.

$$MAE = \frac{1}{h} \sum_{t=1}^h |y_t - \hat{y}_t| \quad (3.1)$$

Onde:

- h número de observações;
- y_t o valor real da i -ésima observação;
- \hat{y}_t o valor predito para a i -ésima observação;
- $|y_t - \hat{y}_t|$ é o erro absoluto da i -ésima observação.

O *RMSE*, Raiz do Erro Quadrático Médio (Equação 3.2), mede a raiz quadrada da média dos quadrados das diferenças entre o valor previsto e o real. Similar ao *MAE*, sintetiza a distância entre os valores preditos dos reais porém, devido à elevação ao quadrado, penaliza os maiores erros. Também de forma similar ao *MAE*, valores menores do *RMSE* significam uma predição mais precisa do modelo.

$$RMSE = \sqrt{\frac{1}{h} \sum_{t=1}^h (y_t - \hat{y}_t)^2} \quad (3.2)$$

Onde:

- h número de observações;
- y_t o valor real da i -ésima observação;
- \hat{y}_t o valor predito para a i -ésima observação;
- $(y_t - \hat{y}_t)^2$ é o erro quadrático da i -ésima observação.

Quando se prevê um valor futuro de um ativo, esta informação irá basear uma tomada de decisão de compra ou venda. Se a magnitude da previsão não estiver correta, ou seja, se o valor predito é maior ou menor do que o valor real, isso significa um ganho maior ou menor. Porém se a tendência predita estiver incorreta, por exemplo, prever uma tendência de alta, mas ocorrer uma queda ou vice-versa, então haverá necessariamente perda no investimento (Sezer; Gudelek; Ozbayoglu, 2020).

A medida *POCID*, Predição da Mudança de Direção (Equação 3.3), apesar de não ser normalmente utilizada, faz-se relevante especificamente na avaliação do desempenho dos modelos no domínio financeiro. Seu objetivo é estimar a precisão das alterações da direção dos dados projetados, ou seja, se o valor futuro irá aumentar ou diminuir em relação ao valor atual (Parmezan, 2016).

$$POCID = \frac{\sum_{t=1}^h \mathbf{D}_t}{h} \times 100 \quad (3.3)$$

Onde:

- h número de observações;
- \mathbf{D}_t armazena o valor 1 se $(\hat{y}_t - \hat{y}_{t-1})(y_t - y_{t-1}) > 0$, e 0 caso contrário;

O índice *POCID* deve ser usado de forma complementar à análise da previsão de erros. Não é aconselhável tomar uma decisão com base apenas nesse valor.

3.7 Configurações

Para os experimentos, será utilizada a seguinte configuração de *hardware*:

- *CPU*: Intel Core i5-4590;
- Memória: 16GB RAM DDR3 1600 MHz;
- *GPU*: NVIDIA PNY GeForce RTX 4060 Ti 16GB RAM;
- Slot PCIe 16x gen 1.

Com relação ao *software*, será utilizada a seguinte configuração de ambiente:

- Sistema Operacional: Ubuntu 22.04.4 LTS, kernel: 5.15.0-119-generic;
- Driver *GPU*: Kernel: 5.15.0-119-generic proprietário desktop;
- Miniconda 3;

- Python 3.10.12.

Quanto às bibliotecas de suporte para a execução dos *notebooks* (ambiente interativo que permite usuários escrever e executar código *Python*) foram utilizadas as configurações especificadas nas respectivas páginas das bibliotecas Nixtla *statsforecast* e *neuralforecast*.

4 AVALIAÇÃO EXPERIMENTAL

Neste capítulo apresentamos os resultados dos testes executados para os modelos descritos na metodologia.

Durante os experimentos foi constatado que o modelo *TimesFM* não pode ser ajustado aos dados devido ao tamanho do conjunto de dados, de apenas 963 observações por série, causando falha de processamento no quinto *batch* de processamento devido ao tamanho diferente deste lote dos demais anteriores. Os conjuntos de dados de exemplo disponíveis junto ao repositório oficial do *TimesFM* tinham entre 17 mil e 63 mil observações. O menor conjunto de dados que foi possível testar com esse modelo possui pelo menos de 6000 observações. Por ser um modelo lançado há pouco tempo, ainda em caráter experimental segundo seus autores, pouca documentação e sem uma comunidade de suporte, foi decidido excluir este modelo do conjunto de teste visto as limitações de tempo para conclusão deste estudo.

Foram executados 15 experimentos, um com o *AutoARIMA* com atualização e sete com os demais modelos *ML* e *Transformers*, em duas versões, com e sem atualização. Em cada grupo de experimentos foi analisado dois aspectos dos modelos: (i) desempenho dos modelos de acordo com as métricas e (ii) consumo dos recursos (CPU, GPU e memória).

4.1 Desempenho Geral dos Modelos

Dos modelos analisados, apenas o *ARIMA* exige a informação do hiperparâmetro de sazonalidade. Foi utilizado o método *Scaterplot* para estimar o valor otimizado igual a 7, com base apenas no segundo elemento do conjunto de dados, o ativo ACN. Todos os demais hiperparâmetros utilizados foram os valores padrão da biblioteca.

Foi acrescentado o sufixo “.u” ao final do nome de cada modelo para diferenciar quando foi executado com a técnica de atualização do conjunto de treinamento.

Como os valores do modelo *FEDFormer* sem atualização são uma ordem de magnitude maior que dos outros modelos, a visualização das demais proporções ficou prejudicada. Para facilitar a visualização, foi gerado o gráfico da Figura 4 sem esse modelo específico.

A análise de todas as três métricas nos gráficos e Tabela 3, *MAE*, *RMSE* e *POCID*, permite levar às seguintes observações:

Desempenho superior da abordagem com atualização dos 15 modelos testados, 13 obtiveram melhores valores quando utilizada a técnica de atualização do conjunto de dados para predição do próximo passo. As principais exceções foram os modelos

Model	MAE	RMSE	POCID
ARIMA.u	0,84 (1,14)	1,19 (1,64)	0,64 (0,43)
LSTM	1,37 (1,98)	1,73 (2,42)	1,09 (0,90)
LSTM.u	2,10 (3,52)	3,26 (5,91)	1,28 (0,70)
MLP	1,55 (2,11)	2,04 (2,78)	1,09 (0,62)
MLP.u	0,84 (1,13)	1,18 (1,63)	0,63 (0,40)
DLINEAR	1,24 (1,88)	1,69 (2,51)	0,95 (0,66)
DLINEAR.u	1,26 (1,58)	1,90 (2,36)	0,92 (0,38)
NLINEAR	1,68 (2,27)	2,08 (2,85)	1,25 (0,89)
NLINEAR.u	0,84 (1,13)	1,19 (1,63)	0,63 (0,40)
INFORMER	1,49 (2,10)	1,85 (2,66)	1,09 (0,66)
INFORMER.u	0,85 (1,15)	1,20 (1,64)	0,65 (0,40)
AUTOFORMER	3,69 (5,08)	3,94 (5,36)	2,65 (2,77)
AUTOFORMER.u	0,86 (1,16)	1,20 (1,65)	0,66 (0,39)
FEDFORMER	39,66 (60,62)	41,03 (61,27)	29,84 (53,82)
FEDFORMER.u	0,85 (1,15)	1,20 (1,64)	0,65 (0,39)

Tabela 2 – *MAE*, *RMSE* e *POCID* de todos os modelos. Os valores são as médias das medidas de avaliação e os desvios padrão estão entre parênteses.

DLinear e *LSTM*, sendo o primeiro ligeiramente superior e o último com diferença mais significativa;

ARIMA, MLP e NLinear são os modelos com melhor desempenho, porém os três modelos baseados em *Transformers* têm desempenho praticamente similar na versão com atualização.

É importante destacar que o modelo *ARIMA* foi o único onde houve análise prévia para estimativa de um de seus hiperparâmetros, com a técnica do *scatterplot*. A estimativa do *Scatterplot* para o *ARIMA* considerou apenas uma das 98 séries, o que pode ter comprometido uma predição geral melhor com este modelo.

4.2 Consumo de Recursos

Tão importante quanto o desempenho nas métricas, é avaliar quanto de recurso cada algoritmo e modelo consomem, em especial na etapa de treinamento. Um algoritmo com mais exigência de memória, de poder de processamento e de energia exige maior custo e tempo para que a predição seja feita e, consequentemente, um número menor de predições pode ser realizado no prazo desejado.

Durante execução dos *notebooks* para cada modelo foi utilizado o script `usage.sh`, também disponibilizado no repositório [github](#), que coletou a cada segundo os valores de consumo de CPU, memória, consumo de GPU, memória dedicada de GPU e consumo de energia da GPU.

Cruzadas com os momentos de início e fim do treinamento, essas informações foram filtradas para o cálculo de quanto tempo cada modelo levou para ser treinado e quanto cada um consumiu de memória total (do computador e GPU), processamento máximo da GPU exigido durante os treinamentos e o consumo total de energia pela GPU. O processamento da CPU foi desconsiderado por ser um recurso comum, com consumo máximo constante e similar a todos os modelos. Foi feita apenas uma tomada de tempo para cada modelo, não tendo sido considerados os efeitos da preemptividade do sistema operacional e plataforma de execução.

A Figura 5 exibe o valor máximo de consumo da GPU medido durante o treinamento. Como esperado, o modelo *ARIMA* não apresentou consumo desse recurso por ser um modelo estatístico leve em comparação aos demais modelos, cuja implementação do algoritmo usa exclusivamente processamento em CPU. Dos modelos de *ML*, com consumo de GPU, os baseados em *MLP*, *DLinear* e *NLinear*, foram os que apresentaram menores valores. Com relação aos *Transformers* foi observado que a versão com atualização teve valores máximos de uso da GPU inferiores com relação à versão sem atualização dos mesmos modelos, sugerindo uma ociosidade do recurso. Não está claro o porquê desse comportamento, porém pode ter influenciado um maior tempo de execução nesses casos.

A quantidade total de memória consumida durante os treinamentos é apresentada na Figura 6. Como as implementações de bibliotecas e *drivers* utilizam tanto a memória do computador quanto a memória dedicada da GPU para armazenar os modelos, foi decidido apresentar a soma destes valores. O valor final foi subtraído do consumo antes do início do treinamento para desconsiderar a memória consumida pelo ambiente.

O modelo *ARIMA* apresentou o menor consumo, com 0.77 GB ao total, seguido novamente pelos modelos baseados em *MLP*, *NLinear* e *DLinear*, entre 1.76 e 1.86 GB. Os modelos baseados em *Transformers* tiveram o consumo ligeiramente superior, entre 2.51 e 3.8 GB, inferior ao modelo *LSTM*, que chegou a consumir 4.51 GB na versão com atualização. Para os padrões atuais de computação, 2 a 4GB de memória são requisitos de *hardware* relativamente baixos e o consumo dos modelos de *Transformers* nesse aspecto não deve ser um fator que impeça seu uso. A diferença de consumo de memória entre os modelos foi a menos acentuada com relação às demais medidas de recursos.

Uma observação se faz necessária com respeito à medida de memória; o ambiente Jupyter utilizado para a execução dos códigos apresentou um consumo elevado de memória, inicialmente em aproximadamente 500 MB, porém incremental. Foi também observado que ao final das execuções, a quantidade de memória livre era menor que a inicial, mesmo com o *kernel* sendo interrompido, o que pode ter comprometido a medida do consumo da memória. Foi feita subtração do valor consumido da memória antes do início da medição, procurando estimar o valor real demandado na geração do modelo, porém como o consumo do ambiente eleva progressivamente durante o treinamento, essa abordagem pode não ter

sido suficiente. Apesar dessas ressalvas, foi decidido manter essa medida no estudo como aproximação do valor de memória consumido de cada modelo.

Os tempos de treinamento dos modelos, apresentado na Figura 7 apresentou valores muito díspares. A quase totalidade dos modelos sem atualização ficaram no grupo dos executados mais rapidamente, com exceção do *ARIMA*. Como este foi executado apenas na versão com atualização, seu tempo de execução foi maior que dos modelos *MLP* e *LSTM* sem atualização. Tanto no grupo sem atualização quanto no com atualização, os modelos baseados em *Transformers* foram os que tiveram os maiores tempos registrados.

O consumo total de energia pelos modelos, Figura 8, é um valor diretamente proporcional ao consumo da GPU e ao tempo de processamento. Esse aspecto foi onde mais ficou evidenciada a desvantagem dos modelos baseados em *Transformers*, que tiveram os maiores valores em ambas as medidas. *ARIMA* e modelos baseados em *MLP* sem atualização ficaram com os melhores desempenhos. Num segundo grupo, os modelos *MLP* com atualização e *Transformers* sem atualização, ficaram com consumo uma ordem de magnitude acima. Por fim o grupo dos *Transformers* com atualização chegaram a consumir até três vezes mais energia que do grupo intermediário e trinta vezes mais que do grupo mais eficiente.

Duas observações esperadas foram confirmadas com as medidas de uso dos recursos. Como previsto para modelos de *DL*, *Transformers* exigem grande esforço computacional, com uma maior quantidade de memória e tempo para o treinamento. A segunda observação foi com relação à técnica de atualização do modelo. Como envolve um treinamento adicional para a predição de cada novo passo, o tempo e processamento se elevam em ordem diretamente proporcional, encarecendo o processo de predição.

A Tabela 3 compila todos os valores medidos com relação aos recursos computacionais consumidos para treinamento dos modelos.

4.3 Amostras de Gráficos de Predições

Para cada modelo foram gerados 98 gráficos, um para cada ativo, contendo os sete dias de horizonte previstos, resultando em 1.470 gráficos individuais de previsões. Desta maneira, serão exibidos apenas alguns gráficos de cada experimento, das duas predições com melhores medidas de erro (valores mais baixos) e das duas com os piores (valores mais altos) para cada métrica.

4.3.1 Tabela de Detalhamento das Predições Individuais

A Tabela 5 contém o detalhamento de quais gráficos obtiveram menor e maior pontuação nas respectivas métricas.

Modelo	CPU%	Mem (GB)	GPU%	GPU (GB)	GPU (wh)	Execução
ARIMA.u	91,90	0,77	0,00	0,00	0,75	05:06
LSTM	98,40	2,17	93,00	2,34	4,80	04:30
LSTM.u	75,00	2,47	90,00	1,16	31,26	30:46
MLP	98,40	1,61	19,00	0,25	1,05	02:25
MLP.u	98,40	2,40	18,00	0,40	7,95	17:43
DLINEAR	98,40	1,60	8,00	0,24	1,06	02:36
DLINEAR.u	98,40	2,10	6,00	0,36	7,87	18:56
NLINEAR	98,40	1,55	6,00	0,21	0,93	02:25
NLINEAR.u	87,10	2,17	6,00	0,36	7,23	17:20
INFORMER	98,30	1,95	61,00	0,56	8,30	09:25
INFORMER.u	98,40	3,34	35,00	0,46	29,32	54:25
AUTOFORMER	98,40	1,88	80,00	1,35	11,53	12:00
AUTOFORMER.u	98,40	2,80	28,00	0,60	17,19	33:17
FEDFORMER	91,90	1,88	89,00	1,07	11,47	11:56
FEDFORMER.u	98,40	2,72	46,00	0,72	25,89	43:02

Tabela 3 – Recursos consumidos pelos algoritmos dos modelos durante o treinamento

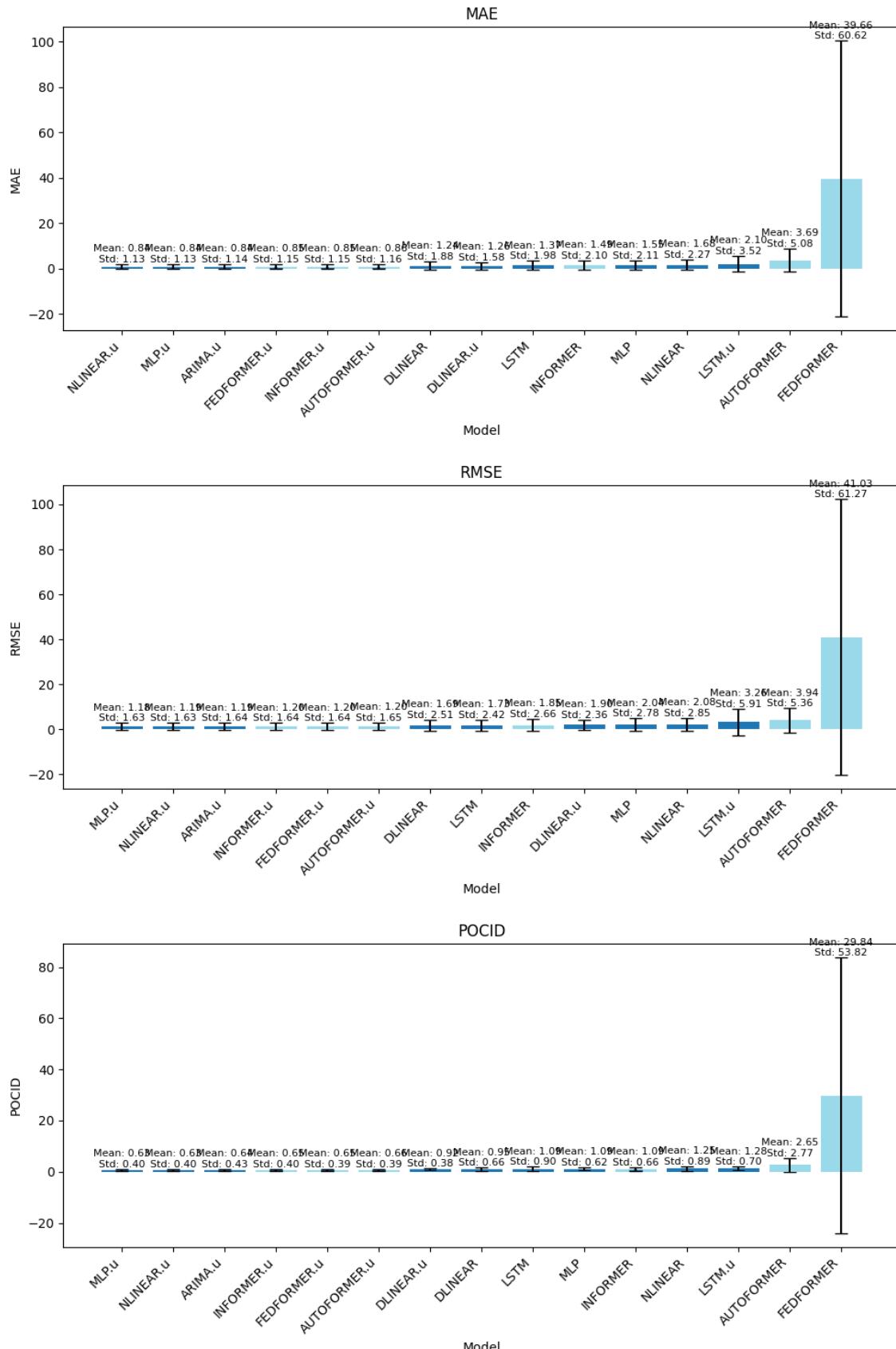


Figura 3 – *MAE*, *RMSE* e *POCID* de todos os modelos. Os modelos de referência estão representados em azul escuro e os modelos *Transformers* avaliados foram destacados em azul claro.

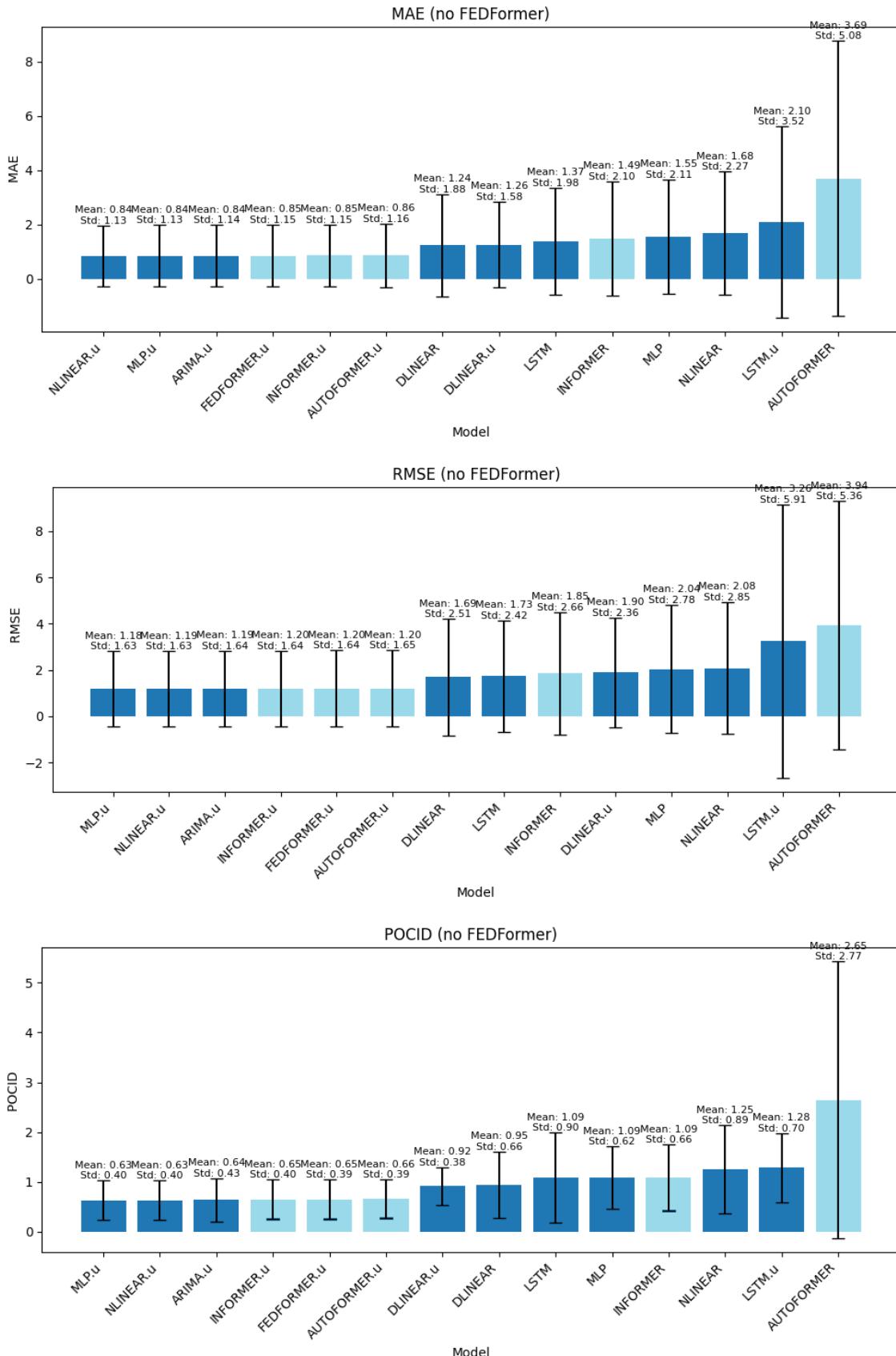


Figura 4 – *MAE*, *RMSE* e *POCID* de todos os modelos exceto *FEDFormer*. Os modelos de referência estão representados em azul escuro e os modelos *Transformers* avaliados foram destacados em azul claro.

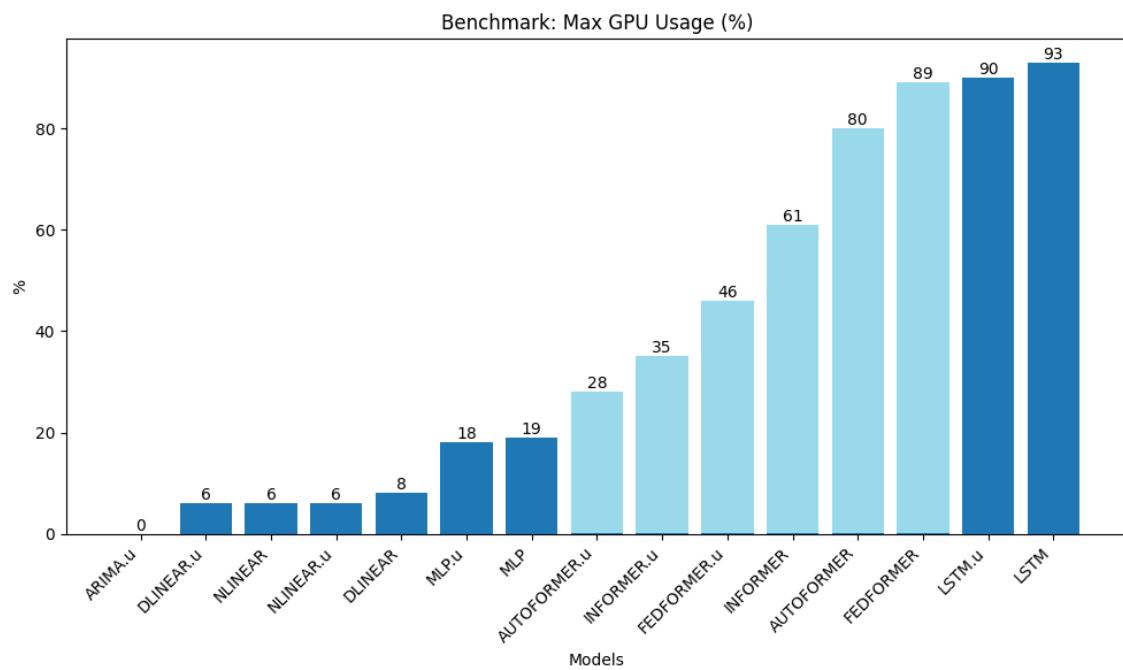


Figura 5 – Processamento máximo de GPU durante o treinamento

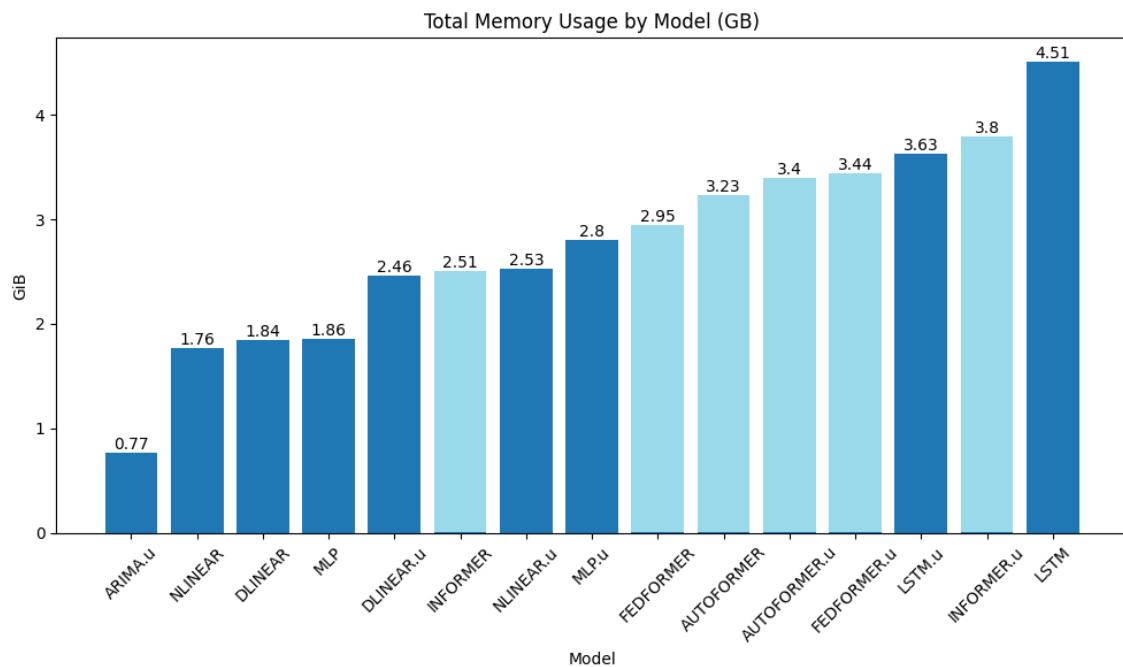


Figura 6 – Quantidade total de memória (do computador e GPU) consumida durante o treinamento

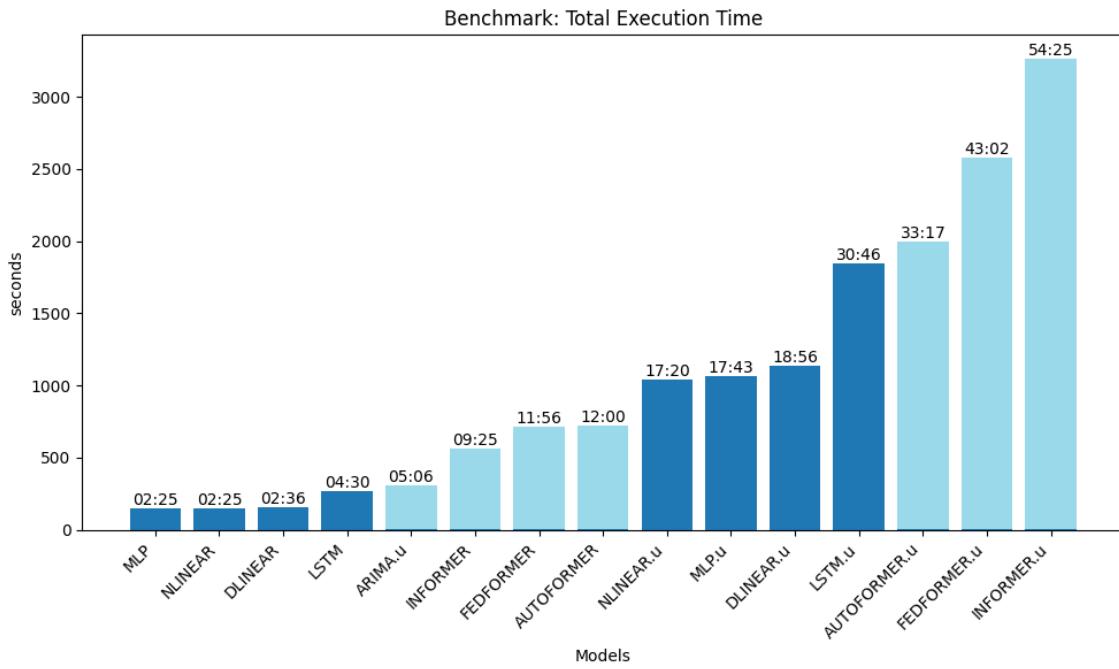


Figura 7 – Tempo de execução do treinamento de ajuste ao conjunto de dados

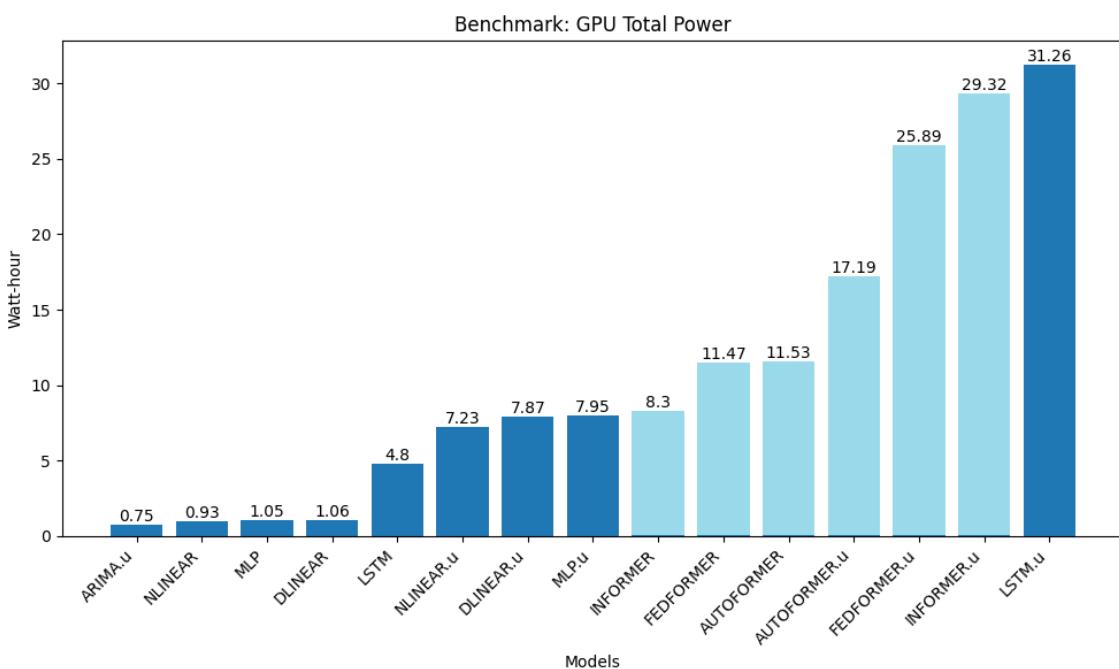


Figura 8 – Consumo total de energia da GPU durante o treinamento

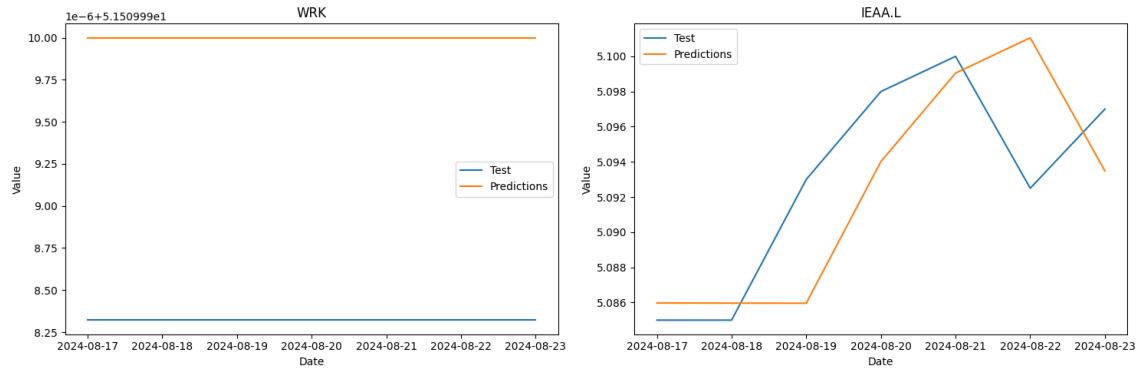


Figura 9 – Predições com menores medidas de erro – *ARIMA*

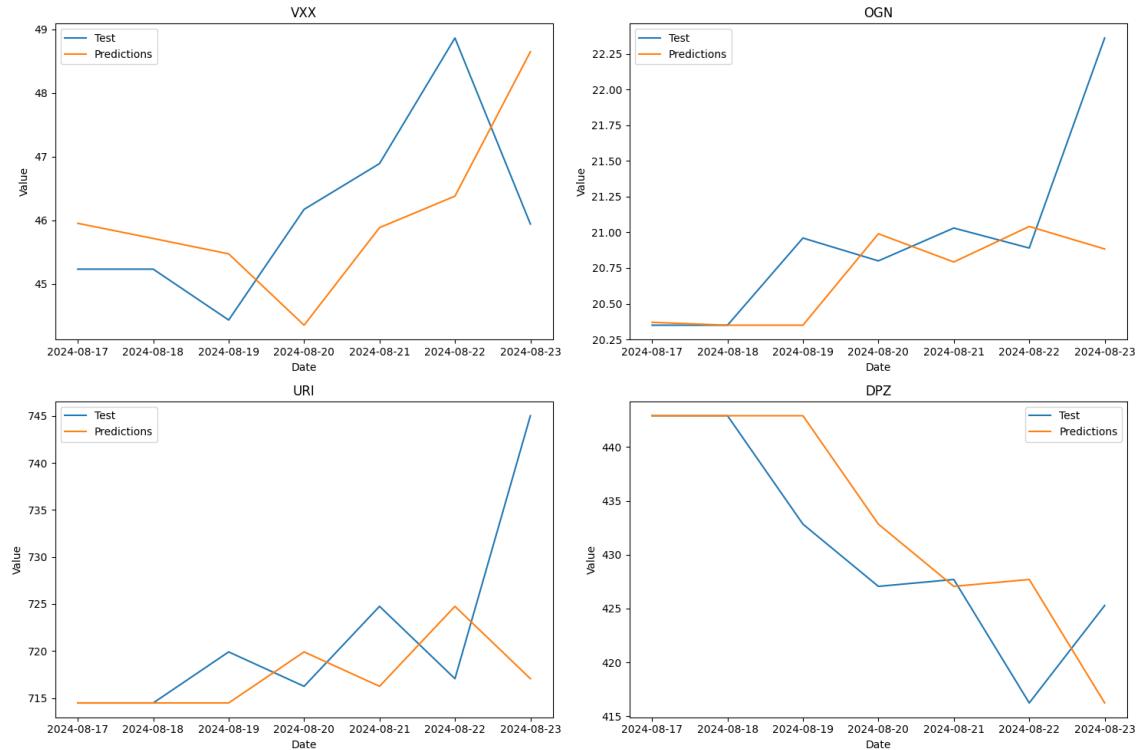


Figura 10 – Predições com maiores medidas de erro – *ARIMA*

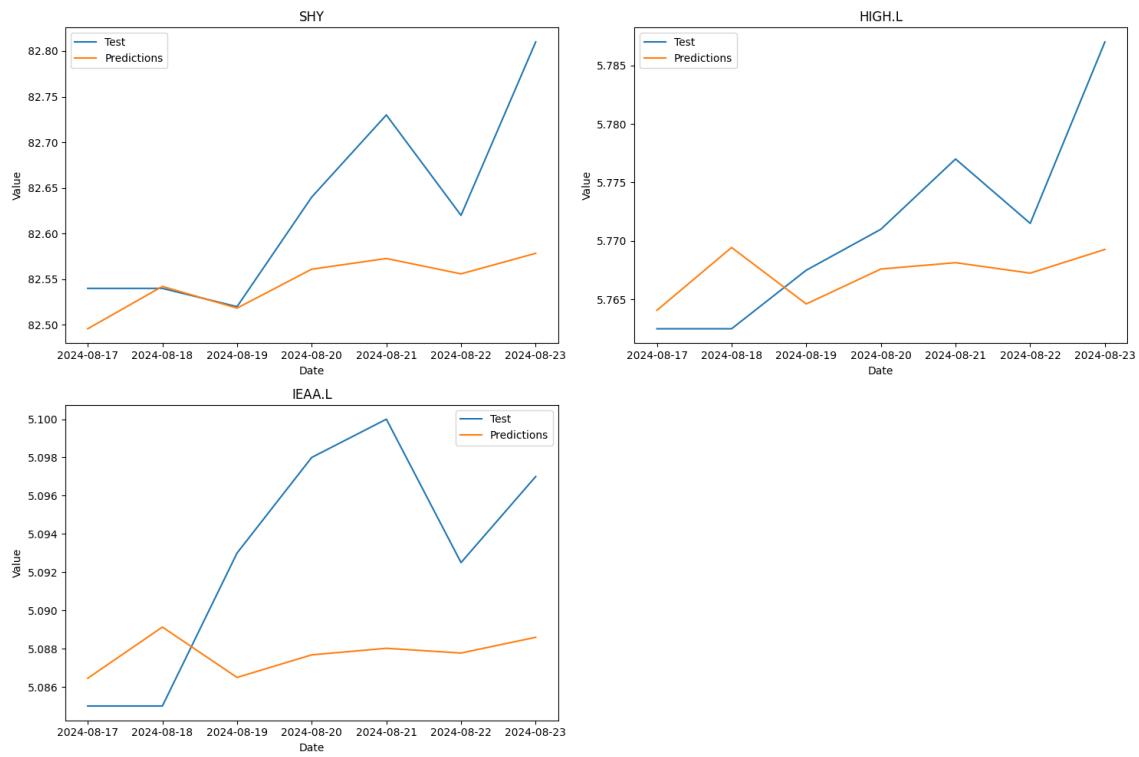


Figura 11 – Predições com menores medidas de erro – *LSTM* sem atualização

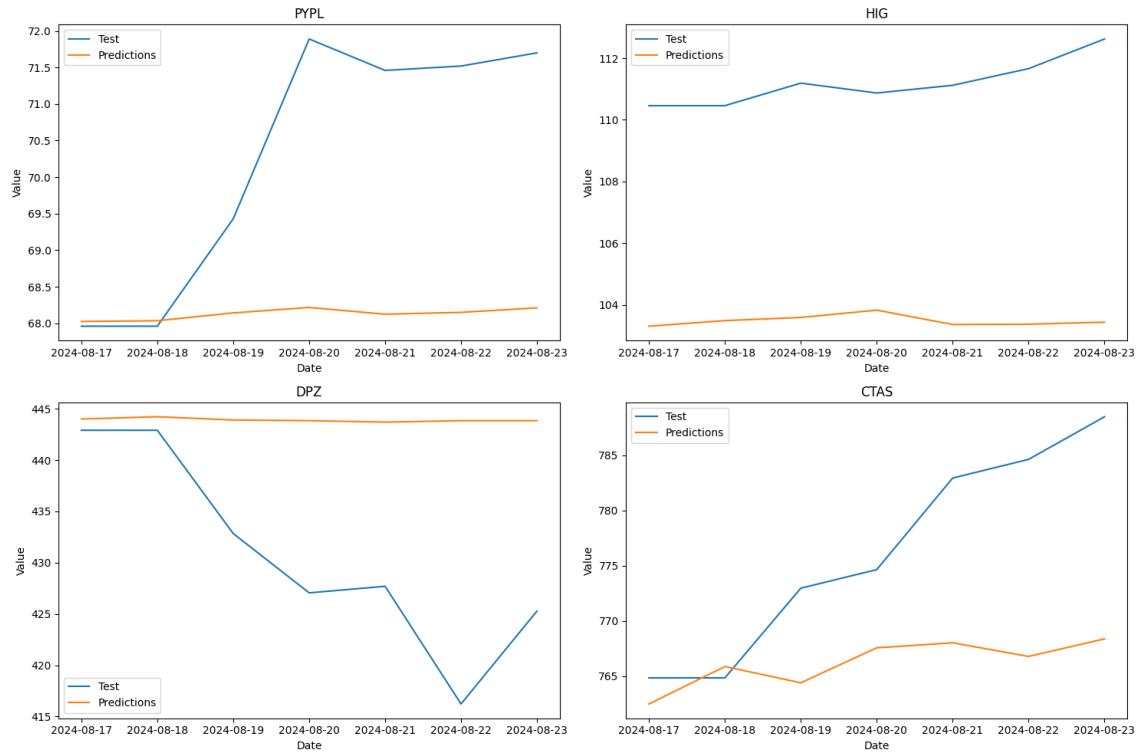


Figura 12 – Predições com maiores medidas de erro – *LSTM* sem atualização

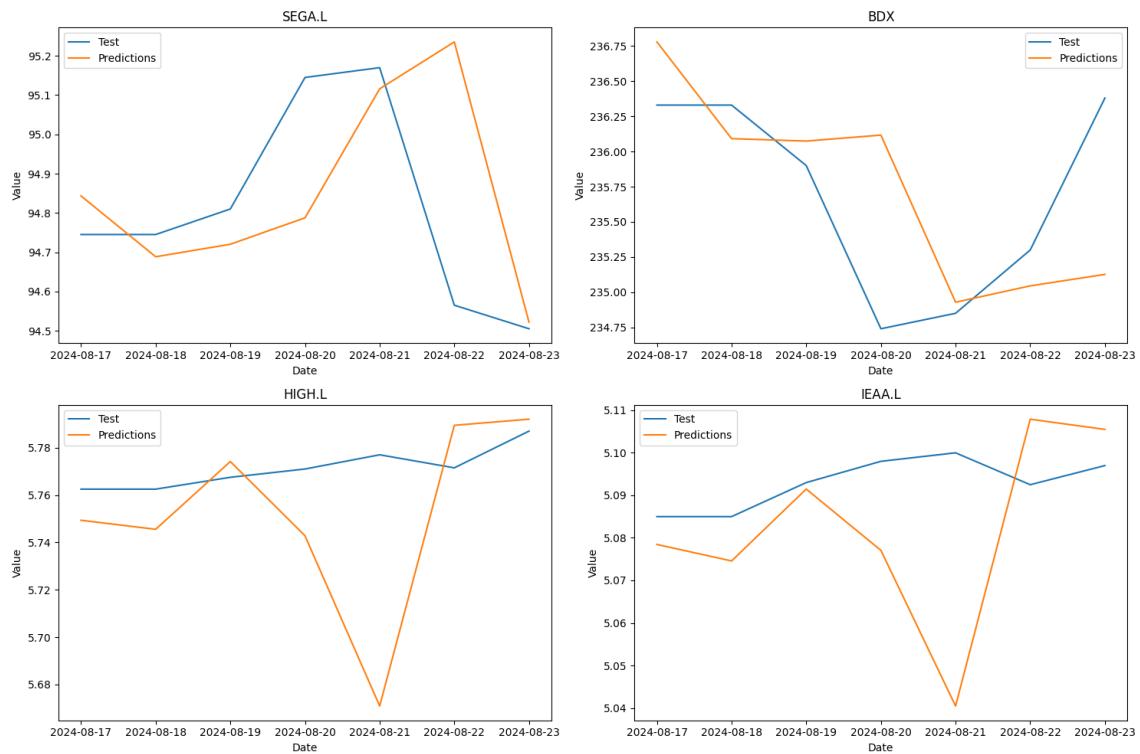


Figura 13 – Predições com menores medidas de erro – *LSTM* com atualização

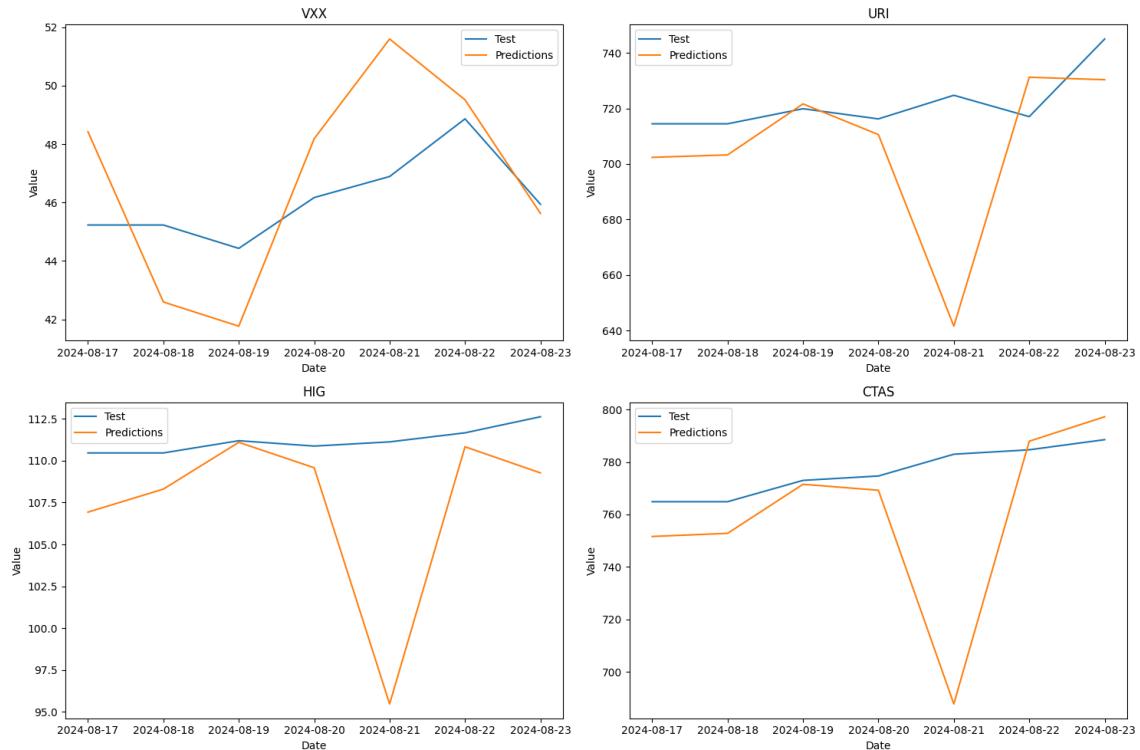


Figura 14 – Predições com maiores medidas de erro – *LSTM* com atualização

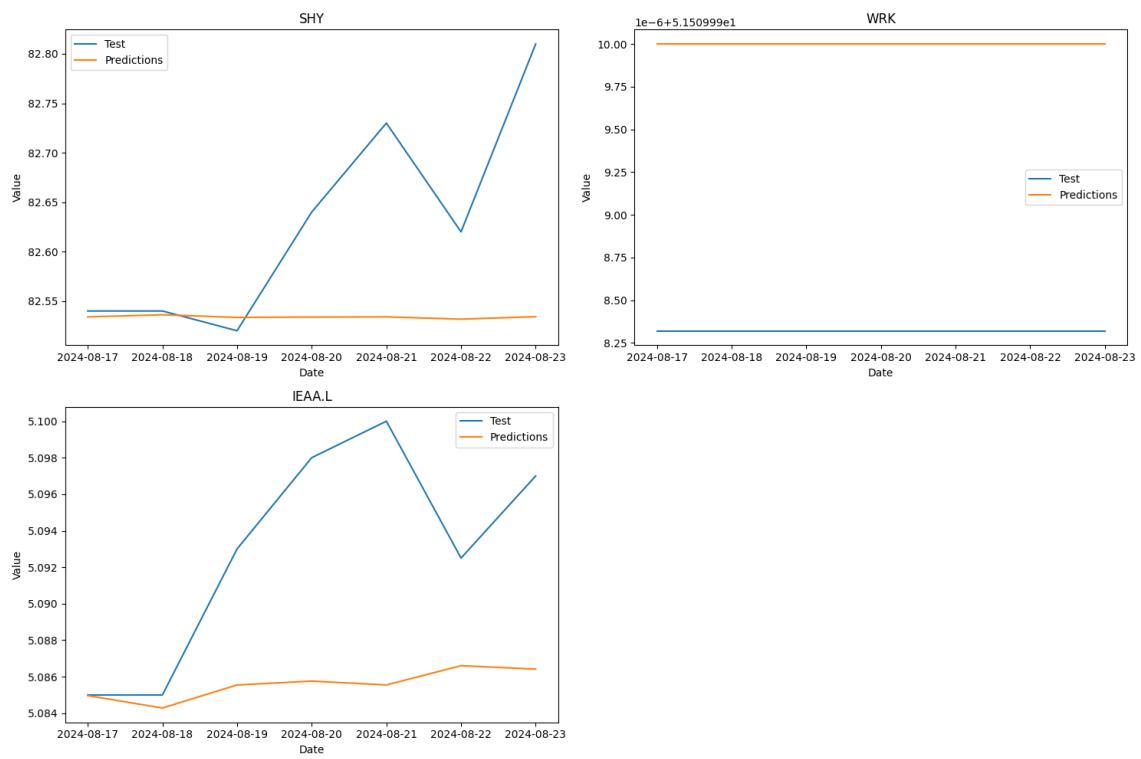


Figura 15 – Predições com menores medidas de erro – *MLP* sem atualização

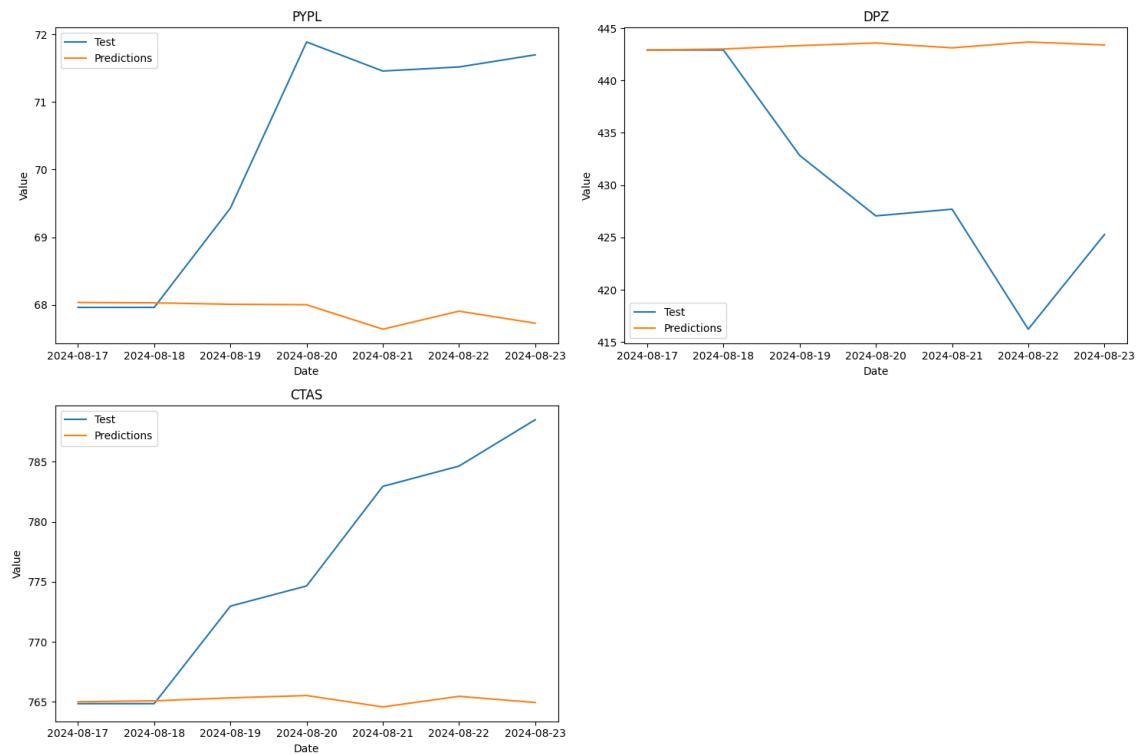


Figura 16 – Predições com maiores medidas de erro – *MLP* sem atualização

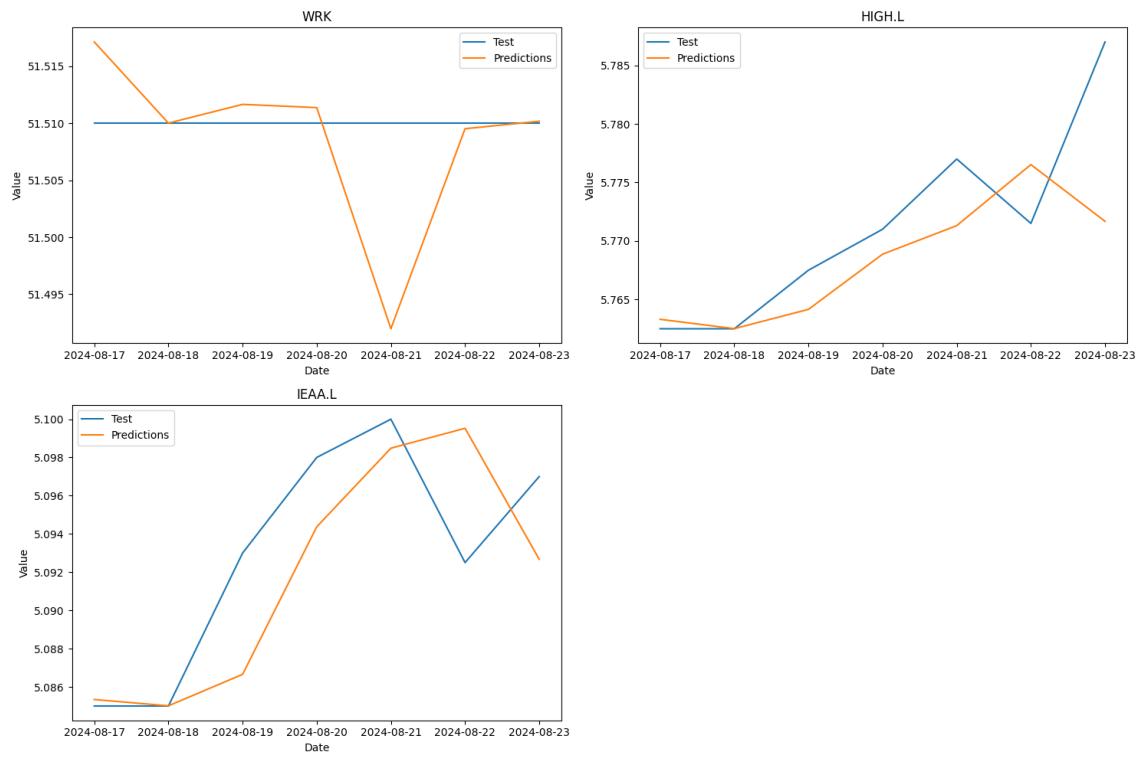


Figura 17 – Predições com menores medidas de erro – *MLP* com atualização

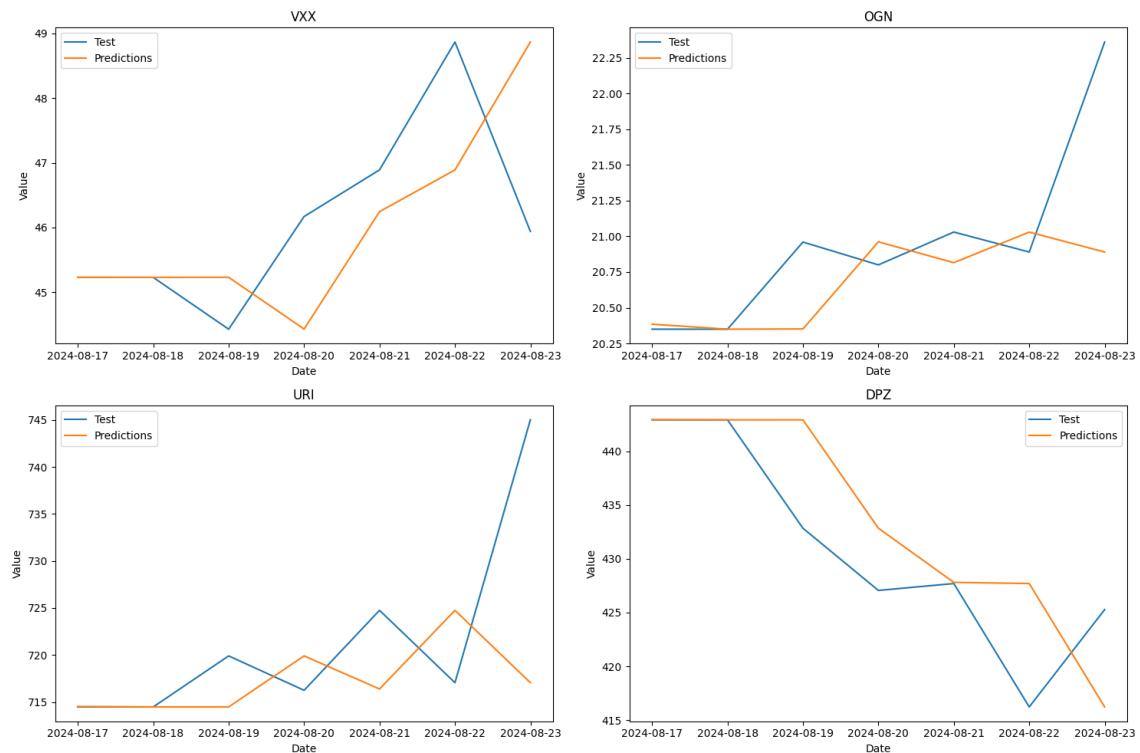


Figura 18 – Predições com maiores medidas de erro – *MLP* com atualização

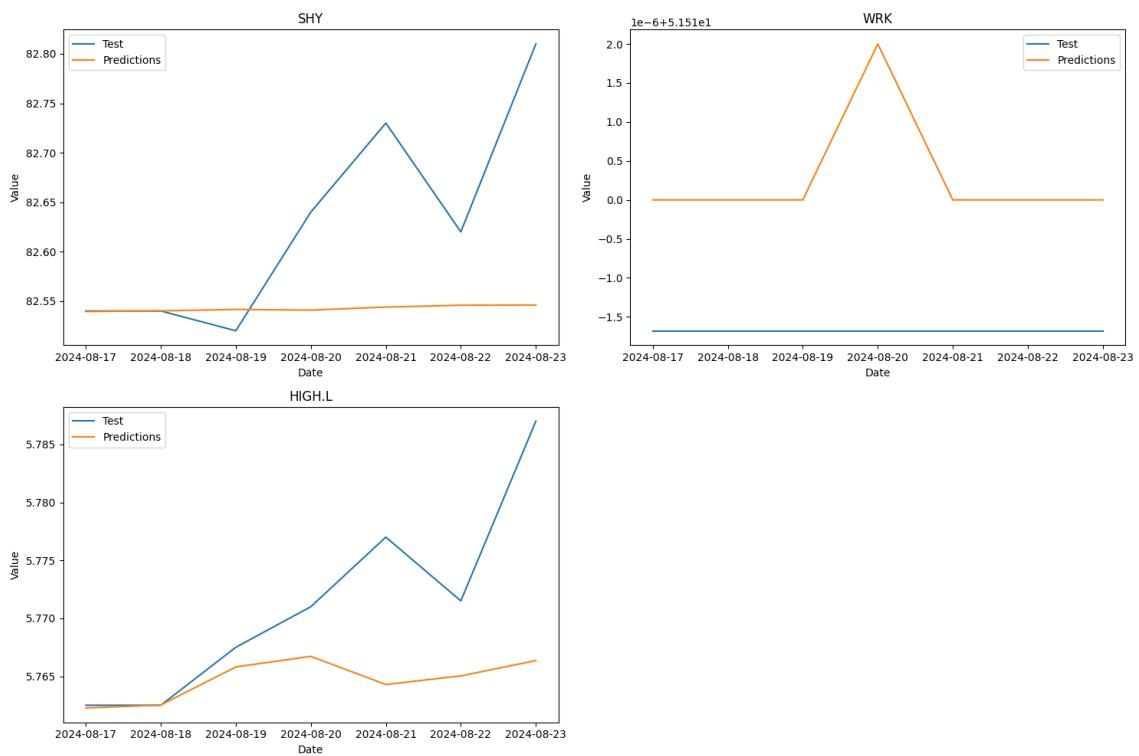


Figura 19 – Predições com menores medidas de erro – *DLinear* sem atualização

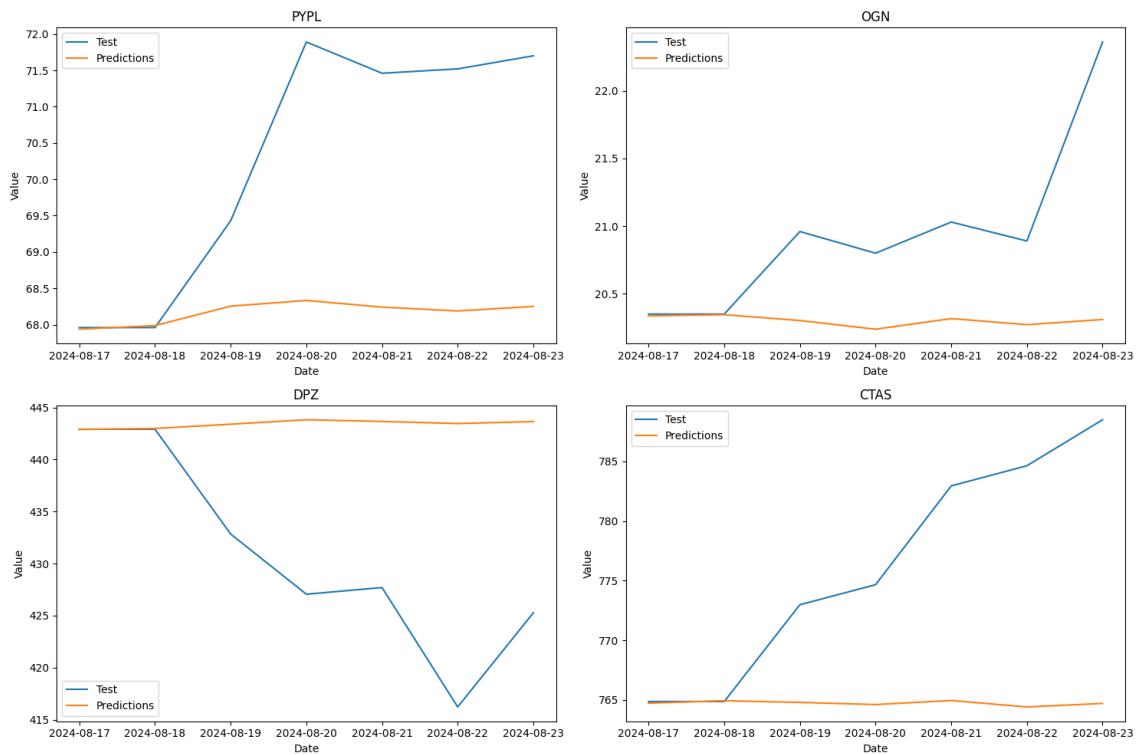


Figura 20 – Predições com maiores medidas de erro – *DLinear* sem atualização

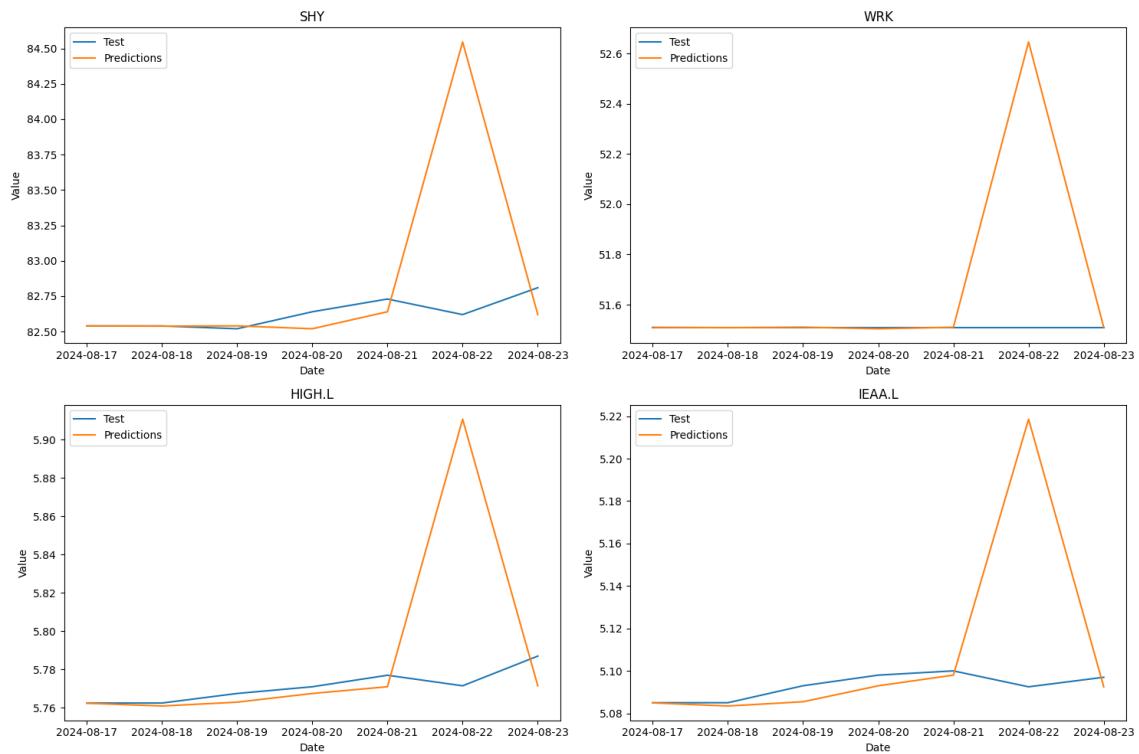


Figura 21 – Predições com menores medidas de erro – *DLinear* com atualização

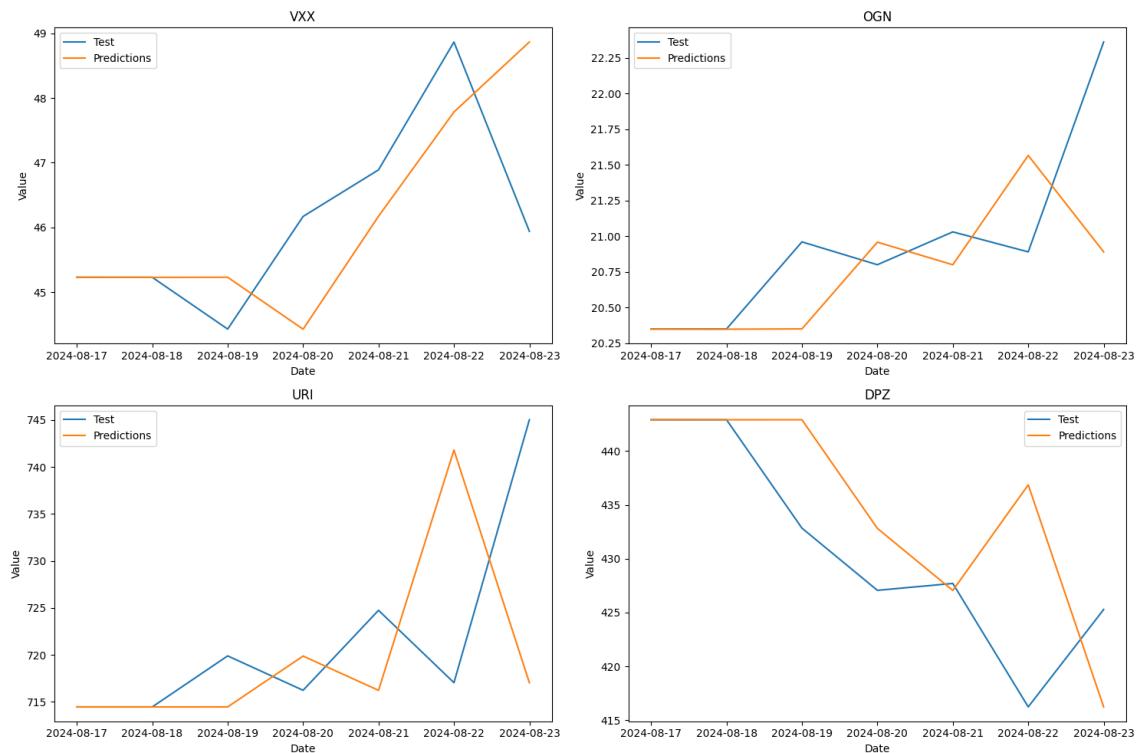


Figura 22 – Predições com maiores medidas de erro – *DLinear* com atualização

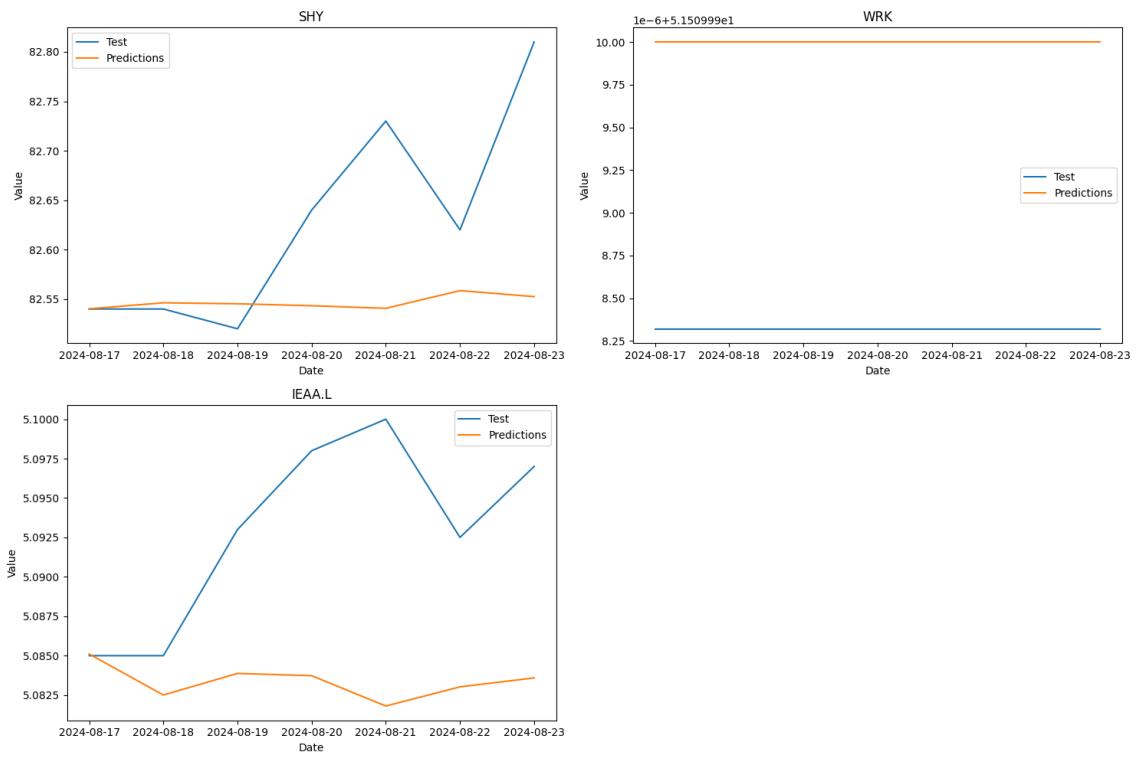


Figura 23 – Predições com menores medidas de erro – *NLinear* sem atualização

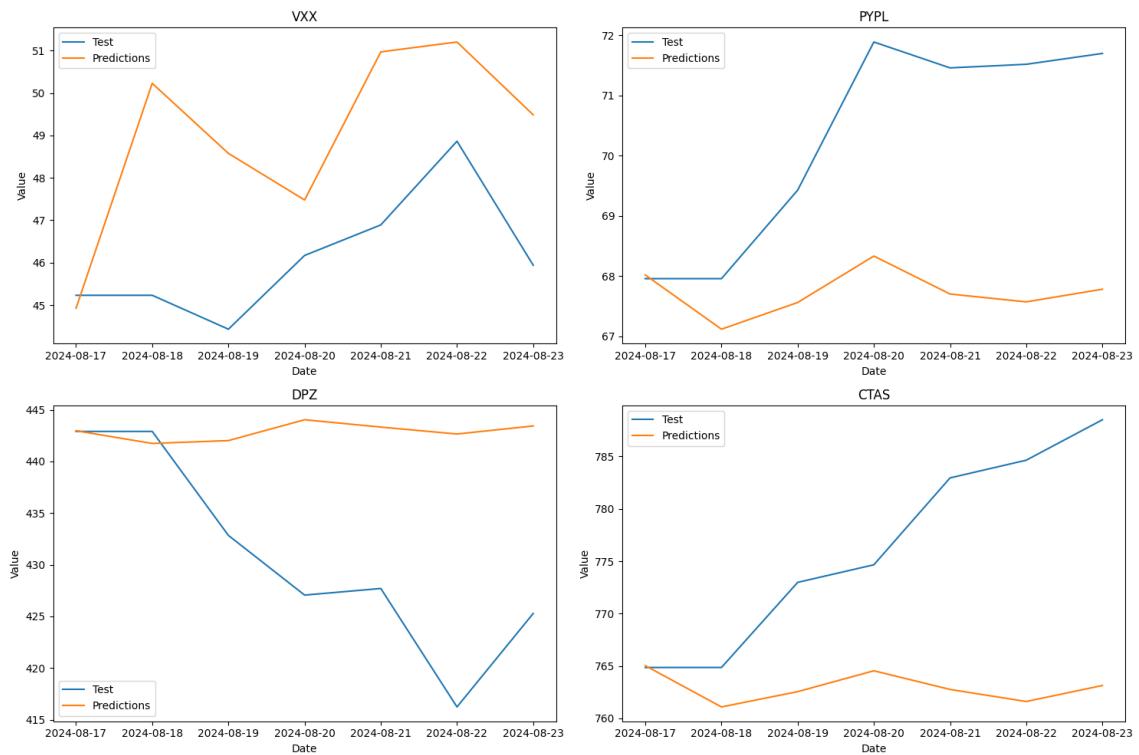


Figura 24 – Predições com maiores medidas de erro – *NLinear* sem atualização

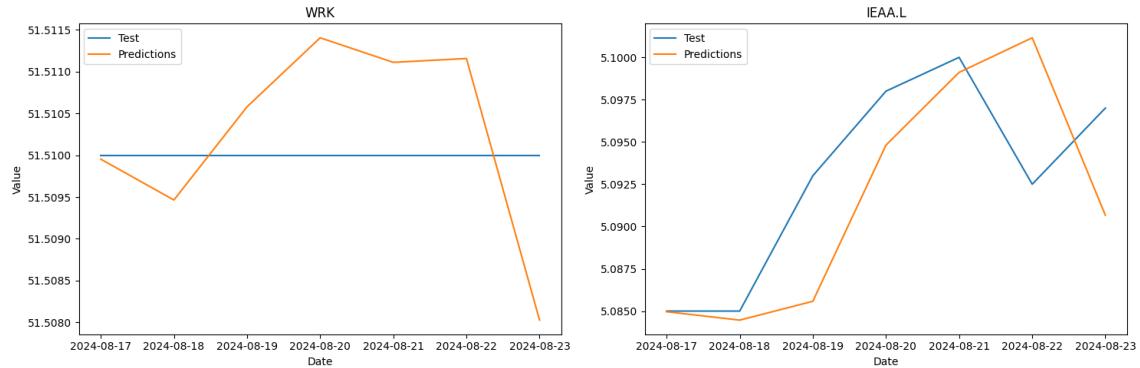


Figura 25 – Predições com menores medidas de erro – *NLinear* com atualização

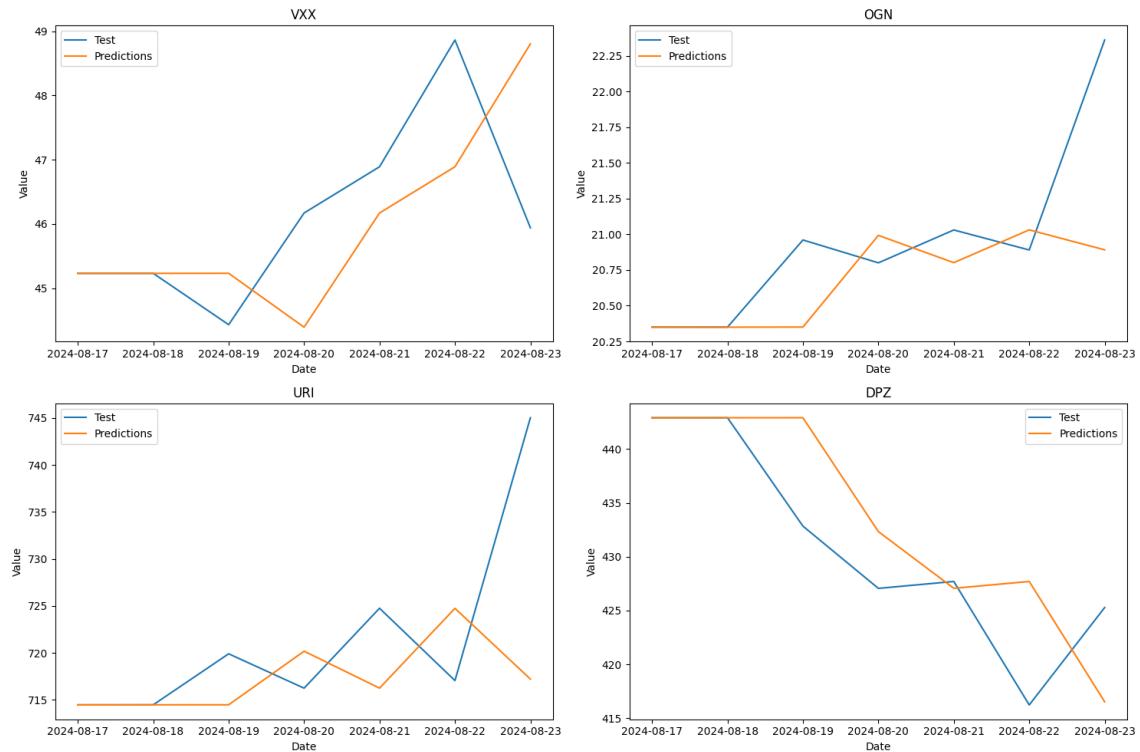


Figura 26 – Predições com maiores medidas de erro – *NLinear* com atualização

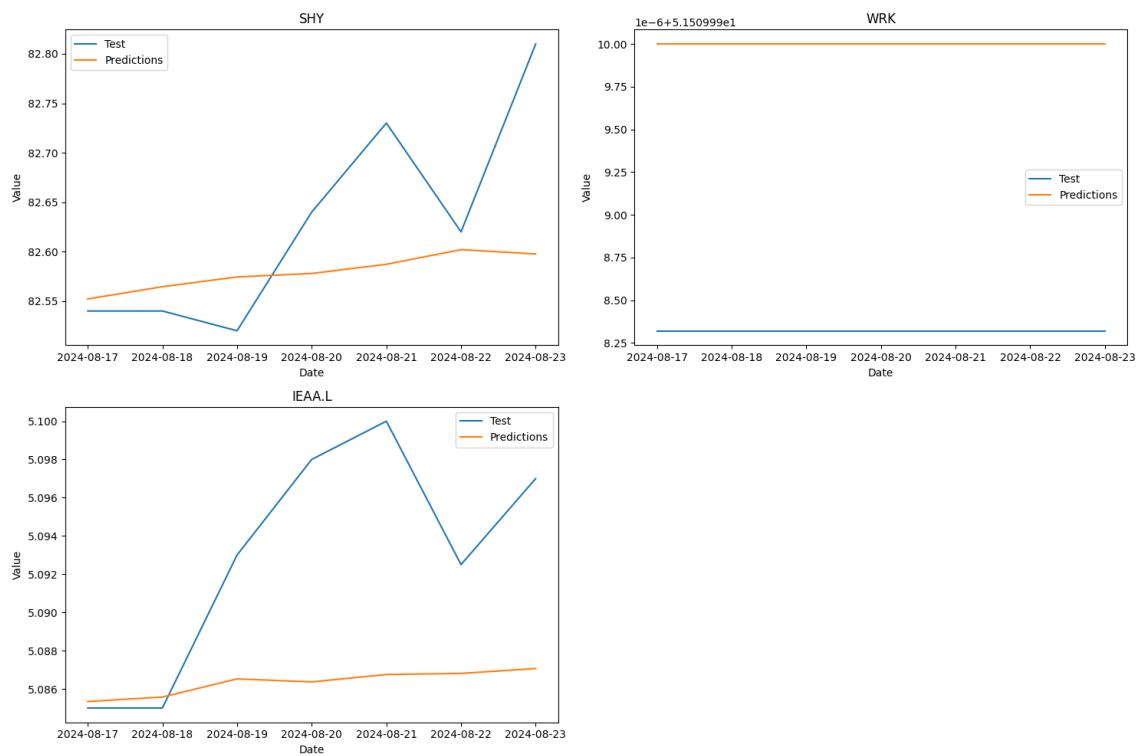


Figura 27 – Predições com menores medidas de erro – *Informer* sem atualização

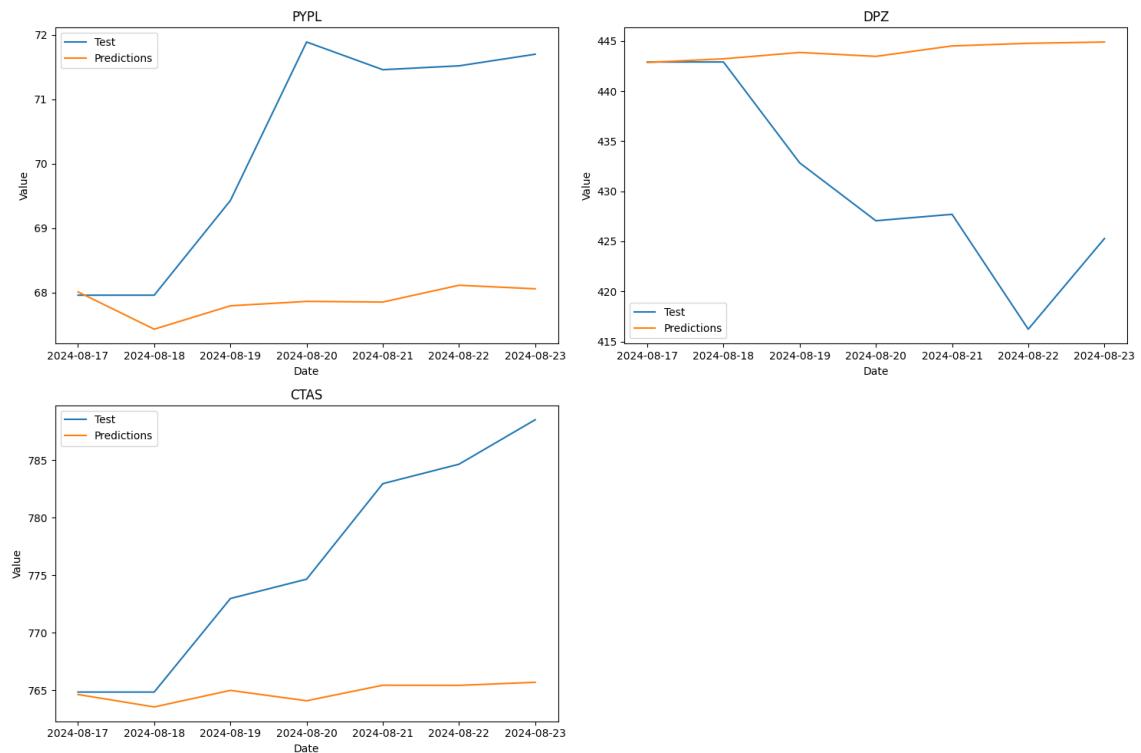


Figura 28 – Predições com maiores medidas de erro – *Informer* sem atualização

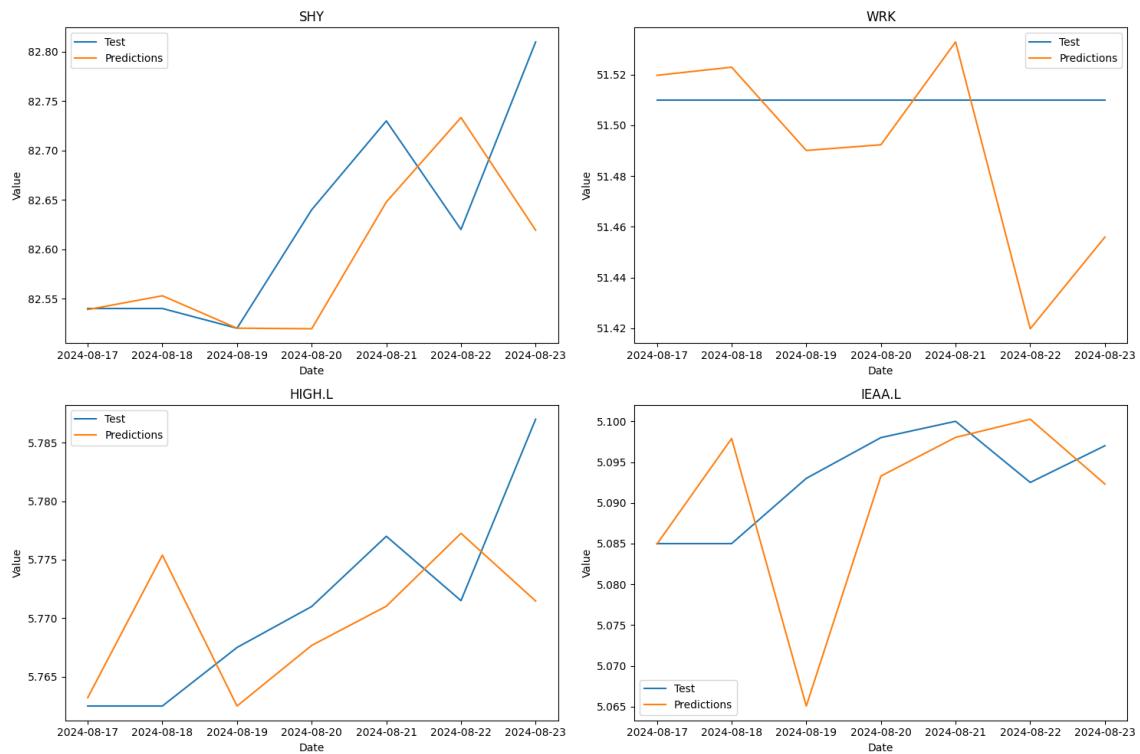


Figura 29 – Predições com menores medidas de erro – *Informer* com atualização

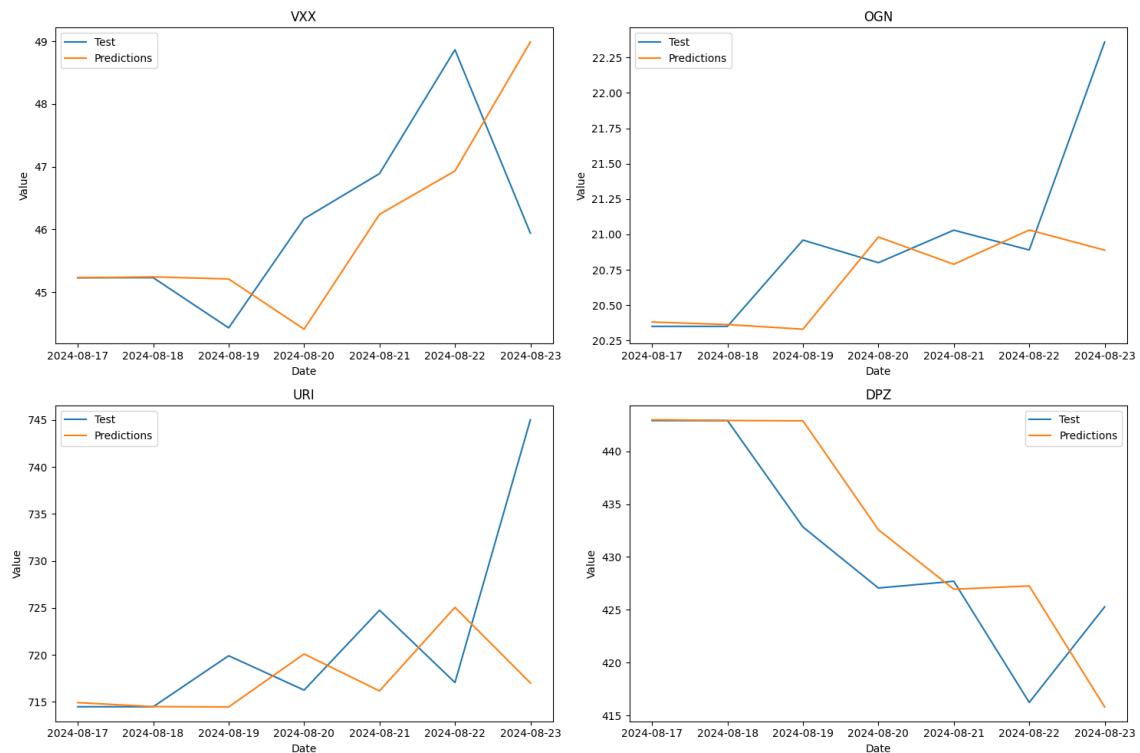


Figura 30 – Predições com maiores medidas de erro – *Informer* com atualização

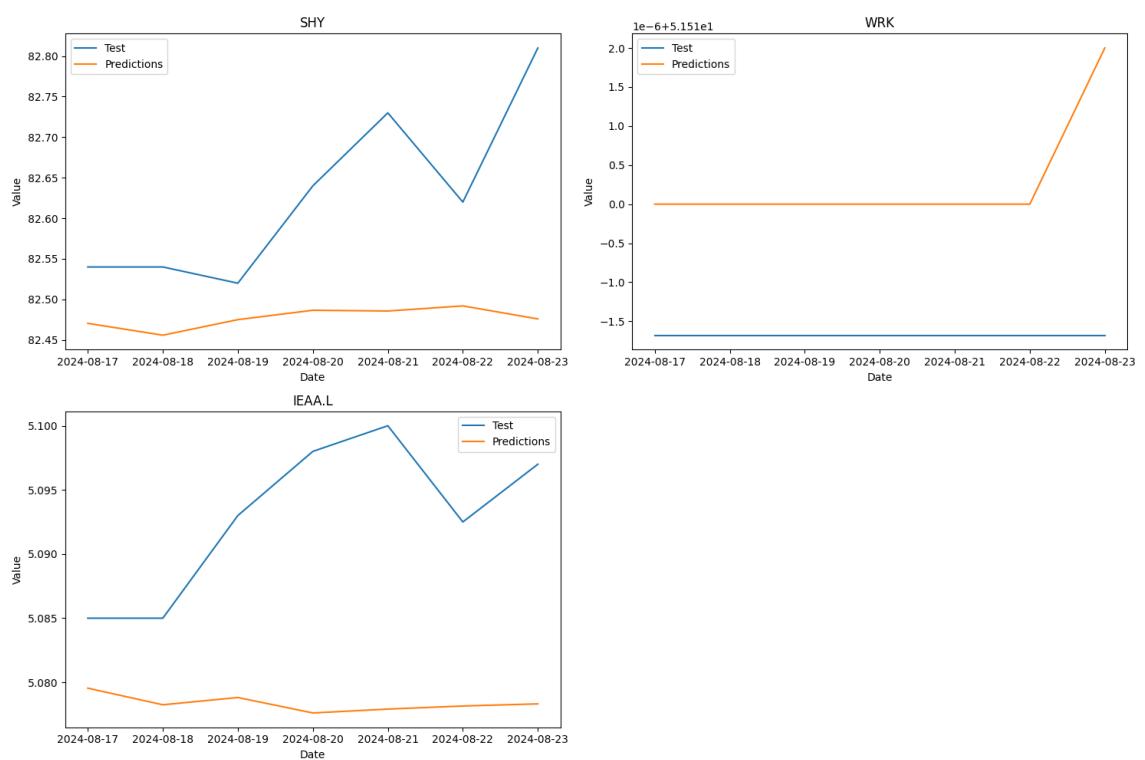


Figura 31 – Predições com menores medidas de erro – *Autoformer* sem atualização

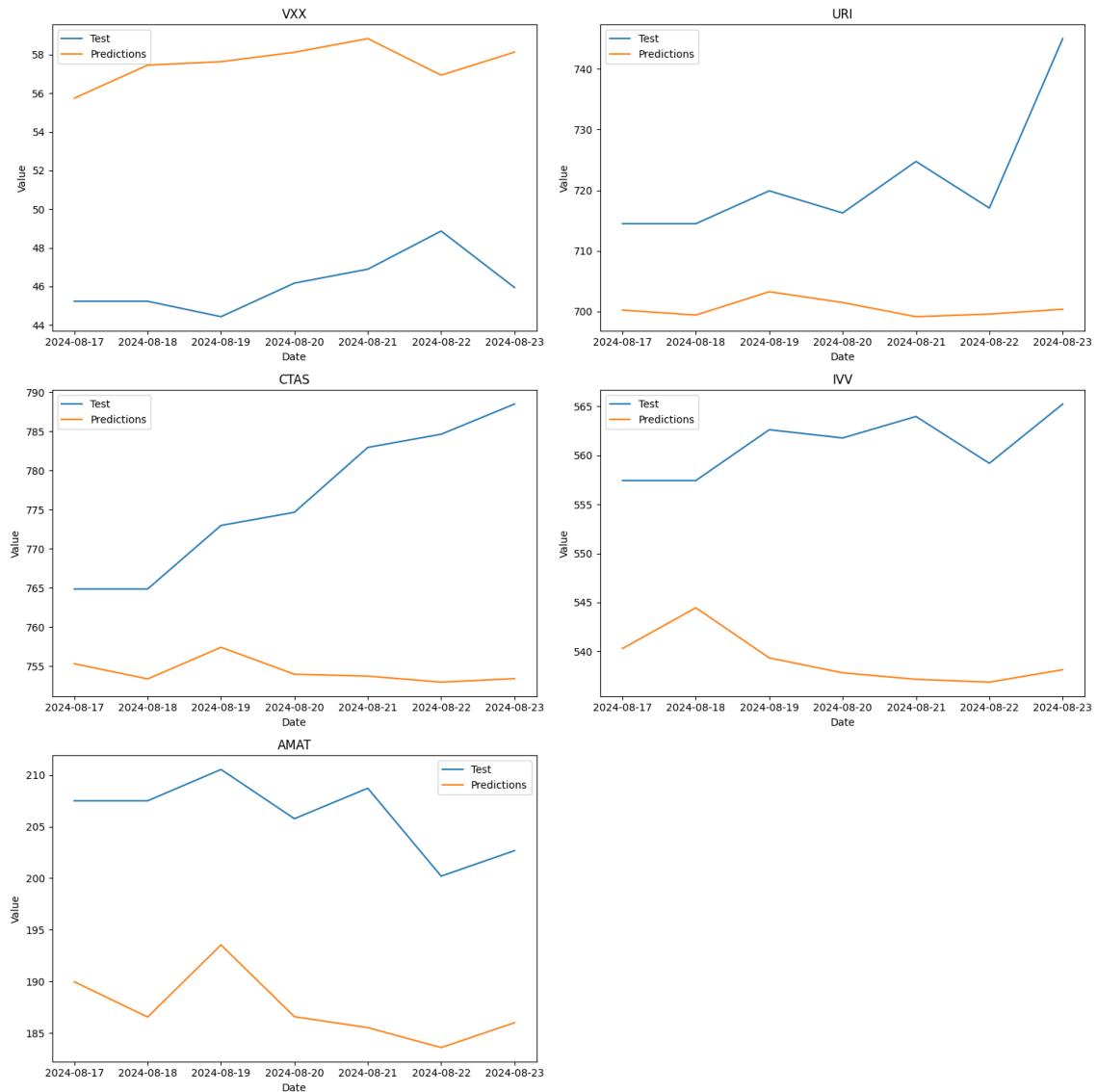


Figura 32 – Predições com maiores medidas de erro – *Autoformer* sem atualização



Figura 33 – Predições com menores medidas de erro – *Autoformer* com atualização

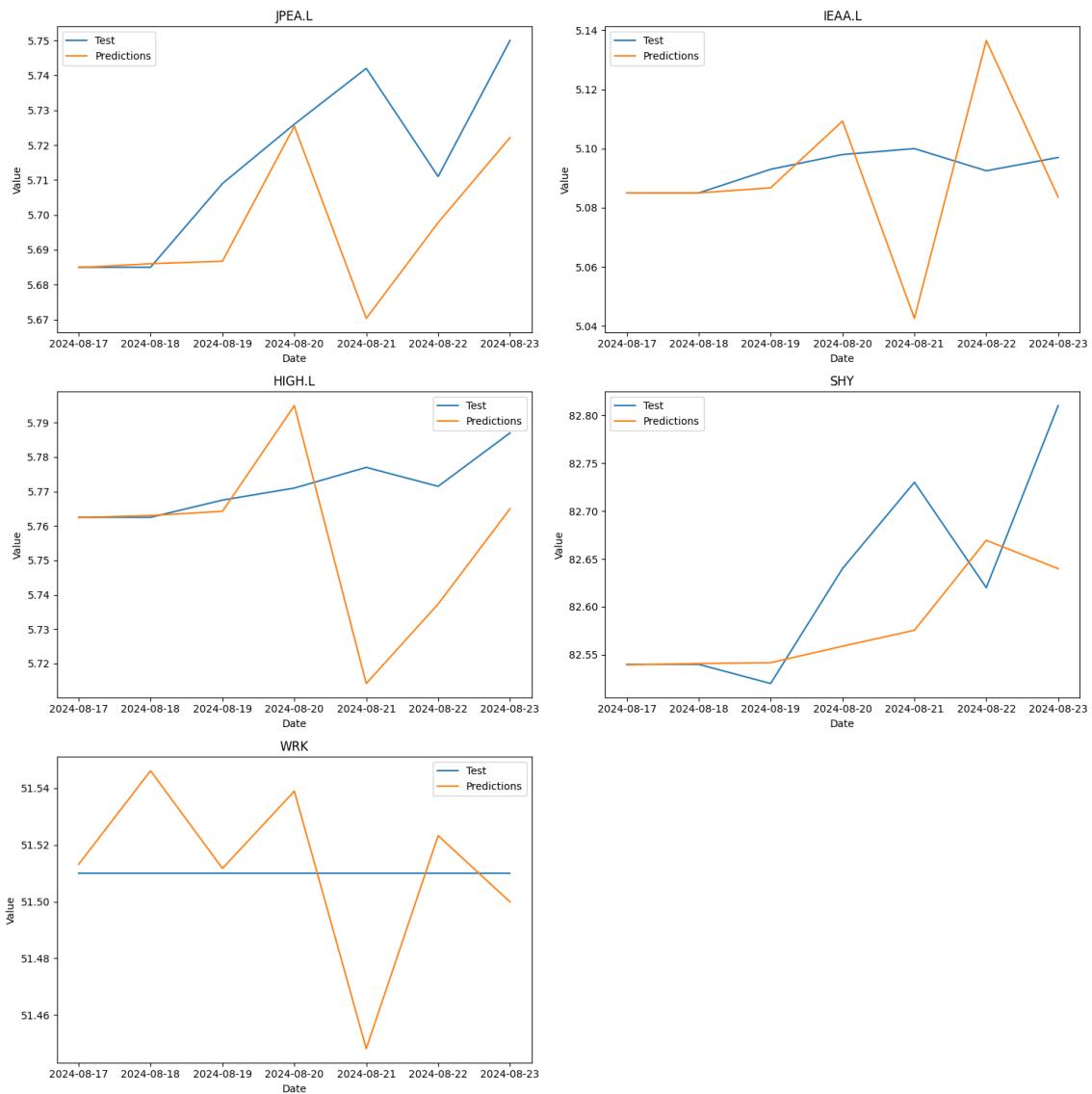


Figura 34 – Predições com maiores medidas de erro – *Autoformer* com atualização

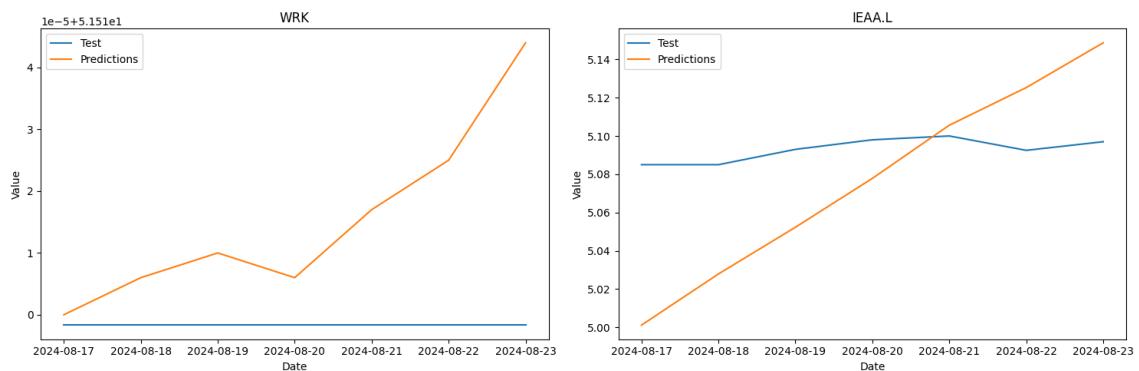


Figura 35 – Predições com menores medidas de erro – *FEDFormer* sem atualização

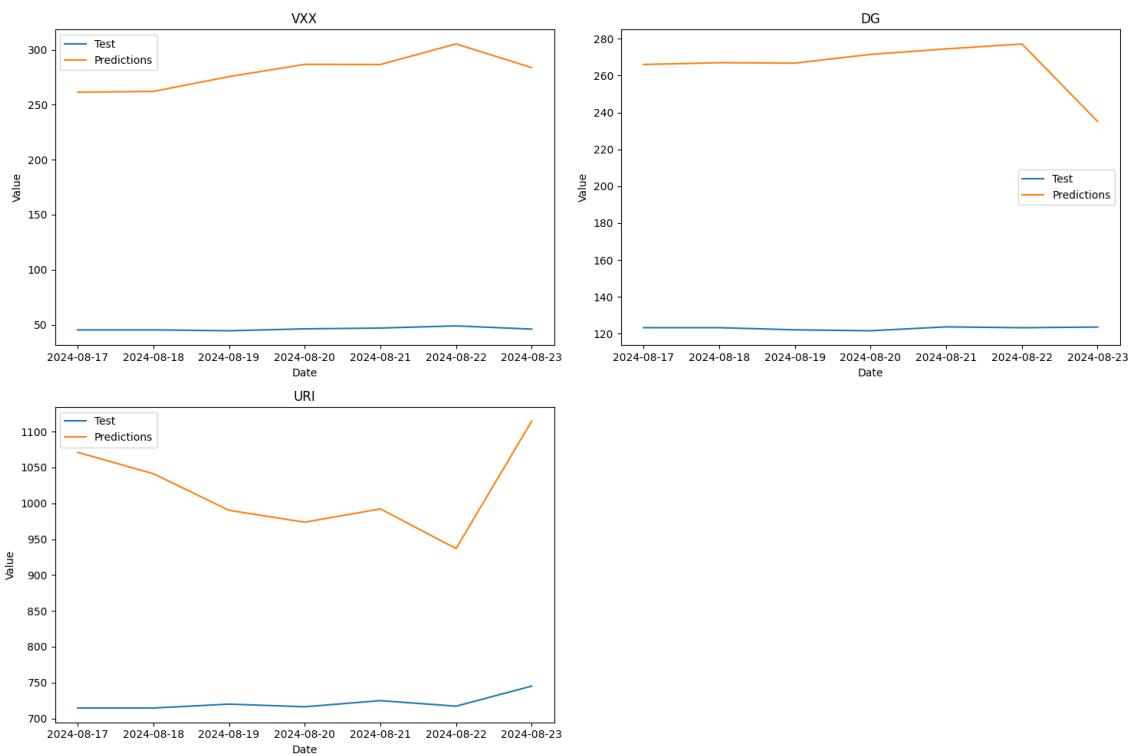


Figura 36 – Predições com maiores medidas de erro – *FEDFormer* sem atualização

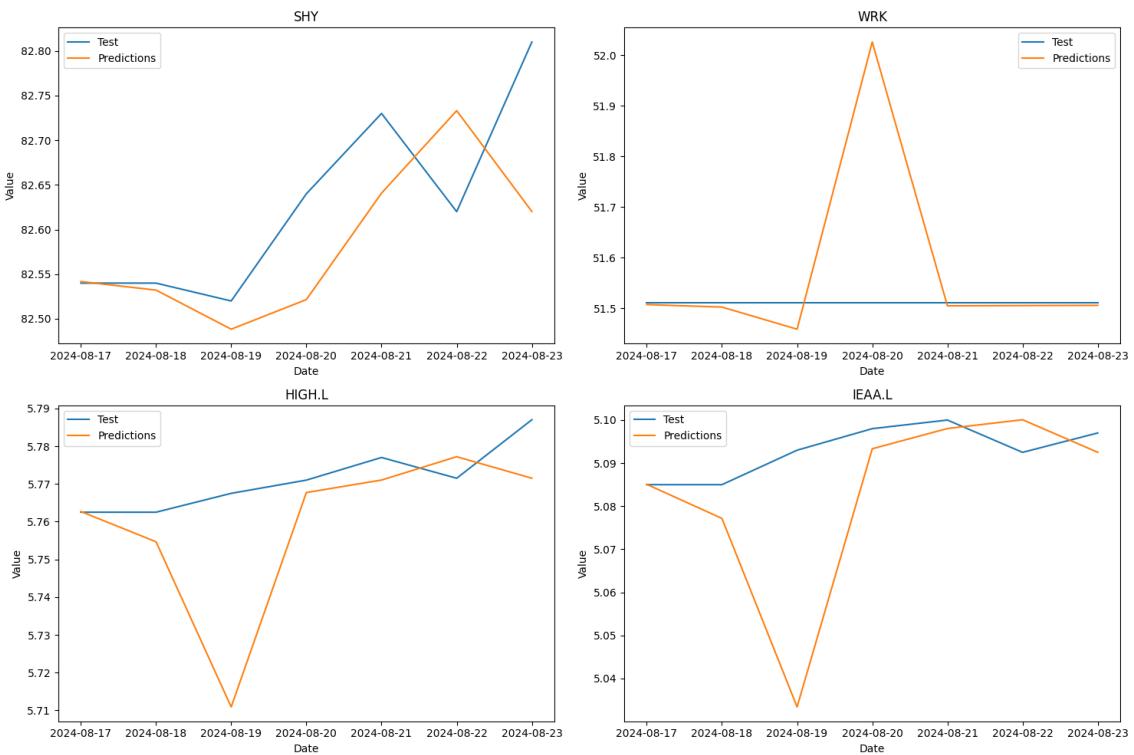


Figura 37 – Predições com menores medidas de erro – *FEDFormer* com atualização

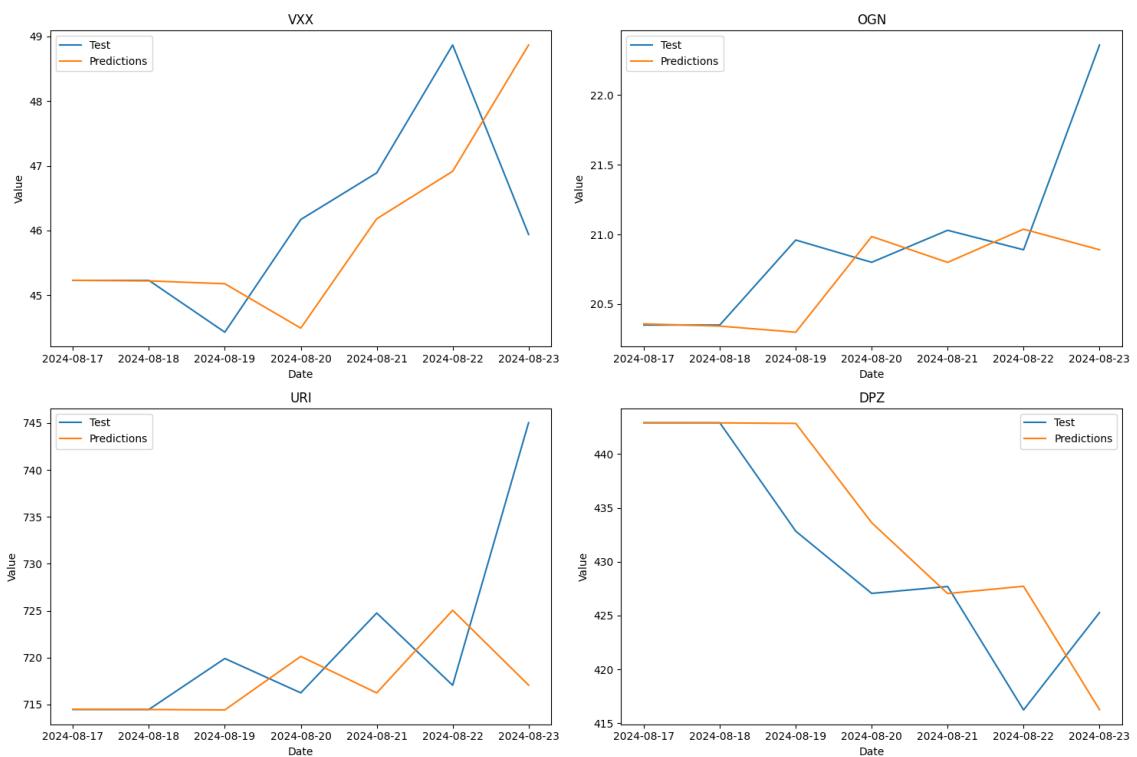


Figura 38 – Previsões com maiores medidas de erro – *FEDFormer* com atualização

Modelo	Top/Bottom	Medida	Assets
ARIMA.u	top	mae	['WRK', 'IEAA.L']
ARIMA.u	top	rmse	['WRK', 'IEAA.L']
ARIMA.u	top	pocid	['WRK', 'IEAA.L']
ARIMA.u	bottom	mae	['URI', 'DPZ']
ARIMA.u	bottom	rmse	['URI', 'DPZ']
ARIMA.u	bottom	pocid	['VXX', 'OGN']
LSTM	top	mae	['HIGH.L', 'IEAA.L']
LSTM	top	rmse	['IEAA.L', 'HIGH.L']
LSTM	top	pocid	['SHY', 'HIGH.L']
LSTM	bottom	mae	['DPZ', 'CTAS']
LSTM	bottom	rmse	['DPZ', 'CTAS']
LSTM	bottom	pocid	['HIG', 'PYPL']
LSTM.u	top	mae	['IEAA.L', 'HIGH.L']
LSTM.u	top	rmse	['IEAA.L', 'HIGH.L']
LSTM.u	top	pocid	['SEGA.L', 'BDX']
LSTM.u	bottom	mae	['URI', 'CTAS']
LSTM.u	bottom	rmse	['CTAS', 'URI']
LSTM.u	bottom	pocid	['VXX', 'HIG']
MLP	top	mae	['WRK', 'IEAA.L']
MLP	top	rmse	['WRK', 'IEAA.L']
MLP	top	pocid	['WRK', 'SHY']
MLP	bottom	mae	['DPZ', 'CTAS']
MLP	bottom	rmse	['DPZ', 'CTAS']
MLP	bottom	pocid	['PYPL', 'DPZ']
MLP.u	top	mae	['IEAA.L', 'WRK']
MLP.u	top	rmse	['IEAA.L', 'HIGH.L']
MLP.u	top	pocid	['WRK', 'IEAA.L']
MLP.u	bottom	mae	['URI', 'DPZ']
MLP.u	bottom	rmse	['URI', 'DPZ']
MLP.u	bottom	pocid	['VXX', 'OGN']
DLINEAR	top	mae	['WRK', 'HIGH.L']
DLINEAR	top	rmse	['WRK', 'HIGH.L']
DLINEAR	top	pocid	['WRK', 'SHY']
DLINEAR	bottom	mae	['DPZ', 'CTAS']
DLINEAR	bottom	rmse	['DPZ', 'CTAS']
DLINEAR	bottom	pocid	['OGN', 'PYPL']
DLINEAR.u	top	mae	['IEAA.L', 'HIGH.L']
DLINEAR.u	top	rmse	['IEAA.L', 'HIGH.L']
DLINEAR.u	top	pocid	['WRK', 'SHY']
DLINEAR.u	bottom	mae	['URI', 'DPZ']
DLINEAR.u	bottom	rmse	['URI', 'DPZ']
DLINEAR.u	bottom	pocid	['VXX', 'OGN']
NLINEAR	top	mae	['WRK', 'IEAA.L']
NLINEAR	top	rmse	['WRK', 'IEAA.L']
NLINEAR	top	pocid	['WRK', 'SHY']
NLINEAR	bottom	mae	['CTAS', 'DPZ']
NLINEAR	bottom	rmse	['CTAS', 'DPZ']
NLINEAR	bottom	pocid	['VXX', 'PYPL']

Tabela 4 – Relação de gráficos com menores e maiores valores de *MAE*, *RMSE* e *POCID*

Modelo	Top/Bottom	Medida	Assets
NLINEAR.u	top	mae	['WRK', 'IEAA.L']
NLINEAR.u	top	rmse	['WRK', 'IEAA.L']
NLINEAR.u	top	pocid	['WRK', 'IEAA.L']
NLINEAR.u	bottom	mae	['URI', 'DPZ']
NLINEAR.u	bottom	rmse	['URI', 'DPZ']
NLINEAR.u	bottom	pocid	['VXX', 'OGN']
INFORMER	top	mae	['WRK', 'IEAA.L']
INFORMER	top	rmse	['WRK', 'IEAA.L']
INFORMER	top	pocid	['WRK', 'SHY']
INFORMER	bottom	mae	['DPZ', 'CTAS']
INFORMER	bottom	rmse	['DPZ', 'CTAS']
INFORMER	bottom	pocid	['PYPL', 'DPZ']
INFORMER.u	top	mae	['HIGH.L', 'IEAA.L']
INFORMER.u	top	rmse	['HIGH.L', 'IEAA.L']
INFORMER.u	top	pocid	['WRK', 'SHY']
INFORMER.u	bottom	mae	['URI', 'DPZ']
INFORMER.u	bottom	rmse	['URI', 'DPZ']
INFORMER.u	bottom	pocid	['VXX', 'OGN']
AUTOFORMER	top	mae	['WRK', 'IEAA.L']
AUTOFORMER	top	rmse	['WRK', 'IEAA.L']
AUTOFORMER	top	pocid	['WRK', 'SHY']
AUTOFORMER	bottom	mae	['IVV', 'CTAS']
AUTOFORMER	bottom	rmse	['CTAS', 'URI']
AUTOFORMER	bottom	pocid	['VXX', 'AMAT']
AUTOFORMER.u	top	mae	['IEAA.L', 'JPEA.L']
AUTOFORMER.u	top	rmse	['IEAA.L', 'HIGH.L']
AUTOFORMER.u	top	pocid	['WRK', 'SHY']
AUTOFORMER.u	bottom	mae	['URI', 'DPZ']
AUTOFORMER.u	bottom	rmse	['URI', 'DPZ']
AUTOFORMER.u	bottom	pocid	['VXX', 'OGN']
FEDFORMER	top	mae	['WRK', 'IEAA.L']
FEDFORMER	top	rmse	['WRK', 'IEAA.L']
FEDFORMER	top	pocid	['WRK', 'IEAA.L']
FEDFORMER	bottom	mae	['URI', 'VXX']
FEDFORMER	bottom	rmse	['URI', 'VXX']
FEDFORMER	bottom	pocid	['VXX', 'DG']
FEDFORMER.u	top	mae	['IEAA.L', 'HIGH.L']
FEDFORMER.u	top	rmse	['HIGH.L', 'IEAA.L']
FEDFORMER.u	top	pocid	['SHY', 'WRK']
FEDFORMER.u	bottom	mae	['URI', 'DPZ']
FEDFORMER.u	bottom	rmse	['URI', 'DPZ']
FEDFORMER.u	bottom	pocid	['VXX', 'OGN']

Tabela 5 – Relação de gráficos com menores e maiores valores de *MAE*, *RMSE* e *POCID*
(cont.)

5 CONCLUSÕES

Neste trabalho foi analisado o desempenho de três modelos baseados na arquitetura de *Transformers* para predição de séries temporais — *Informer*, *AutoFormer* e *FedFormer* — em especial no domínio de séries financeiras. Com a finalidade de propiciar a comparação com outros estudos, também foram escolhidos modelos de predição mais conhecidos e com mais estudos publicados, o modelo estatístico *ARIMA* e os modelos de *Deep Learning* *MLP*, *LSTM* e dois modelos também baseados em *MLP*, *NLinear* e *DLinear*, propostos no estudo (Zeng *et al.*, 2023) que questionou a eficiência de *Transformers* para a tarefa de predição de *TS*. A escolha do conjunto de dados M6 também teve como propósito de facilitar a comparação deste trabalho com outros modelos utilizados nessa competição e seus respectivos resultados.

O desempenho foi medido por dois aspectos: proximidade dos valores preditos com os reais, através das medidas de erro *MAE*, *RMSE* e *POCID*; e dos principais recursos consumidos: processamento, memória e tempo de execução. Com relação às medidas de erro, os três modelos baseados em *Transformers* obtiveram resultados similares aos modelos com as menores medidas — *ARIMA*, *MLP*, *NLinear* e *DLinear* — superando o *LSTM*, especificamente utilizado para esse fim. Sob esse aspecto, os modelos de *Transformers* podem ser considerados dentre os mais precisos com relação aos valores preditos.

Por outro lado, considerando os recursos necessários para as predições, o desempenho dos modelos de *Transformers* foram compatíveis com o esperado para qualquer modelo de *DL*. Apresentando os maiores valores de consumo de memória, processamento de *GPU* e tempo de execução — de 10 a 30 vezes superior ao modelos menos custosos — o uso de *Transformers* para a predição de *FTS* não se mostrou viável.

O estudo encerra com a constatação que os modelos com melhor custo-benefício para a tarefa de predição de *FTS*, combinando as medidas mais precisas com o menor consumo de recursos computacionais e no menor tempo, foram os modelos *ARIMA*, *MLP* e suas variações *NLinear* e *DLinear*. O modelo estatístico *ARIMA* apresenta a vantagem de possuir o menor consumo de recurso, sem necessidade de uso de *GPU*, porém ao custo de exigir o hiperparâmetro de sazonalidade estimado externamente. Apesar dos modelos *MLP* possuírem metade da quantidade de hiperparâmetros com relação ao *ARIMA*, eles consomem mais recursos, porém ainda modestos comparados ao padrão computacional disponível atualmente e fazem o ajuste dos hiperparâmetros de forma totalmente automatizada durante a etapa de treinamento. Ambas as abordagens possuem suas vantagens específicas e a escolha da mais apropriada deve levar em consideração o contexto da sua aplicação.

5.1 Limitações do Estudo

O tempo disponível para o estudo junto à quantidade de modelos escolhidos para serem avaliados foram os principais fatores restritivos. Para viabilizar este trabalho, foi decidido pelo uso de implementações que além de ajustarem os modelos também fazem a otimização de seus hiperparâmetros. Desta forma, dentro do escopo deste estudo admite-se que ainda haja espaço para melhoria dos resultados obtidos pelas previsões com técnicas atualmente conhecidas.

5.2 Trabalhos Futuros

Durante o desenvolvimento deste estudo, muitas outras possibilidades consideradas importantes não puderam ser incluídas em vista da limitação do escopo. Dentre as principais, são destacadas:

Conjunto de dados maior modelos mais complexos de *DL*, como *LSTM* e *Transformers*, exigem maiores quantidades de dados para seu ajuste. O conjunto de dados M6 escolhido possui observações diárias durante um período inferior a três anos; cada uma de suas séries possuía menos de 1.000 observações. *FTS* disponibilizadas com diferentes períodos de aquisição, por exemplo 15 minutos ou a cada hora, podem ter resultados diferentes dos observados neste estudo, especialmente nos modelos mais complexos como *Transformers*;

Uso de técnicas complementares disponíveis para análise de séries temporais não foram abordadas nesse estudo, seu uso pode ter impacto direto na melhoria das medidas de erro, como uso de variáveis exógenas, séries multivariadas e treinamento com cross-validation;

Análise de sentimento muitos estudos em previsão de *FTS* se valem de *SA* como variáveis exógenas para melhorar os resultados das previsões. Esta possibilidade, não disponível para modelos mais simples como o *ARIMA*, podem aprimorar precisão das previsões de *Transformers* e justificar seu maior custo computacional;

Comparação com outros modelos o conjunto de bibliotecas utilizadas possuem uma ampla variedade de outros modelos que podem ser igualmente medidos pela pipeline proposta neste estudo, podendo revelar modelos tão eficientes quanto os apontados aqui;

Modelos pré-treinados em séries temporais o modelo TimesFM, assim como outras iniciativas similares com milhões de parâmetros ajustados a partir de bilhões de observações, começaram mais recentemente a serem disponibilizados publicamente. Devido ao tamanho reduzido do conjunto de dados M6, este modelo não pode ser incluído, consistindo uma linha interessante para estudos futuros.

REFERÊNCIAS

- ALDRIDGE, I. **High-frequency trading: a practical guide to algorithmic strategies and trading systems.** [S.l.: s.n.]: John Wiley & Sons, 2013. v. 604.
- ARACI, D. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*, 2019.
- ARLT, J.; ARLTOVÁ, M. Financial time series and their features. *Acta oeconomica pragensia*, v. 9, n. 4, p. 7–20, 2001.
- AROUSSI, R. **GitHub - ranaroussi/yfinance: Download market data from Yahoo! Finance's API — github.com.** 2019. <https://github.com/ranaroussi/yfinance>. [Accessed 28-05-2024].
- CHATFIELD, C.; XING, H. **The analysis of time series: an introduction with R.** [S.l.: s.n.]: Chapman and hall/CRC, 2019.
- CHEN, W. *et al.* Learning to rotate: Quaternion transformer for complicated periodical time series forecasting. *In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining.* [S.l.: s.n.], 2022. p. 146–156.
- DAS, A. *et al.* A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.
- ESTEVEZ, E. **What Is Reflexivity? How It Works, History, and Opposing Theories.** [S.l.: s.n.]: Investopedia. <https://www.investopedia.com/terms/r/reflexivity.asp>, 2020.
- FRYZLEWICZ, P. Lecture notes: Financial time series, arch and garch models. **University of Bristol**, 2007.
- FU, T. chung. A review on time series data mining. **Engineering Applications of Artificial Intelligence**, v. 24, n. 1, p. 164–181, 2011. ISSN 0952-1976. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0952197610001727>.
- GOOGLE. **Google Finance - Stock Market Prices, Real-time Quotes & Business News — google.com.** 2024. <https://www.google.com/finance/>. [Accessed 15-04-2024].
- HAYES, A. **Herd Instinct: Definition, Stock Market Examples, How to Avoid.** [S.l.: s.n.]: Investopedia. <https://www.investopedia.com/terms/h/herdinstinct.asp>, 2022.
- HAYES, A. **Technical Analysis: What It Is and How to Use It in Investing - investopedia.com.** 2024. <https://www.investopedia.com/terms/t/technicalanalysis.asp>. [Accessed 08-04-2024].
- KENTON, W. **What Is Behavioral Economics? Theories, Goals, and Applications.** [S.l.: s.n.]: Investopedia. <https://www.investopedia.com/terms/b/behavioraleconomics.asp>, 2023.

- KRAMER, L. **Long Position vs. Short Position: What's the Difference?** — **investopedia.com**. 2024. <https://www.investopedia.com/ask/answers/100314/whats-difference-between-long-and-short-position-market.asp>. [Accessed 10-04-2024].
- LI, S. *et al.* Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. **Advances in neural information processing systems**, v. 32, 2019.
- LI, Y. *et al.* Large language models in finance: A survey. *In: Proceedings of the Fourth ACM International Conference on AI in Finance*. [S.l.: s.n.], 2023. p. 374–382.
- LIN, T. C. The new investor. **UCLA L. Rev.**, HeinOnline, v. 60, p. 678, 2012.
- LIU, S. *et al.* Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. *In: International conference on learning representations*. [S.l.: s.n.], 2021.
- LIU, Y. *et al.* Stock price movement prediction from financial news with deep learning and knowledge graph embedding. *In: SPRINGER. Knowledge Management and Acquisition for Intelligent Systems: 15th Pacific Rim Knowledge Acquisition Workshop, PKAW 2018, Nanjing, China, August 28-29, 2018, Proceedings 15*. [S.l.: s.n.], 2018. p. 102–113.
- MANNING, C. Artificial intelligence definitions. **HAI Stanford University**, 2020.
- MINH, D. L. *et al.* Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network. **Ieee Access**, IEEE, v. 6, p. 55392–55404, 2018.
- MISWAN, N. H. *et al.* Comparative performance of arima and garch models in modelling and forecasting volatility of malaysia market properties and shares. **Applied Mathematical Sciences**, v. 8, n. 140, p. 7001–7012, 2014.
- MITCHELL, C. **Understanding an OHLC Chart and How to Interpret It** — **investopedia.com**. 2024. <https://www.investopedia.com/terms/o/ohlcchart.asp>.
- MUSA, T. H. A.; BOURAS, A. Anomaly detection: A survey. *In: .* [S.l.: s.n.], 2022. v. 217. ISSN 23673389.
- NATURE. **Statistical methods - Latest research and news | Nature** — **nature.com**. 2024. <https://www.nature.com/subjects/statistical-methods>. [Accessed 31-03-2024].
- NIXTLAVERSE - Nixtla — nixtlaverse.nixtla.io. <https://nixtlaverse.nixtla.io/>. [Accessed 15-09-2024].
- PARMEZAN, A. R.; SOUZA, V. M.; BATISTA, G. E. Time series prediction via similarity search: exploring invariances, distance measures and ensemble functions. **IEEE Access**, IEEE, v. 10, p. 78022–78043, 2022.
- PARMEZAN, A. R. S. **Predição de séries temporais por similaridade**. 2016. Tese (Doutorado) — Universidade de São Paulo, 2016.
- PARMEZAN, A. R. S.; BATISTA, G. E. d. A. P. A. Descrição de modelos estatísticos e de aprendizado máquina para predição de séries temporais. 2016.

- PATEL, M.; JARIWALA, K.; CHATTOPADHYAY, C. Deep learning techniques for stock market forecasting: Recent trends and challenges. *In: Proceedings of the 2023 6th International Conference on Software Engineering and Information Management.* [S.l.: s.n.], 2023. p. 1–11.
- PROSKY, J. *et al.* Sentiment predictability for stocks. *arXiv preprint arXiv:1712.05785*, 2017.
- RUNDO, F. *et al.* Machine learning for quantitative finance applications: A survey. *Applied Sciences*, MDPI, v. 9, n. 24, p. 5574, 2019.
- SECURITIES, U.; COMMISSION, E. **Trading Basics - Understanding the Different Ways to Buy and Sell Stock.** 2011. <https://www.sec.gov/tm/investor-alerts-bulletins/investoralertstrading101basicspdf>. [Accessed 09-04-2024].
- SECURITIES, U.; COMMISSION, E. **SEC.gov | Leveraged Investing Strategies & Know the Risks Before Using These Advanced Investment Tools — sec.gov.** 2021. https://www.sec.gov/oiea/investor-alerts-and-bulletins/ib_leveragedinvesting. [Accessed 09-04-2024].
- SEGAL, T. **Fundamental Analysis: Principles, Types, and How to Use It - investopedia.com.** 2023. <https://www.investopedia.com/terms/f/fundamentalanalysis.asp>. [Accessed 07-04-2024].
- SEGAL, T. **Spreads in Finance: The Multiple Meanings in Trading Explained — investopedia.com.** 2023. <https://www.investopedia.com/terms/s/spread.asp>. [Accessed 10-04-2024].
- SEWELL, M. Characterization of financial time series. *Rn*, v. 11, n. 01, p. 01, 2011.
- SEZER, O. B.; GUDELEK, M. U.; OZBAYOGLU, A. M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, Elsevier, v. 90, p. 106181, 2020.
- SONKIYA, P.; BAJPAI, V.; BANSAL, A. Stock price prediction using bert and gan. *arXiv preprint arXiv:2107.09055*, 2021.
- TAYLOR, S. J. **Modelling financial time series.** [S.l.: s.n.]: world scientific, 2008.
- TSAY, R. S. **Analysis of financial time series.** [S.l.: s.n.]: John wiley & sons, 2005.
- UDDIN, M. *et al.* The effect of covid-19 pandemic on global stock market volatility: Can economic strength help to manage the uncertainty? *Journal of Business Research*, Elsevier, v. 128, p. 31–44, 2021.
- UNIC. **M6 Competition — unic.ac.cy.** 2024. <https://www.unic.ac.cy/iiff/research/forecasting/m-competitions/m6/>.
- VASWANI, A. *et al.* Attention is all you need. *Advances in neural information processing systems*, v. 30, 2017.
- WFE. **World Federation of Exchanges.** 2024. <https://www.world-exchanges.org/>. [Accessed 23-10-2024].

WU, S. *et al.* Adversarial sparse transformer for time series forecasting. **Advances in neural information processing systems**, v. 33, p. 17105–17115, 2020.

XING, F. Z.; CAMBRIA, E.; WELSCH, R. E. Natural language based financial forecasting: a survey. **Artificial Intelligence Review**, Springer, v. 50, n. 1, p. 49–73, 2018.

YAHOO. **Yahoo Finance - Stock Market Live, Quotes, Business & Finance News** — finance.yahoo.com. 2024. <https://finance.yahoo.com/>. [Accessed 15-04-2024].

YOSHINAGA, C. E.; CASTRO, F. H. Inteligência artificial: a vanguarda das finanças. **GV-EXECUTIVO**, v. 22, n. 3, 2023.

ZENG, A. *et al.* Are transformers effective for time series forecasting? *In: Proceedings of the AAAI conference on artificial intelligence*. [S.l.: s.n.], 2023. v. 37, n. 9, p. 11121–11128.

ZHOU, H. *et al.* Informer: Beyond efficient transformer for long sequence time-series forecasting. *In: Proceedings of the AAAI conference on artificial intelligence*. [S.l.: s.n.], 2021. v. 35, n. 12, p. 11106–11115.

ZHOU, T. *et al.* Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *In: PMLR. International conference on machine learning*. [S.l.: s.n.], 2022. p. 27268–27286.

ZUCCHI, K. **4 Common Active Trading Strategies** — [investopedia.com](https://www.investopedia.com/articles/active-trading/11/four-types-of-active-traders.asp). 2023. <https://www.investopedia.com/articles/active-trading/11/four-types-of-active-traders.asp>. [Accessed 10-04-2024].