

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO**

Natália Maria Puggina Bianchesi

**Previsão de Séries Temporais Não Lineares Baseada em Redes Neurais
e Sinais de Rastreamento**

Itajubá, dezembro de 2022

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO**

Natália Maria Puggina Bianchesi

**Previsão de Séries Temporais Não Lineares Baseada em Redes Neurais
e Sinais de Rastreamento**

**Tese submetida ao programa de Pós-
Graduação em Engenharia de Produção
como parte dos requisitos para obtenção do
Título de Doutor em Ciências em
Engenharia de Produção.**

Área: Engenharia de Produção

Orientador: Prof. Dr. Antônio Fernando
Branco Costa

Co-orientador: Prof. Dr. Pedro Paulo
Balestrassi

**Dezembro de 2022
Itajubá**

**UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS GRADUAÇÃO EM
ENGENHARIA DE PRODUÇÃO**

Natália Maria Puggina Bianchesi

**Previsão de Séries Temporais Não Lineares Baseada em Redes Neurais
e Sinais de Rastreamento**

Banca examinadora:

Antônio Fernando Branco Costa
Pedro Paulo Balestrassi
Carlos Henrique Pereira Mello
Juliana Helena Daroz Gaudêncio
Claudimar Pereira da Veiga
Wesley Vieira da Silva

**Itajubá
2022**

DEDICATÓRIA

A Deus, ao meu marido Pedro,
aos meus pais Natálio e Lúcia,
aos meus familiares e aos meus
amigos.

AGRADECIMENTOS

A Deus, por sempre estar ao meu lado nos momentos de felicidades e dificuldades.

Ao meu marido, Pedro, pelo apoio incondicional, pelo companheirismo, pela paciência e amor em todos os momentos, e ao nosso filho que está chegando, mas já me dá forças para sempre entregar o melhor de mim.

Aos meus pais, Natálio e Lúcia, pelo incentivo, pelo apoio e por tudo o que já fizeram e ainda fazem por mim.

Sou muito grata aos professores Antônio Fernando Branco Costa e Pedro Paulo Balestrassi pela confiança, orientação e conselhos profissionais.

Aos meus amigos e colegas de doutorado: Cláudia, Estevão e Simone, que contribuíram muito com ideias, sugestões e críticas.

Por fim, agradeço à UNIFEI pelo apoio estrutural à pesquisa brasileira viabilizando a realização deste trabalho e de muitos outros.

EPÍGRAFE

“Feliz aquele que transfere o que sabe e aprende o que ensina”.

Cora Coralina

RESUMO

A previsão de séries temporais não lineares é amplamente utilizada em diversas áreas do conhecimento para fazer boas inferências sobre o futuro e apoiar decisões. Muitos exemplos de séries temporais não lineares incluem observações médicas, registros financeiros e dados meteorológicos. A precisão das previsões é determinada considerando o desempenho do modelo em novos dados que não foram usados no ajuste do modelo e o monitoramento dos erros de previsão é essencial para garantir a precisão da previsão. Portanto, esta tese apresenta uma metodologia de previsão de séries temporais não lineares utilizando Redes Neurais e Sinais de Rastreamento para detectar viés e sua responsividade a mudanças não aleatórias na série temporal. Os conjuntos de dados foram gerados de modo a simular diferentes situações com base em séries temporais não lineares, alterando o erro da série; os conjuntos de dados também foram previstos por Rede Neural Artificial, *Multilayer Perceptron*, e os erros de previsão foram monitorados pelos Sinais de Rastreamento, *Cumulative Sum Tracking Signals*. Diferente de muitos estudos publicados na área, a metodologia de Planejamento de Experimentos foi aplicada para avaliar os sinais de rastreamento com base no número médio de observações. Em seguida, a metodologia foi aplicada em dados baseados em óleo e graxa total e comparada com a aplicação de outras metodologias tradicionais. Os resultados mostraram que a metodologia de previsão proposta é uma forma eficaz de detectar viés no processo quando um erro é introduzido na série temporal não linear devido à média e ao desvio padrão do erro ter um impacto significativo no Número Médio de Observações. Este estudo contribui para uma discussão sobre a metodologia de previsão de séries temporais, uma vez que esta nova técnica pode ser amplamente utilizada em diversas áreas para melhorar a precisão das previsões.

Palavras-chave: Séries temporais não lineares, Previsão de séries temporais, Redes neurais, Sinais de rastreamento, Planejamento de Experimentos.

ABSTRACT

Nonlinear time series forecasting is widely used in several areas to make good inferences about the future and to support decisions. Many examples of nonlinear time series include medical observations, financial recordings, and weather data. The accuracy of forecasts is determined by considering how well a model performs on new data that were not used when fitting the model and the monitoring of forecast errors is essential to ensure forecasting accuracy. Therefore, this thesis presents a nonlinear time series prediction methodology using Neural Networks and Tracking Signals method to detect bias and their responsiveness to non-random changes in the time series. Datasets were generated to simulate different nonlinear time series by changing the error of the series. The datasets were predicted by Artificial Neural Network, Multilayer Perceptron, and the forecast errors were monitored by Cumulative Sum Tracking Signals. Different from many studies published in the area, the statistical methodology of Design of Experiments was applied to evaluate the tracking signals based on Average Run Length. After, the methodology was applied in data based on total oil and grease and it was compared with the application of other traditional methodologies. The results showed that the proposed prediction methodology is an effective way to detect bias in the process when an error is introduced in the nonlinear time series because the mean and the standard deviation of the error have a significant impact on the Average Run Length. This study contributes to a discussion about time series prediction methodology since this new technique could be widely used in several areas to improve forecast accuracy.

Key words: Nonlinear time series, Time series forecasting, Neural networks, Tracking signals, Design of Experiments.

LISTA DE FIGURAS

Figura 1.1 - Citações e publicações ao longo do tempo.	15
Figura 1.2 – Quantidade de publicações por área de conhecimento.	15
Figura 1.3 – O Problema de Pesquisa.	17
Figura 2.1- Modelo matemático de um neurônio.	23
Figura 2.2 – Estrutura de rede MLP.	24
Figura 2.3- Pontos Centrais em um experimento com (a) 2 e (b) 3 fatores.	30
Figura 2.4- Gráficos de representação da metodologia RSM. (a) Superfície de resposta; (b) contorno.	30
Figura 3.1 - Classificação da pesquisa.	31
Figura 3.2 – Procedimento da Pesquisa.	32
Figura 4.1 – Previsor Proposto.	34
Figura 4.2 – Representação de Cohen.	36
Figura 4.3 – Relação da Carta de Controle com a Curva Normal.	37
Figura 4.4 – STAR1 – Previsão de Séries Temporais.	40
Figura 4.5 – BL1 – Previsão de Séries Temporais.	41
Figura 4.6 – Amostras de treinamento.	42
Figura 4.7 – Série temporal original.	42
Figura 4.8 – Série temporal modificada.	43
Figura 4.9 – SATR1 - Gráfico de Pareto dos efeitos de NMO.	46
Figura 4.10 – BL1 - Gráfico de Pareto dos efeitos de NMO.	46
Figura 4.11 – NMA - Gráfico de Pareto dos efeitos de NMO.	47
Figura 4.12 – STAR1 – Gráfico dos Principais Efeitos para NMO.	47
Figura 4.13 – BL1 – Gráfico dos Principais Efeitos para NMO.	48
Figura 4.14 – NMA – Gráfico dos Principais Efeitos para NMO.	48
Figura 4.15 – STAR1 – Gráfico de Interações de NMO.	49
Figura 4.16 – BL1 – Gráfico de Interações de NMO.	49
Figura 4.17 – NMA – Gráfico de Interações de NMO.	49
Figura 4.18 – STAR1 - Gráfico de Superfície de NMO.	51
Figura 4.19 – BL1 - Gráfico de Superfície de NMO.	51
Figura 4.20 – NMA - Gráfico de Superfície de NMO.	51
Figura 4.21 – STAR 1 - Otimização da resposta de NMO.	52
Figura 4.22 – Previsão dados TOG – Amostra 1 a 54.	54
Figura 4.23 – Previsão dados TOG – Amostra 55 a 84.	54
Figura 4.24 – Previsão dados TOG – Amostra 85 a 100.	55
Figura 4.25 – Análise de Tendência TOG – Amostra 1 a 54.	56
Figura 4.26 – Análise de Tendência TOG – Amostra 55 a 84.	56
Figura 4.27 – Previsão dados TOG – Amostra 85 a 100.	57
Figura 4.28 – CUSUM aplicado aos dados TOG.	57
Figura 4.29 – EWMA aplicado aos dados TOG.	58

LISTA DE TABELAS

Tabela 2.1 – Séries Temporais Não Lineares.	21
Tabela 4.1 – Fatores e níveis do DOE.	37
Tabela 4.2 – Matriz de Experimentos.	38
Tabela 4.3 – Resultados de NMO.	43
Tabela 4.4 – STAR1 - Coeficientes estimados 1.	44
Tabela 4.5 – STAR1 - Coeficientes estimados 2.	44
Tabela 4.6 – BL1 - Coeficientes estimados 2.	44
Tabela 4.7 – NMA - Coeficientes estimados 2.	45
Tabela 4.8 – Resultados da Análise ‘ <i>Desirability</i> ’.	52
Tabela 4.9. Comparativo <i>Tracking Signals</i> x CUSUM x EWMA.	58

LISTA DE ABREVIATURAS E SIGLAS

ARL	<i>Averga Run Length</i> - Número Médio de Observações
BL1	<i>Bilinear Time Series</i> – Série temporal Bilinear
DOE	<i>Design of Experiments</i> – Planejamento de experimentos
MLP	<i>Multilayer Perceptron</i> – Perceptron Multicamadas
NMA	<i>Nonlinear Moving Average</i> – Série temporal média móvel não linear
NMO	Número Médio de Observações
RNA	Redes Neurais Artificiais
STAR1	<i>Smooth Transition Autoregressive</i> – Série temporal transição suave autorregressiva
TOG	Total de Óleo e Graxa (petróleo)
TS	<i>Tracking Signals</i> – Sinais de Rastreamento

SUMÁRIO

1. INTRODUÇÃO	13
1.1 Contexto	13
1.2 Relevância	14
1.3 Problema de pesquisa	16
1.4 Justificativa e Contribuição	17
1.5 Objetivos	18
1.6 Estrutura do trabalho	18
2 FUNDAMENTAÇÃO TEÓRICA	19
2.1 Séries Temporais	19
2.1.1 <i>Séries temporais não lineares</i>	20
2.2 Redes Neurais Artificiais	22
2.3 Sinais de Rastreamento	25
2.4 Projeto e Análise de Experimentos	27
3 MÉTODO DE PESQUISA	31
3.1 Classificação da pesquisa	31
3.2 Procedimento da pesquisa	31
4 DESENVOLVIMENTO DO PREVISOR	34
4.1 Método de aplicação.....	34
4.2 Estudo de Modelagem e Simulação	35
4.2.1 <i>Geração de dados sintéticos por DOE</i>	35
4.2.2 <i>Aplicação do previsor de séries temporais</i>	39
4.2.3 <i>Avaliação de performance do previsor</i>	43
4.2.4 <i>Validação da Metodologia</i>	53
5 RESULTADOS	59
6 CONCLUSÕES	59
REFERÊNCIAS BIBLIOGRÁFICAS	62
APÊNDICE A – Pseudocódigo	67
APÊNDICE B – Aplicação do Previsor nas Séries Temporais em MATLAB	68
APÊNDICE C – Aplicação do Previsor nos dados de TOG em MATLAB	81

1. INTRODUÇÃO

No presente capítulo, será apresentado o contexto em que o estudo está inserido, a relevância da pesquisa na literatura, o problema da pesquisa, assim como suas contribuições e objetivos. Por fim, será apresentado a estrutura em que os próximos capítulos do trabalho estão organizados.

1.1 Contexto

A modelagem de Séries Temporais é extremamente importante para fazer boas inferências sobre o futuro, o que fornece uma forte base teórica para processamento de informações e análise de decisões, que tem sido um importante domínio de pesquisa (PANT & KUMAR, 2021). Análise e previsão para séries temporais fornecem um melhor método de suporte à decisão (HU *et al.*, 2020).

As atividades humanas e a natureza produzem séries temporais não lineares todos os dias, como observações médicas, registros financeiros, sinais fisiológicos e dados climáticos (QIAN *et al.*, 2020). Em geral, as séries temporais são encontradas em qualquer domínio da ciência aplicada e da engenharia que envolva medições temporais.

Os problemas de previsão de séries temporais têm recebido grande atenção desde que foi proposto (HU *et al.*, 2020). Muitos exemplos são encontrados na literatura em diversas áreas do conhecimento. Existem trabalhos sobre o uso de séries temporais para previsão do nível médio diário de água de uma estação hidrológica (WANG & LOU, 2019); os efeitos da terapia endócrina adjuvante na qualidade de vida relacionada à saúde (XIAO *et al.*, 2020); cadeias de abastecimento (MIRCETIC, 2021) e abordagens econométricas (MATTA *et al.*, 2021).

Os pesquisadores geralmente procuram o melhor método de previsão que resulte no valor previsto mais aproximado dos dados reais. Alguns estudos foram conduzidos para avaliar métodos de previsão de séries temporais como regressão de séries temporais, média móvel integrada autorregressiva (ARIMA) e suavização exponencial (LIU *et al.*, 2021; MAO & XIAO, 2019), os quais são baseadas em modelos lineares (KUMAR & MURUGAN, 2017).

Todos esses métodos assumem uma relação linear entre os valores passados da variável de previsão e, portanto, padrões não lineares não podem ser capturados por esses modelos (WONG *et al.*, 2010). Os modelos de Rede Neural Artificial (RNA) têm sido

propostos durante os últimos anos para obter resultados precisos de previsão (MATTA *et al.*, 2021), e melhorar as abordagens convencionais lineares e não lineares.

Nesse contexto, abordagens baseadas em RNA para previsão de séries temporais têm produzido resultados convincentes nas últimas décadas para modelos não lineares (GERALD & DIMITRI, 2007; HIPPERT & TAYLOR, 2010; XIAO *et al.*, 2012).

Considerando a característica dinâmica da previsão de séries temporais, não é incomum descobrir que o modelo precisa ser atualizado explicitamente após a passagem de um certo número de períodos de tempo (DENG; JARAIEDI; ISKANDER, 2004).

Portanto, um sistema de monitoramento da previsão desempenha um papel importante na precisão da previsão (MAKRIDAKIS & WHEELWRIGHT, 1989). O monitoramento é importante para determinar se um desvio ocorre na série temporal e para identificar se uma ação corretiva no modelo precisa ser tomada para garantir que o processo de previsão volte ao controle. O monitoramento inadequado dos resultados pode implicar em previsões incertas e decisões incorretas. Então, há uma preocupação crescente em tornar os métodos de monitoramento adequados para identificar o desvio da série temporal o mais rápido possível.

Diferentes abordagens de monitoramento têm sido propostas na área de previsão. Os métodos do sinal de rastreamento (*Tracking Signals*) têm sido usados para verificar o viés dos métodos de previsão (SABETI; AL-SHEBEEB; JARIDI, 2016) e, também, alertar quando há resultados inesperados da previsão (KUMAR & MURUGAN, 2017). Existem várias medidas de sinal de rastreamento, sendo uma das primeiras o sinal de rastreamento de soma cumulativa (CUSUM) proposto por Brown (1959).

Embora o rastreamento de sinais tenha sido uma prática comum na previsão tradicional de séries temporais, esse assunto não tem sido amplamente abordado na previsão de séries temporais com RNA. Alguns estudos utilizaram sinais de rastreamento durante o treinamento da RNA (YU & LAI, 2005; KUMAR & MURUGAN, 2017) para selecionar o melhor modelo, mas não são comumente usados para monitorar da previsão a fim de definir o momento ideal de calibrar o modelo novamente.

1.2 Relevância

A previsão de séries temporais é explorada por vários autores por meio de diversos métodos. Entretanto, ao se tratar de um previsor de séries temporais, encontra-se uma lacuna na literatura. A maioria dos artigos estão limitados a realizar as previsões sem

considerar o monitoramento dos resultados. Os poucos artigos que realizam monitoramento das previsões não utilizam o método de sinais de rastreamento.

Em 16 de agosto de 2021, foi realizada uma análise bibliométrica nas bases de dados Web of Science e Scopus. Tais bases possuem cobertura abrangente de campos interdisciplinares e permitem acesso a periódicos, trabalhos de conferências, seminários, convenções e simpósios, e também possibilitam buscas por tendências e terminologias de publicação.

Ao pesquisar na *Web of Science* trabalhos intitulados com as palavras *time-series*, e *neural network*, foram encontrados 1.712 artigos, como ilustra a Figura 1.1. Observa-se que a partir de 2018, o número de publicações e citações aumentaram significativamente. As principais áreas de aplicação desses métodos podem ser observadas na Figura 1.2.

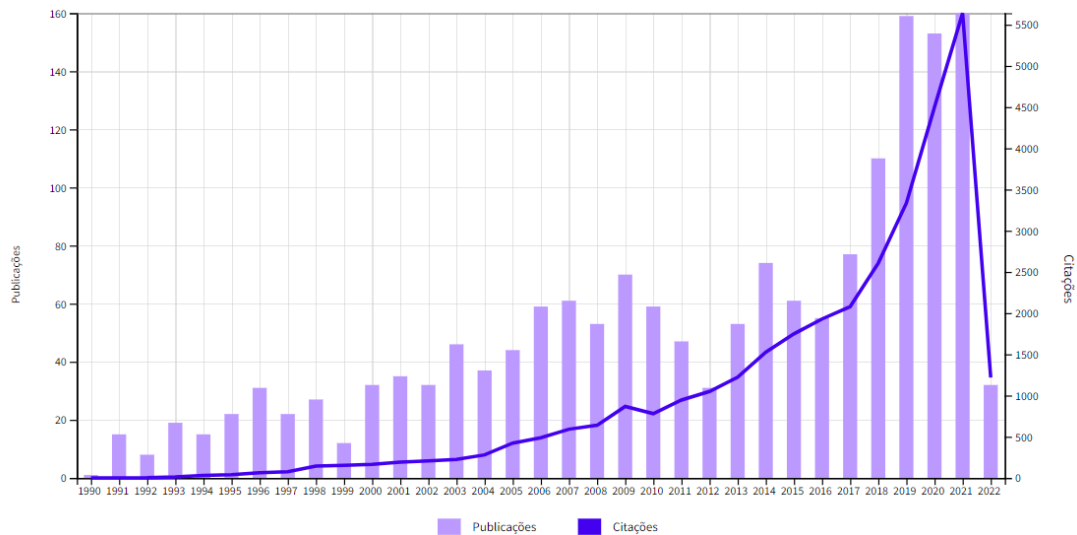


Figura 1.1 - Citações e publicações ao longo do tempo. Fonte: *Web of Science* (2021)

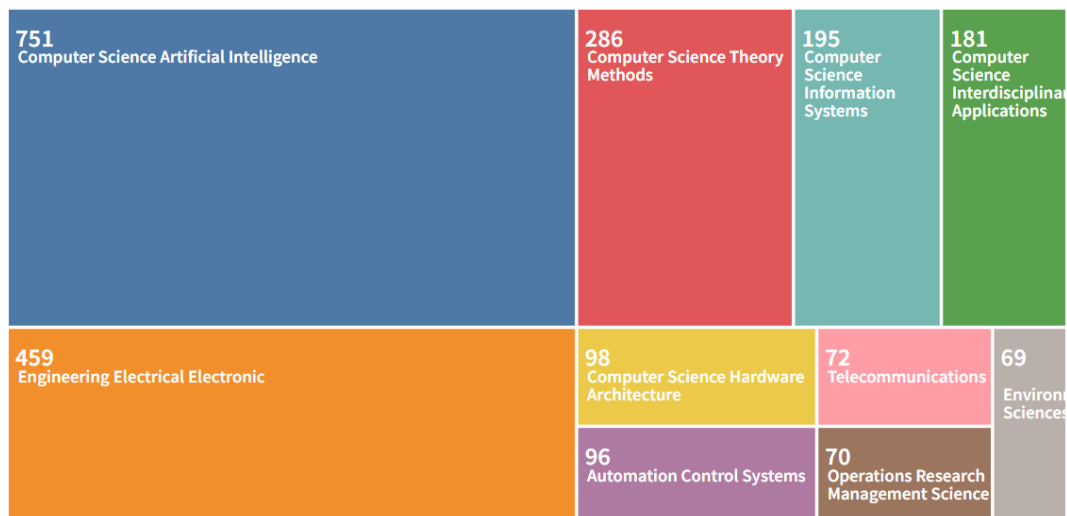


Figura 1.2 – Quantidade de publicações por área de conhecimento. Fonte: *Web of Science* (2021)

A busca foi refeita com os termos *time-series*, e *tracking signals*, e foram encontrados apenas dois artigos. Também foram pesquisados artigos com os termos *neural network*, e *tracking signals*, sendo encontrados outros dois artigos relacionados com os temas. Ainda, ao incluir em todas as buscas o termo *forecaster*, nenhum artigo foi encontrado.

As mesmas buscas foram realizadas na base de dados *Scopus*, sendo que nesta foram encontrados 149 artigos com os termos *time-series*, *neural network* e *forecaster*.

Portanto, percebe-se que a busca por artigos relacionados a previsão de séries temporais por redes neurais está em crescimento. Porém, a noção de que um mecanismo de redes neurais é capaz de resolver todos os problemas automaticamente, a partir de inferências de dados históricos, é um equívoco (DENG; JARAIEDI; ISKANDER, 2004).

Apesar de questões semelhantes terem sido observadas por alguns autores (BERRY e LINOFF, 1997; DEBOECK, 1994; GARDNER, 1985; SUPERVILLE, 2019; BRENCÉ & MASTRANGELO, 2006), muitos poucos relatos de pesquisas são encontrados na literatura que abordaram essa questão ao usar redes neurais artificiais e sinais de rastreamento.

Além disso, os estudos de previsão geralmente consideram um único cenário que é o do problema proposto, impossibilitando a generalização das conclusões para outras situações diferentes da estudada.

1.3 Problema de pesquisa

Este trabalho propõe uma metodologia de previsão de séries temporais não lineares utilizando o método do sinal de rastreamento para detectar viés e sua responsividade a mudanças na série temporal, Figura 1.3. Para isso, dados sintéticos de diferentes séries temporais são gerados por simulação, em que erros modificados são inseridos nas séries temporais de maneira controlada por DOE. As previsões são obtidas utilizando Rede Neural Artificial, *Multilayer Perceptron (MLP)*, as quais são monitoradas por sinais de rastreamento. Se um erro for inserido na série temporal, os Sinais de Rastreamento detectam a mudança e o modelo de previsão é calibrado novamente. A performance do previsor é avaliada pelo número médio de observações (NMO) e utilizando o DOE é possível verificar o comportamento do previsor diante dessa mudança na série temporal.

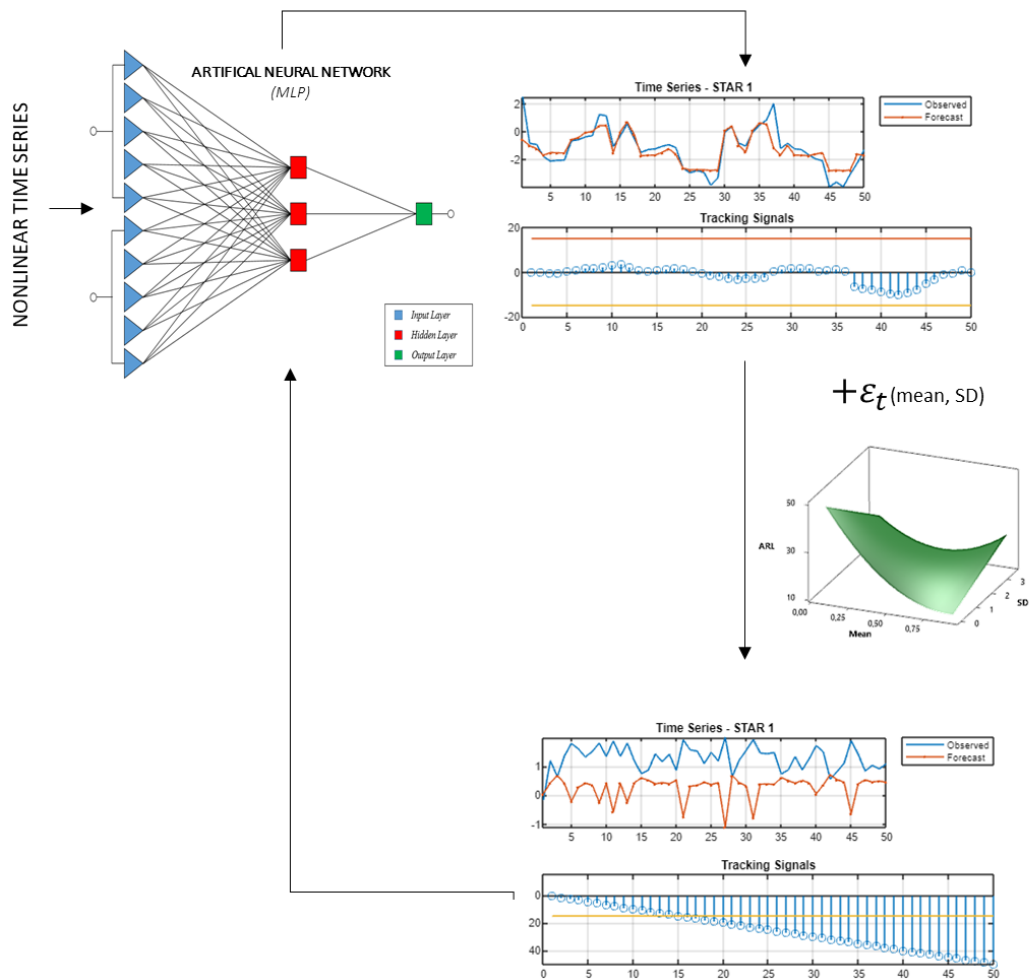


Figura 1.3.3 – O Problema de Pesquisa. Fonte: A autora

1.4 Justificativa e Contribuição

A previsão de séries temporais é amplamente usada em várias áreas para fazer inferências razoáveis sobre o futuro e o monitoramento de erros de previsão é essencial para garantir a precisão da previsão.

Esta tese apresenta uma metodologia de previsão de séries temporais não lineares simples e funcional, uma vez que esta técnica pode ser amplamente utilizada em inúmeras áreas e aplicada em diversas séries temporais não lineares. Na área de Engenharia de Produção, a previsão de séries temporais pode utilizada em todo o sistema de produção, a fim de planejar níveis de produção, aquisições de materiais e serviços, programação da mão-de-obra e da produção, estoques, capacidade a longo prazo e em outras áreas de operações.

Este trabalho difere de outros principalmente no que diz respeito ao método utilizado para avaliar os resultados de NMO. A técnica de planejamento de experimento

(DOE) foi aplicada para gerar conjuntos de dados sintéticos usando um arranjo de experimentos com níveis controlados dos fatores, possibilitando a generalização dos resultados e permitido identificar em quais condições o previsor é eficiente. No entanto, outros trabalhos apresentam uma abordagem usual de NMO em que um número maior de conjuntos de dados diferentes é testado e pode levar a conclusões restritas sem uma análise formal e estatística, o que pode ser alcançado usando DOE.

Portanto, esta metodologia pode ser uma referência para pesquisadores que buscam melhorar a eficiência e eficácia no monitoramento de sistemas de previsão.

1.5 Objetivos

O objetivo geral deste trabalho é desenvolver e avaliar uma metodologia de previsão de séries temporais não lineares usando o método do sinal de rastreamento para detectar viés e sua capacidade de resposta a mudanças não aleatórias na série temporal.

De modo a cumprir com o objetivo geral deste trabalho, têm-se como objetivos específicos: (1) desenvolvimento de um previsor séries temporais através de redes neurais artificiais e sinais de rastreamento; (2) avaliação do previsor quando submetido a mudança na série temporal não linear original utilizando delineamento de experimento; (3) validação do previsor por meio de sua aplicação em dados reais.

1.6 Estrutura do trabalho

No capítulo 1 está descrita a introdução da tese, abordando o contexto, o problema de pesquisa, a justificativa e os objetivos do trabalho. No segundo capítulo contém uma fundamentação teórica sobre séries temporais, redes neurais artificiais, a qual é aprofundada no método MLP, e sinais de rastreamento. A fundamentação teórica também apresenta uma breve explicação sobre planejamento de experimentos, visto que este será empregado para avaliar os resultados de NMO. No terceiro capítulo os procedimentos metodológicos e o previsor de séries temporais proposto são descritos. No quarto capítulo são expostos os resultados e a aplicação da metodologia em um conjunto de dados reais. Por fim, o quinto capítulo contém discussões e conclusões sobre o trabalho e, também, sugestões para trabalhos futuros. Em seguida, são apresentados as Referências e os Apêndices, os quais contêm dados e informações necessárias à compreensão dos resultados obtidos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os aspectos gerais das séries temporais, principalmente as não lineares. Após, será abordado os princípios das redes neurais com foco no método *Multilayer Perceptron* (MLP). Em seguida, será apresentado o método de rastreamento de sinais que será utilizado nesse estudo. Por fim, será abordada a metodologia de Planejamento de Experimentos para a geração de dados simulados.

2.1 Séries Temporais

A série temporal é uma coleção de valores observados sequencialmente ao longo do tempo (ARMSTRONG, 2001).

Existem duas formas de estudar séries temporais: análise e previsão. A análise da série temporal é um método para tentar entender a estrutura que gerou a série e também identificar ciclos e tendência. Nos modelos de séries temporais, os dados históricos da variável a ser prevista são analisados na tentativa de identificar um padrão de dados. Então, supondo que continuará no futuro, esse padrão é extrapolado para produzir previsões (KRISHNAMURTHY, 2006). Uma previsão a partir da série temporal procura construir um modelo matemático a partir do qual seja possível prever valores futuros da série.

Os modelos clássicos de séries temporais podem ser classificados em duas categorias: modelos lineares e modelos não lineares.

Modelos de séries temporais lineares têm sido amplamente usados nos últimos anos. De acordo com Balestrassi *et al.* (2009), uma série temporal estocástica Y_t é dita linear se puder ser escrita como mostrado na Equação 1. Se a série temporal não pode ser bem definida usando este modelo, então é considerada não linear.

$$Y_t = \mu + \sum_{i=0}^{\infty} \varphi_i \alpha_{t-i} \quad (1)$$

sendo, μ é uma constante, φ_i são números reais com $\varphi_0 = 1$ e $\{\alpha_t\}$ é uma sequência de variáveis aleatórias, independentes e identicamente distribuídas. A distribuição de α_t é contínua e $E(\alpha_t) = 0$.

2.1.1 Séries temporais não lineares

Os problemas de série temporal encontrados nas atividades humanas e na natureza são em sua maioria não lineares. Assim, a preocupação com a não linearidade dos dados aumentou consideravelmente, o que resultou no desenvolvimento de modelos não lineares amplamente adotados em aplicações de pesquisa e engenharia.

O objetivo das análises em séries temporais é construir modelos para a série com propósitos determinados, tais como descrever apenas o comportamento da série, investigar o mecanismo gerador da série temporal, fazer previsão de valores futuros e procurar periodicidades relevantes nos dados.

Um modelo clássico para séries temporais supõe que a série pode ser escrita como na Equação 2 (BATISTA, 2009):

$$Z_t = T_t + S_t + A_t \quad (2)$$

em que Z_t é a série temporal, T_t é a tendência, S_t é a sazonalidade e A_t é um componente aleatório.

Segundo Morettin e Tolo (2006), a tendência pode ser entendida como o aumento ou diminuição gradual das observações ao longo do período. A sazonalidade mostra flutuações ocorridas em períodos menores ou iguais a doze meses e a componente aleatória mostra oscilações aleatórias irregulares. A suposição usual é a de que A_t seja uma série puramente aleatória ou um ruído branco independente, isto é, tenha média igual a zero e variância constante.

A modelagem de séries temporais não lineares é aplicada a diversos campos, não se restringindo a apenas uma determinada área de pesquisa como, por exemplo, Guan *et al.* (2018) que aplicaram séries temporais não lineares para caracterizar oscilações termo acústicas em motores de combustíveis sólidos de foguetes e Bonsall *et al.* (2012) que caracterizaram a variabilidade no humor de pacientes que sofriam de bipolaridade por meio de uma abordagem não linear baseada em processos autorregressivos.

Muitos modelos de séries temporais não lineares foram propostos na literatura estatística, como *Threshold Autoregressive Nonlinear Autoregressive*, *Smooth Transition Autoregressive* e *Bilinear Model* (TONG, 1978; AMIRI, 2015; HAMILTON, 1989). Uma teoria mais profunda sobre séries temporais não lineares pode ser encontrada em Priestley (1980). A ideia básica subjacente a esses modelos não lineares é deixar a média

condicional evoluir ao longo do tempo de acordo com alguma função não linear paramétrica simples.

A Tabela 2.1 mostra algumas séries temporais não lineares implementadas e simuladas para o presente estudo. Em cada caso, $\varepsilon_t \sim N(0,1)$ é considerado independente e identicamente distribuído. Esses três modelos de séries temporais são escolhidos para representar uma variedade de problemas que possuem diferentes características de séries temporais. Por exemplo, algumas das séries têm estruturas de correlação autorregressiva (AR) ou média móvel (MA). AR envolve a regressão da variável em seus próprios valores defasados, enquanto que MA envolve a modelagem do termo de erro como uma combinação linear de termos de erro ocorrendo simultaneamente e em vários momentos no passado.

Tabela 2.1 Séries Temporais Não Lineares. Fonte: A autora.

Modelo	Equação
<i>Bilinear (BLI)</i>	$y_t = 0.7y_{t-1}\varepsilon_{t-2} + \varepsilon_t$
<i>Nonlinear moving average (NMA)</i>	$y_t = \varepsilon_t - 0.3\varepsilon_{t-1} + 0.2\varepsilon_{t-2} + 0.4\varepsilon_{t-1}\varepsilon_{t-2} - 0.25\varepsilon_{t-2}^2$
<i>Smooth transition autoregressive (STAR1)</i>	$y_t = 0.8y_{t-1} - 0.8y_{t-1}[1 + \exp(-10y_{t-1})]^{-1} + \varepsilon_t$

Em contraste com o modelo linear tradicional por partes, que permite que as mudanças do modelo ocorram no espaço de tempo, o modelo TAR usa o espaço limite para melhorar a aproximação linear. Diz-se que uma série temporal segue um modelo TAR auto excitante de regime k (SETAR) com uma variável de limiar. Uma crítica ao modelo SETAR é que sua equação média condicional não é contínua. Os limites são os pontos de descontinuidade da função média condicional. Em resposta a esta crítica, modelos TAR suaves (STAR) foram propostos (CHAN e TONG, 1986).

O modelo NMA especifica que a variável de saída depende não linearmente dos valores atuais e passados de um termo estocástico (DE GOOIJER e HYNDMAN, 2006).

O modelo Bilinear é uma extensão natural da não linearidade empregando os termos de segunda ordem na expansão para melhorar a aproximação (TSAY, 2005). Este modelo foi introduzido por Granger e Anderson (1978) e tem sido amplamente estudado.

Embora as propriedades desses modelos tendam a se sobrepor um pouco, cada um é capaz de capturar uma ampla variedade de comportamentos não lineares. Na maioria das séries temporais, entretanto, esse tipo de modelagem é ainda mais complexo devido a algumas características como alta frequência, sazonalidade diária e semanal, efeito

calendário nos finais de semana e feriados, alta volatilidade e presença de *outliers* (BALESTRASSI, 2009).

2.2 Redes Neurais Artificiais

O cérebro humano pode ser entendido como um sistema de processamento de informação altamente complexo, não linear e paralelo. Ele tem a capacidade de reconhecer padrões, por meio da estruturação de seus neurônios, muito mais rápido do que qualquer computador digital existente. Toda essa capacidade deve-se ao fato da grande habilidade que o cérebro possui de criar suas próprias regras a partir das inúmeras experiências vividas (HAYKIN, 2009).

Uma das grandes habilidades no processo de aprendizado humano é a regra de aprendizado empírico. Assim como na vida real, as redes neurais artificiais são um tipo de tecnologia computadorizada que aprende a reter conhecimento continuamente por tentativa e erro (CHANG; TSENG, 2017). Dessa forma, assemelham-se ao cérebro já que o conhecimento é adquirido por um processo de aprendizagem a partir do ambiente no qual elas se encontram inseridas e existem forças de conexão entre os neurônios, chamadas de pesos sinápticos, que são utilizadas para reter o conhecimento (HAYKIN, 2009).

As Redes Neurais Artificiais (RNA) são um dos métodos de previsão mais utilizados e são amplamente aceitas como uma tecnologia que oferece uma forma alternativa de lidar com problemas complexos e mal definidos (YU & LAI, 2005). A principal razão para o aumento da popularidade das RNAs é que esses modelos são capazes de aproximar arbitrariamente quase qualquer função não linear (CHEN & CHEN, 1995; HORNIK, 1993).

Uma RNA é uma estrutura computacional modelada com os princípios dos processos biológicos (ADYA & COLLOPY, 1998). Esse método foi inspirado na estrutura de funcionamento da rede de neurônios do cérebro pelo fato da facilidade e eficácia com que o cérebro realiza tarefas difíceis e complexas e principalmente pela sua capacidade de aprender.

O primeiro modelo de redes neurais foi proposto no trabalho de McCulloch e Pitts (1943), no qual descrevem o cálculo lógico das redes neurais que unificava o estudo de neurofisiologia e de lógica matemática.

O objetivo da RNA não é replicar a operação do sistema biológico, mas fazer o uso do que é conhecido sobre a funcionalidade das redes biológicas para resolver problemas complexos (BASHEER & HAJMEER, 2000).

Um neurônio pode ser entendido como a unidade de processamento de informação para a operação de uma rede neural composta por três elementos básicos: sinapse, somador e função de ativação. Uma simplificação do modelo de neurônio artificial apresentado por Haykin (2009) está representada na Figura 2.1.

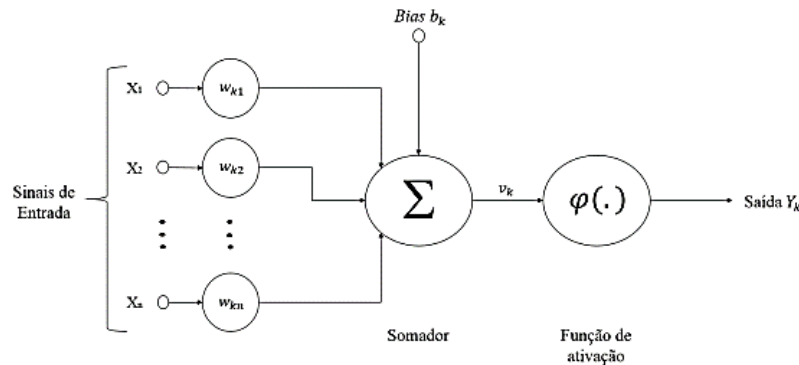


Figura 2.1- Modelo matemático de um neurônio. Fonte: Haykin (2009)

Pode-se descrever matematicamente um neurônio k com as Equações 3 e 4:

$$u_k = \sum_{j=0}^m w_{kj} x_j \quad (3)$$

$$y_k = \varphi(u_k + b_k) \quad (4)$$

Onde: x_1, x_2, \dots, x_m são sinais de entrada, $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos sinápticos do neurônio k , u_k é a saída do combinador linear, b_k é o bias (desvio), $\varphi(.)$ é a função de ativação e y_k é a função de saída do neurônio (HAYKIN, 2009).

O neurônio recebe sinais de entrada (x), que podem ser informações de outros neurônios ou estímulos externos, através de conexões simuladas por pesos (w). O efeito de um sinal é determinado pela multiplicação do valor do sinal pelo peso da conexão correspondente ($x \times w$). Em seguida, é efetuada a soma dos valores de todas as conexões, resultando em um valor u_k . Esse valor é somado a um elemento polarizador b_k (bias), que tem o efeito de aumentar ou diminuir o argumento da função de ativação. O valor resultante é processado por uma função de ativação, que produz um sinal de saída (y).

Do ponto de vista estrutural, as RNAs podem ser classificadas como neurais estáticas ou dinâmicas, segundo Haykin (2009). A principal diferença funcional entre as duas estruturas é a presença ou não de conexões que realimentam os neurônios presentes

na arquitetura da rede. Quando não existe dependência das saídas atuais com seus valores passados, as redes são estáticas. Entre essas estruturas, destacam-se as redes neurais multicamadas (*Multilayer Perceptron – MPL*).

MLP é um dos tipos de rede mais populares (AIZENBERG *et al.*, 2016; OLSON *et al.*, 2012) e, em muitos problemas, parecem oferecer o melhor desempenho possível para descrever uma relação entre variáveis independentes e dependentes (KIALASHAKI & REISEL, 2013). MLP são redes treinadas com algoritmos de aprendizado de retropropagação (ZHAI *et al.*, 2016), e consistem em uma camada de entrada (*input layer*), uma ou mais camadas ocultas (*hidden layers*) e uma camada de saída (*output layers*), conforme mostrado na Figura 2.2. Uma camada oculta é um grupo de neurônios (*neurons*) que possuem uma função específica e são processados como um todo. Resultados teóricos prescrevem que uma MLP com uma camada oculta (MLP de três camadas) é capaz de aproximar qualquer função contínua (HORNIK, 1993).

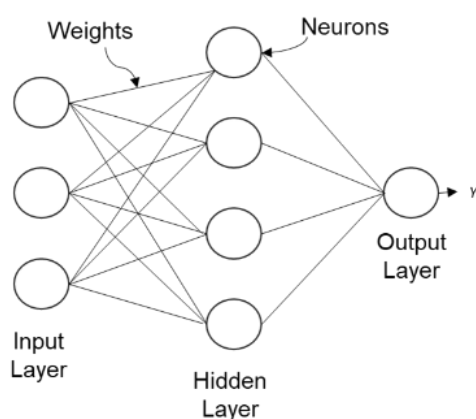


Figura 2.2 – Estrutura de rede MLP. Fonte: Balestrassi *et al.* (2009).

No momento do treinamento, os valores dos pesos (*weights*) são ajustados continuamente de acordo com o resultado obtido do valor real da variável resposta (MO *et al.*, 2007), e o peso e vieses são otimizados com base na minimização da soma dos quadrados da diferença entre a saída desejada e uma saída estimada (BALESTRASSI *et al.*, 2009). O treinamento da rede é realizado até não haver mais alterações significativas nos valores dos pesos sinápticos (HAYKIN, 2009). A taxa de aprendizado representa a taxa na qual os pesos são ajustados. Uma maior taxa de aprendizado permite que a rede convirja mais rapidamente, porém, as chances de uma solução não ótima são maiores.

Portanto, a RNA pode ser amplamente utilizada com o propósito de modelar problemas não lineares. Uma das principais vantagens da RNA é que não é necessário conhecer com antecedência um modelo matemático que represente o conjunto de dados (CHANG &

TSENG, 2017). Assim, a RNA pode descrever processos não lineares com boa precisão (SUN *et al.*, 2017).

2.3 Sinais de Rastreamento

Uma perspectiva importante da previsão é monitorar o processo. Nos campos de previsão e série temporal, sinais de rastreamento (*Tracking Signals*) são usados para monitorar os sistemas de previsão. *Tracking Signals* têm sido aplicados para prever erros e têm se mostrado úteis para determinar se os processos devem continuar ininterruptos ou se a intervenção é necessária para trazer o processo de volta ao controle (KRISHNAMURTHY, 2006).

Tracking Signals, em geral, são razões dos erros de previsão pelo desvio médio absoluto. Se o índice exceder um limite pré especificado, é necessário verificar se o modelo de previsão precisa ser revisado (KRISHNAMURTHY, 2006).

A primeira proposta de *Tracking Signals* foi feita por Brown (1963) e posteriormente analisada por vários pesquisadores, incluindo Trigg (1964) e Gardner (1985).

Um dos *Tracking Signals* mais comum compara a soma cumulativa (CUSUM) dos erros no final de cada período com uma média absoluta não suavizada (MAD) mostrada nas Equações 5 - 9. O sinal de rastreamento CUSUM é apresentado na maioria dos artigos sobre gerenciamento de produção e recomendado pela *Association for Operations Management* (RAVI, 2014).

$$CTS_t = \frac{CUSUM_t}{MAD_t} \quad (5)$$

onde,

$$CUSUM_t = e_t + CUSUM_{t-1} \quad (6)$$

$$e_t = X_t - F_t \quad (7)$$

$$MAD_t = \frac{AD_t}{t} \quad (8)$$

$$AD_t = |e_t| + AD_{t-1} \quad (9)$$

sendo, o erro de previsão, e_t , o valor real da série temporal, X_t , menos a previsão F_t .

O erro cumulativo pode ser positivo ou negativo, então os *Tracking Signals* também podem ser positivos ou negativos. Se o *Tracking Signals* estiver dentro dos limites de

controle, então o modelo de previsão está funcionando corretamente (KUMAR & MURUGAN, 2017).

Os limites de uma carta de controle são determinados pelo conceito de número médio de observações (NMO). NMO é o tempo médio até que um dado exceda os limites de controle, a partir de um ponto no tempo em que foi realizada uma alteração na série (SUN *et al.*, 2017). Portanto, o NMO determina a probabilidade de detectar mudanças na série de dados. O número médio de observações (NMO_0) mede também a probabilidade de alarme falso, definido como erro Tipo I. O alarme falso é a probabilidade de um dado exceder os limites de controle sem que haja alteração na série. Assim, os limites de controle devem ser definidos por simulação para gerar a probabilidade desejada de obter um alarme falso.

Além do Tracking Signals, existem também outras técnicas tradicionais de monitoramento que incluem o Gráfico de Controle CUSUM (cumulative sum) e o Gráfico de Controle EWMA (exponentially weighted moving average), as quais serão aplicadas no estudo a fim de comparação com a metodologia proposta por *Tracking Signals*. Esses gráficos de controles são adaptados a partir do contexto do controle Estatístico de Processos, e vários autores já estudaram a utilização desses gráficos dessas cartas para detecção de viés de previsões.

Segundo Montgomery, Jennings e Kulahci (2011), o gráfico de controle CUSUM é muito eficaz na detecção de mudanças da variável que é monitorada. Seu funcionamento ocorre através da acumulação de desvios dos erros de previsão que estão acima ou abaixo do valor-alvo T desejado. Este valor-alvo desejado possui estatísticas C^+ e C^- que são os chamados CUSUMs superior e inferior. Ainda segundo os autores, estes valores são calculados seguindo as Equações 10 e 11.

$$C_t^+ = \max[0, e_t - (T + K) + C_{t-1}^+] \quad (10)$$

$$C_t^- = \min[0, e_t - (T - K) + C_{t-1}^-] \quad (11)$$

sendo, e_t é o erro de previsão e K um valor de referência. Segundo os autores, normalmente $K = 0.5\sigma_e$, sendo σ_e o desvio padrão dos erros.

A lógica segundo Montgomery, Jennings e Kulahci (2011) é que se os erros começam a cair de um lado do valor-alvo, um dos CUSUMs aumenta em magnitude gerando um sinal de falta de controle. Este sinal indica que a série possui anormalidades e não está satisfatória.

Os limites da carta de controle inferior (LCL) e superior (UCL) são calculados conforme Equações 12 e 13.

$$LCL = -h\sigma_e \quad (12)$$

$$UCL = h\sigma_e \quad (13)$$

sendo, h o número de desvios padrão entre a linha central e os limites de controle, com valor padrão de 4.

Tendo um desempenho similar ao gráfico de controle CUSUM, a carta de EWMA é usada, tipicamente, em observações individuais (LUCAS E SACCUCCI, 1990). Segundo Montgomery (2009), a carta EWMA é eficiente em situações de pequenas mudanças no processo. Segundo Montgomery (2009), a média móvel ponderada é definida como na Equação 14:

$$Z_t = \lambda e_t + (1 - \lambda)Z_{t-1} \quad (14)$$

onde, $0 < \lambda \leq 1$ é uma constante definida como peso.

Os limites da carta de controle inferior (LCL) e superior (UCL) são calculados conforme Equações 15 e 16.

$$LCL = \mu - k\sigma_e \quad (15)$$

$$UCL = \mu + k\sigma_e \quad (16)$$

sendo, k constante com valor padrão igual a 3.

2.4 Projeto e Análise de Experimentos

Montgomery (2009) define Planejamento de Experimentos, que vem do inglês *Design of Experiments* (DOE), como um processo onde se planejam os experimentos para que dados apropriados sejam coletados e depois analisados por métodos estatísticos, resultando em conclusões válidas e objetivas.

DOE é definido também como um teste, ou uma série de testes, em que um conjunto de variáveis de entrada ou fatores são alterados pelo experimentador de maneira controlada a fim de observar e identificar como as respostas desse sistema são afetadas

devido as alterações (STAICULESCU *et al.*, 2005). Assim, é possível compreender quais fatores são significativos e como eles interagem um com o outro.

Para Gomes (2010), as técnicas do DOE têm encontrado uma ampla aplicação nas mais variadas áreas do conhecimento, mostrando-se como um conjunto de ferramentas de grande importância para o desenvolvimento de produtos e processos.

Na simulação, a técnica DOE aumenta a transparência do comportamento do modelo de simulação e a eficácia dos relatórios dos resultados da simulação (LORSCHIED; HEINE & MEYER, 2012). Além disso, permite controlar os fatores que serão utilizados na simulação e apresentar resultados melhores e mais rápidos do que a simulação de tentativa e erro.

Neste trabalho, o DOE será usado para simular conjuntos de dados de séries temporais não lineares por meio da inserção de erros na série com média e desvio padrão controlados, a fim de simular diferentes modificações em que a série temporal pode ser submetida.

Para isso, será utilizado o método Metodologia de Superfície de Resposta (RSM), o qual é uma classe de DOE amplamente utilizada devido à sua simplicidade e eficácia para minimizar o número de experimentos projetados (MONTGOMERY, 2009). O objetivo do RSM é explorar a relação entre a resposta e os fatores envolvidos em um experimento (MONTGOMERY, 2009). Os fatores e níveis aplicados no planejamento de experimentos estão descritos no capítulo 4.2.1.

A metodologia de superfície de resposta, MSR, é uma coleção de técnicas matemáticas e estatísticas úteis para a modelagem e análise de um processo ou sistema no qual a resposta de interesse é influenciada por diversas variáveis (MYERS; MONTGOMERY; ANDERSON-COOK, 1995).

De acordo com Mason, Gunst e Hess (2003) existem quatro razões importantes para utilização da metodologia de superfície de resposta.

- A função de resposta é caracterizada em uma região de interesse;
- Inferências estatísticas podem ser feitas sobre a sensibilidade da resposta aos fatores de interesse;
- Níveis de fatores podem ser determinados para que a variável da resposta seja otimizada (por exemplo, máximo ou mínimo);
- Níveis de fatores podem ser determinados a fim de simultaneamente, otimizar diversas respostas; se a otimização simultânea não for possível, compensações são facilmente perceptíveis.

O modelo de aproximação baseia-se nos dados observados do sistema ou processo, constituindo-se no modelo empírico. Em geral, assume-se, inicialmente, um modelo de regressão polinomial de baixa ordem (primeira ordem), em que a relação entre a(s) variável(eis) x_i e a resposta y pode ser escrita como na Equação 17:

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \varepsilon \quad (17)$$

Caso haja interação entre os fatores, pode-se aplicar o modelo segundo Equação 18:

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{j=2}^k \sum_{i=1}^{j-1} \beta_{ij} x_i x_j + \varepsilon \quad (18)$$

o qual introduzirá curvatura a superfície de resposta.

Entretanto, dependendo do caso em estudo, a curvatura da superfície de resposta pode ser suficientemente forte ao ponto do modelo de primeira ordem ser inadequado, mesmo quando há inclusão do termo de interação. Diante desta situação, emprega-se um modelo de segundo grau (quadrático), conforme Equação 19:

$$y = \beta_0 + \sum_{i=1}^k \beta_i x_i + \sum_{i=1}^k \beta_{ii} x_i^2 + \sum_{i=1}^{k-1} \sum_{j=i+1}^k \beta_{ij} x_i x_j + \varepsilon \quad (19)$$

onde β_0 , β_i , β_{ii} e β_{ij} são parâmetros dos polinômios estimados pelo método dos mínimos quadrados.

Dentre os experimentos mais utilizados na metodologia da superfície de resposta estão os experimentos de BoxBehnken (*BBD – Box-Behnken Design*) e compostos centrais (*CCD – Central Composite Design*). Neste trabalho será utilizado o método CDD, pois este podem ajustar um modelo quadrático completo. Experimentos centrais compostos são experimentos fatoriais completos ou fracionados com pontos centrais, ampliados com um grupo de pontos axiais (também chamados pontos estrela) que permitem estimar curvas, Figura 2.3.

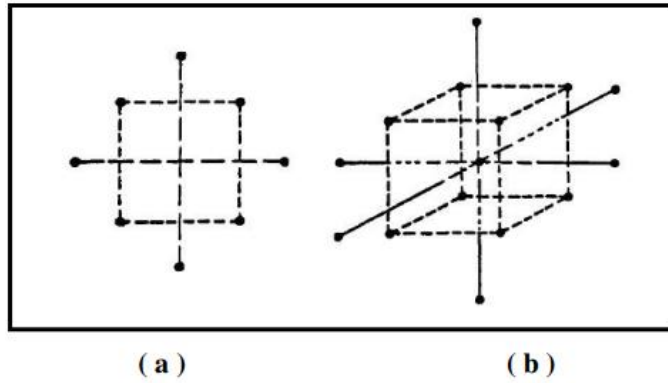


Figura 2.3- Pontos Centrais em um experimento com (a) 2 e (b) 3 fatores. Fonte: Draper e Lin (1996).

Os resultados podem ser apresentados em forma de gráficos de superfície de resposta que apresentam uma superfície tridimensional sobre um plano com dois parâmetros ou gráficos de contorno que são bidimensionais em relação a um parâmetro, conforme Figura 2.4.

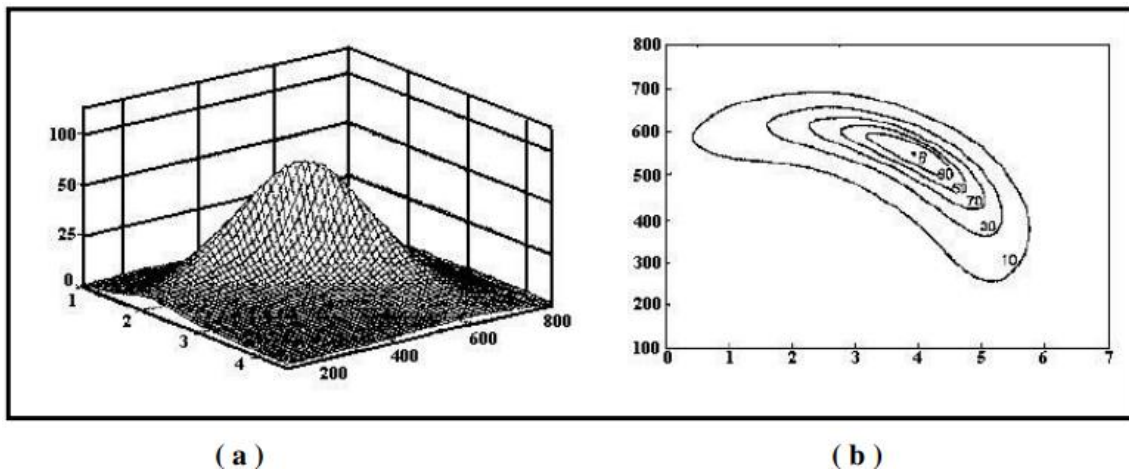


Figura 2.4- Gráficos de representação da metodologia RSM. (a) Superfície de resposta; (b) contorno. Fonte: Myers; Montgomery; Anderson-Cook (1995).

3 MÉTODO DE PESQUISA

Este capítulo tem o objetivo apresentar a classificação da pesquisa e procedimento metodológico que será adotado no estudo.

3.1 Classificação da pesquisa

A classificação da pesquisa, apresentada na Figura 3.1, baseou-se em Miguel *et al.* (2014), que sugerem a classificação da pesquisa científica segundo os critérios: natureza, objetivos, abordagem e método.

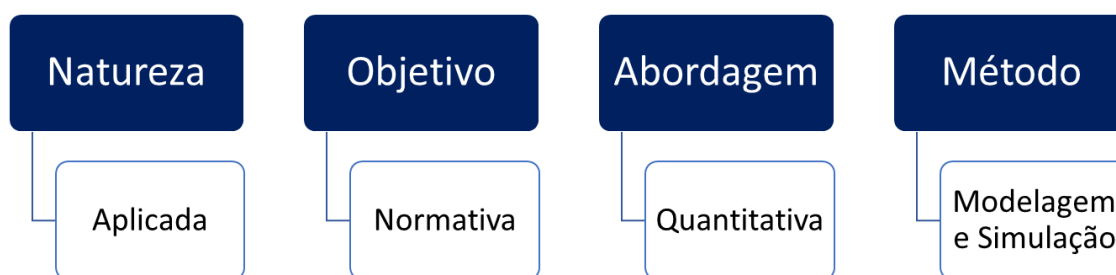


Figura 3.1 - Classificação da pesquisa. Fonte: Bertrand e Fransoo (2002).

Esta é uma pesquisa de natureza aplicada, pois tem como foco a solução de problemas reais com a utilização da metodologia de previsão proposta (APOLINÁRIO, 2006). O objetivo da pesquisa é classificado como normativo, pois procura desenvolver novas maneiras, estratégias e ações para resolver determinado problema. A pesquisa normativa está primariamente interessada no desenvolvimento de políticas, estratégias e ações para aperfeiçoar os resultados disponíveis na literatura existente, para encontrar uma solução ótima para novas definições de problemas ou para comparar várias estratégias relativas a um problema específico (BERTRAND & FRANSOO, 2002). A abordagem empregada na pesquisa é quantitativa, e o método utilizado é o de modelagem e simulação, pois se deseja prever o efeito de mudanças em um sistema e avaliar seu desempenho ou comportamento por meio das etapas: Conceitualização, Modelagem, Modelo de Solução, Implementação, *Feedback* e Validação (BERTRAND & FRANSOO, 2002).

3.2 Procedimento da pesquisa

O presente estudo, baseado na metodologia de modelagem e simulação, utilizou como procedimento de pesquisa o modelo proposto na Figura 3.2.

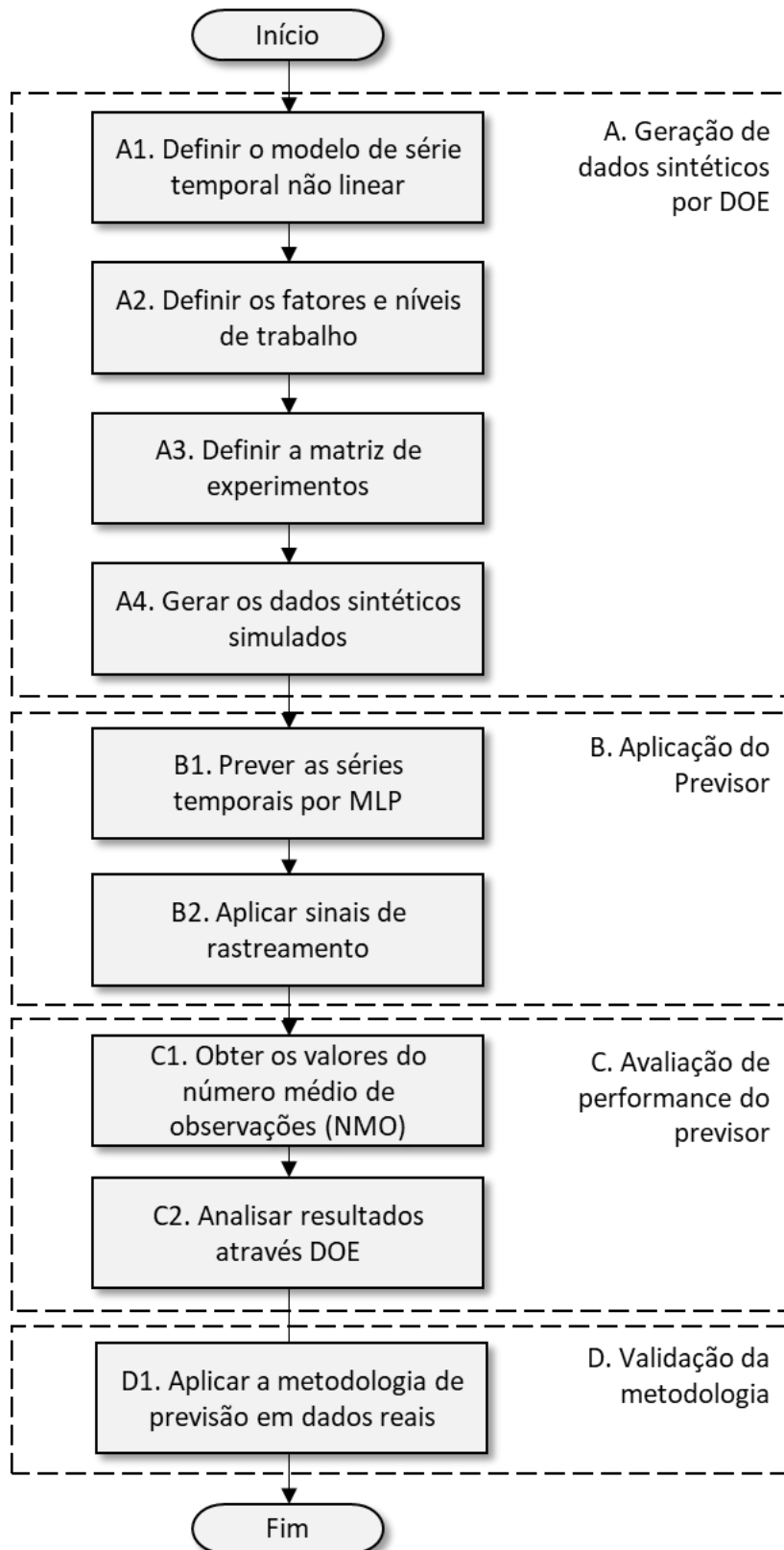


Figura 3.22 – Procedimento da Pesquisa. Fonte: A autora.

Conceitualização:

A1: A geração de dados sintéticos consiste, inicialmente, na definição do modelo de série temporal e especificação dos parâmetros de sua equação.

A2: Em seguida, é necessário definir os parâmetros da equação (fatores e níveis) que serão modificados para simular diferentes condições de mudança ao longo da série temporal.

Modelagem:

A3: Após, deve ser definida a matriz de experimentos que será utilizada no planejamento de experimentos. Com isso, os experimentos serão gerados conforme combinação dos fatores definidos na etapa anterior. A utilização da ferramenta DOE permite que esses dados englobem diversas alterações na série temporal, de modo a permitir a generalização dos cenários trabalhados.

A4: Então, deverão ser gerados os bancos de dados sintéticos de acordo com cada experimento.

Modelo de Solução e Implementação:

B1: Essa etapa consiste na aplicação do previsor. Primeiramente, é necessário realizar o modelo no software estatístico e aplicá-lo em para prever a série temporal original utilizando o método MLP. Após definido o modelo de previsão, as demais séries temporais modificadas serão previstas também pelo modelo gerado;

B2: Em seguida, deverá ser aplicado o sinal de rastreamento e definido os limites de controle para a série temporal original. Após, as demais séries modificadas deverão ser inseridas na mesma carta de controle.

Feedback:

C1: Nesse momento, deverá ser calculado o número médio de observações (NMO) que será a métrica para determinar a probabilidade de o sinal de rastreamento detectar as mudanças simuladas na série temporal.

C2: Nesta etapa, os valores de NMO deverão ser atribuídos como resposta do DOE e deve realizar a ANOVA para comparar os resultados encontrados ao utilizar os sinais de rastreamento para monitorar a série temporal com modificações ao longo do tempo. Deve-se realizar a análise de efeitos principais e das interações para determinar a eficiência dos sinais de rastreamento para detectar a alteração nos parâmetros da série temporal. Além disso, é possível utilizar algoritmos de otimização para identificar em que circunstâncias o previsor possui melhor desempenho.

Validação:

D1: Na última etapa, a metodologia é aplicada em dados reais para validar os seus passos e comprovar os resultados.

4 DESENVOLVIMENTO DO PREVISOR

Este capítulo tem o objetivo de apresentar o previsor proposto, realizar as etapas do procedimento de pesquisa, validar a metodologia através da aplicação prática do previsor e apresentar discussões sobre os resultados.

4.1 Método de aplicação

O previsor proposto tem como objetivo realizar a previsão de séries temporais não lineares e monitorar os erros para detectar viés no modelo de séries temporais garantindo a precisão da previsão. A aplicação do previsor, mencionada na Figura 3.2 como item B, é detalhado em cinco etapas, conforme Figura 4.1.

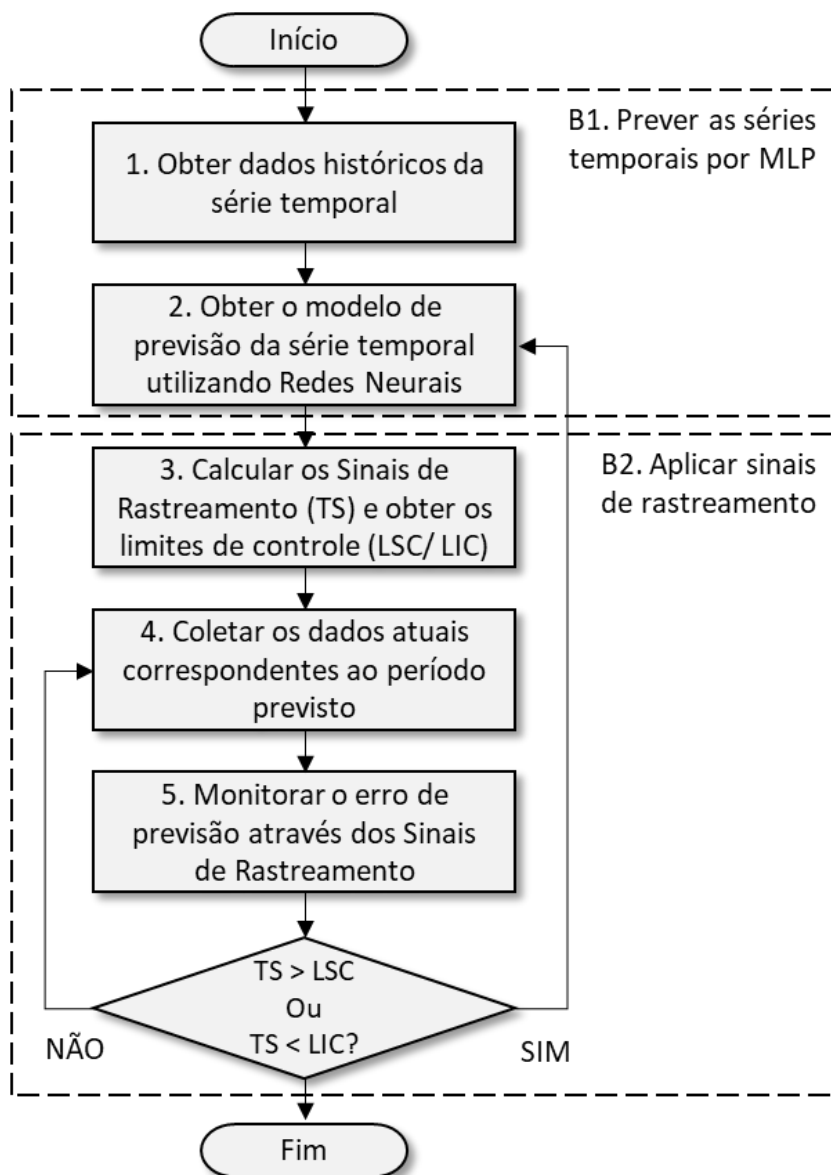


Figura 4.1 – Previsor Proposto. Fonte: A autora

1) O primeiro passo é definir a variável que será controlada e obter os dados históricos das séries temporais;

2) Considerando os dados selecionados, é possível realizar a previsão de séries temporais utilizando MLP;

3) Obtendo os valores real e previsto, é possível calcular o erro de previsão e obter o *Tracking Singals* (TS). Assim, o limite de controle superior (UCL) e o limite de controle inferior (LCL) da carta de controle também são definidos;

4) Os dados reais correspondentes aos períodos previstos no modelo de previsão são coletados e comparados com os valores previstos;

5) Em seguida, os valores de TS são calculados e os erros de previsão são monitorados pela carta de controle. Caso o TS ultrapasse os limites de controle, o MLP é executado novamente, caso contrário retorna ao passo 4.

O previsor também pode ser descrito por pseudocódigo (Apêndice A), associado às etapas do fluxograma na Figura 4.1.

4.2 Estudo de Modelagem e Simulação

Nesta seção, o estudo de simulação para o previsor de séries temporais não lineares é examinado para testar diferentes cenários e generalizar os resultados. Primeiramente, é apresentado o DOE aplicado na simulação de conjuntos de dados de séries temporais não lineares. Os conjuntos de dados contêm uma parte da série temporal original e uma parte na qual o erro de série (ϵ_t) é modificado. Em seguida, a previsão de séries temporais e os sinais de rastreamento são aplicados. Após, os resultados de NMO dos sinais de rastreamento são analisados pelo DOE e é possível determinar o quanto o erro da série (ϵ_t) pode ser modificado para que os sinais de rastreamento ainda sejam eficientes na detecção do deslocamento de série. Por fim, a metodologia é validada através da sua aplicação em dados reais de óleo e graxa totais.

4.2.1 Geração de dados sintéticos por DOE

No que se refere à geração de conjuntos de dados de séries temporais distintos do modelo original, o erro da série presente no modelo de série temporal foi modificado alterando sua média e desvio padrão de forma controlada pela técnica DOE. erro de série original (ϵ_t) assume uma distribuição normal $N(0;1)$.

Os níveis assumidos para modificar a média do erro de série (ϵ_t) foram definidos considerando o conceito de tamanho do efeito. O tamanho do efeito (TDE) é definido

como a estimativa da magnitude da relação entre as variáveis ou a diferença entre duas amostras (ROSENTHAL, 1994). Para calcula o TDE usa-se normalmente a medida de “*d de Cohen*”, Equações 20 e 21, (COHEN, 1988).

$$d = \frac{x_1 - x_2}{s_p} \quad (20)$$

onde,

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \quad (21)$$

Cohen (1988) classificou o tamanho do efeito como pequeno ($d = 0,2$) quando a diferença entre duas amostras é difícil de ver a ‘olho nu’, e como grande ($d = 0,8$) quando a diferença entre duas amostras é evidente para ver com o ‘olho nu’.

Segundo Cohen (1988), para $d = 0,2$, 58% do erro da série original estará acima da média do erro da série modificada; 92% das duas séries de erro serão sobrepostas, e há 56% de chance de que um elemento escolhido aleatoriamente do erro da série original seja maior que um elemento escolhido aleatoriamente do erro da série modificada. Para $d = 0,8$, 79% do erro da série original estará acima da média do erro da série temporal modificada; 69% das duas séries de erro serão sobrepostas e há 71% de chance de que um elemento escolhido aleatoriamente do erro da série original seja maior que um elemento escolhido aleatoriamente do erro da série modificada. Isso está representado na Figura 4.2.

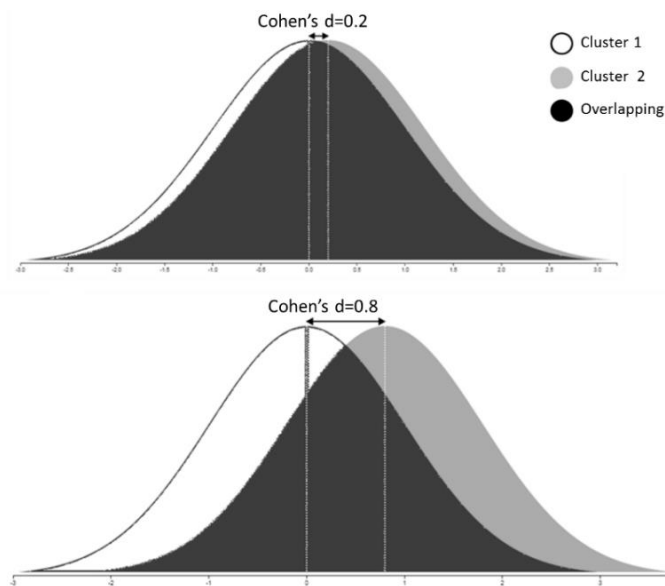


Figura 4.2 – Representação de Cohen. Fonte: Bianchesi *et al.* (2019).

O desvio padrão do erro de série (ϵ_t) foi simulado entre 0,5 e 3,0, de acordo com o conceito de normalidade dos dados. A curva normal apresenta 68,26% de todos os dados dentro de mais ou menos um desvio padrão da média, 95,44% de todos os dados dentro de mais ou menos dois desvios padrão da média e 99,73% dos dados dentro de mais ou menos três desvios padrão da média (MONTGOMERY, 2009), conforme mostrado na Figura 4.3. Desse modo, utilizou-se um desvio padrão de 0,5 para simular uma pequena dispersão da média e um desvio padrão de 3,0 em que a média pode variar nos limites da curva normal.

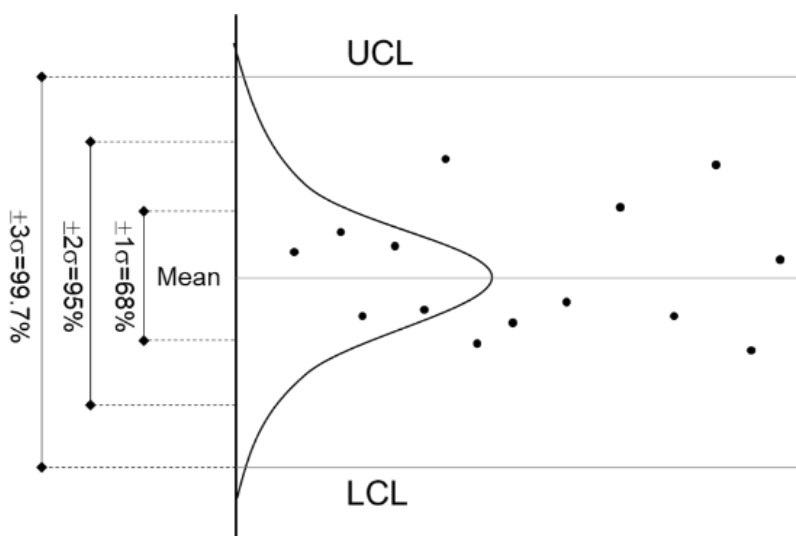


Figura 4.3 – Relação da Carta de Controle com a Curva Normal. Fonte: Montgomery (2009).

Assim, um resumo dos fatores DOE e seus níveis são detalhados na Tabela 4.1.

Tabela 4.1 – Fatores e níveis do DOE. Fonte: A autora.

	Média (Mean)	Desvio Padrão (SD)
(-1)	0.2	0.5
(+1)	0.8	3.0

O RMS (*Response Surface Methodology*) é uma classe de DOE amplamente utilizada devido à sua simplicidade e eficácia para projetar os experimentos e minimizar o número de experimentos para um número específico de fatores e seus níveis (MONTGOMERY, 2009). O objetivo do RSM é explorar a relação entre a resposta e os fatores estudados envolvidos em um experimento (AMDOUN *et al.*, 2018). O modelo matemático do RSM é uma equação polinomial de segunda ordem, cuja vantagem é a fácil estimação e boa acuracidade da resposta (CUI *et al.*, 2012).

Um experimento central composto consiste de uma porção do "cubo" formada pelos pontos do experimento a partir de um experimento fatorial de 2^k ; pontos axiais de 2^k e pontos centrais (em que k é o número de fatores). Os pontos centrais são ensaios experimentais em que todos os fatores são definidos a meio caminho entre os níveis baixo (-1) e alto (+1).

Considerando os fatores e níveis da Tabela 4.1, construiu-se a matriz de DOE, Tabela 4.2, usando o software estatístico MINITAB®. Além disso, foram utilizados 5 pontos centrais de modo a formar blocos ortogonais, e foi adicionada uma réplica a cada experimento a fim de obter parâmetros estatísticos válidos na análise de variância. Desse modo, a matriz de RSM resultou em 26 experimentos. Neste caso, não foi necessário randomizar os experimentos porque os mesmos são usados apenas como simulação de dados.

Tabela 4.2 – Matriz de Experimentos. Fonte: A autora.

Ordem	Réplicas	Média	Desvio Padrão
1	1	0.20	0.50
2	1	0.80	0.50
3	1	0.20	3.00
4	1	0.80	3.00
5	1	0.08	1.75
6	1	0.92	1.75
7	1	0.50	0.02
8	1	0.50	3.52
9	1	0.50	1.75
10	1	0.50	1.75
11	1	0.50	1.75
12	1	0.50	1.75
13	1	0.50	1.75
14	2	0.20	0.50
15	2	0.80	0.50
16	2	0.20	3.00
17	2	0.80	3.00
18	2	0.08	1.75
19	2	0.92	1.75
20	2	0.50	0.02
21	2	0.50	3.52
22	2	0.50	1.75
23	2	0.50	1.75
24	2	0.50	1.75
25	2	0.50	1.75
26	2	0.50	1.75

A matriz de projeto é um guia para indicar a combinação de média e desvio padrão para gerar o erro de série para conjuntos de dados que serão usados para a simulação de série temporal. Portanto, de acordo com cada experimento da matriz, gerou-se 26 conjuntos de dados contendo os diferentes erros de série. Por exemplo, o ‘banco de dados1’ é composto por 100 amostras da série original e 50 amostras da série com o erro modificado com distribuição normal $N(20, 0.5)$. No total, foram geradas 100 amostras de cada 26 conjuntos de dados para cada série temporal.

4.2.2 Aplicação do previsor de séries temporais

Após a obtenção dos conjuntos de dados, o MLP foi implementado em cada modelo de série temporal usando o software Matlab®, Apêndice B. Para cada série temporal original, 50 amostras foram treinadas usando 80% de observações aleatórias do conjunto de dados original e 10% para validação. Assim, os 10% restantes das observações foram alocados para o conjunto de testes. No processo de treinamento, usamos o ‘*seed for sampling*’ igual a um número constante para obter sempre as mesmas amostras aleatórias. A rede neural utilizada foi o perceptron de três camadas por apresentar excelente desempenho ao lidar com conjuntos de dados não lineares (KIALASHAKI & REISEL, 2013), e resultados teóricos prescrevem que, conforme Teorema da Aproximação Universal, uma MLP com uma camada oculta é capaz de aproximar qualquer função contínua (MO *et al.*, 2017). O número de unidades ocultas foi definido como 10, a função de treinamento utilizada foi a Regularização Bayesiana e a taxa de aprendizado foi definida como 0,01.

O número de ‘*time steps*’ utilizados como entrada foram definidos como 1 ou 2, de acordo com a sazonalidade de defasagem do modelo. O número de ‘*steps ahead*’, ou horizonte de previsão, representam o número de etapas à frente dos valores de entrada defasados em que a saída prevista se encontra. Neste caso, devido ao tamanho pequeno da série temporal sintética e considerando a propagação do erro ao longo da previsão dos passos, foi utilizado ‘*steps ahead*’ igual a um. A saída da rede pode ser combinada com valores de entrada anteriores, passos únicos deslocados e previsões repetidas feitas. Como o tempo de execução depende principalmente do erro mínimo a ser alcançado e esse erro não é linear, não é correto dizer que prever dois passos à frente dobra o tempo de execução de prever um passo à frente (BALESTRASSI, 2009).

Após a implementação do MLP, outras 50 amostras da série temporal original foram previstas e os sinais de rastreamento foram calculados pela Equação 4. Os valores

iniciais de CUSUM (Equação 5) foram definidos como zero, conforme sugerido por Gardner (1985) e McClain (1988). Em seguida, os limites de controle (LC) para os sinais de rastreamento foram determinados com base em simulação para detectar a probabilidade desejada de obter um alarme falso, definido como erro Tipo I (GARDNER, 1985). O erro tipo I é a probabilidade de um sinal de rastreamento exceder os limites de controle quando a série temporal não apresenta alterações. Neste trabalho, a simulação foi realizada assumindo uma taxa de falso alarme desejado entre 0,01 e 0,0105 considerando um intervalo de confiança (IC) de 99%, o que representa nesse estudo um NMO_0 entre 100 e 95 amostras da série original, ou seja, como serão adicionadas 100 amostras modificadas, o desejado é que não ocorra falso alarme.

Em seguida, a parte modificada da série foi implementada no modelo de previsão e 50 amostras foram previstas. Em seguida, os sinais de rastreamento para a série temporal modificada também foram calculados.

A Figura 4.4 mostra um conjunto de dados gerado e a previsão de séries temporais usando redes neurais (ANN) para o modelo STAR1 e a Figura 4.5 mostra um exemplo para o modelo Bilinear. Pode-se notar que, em ambos os casos, a partir da amostra 100, em que a série temporal com erro (ε_t) modificado foi inserida, o modelo de previsão não se ajusta bem aos dados.

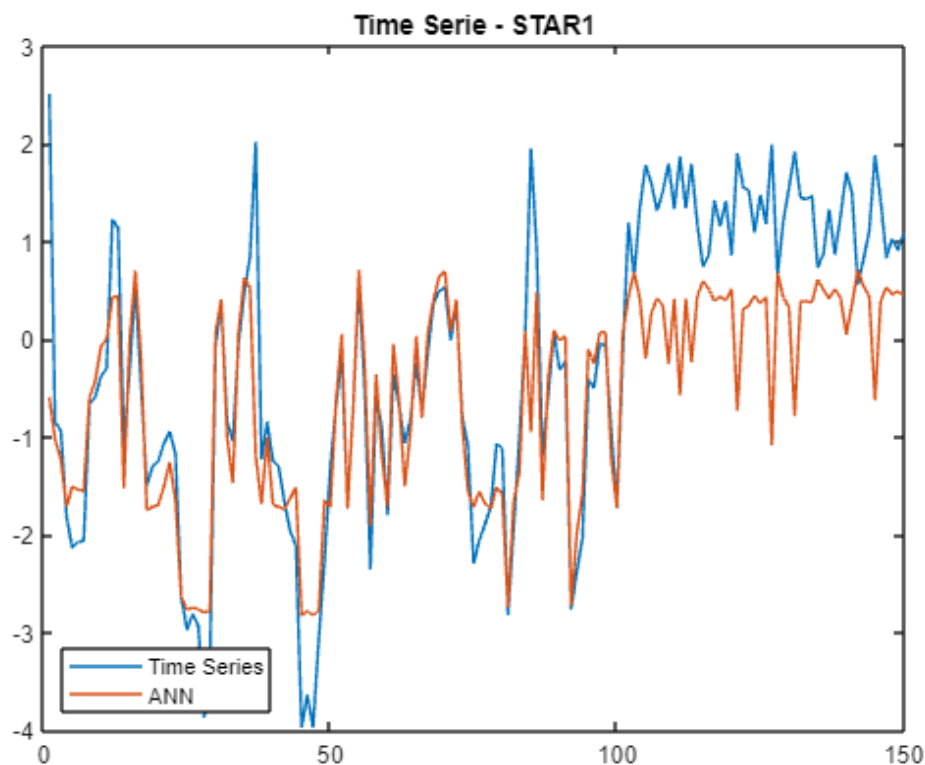


Figura 4.4 – STAR1 – Previsão de Séries Temporais. Fonte: A autora.

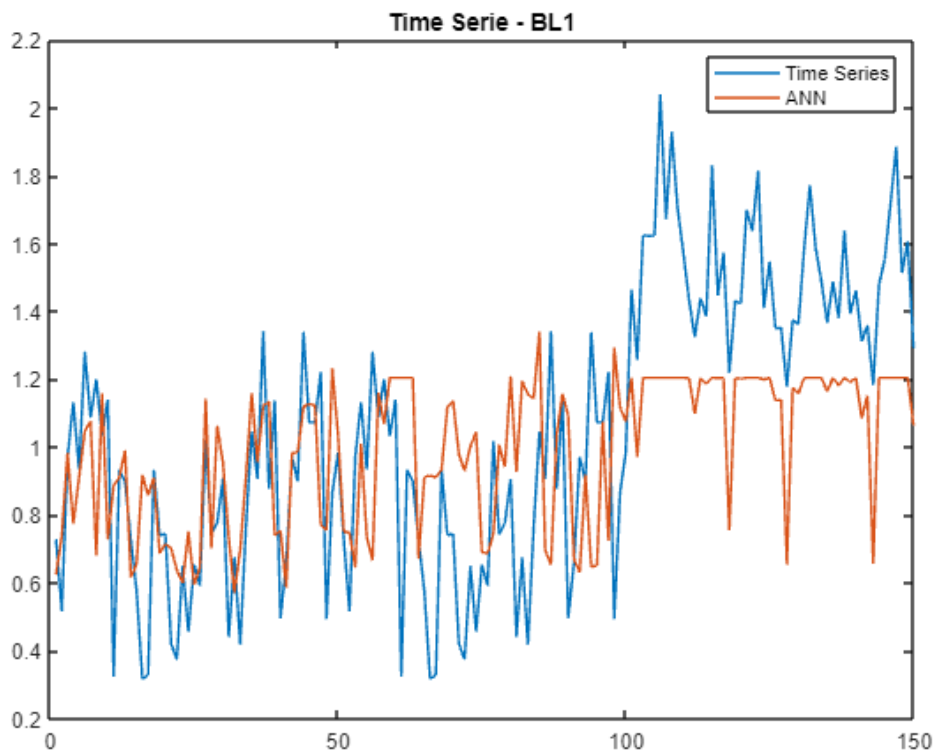


Figura 4.5 – BL1 – Previsão de Séries Temporais. Fonte: A autora.

Em seguida, também foram calculados os sinais de rastreamento para as séries temporais modificadas. As Figuras 4.6, 4.7 e 4.8 mostram um exemplo da série STAR e da previsão de ANN para as amostras de treinamento, a série original e a série modificada, respectivamente.

O erro de previsão (e_t) também é apresentado nas Figuras 4.6, 4.7 e 4.8. Os erros de previsão (e_t) das amostras de treinamento e da série original mostram-se aleatórios, enquanto que o erro de previsão (e_t) da série modificada apresenta-se com tendência e valores maiores. Além disso, os sinais de rastreamento são representados e é possível notar que na Figura 4.7 estes ultrapassam os limites de controle rapidamente.

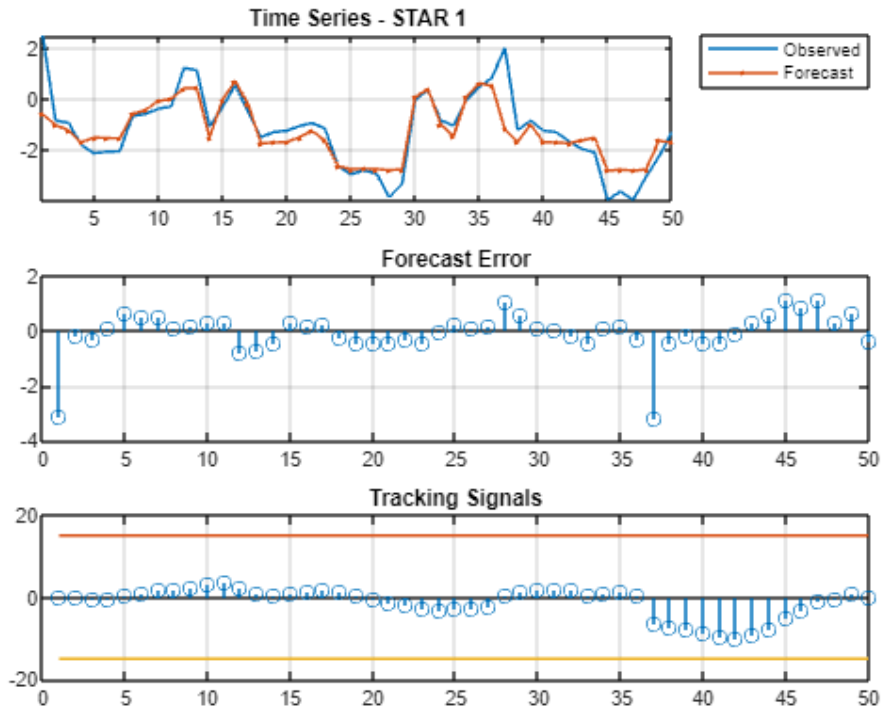


Figura 4.6 – Amostras de treinamento. Fonte: A autora.

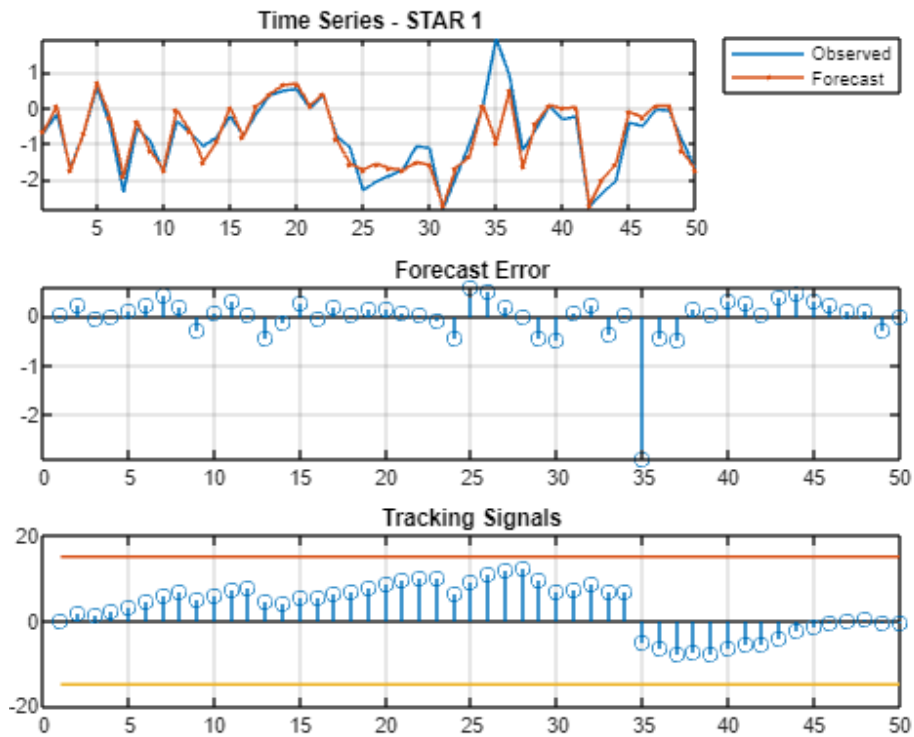


Figura 4.7 – Série temporal original. Fonte: A autora.

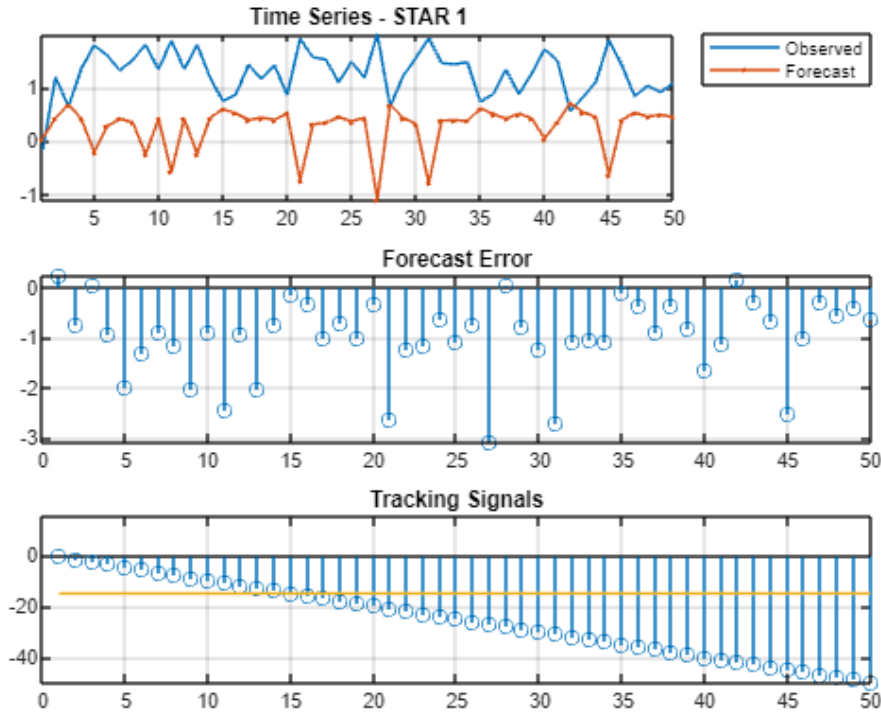


Figura 4.8 – Série temporal modificada. Fonte: A autora.

4.2.3 Avaliação de performance do previsor

A partir dos sinais de rastreamento, o NMO foi obtido para cada conjunto de dados e os resultados médios estão na Tabela 4.3. Após, o Método dos Mínimos Quadrados Ordinários (OLS) e Análise de Variância (ANOVA) foram aplicados, Tabela 4.4.

Tabela 4.3 – Resultados de NMO. Fonte: A autora.

Ordem	NMO			Ordem	NMO		
	STAR1	BL1	NMA		STAR1	BL1	NMA
1	35.76	32.26	25.59	14	47.94	31.25	26.38
2	18.15	16.11	20.56	15	28.55	16.20	20.63
3	17.76	16.00	17.12	16	18.63	16.00	17.46
4	16.94	16.00	29.65	17	16.94	16.00	30.34
5	23.84	16.95	18.21	18	41.83	17.82	18.49
6	17.04	16.00	17.17	19	17.61	16.00	16.97
7	17.66	18.74	17.83	20	16.23	18.53	18.03
8	17.07	16.00	19.90	21	16.83	16.00	20.05
9	17.60	16.01	19.90	22	18.27	16.00	20.05
10	17.60	16.01	19.90	23	18.27	16.00	20.05
11	17.60	16.01	19.90	24	18.27	16.00	20.05
12	17.60	16.01	19.90	25	18.27	16.00	20.05
13	17.60	16.01	19.90	26	18.27	16.00	20.05

Tabela 4.4 – STAR1 - Coeficientes estimados 1. Fonte: A Autora.

Termo	Coef	EP de		Valor-T	Valor-P	VIF
		Coef	Coef			
Constante	17,93	1,71		10,47	0,000	
Mean	-5,21	1,35		-3,85	0,001	1,00
SD	-3,76	1,35		-2,78	0,012	1,00
Mean*Mean	4,59	1,45		3,16	0,005	1,02
SD*SD	0,52	1,45		0,36	0,722	1,02
Mean*SD	4,31	1,91		2,25	0,036	1,00

Os resultados indicam que apenas o termo ‘SD*SD’ (interação entre o desvio padrão) não é significativo (valor-p >0,05). Portanto, este termo foi retirado do modelo e novos coeficientes foram estimados, conforme Tabela 4.5, sendo todos os termos significativos (valor-p <0,05).

Tabela 4.5 – STAR1 - Coeficientes estimados 2. Fonte: A Autora.

Termo	Coef	EP de		Valor-T	Valor-P	VIF
		Coef	Coef			
Constante	17,42	0,69		25,11	0,000	
Mean	-6,55	0,81		-8,05	0,000	1,10
SD	-2,04	0,72		-2,82	0,011	1,01
Mean*Mean	5,04	0,80		6,31	0,000	1,04
Mean*SD	4,70	1,13		4,17	0,001	1,06

O Método dos Mínimos Quadrados Ordinários (OLS) e Análise de Variância (ANOVA) também foram aplicados para os modelos BL1 e NMA. Os coeficientes são apresentados nas Tabelas 4.6 e 4.7, respectivamente.

Tabela 4.6 – BL1 - Coeficientes estimados. Fonte: A Autora.

Termo	Coef	EP de		Valor-T	Valor-P	VIF
		Coef	Coef			
Constante	16,00	0,04		362,36	0,000	
Mean	-0,76	0,05		-15,79	0,000	1,35
SD	-0,93	0,04		-22,49	0,000	1,21
Mean*Mean	0,41	0,04		9,29	0,000	1,20
SD*SD	0,61	0,04		15,67	0,000	1,04
Mean*SD	0,85	0,07		12,84	0,000	1,32

Tabela 4.7 – NMA - Coeficientes estimados. Fonte: A Autora.

Termo	Coef	EP de		Valor-T	Valor-P	VIF
		Coef	Coef			
Constante	19,98	0,31	63,42	0,000		
Mean	-3,84	0,29	-13,34	0,000	1,15	
SD	-0,25	0,29	-0,85	0,406	1,15	
Mean*Mean	2,51	0,28	9,09	0,000	1,07	
SD*SD	-0,82	0,28	-2,97	0,008	1,07	
Mean*SD	-0,86	0,46	-1,89	0,075	1,22	

Para o modelo BL1, todos os termos são significativos (valor-p<0,05), enquanto que para o modelo NMA a interação ‘Mean*SD’ e o termo ‘SD’ apareceram como não significativos, porém são necessários para obter um bom ajuste da equação do modelo.

Em seguida, realizou-se a análise dos resíduos. Os resíduos são definidos como a diferença entre o valor previsto do modelo e o valor experimental observado para cada condição e sua análise é importante para garantir que os modelos matemáticos desenvolvidos representam bem as respostas de interesse (GOMES, 2010).

Segundo Montgomery (2009), os resíduos devem ser normais, aleatórios e não correlacionados. Dessa forma, os resíduos foram analisados e apresentaram distribuições normais, sem correlação e sem causas especiais.

Então, foi possível obter as equações gerais dos modelos, Equações 22, 23 e 24, com nível de significância de 5% e R² (adj.) iguais a 81,25%, 97,11% e 92,65% para os modelos STAR1, Bilinear (BL1) e NMA, respectivamente, indicando boa confiabilidade e previsibilidade do modelo.

$$ARL (STAR1) = 56.14 - 99.7 * X - 7.90 * SD + 55.98 * X^2 + 12.53 * X * SD \quad (22)$$

$$ARL (BL1) = 22.90 - 11.10 * X - 3.25 * SD + 4.61 * X^2 + 0.39 * SD^2 + 2.28 * X * SD \quad (23)$$

$$ARL (NMA) = 30.08 - 36.71 * X + 2.79 * SD + 27.93 * X^2 - 0.53 * SD^2 - 2.29 * X * SD \quad (24)$$

A partir do desenvolvimento dos modelos matemáticos finais, torna-se possível analisar a maneira como as repostas se comportam devido às alterações nos parâmetros de entrada (GOMES, 2010).

Portanto, deve-se extrair as conclusões práticas dos resultados, descrevê-lo em gráficos e questionar se as repostas satisfazem as questões experimentais. Assim, é

possível analisar quais fatores são mais significativos e têm maior influência no resultado final. Para melhor visualização da significância dos fatores, as Figuras 4.9, 4.10 e 4.11 apresentam o Gráfico de Pareto para cada método. Os modelos STAR1 e NMA mostram que a média apresenta um efeito relativamente maior que o desvio padrão, enquanto para BL1 o desvio padrão apresenta mais influência.

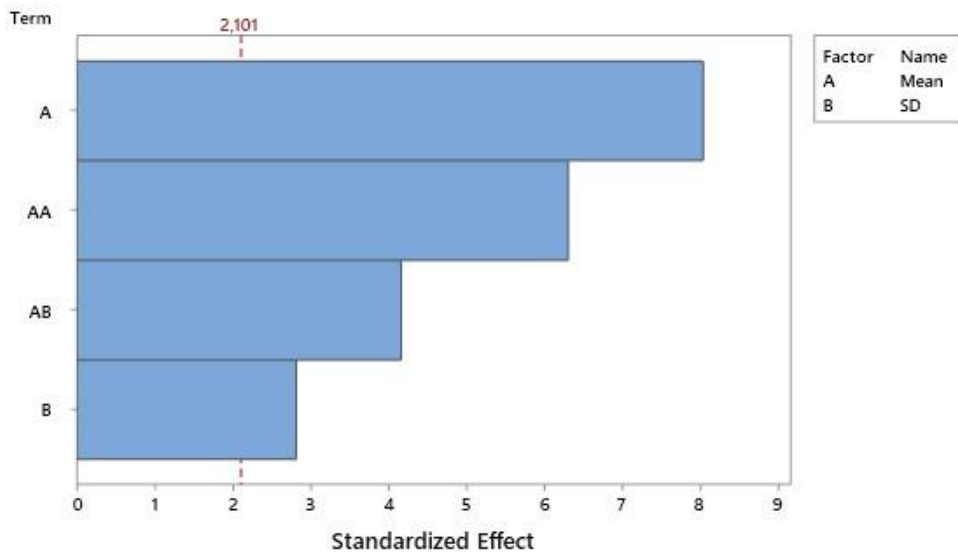


Figura 4.9 – SATR1 - Gráfico de Pareto dos efeitos de NMO. Fonte: A autora.

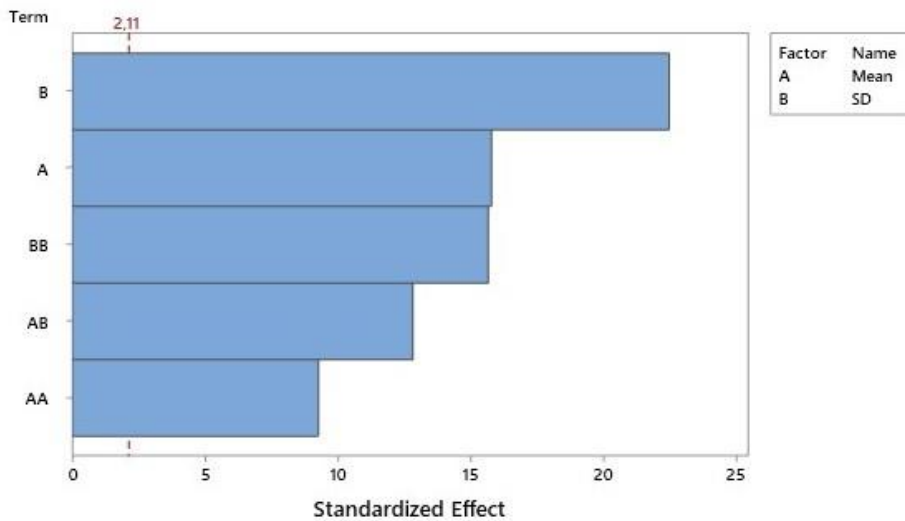


Figura 4.10 – BL1 - Gráfico de Pareto dos efeitos de NMO. Fonte: A autora.

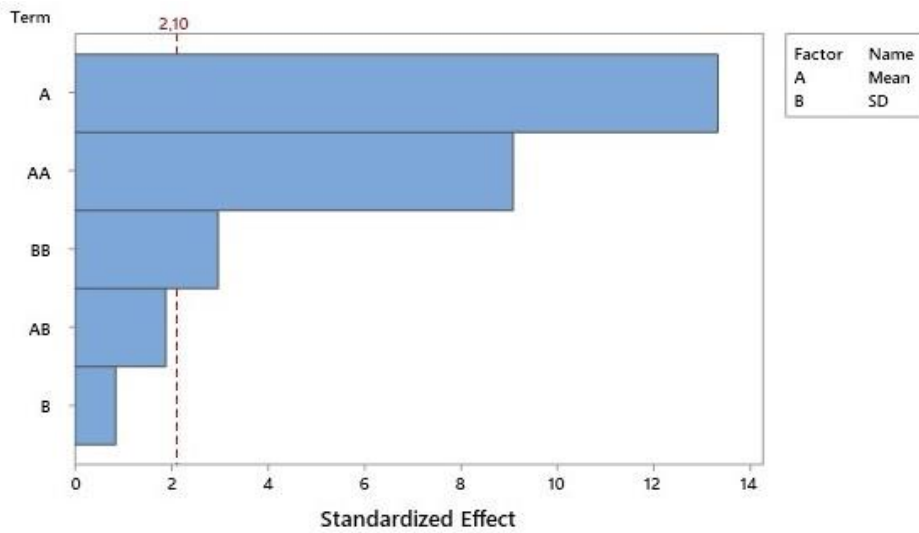


Figura 4.11 – NMA - Gráfico de Pareto dos efeitos de NMO. Fonte: A autora.

Neste momento, é necessário entender como cada fator significativo influencia o modelo. Então, foi desenvolvida uma análise dos principais efeitos, Figuras 4.12, 4.13 e 4.14, e análise de interação, Figuras 4.15, 4.16 e 4.17 e 4.10, para os modelos STAR1, BL1 e NMA, respectivamente.

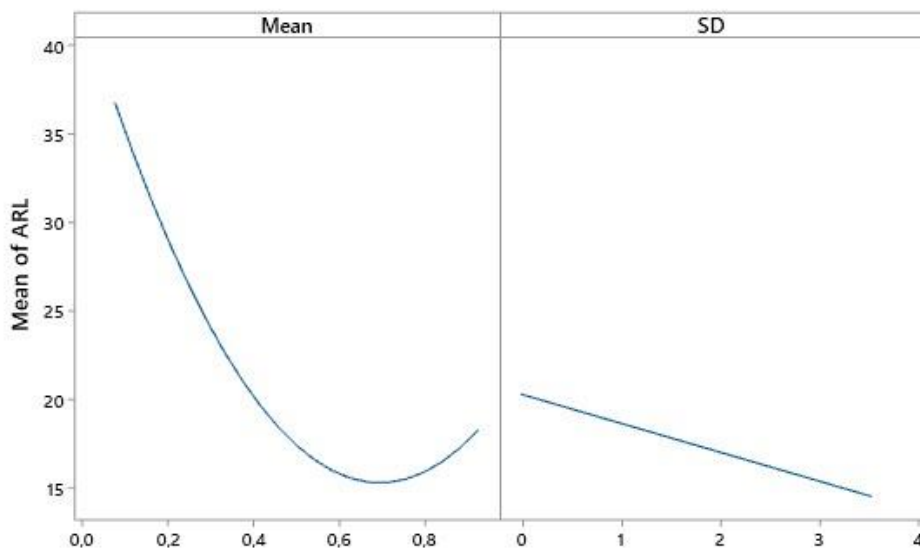


Figura 4.12 – STAR1 – Gráfico dos Principais Efeitos para NMO. Fonte: A autora.

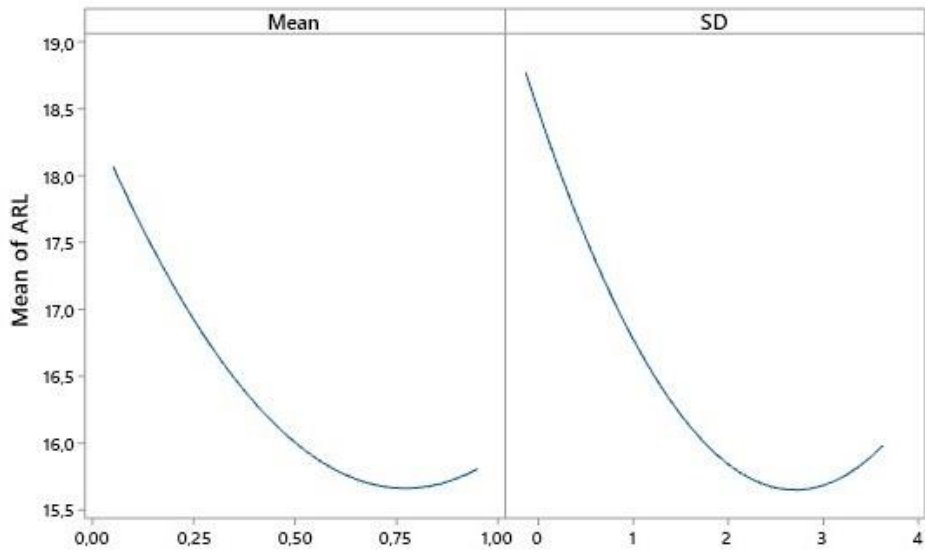


Figura 4.13 – BL1 – Gráfico dos Principais Efeitos para NMO Fonte: A autora.

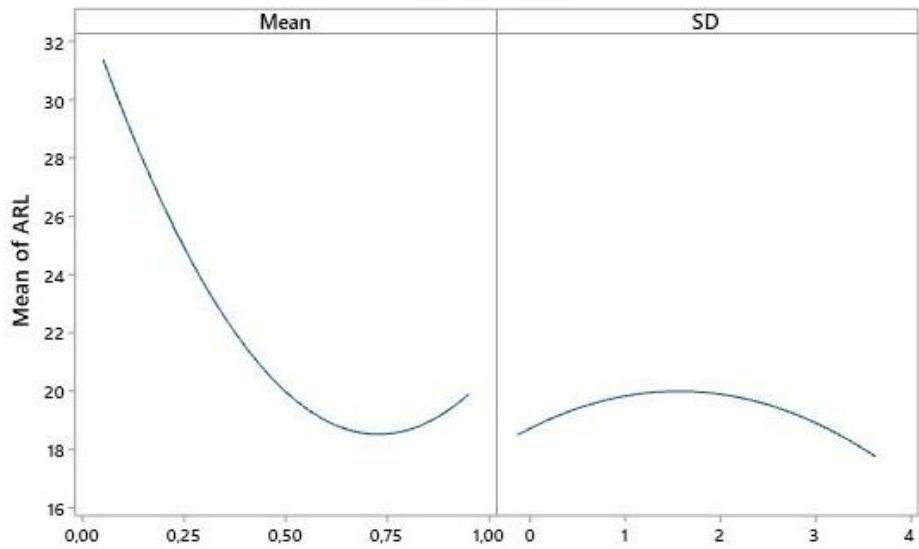


Figura 4.14 – NMA – Gráfico dos Principais Efeitos para NMO. Fonte: A autora.

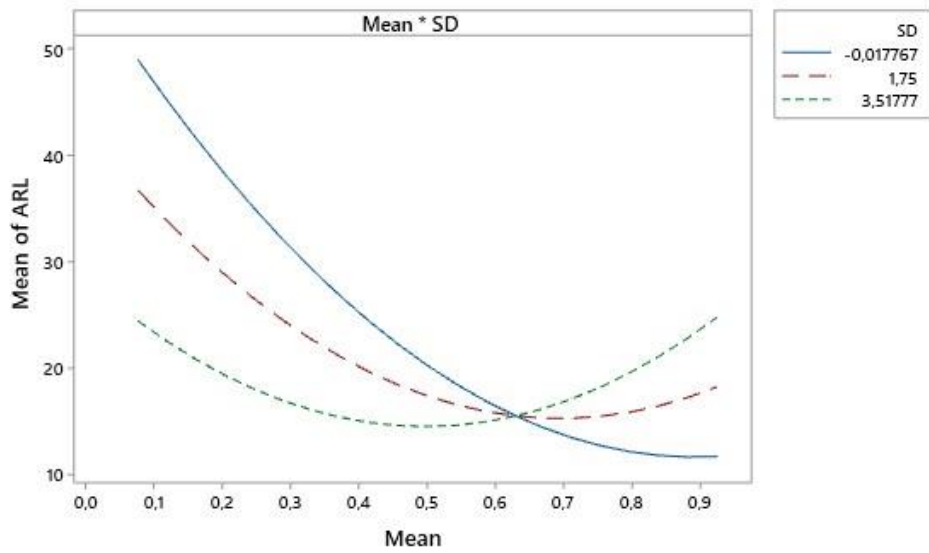


Figura 4.15 – STAR1 – Gráfico de Interações de NMO. Fonte: A autora.

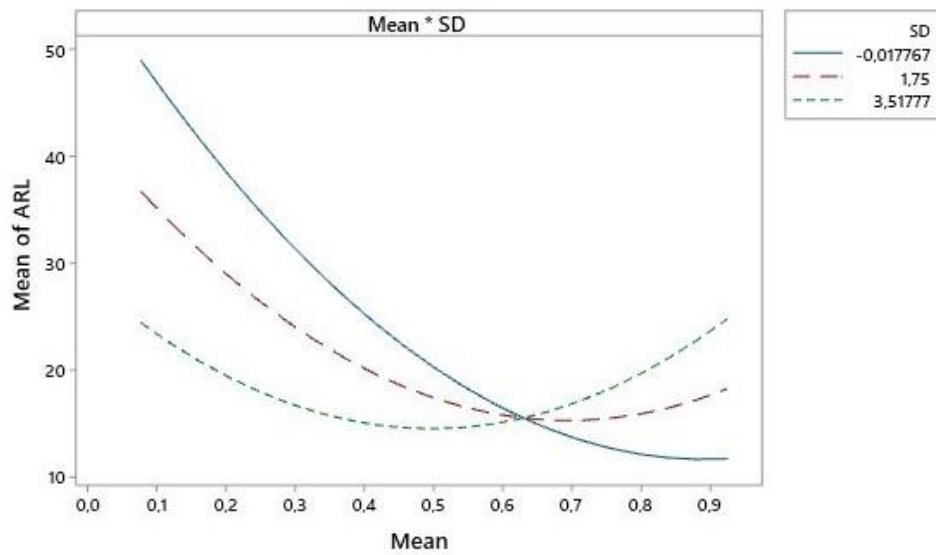


Figura 4.16 – BL1 – Gráfico de Interações de NMO. Fonte: A autora.

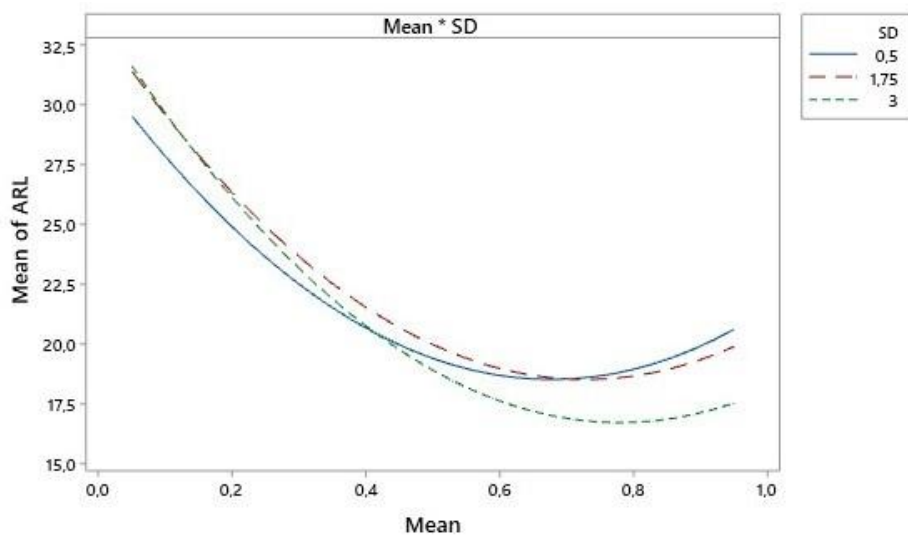


Figura 4.17 – NMA – Gráfico de Interações de NMO. Fonte: A autora.

Conforme inferido da Figura 4.12, para a série STAR1, o NMO diminui até que a média do erro atinja o valor de 0,7. Além disso, o aumento do desvio padrão do erro implica em uma diminuição nos resultados de NMO, sendo mais fácil detectar a mudança na série temporal. Porém, há uma interação significativa entre a média e o desvio padrão do erro, conforme mostrado na Figura 4.15. O menor NMO ocorre quando a média é maior que 0,65 e o desvio padrão é igual a 0,5, mas se a média for menor que 0,65, o melhor NMO é obtido com desvio padrão igual a 3.

Analisando as Figuras 4.13 e 4.14, o efeito da média no NMO para BL1 e NMA segue a mesma tendência do STAR1. Para o modelo BL1, os resultados de NMO apresentam melhor desempenho quando o desvio padrão atinge 2,8 e depois aumenta novamente até atingir 3,5. Além disso, a Figura 4.16 também mostra uma interação entre a média e o desvio padrão. O menor NMO ocorre quando a média é cerca de 0,5 e o desvio padrão é igual a 3, mas se a média for maior que 0,5, o melhor NMO é obtido com o desvio padrão igual a 1,75.

O modelo NMA apresenta uma pequena variação nos valores de NMO à medida que o desvio padrão muda, apresentando um NMO melhor com desvio padrão de 3,5. A Figura 4.17 também mostra uma interação entre a média e o desvio padrão para o NMA, que apresenta um NMO menor com média superior a 0,5 e desvio padrão de 3,5. Para valores médios menores que 0,5, o melhor NMO é alcançado com desvio padrão de 0,5.

O efeito nos resultados de NMO quanto à variação da média e desvio padrão do erro de série também pode ser observado nas Figuras 4.18, 4.19 e 4.20. O modelo STAR1 é o que apresenta a maior variação dos resultados de NMO de acordo com as variações de o erro de série.

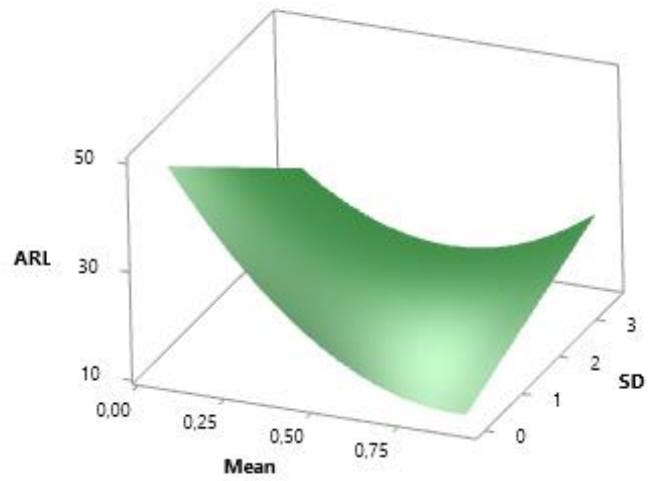


Figura 4.18 – STAR1 - Gráfico de Superfície de NMO. Fonte: A autora.

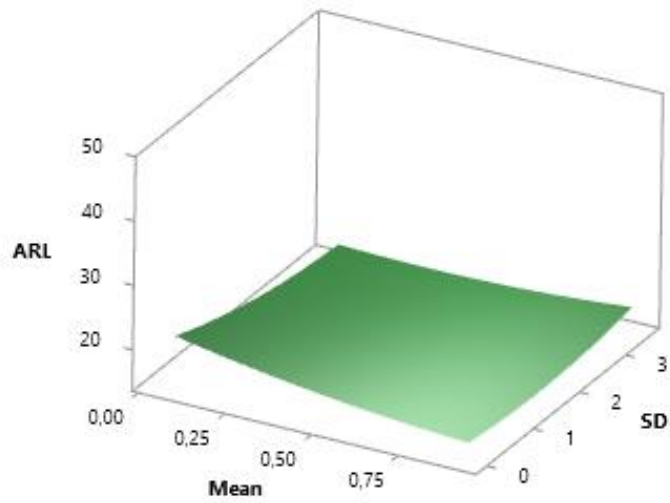


Figura 4.19 – BL1 - Gráfico de Superfície de NMO. Fonte: A autora.

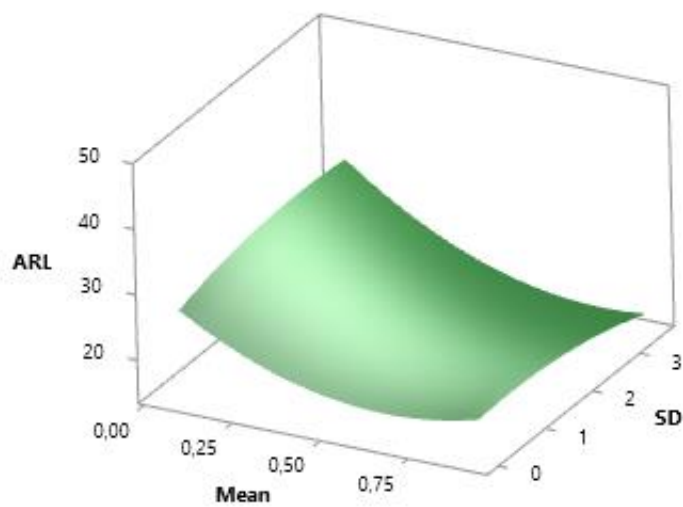


Figura 4.20 – NMA - Gráfico de Superfície de NMO. Fonte: A autora.

Em seguida, foi realizada a análise de otimização para encontrar a solução ótima para minimizar a resposta NMO. A função de *Desirability*, definida por Harrington (1965) e Derringer & Suich (1980), é uma das abordagens utilizadas para otimização fatorial (CANDIOTI *et al.*, 2014). Baseia-se na transformação de todas as respostas obtidas de diferentes escalas em um valor livre de escala. Os valores das funções de desejabilidade (D) situam-se entre 0 e 1. O valor 0 é atribuído quando os fatores dão uma resposta indesejável, enquanto o valor 1 corresponde ao desempenho ótimo (DERRINGER & SUICH, 1980).

A Tabela 4.8 apresenta os resultados da análise de desejabilidade para as três séries temporais não lineares. Eles foram obtidos de acordo com um procedimento semelhante ao mostrado para o modelo STAR1, Figura 4.21. Para STAR1, o menor NMO=11,79 é obtido com média igual a 0,89 e desvio padrão igual a 0,02, com D=1. Para BL1, o menor NMO =15,62 é obtido com média igual a 0,59 e desvio padrão igual a 2,41, com D=1. Para NMA, o menor NMO =15,45 é obtido com média igual a 0,80 e desvio padrão igual a 3,52, com D=1.

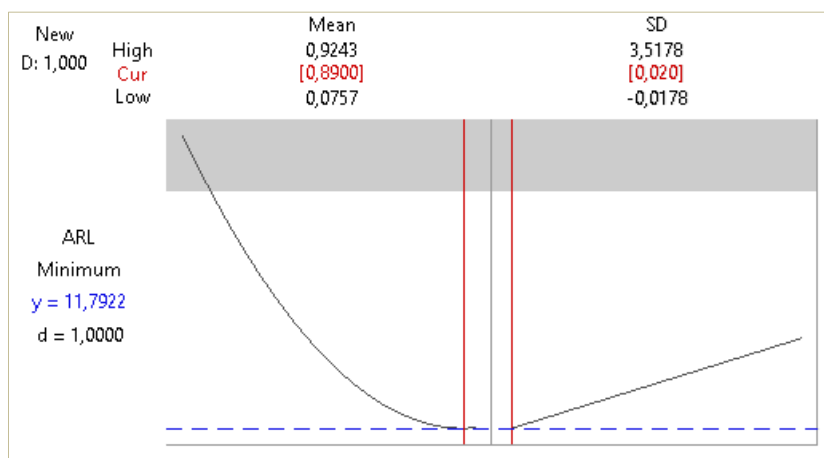


Figura 4.21 – STAR 1 - Otimização da resposta de NMO. Fonte: A autora.

Tabela 4.8 – Resultados da Análise ‘*Desirability*’. Fonte: A autora.

Factors	STAR1	BL1	NMA
Mean	0.89	0.59	0.80
SD	0.02	2.41	3.52
ARL	11.79	15.62	15.45

4.2.4 Validação da Metodologia

Para validar a metodologia proposta, foram realizadas análises de previsão utilizando dados reais baseados no total de óleo e graxa do petróleo (TOG). Os resultados foram comparados com aplicações de métodos tradicionais utilizados para monitoramento de previsão: Cartas de Controle de Soma Acumulada (CUSUM) e Exponencialmente Ponderada (EWMA).

O processamento primário do petróleo consiste na separação do óleo, gás e água obtidos a partir da extração do petróleo bruto. Nesse processo, a água produzida é uma mistura complexa que pode ser caracterizada pelo total de óleo e graxa (TOG) (YANG, 2011; RAY & ENGELHARDT, 1992). O TOG também é amplamente utilizado para fins de vigilância ambiental (CONAMA, 2007).

O valor TOG é obtido a partir de uma análise química da água produzida e é extremamente dependente do método. Segundo Yang (2011), os principais métodos de referência para mensuração do TOG são: absorção no infravermelho, gravimetria e cromatografia gasosa. Dentre estas, a gravimétrica é considerada no presente estudo. Algumas vantagens da análise gravimétrica estão relacionadas à sua simplicidade e baixo custo. No entanto, apresenta desvantagem em termos de sensibilidade, pois seu limite inferior de detecção varia de 5 a 10 mg/L (EBENEZER & GEORGEM, 2014).

O valor de TOG obtido pelo método gravimétrico considera tanto a fração de óleo dispersa, que representa o óleo na água produzida na forma de pequenas gotas, quanto a fração de óleo dissolvida, que é definida como a quantidade de óleo na água produzida em uma forma solúvel (RAY & ENGELHARDT, 1992).

Para aplicar a metodologia proposta, foram obtidos 100 valores de TOG de uma plataforma de petróleo no Brasil. Primeiramente, 50 amostras foram treinadas usando MLP com os parâmetros definidos conforme mostrado no estudo de simulação, gerando o modelo de previsão. Depois, os sinais de rastreamento foram calculados usando os erros de previsão obtidos através do modelo. Após, foi possível determinar os limites de controle para os sinais de rastreamento, assumindo uma taxa de falso alarme desejado de 0,01. Os limites de controle calculados foram iguais a -4 e 4. A aplicação do predictor nos dados de TOG em Matlab® está descrita no Apêndice C.

Em seguida, o modelo de previsão foi alimentado com novos dados e novas previsões foram geradas, assim como novos sinais de rastreamento. Este processo foi feito até que um sinal de rastreamento excedesse o limite de controle na 54ª amostra. A Figura

4.22 mostra a série TOG real e sua previsão, bem como o erro de previsão e o sinal de rastreamento.

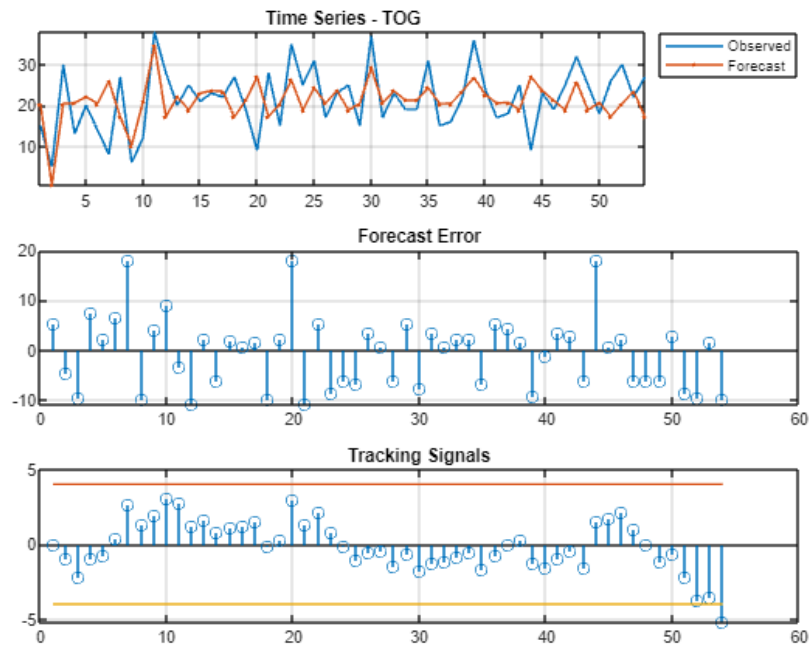


Figura 4.22 – Previsão dados TOG – Amostra 1 a 54. Fonte: A autora.

Portanto, foi necessário aplicar novamente o MLP e atualizar o modelo de previsão. Assim, os sinais de rastreamento e os limites de controle foram recalculados, e outras 30 amostras foram previstas usando este modelo de previsão até que o sinal de rastreamento transpusesse o limite superior, conforme mostrado na Figura 4.23.

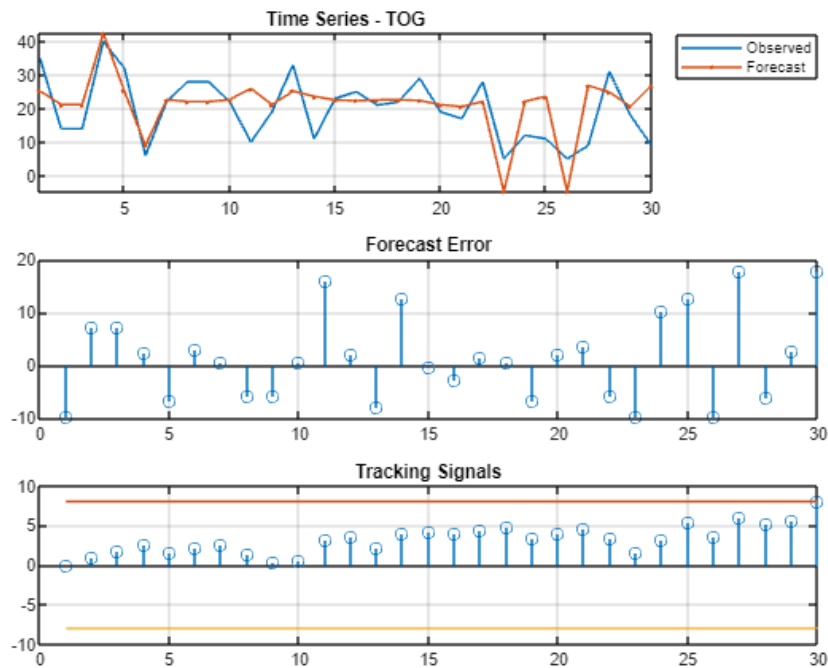


Figura 4.23 – Previsão dados TOG – Amostra 55 a 84. Fonte: A autora.

Assim, o modelo de previsão foi atualizado novamente e utilizado nas demais amostras, estando todas dentro dos limites de controle do gráfico de sinais de rastreamento, Figura 4.24.

Diferente de muitos estudos que atualizam o modelo de previsão para cada amostra ou utilizam o mesmo modelo para todas as amostras, o exemplo mostrou que o previsor conseguiu detectar mudanças nos dados do TOG e foi necessário atualizar o modelo de previsão apenas duas vezes.

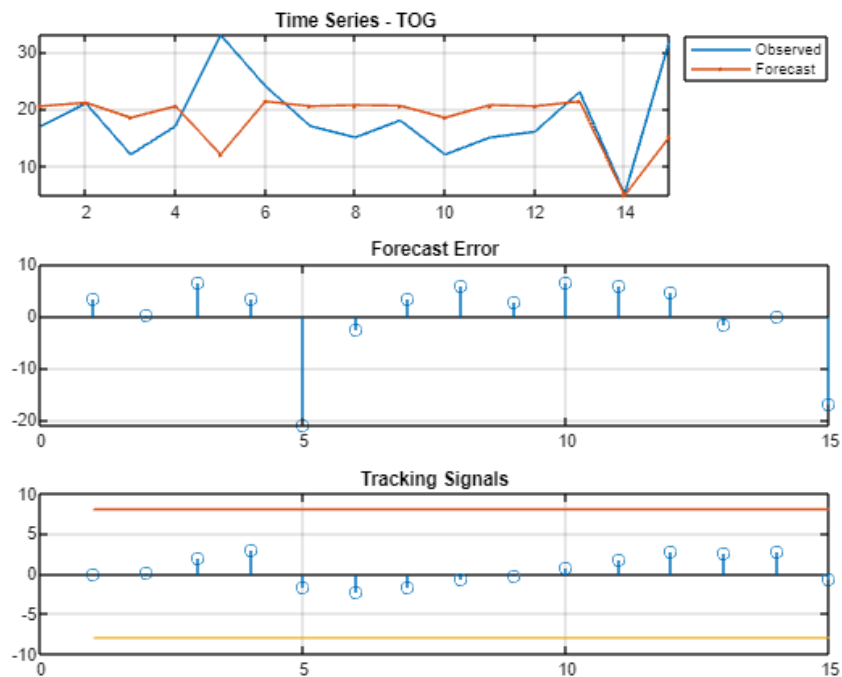


Figura 4.24 – Previsão dados TOG – Amostra 85 a 100. Fonte: A autora.

Por meio de uma análise de tendência, é possível observar essa mudança de comportamento na série temporal. Na Figura 4.25, a série apresenta uma tendência de crescimento nos valores de TOG considerando as amostras de 1 a 54, enquanto que na Figura 4.26, as amostras de 55 a 84 apresentam um decréscimo nos valores de TOG e, por fim, as amostras de 85 a 100 apresentam uma inclinação suave de decréscimo próximo a estabilidade, Figura 4.27.

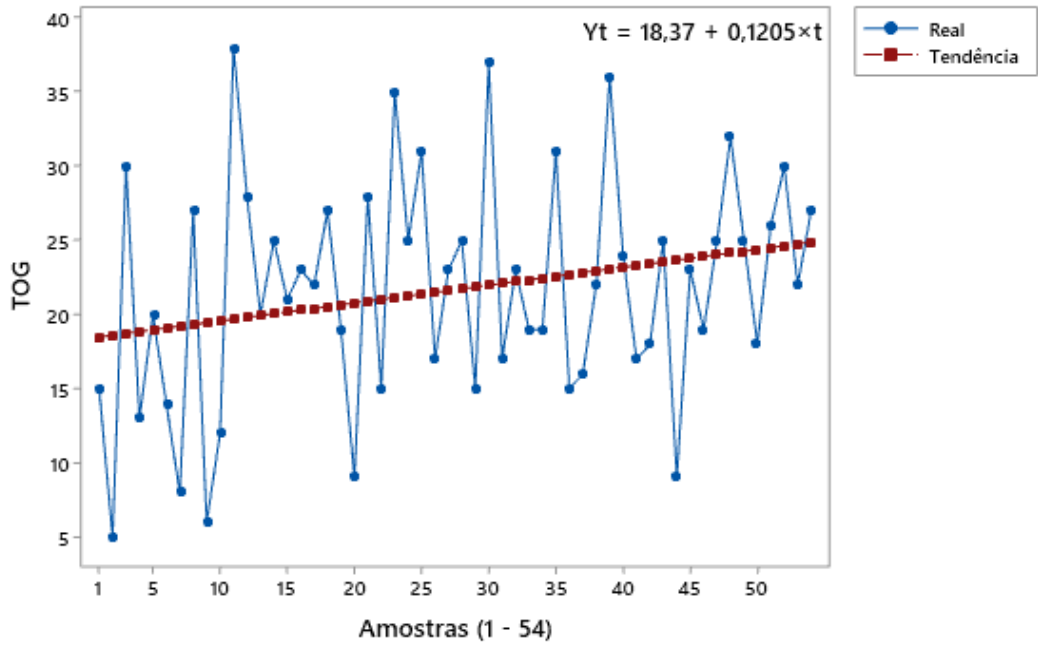


Figura 4.25 – Análise de Tendência TOG – Amostra 1 a 54. Fonte: A autora.

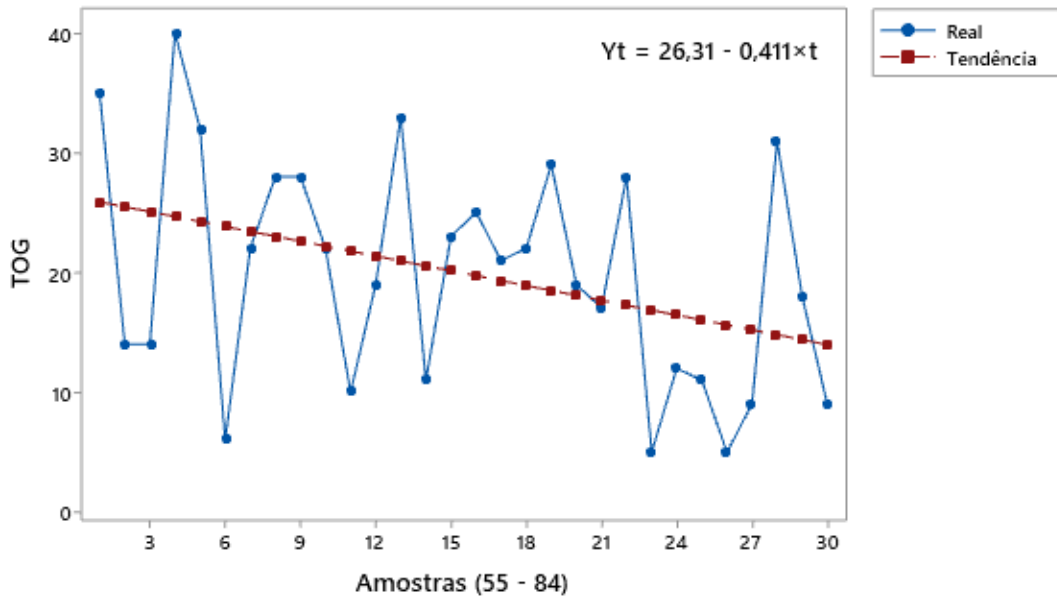


Figura 4.26 – Análise de Tendência TOG – Amostra 55 a 84. Fonte: A autora.

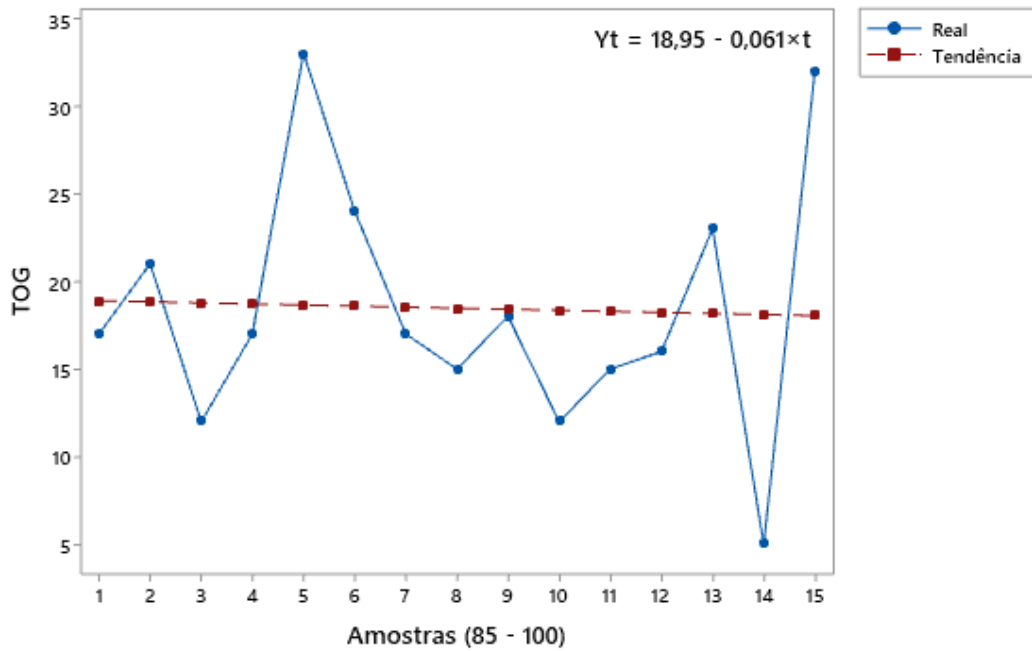


Figura 4.27 – Previsão dados TOG – Amostra 85 a 100. Fonte: A autora.

Com o objetivo de desafiar o método proposto, os métodos EWMA e CUSUM foram aplicados na mesma série temporal TOG.

A Figura 4.28 traz a aplicação dos erros de previsão na carta CUSUM, realizada no software Minitab®. Pode-se notar que não houve indícios de anormalidade na série, portanto, o método de previsão não foi reajustado no decorrer das 100 amostras.

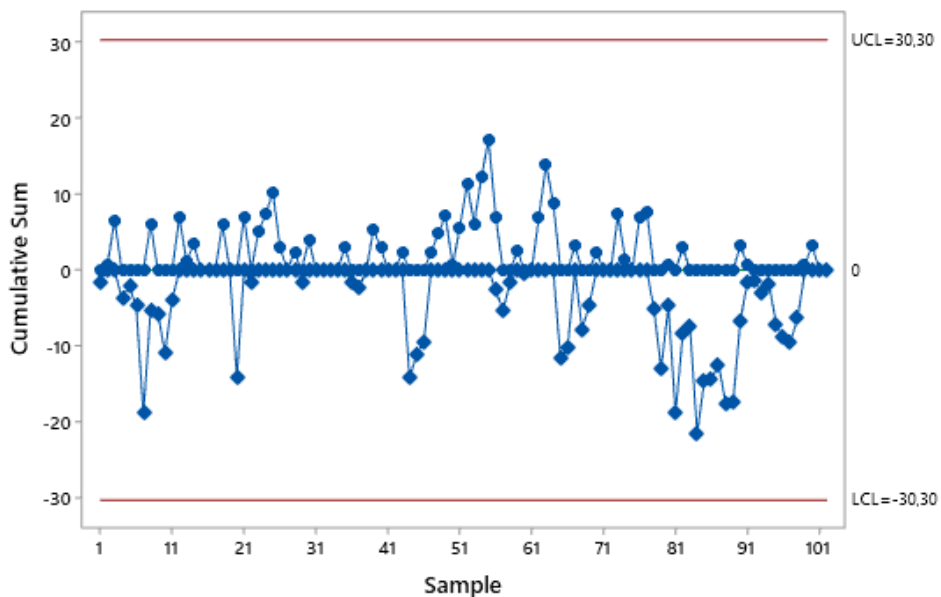


Figura 4.28 – CUSUM aplicado aos dados TOG. Fonte: A autora.

A Figura 4.29 mostra os erros de previsão inseridos na carta EWMA, a qual não indicou nenhum desvio na série, semelhantes ao resultado obtido pela carta CUSUM.

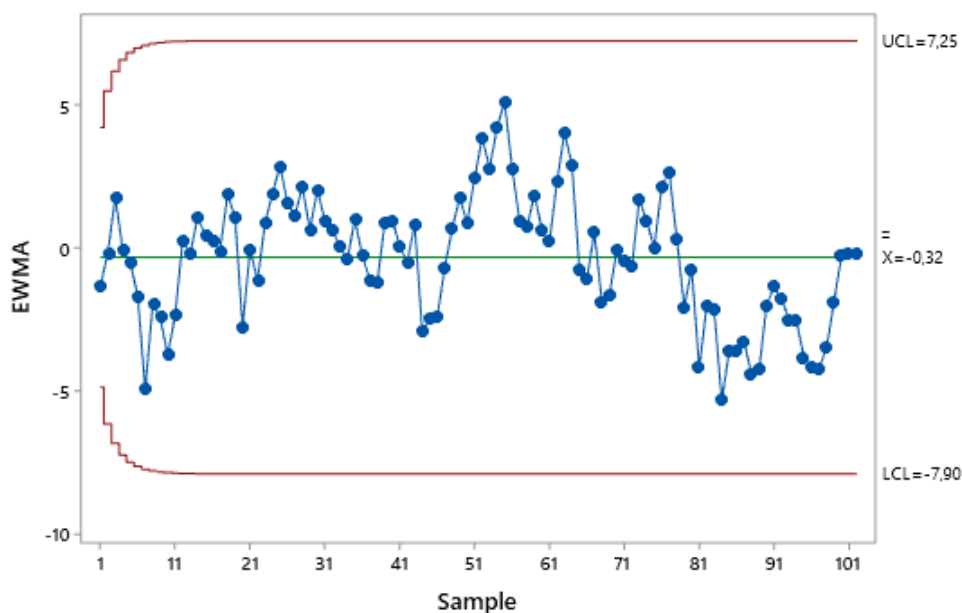


Figura 4.29 – EWMA aplicado aos dados TOG. Fonte: A autora.

Os resultados obtidos nos três métodos estão descritos na Tabela 4.9. Nota-se que o método utilizando *Tracking Signals* indicou a necessidade de 2 ajustes no modelo de previsão, resultando em uma previsão com maior acuracidade. A comparação da precisão da previsão foi realizada através do Erro Médio Quadrado (MSE) e Erro Absoluto Médio Percentual (MAPE), cujas fórmulas estão descritas nas Equações 25 e 26, respectivamente, as quais indicam o quanto os valores previstos se aproximam dos valores reais.

Tabela 4.9. Comparativo *Tracking Signals* x CUSUM x EWMA. Fonte: A autora.

Método	Ajustes do Modelo	MSE	MAPE
TRACKING SIGNALS	2 vezes	44,38	0,34
CUSUM	0	56,43	0,43
EWMA	0	56,43	0,43

$$MSE = \frac{\sum (y_i - \hat{y}_i)^2}{N} \quad (25)$$

$$MAPE = \frac{1}{N} \sum \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (26)$$

onde, y_i é o valor observado, \hat{y}_i é o valor previsto e N é o número de observações.

5 RESULTADOS

As principais conclusões dos resultados experimentais estão resumidas a seguir:

- A metodologia de previsão apresenta melhor desempenho quando aplicada no modelo STAR1, uma vez que a mudança na série temporal pode ser detectada precocemente do que em outros modelos;
- Para STAR1 e NMA, a mudança na média do erro tem um efeito significativo nos resultados de NMO, obtendo o menor NMO quando a média tem uma variação alta, enquanto o BL1 apresenta um NMO menor quando há uma variação moderada na média;
- Os resultados também mostram que o desvio padrão tem um efeito significativo nos resultados de NMO, principalmente em relação aos modelos BL1 e NMA, que apresentam o melhor resultado de NMO quando o desvio padrão tem uma variação de moderada a alta enquanto o menor NMO para STAR1 é obtido quando ocorre uma pequena mudança no desvio padrão. Isso poderia ser teoricamente esperado uma vez que BL1 e NMA possuem uma correlação de média móvel (MA) e uma mudança no desvio padrão do erro pode implicar em uma maior variabilidade nos valores de y .
- A metodologia de previsão proposta pode ser generalizada para outras séries temporais não lineares, sendo possível detectar viés na série temporal de forma eficaz.

6 CONCLUSÕES

A previsão de séries temporais é amplamente utilizada em diversas áreas para fazer inferências razoáveis sobre o futuro e o monitoramento dos erros de previsão é essencial para garantir a precisão da previsão. Embora ferramentas inteligentes como as redes neurais tenham sido aplicadas na previsão de séries temporais há algum tempo, o problema de monitorar o processo de previsão não tem sido amplamente abordado. Às vezes, pode não haver dados atuais suficientes disponíveis ou, em alguns casos, com o passar do tempo, dados mais recentes se tornam disponíveis, enquanto algumas observações históricas antigas podem distorcer a estrutura atual do sistema, ou o padrão existente ou as relações obtidas do passado podem não continuar no futuro. Portanto,

detectar quaisquer alterações no sistema e retomar o controle do processo é muito importante para garantir certas previsões e tomadas de decisões corretas.

Motivado por isso, esta tese teve como objetivo avaliar uma metodologia simples e funcional de previsão de séries temporais não lineares usando o método do sinal de rastreamento para detectar viés e sua responsividade a mudanças não aleatórias na série temporal.

A análise do previsor foi baseada em 100 amostras de 26 conjuntos de dados sintéticos para as séries temporais não lineares STAR1, BL1 e NMA. Os dados foram gerados para simular diferentes situações através da alteração da média e o desvio padrão do erro da série inicialmente definido como distribuição normal $N(0;1)$. Cada conjunto de dados contém 100 dados do modelo original e 50 dados com o erro da série modificado. O MLP foi implementado nas amostras de séries temporais originais com os parâmetros de treinamento cuidadosamente definidos. Os sinais de rastreamento foram aplicados nos erros de previsão, e os resultados foram plotados em gráficos de controle. Assim, as séries temporais modificadas foram inseridas no modelo de previsão e os NMO foram obtidos e comparados por meio de ANOVA com nível de significância de 5%.

Os resultados apresentados neste trabalho mostram que a metodologia de previsão apresentou melhor desempenho aplicada a série STAR1, pois foi capaz de detectar a mudança na série antes dos demais modelos. Para STAR1 e NMA, os menores resultados de NMO são obtidos quando a média tem alta variação, enquanto o BL1 apresenta menor NMO quando há variação moderada na média. Os resultados também mostram que, para os modelos BL1 e NMA, o melhor NMO ocorre quando o desvio padrão tem uma variação moderada a alta enquanto o menor NMO para STAR1 é obtido com uma pequena variação no desvio padrão.

Além disso, este estudo apresentou a aplicação da metodologia utilizando dados TOG e comparou os resultados com as metodologias de cartas e controle CUSUM e EWMA, sendo que foi possível observar melhor precisão nos resultados de previsão ao utilizar a metodologia proposta, a qual a apresentou MAPE de 34% enquanto que as demais metodologias apresentaram MAPE de 43%.

Em geral, a metodologia de previsão proposta pode ser utilizada em diversas séries temporais não lineares e é uma maneira eficaz de detectar viés no processo quando um erro é introduzido na série temporal, pois a média e o desvio padrão do erro da série têm um impacto significativo no NMO. Ao usar esta metodologia, o modelo de previsão pode ser mantido atualizado e, assim, melhorar a precisão da previsão.

Este estudo difere-se dos demais encontrados na literatura principalmente em relação a metodologia aplicada para previsão de séries temporais não lineares, a qual utiliza de redes neurais artificiais e sinais de rastreamento para detectar viés e sua capacidade de resposta a mudanças não aleatórias na série temporal.

Além disso, esta tese difere-se no que diz respeito ao método utilizado para comparar os resultados de NMO. A técnica DOE foi aplicada para simular os conjuntos de dados usando um arranjo de experimentos com níveis controlados dos fatores e para identificar em quais condições os sinais de rastreamento são eficientes. No entanto, outros trabalhos apresentam uma abordagem usual de NMO em que um número maior de conjuntos de dados diferentes é testado e pode levar a conclusões restritas sem uma análise formal e estatística, o que pode ser alcançado usando DOE.

Outra diferença é o cálculo dos limites de controle que foram determinados utilizando o conceito de taxa de falsos alarmes, diferente de outros trabalhos que utilizam limite de controle fixo.

Pode-se concluir que os objetivos da tese foram atingidos, pois foi possível (1) desenvolver de um previsor séries temporais através de redes neurais artificiais e sinais de rastreamento; (2) utilizar delineamento de experimentos para avaliar o comportamento do previsor quando submetido a mudança na série; (3) implementar e validar a aplicação do previsor em dados reais.

Uma limitação apresentada pelo trabalho foi que só foram estudadas três modelos de séries temporais não lineares, não abordando séries lineares e auto correlacionadas. Além disso, só foi estudado um método de sinais de rastreamento. Outra limitação foi a configuração dos parâmetros das redes neurais e planejamento de experimentos devido às várias possibilidades que estes oferecem.

A partir desse estudo, muitos outros ainda podem ser realizados, exemplos: (1) análise de outras séries temporais não lineares, (2) avaliação do previsor de séries temporais usando outro método que não o MLP, (3) ou usando outro método de sinais de rastreamento que não CUSUM.

Por fim, esta metodologia pode ser amplamente aplicada em análises de previsão em diversas áreas do conhecimento, como áreas ambientais em que são realizadas previsões climáticas e hidrológicas e áreas sociais em que previsões de estudos clínicos e medicinais podem ser de extrema importância. Este estudo também pode ser uma referência para futuras pesquisas que buscam melhorar a precisão da previsão de séries temporais não lineares.

REFERÊNCIAS BIBLIOGRÁFICAS

- AIZENBERG, I. et al. Multilayer Neural Network with Multi-Valued Neurons in time series forecasting of oil production. **Neurocomputing**, 175, p. 980-989, 2016.
- ADYA, M.; COLLOPY, F.L. How effective are neural networks at forecasting and prediction? A review and evaluation. **Journal of forecasting**, v.17, n.5-6, p.481-495, 1998.
- AMDOUN, R. et al. The Desirability Optimization Methodology a Tool to Predict Two Antagonist Responses in Biotechnological Systems: Case of Biomass Growth and Hyoscyamine Content in Elicited *Datura stramonium* Hairy Roots. **Iranian J Biotech**, v. 16, 2018.
- AMIRI, E. Forecasting Daily River Flows Using Nonlinear Time Series Models. **Journal of Hydrology**, v. 527, p. 1054-1072, 2015.
- APPOLINÁRIO, F. **Metodologia da ciência: filosofia e prática da pesquisa**. São Paulo: Pioneira Thomson Learning, 2006.
- ARMSTRONG, J. **Principles of Forecasting: A Handbook for Researchers and Practitioners**, New York: Springer Science + Business Media, 2001.
- BALESTRASSI, P.P.; POPOVA E.; PAIVA, A.P.; LIMA, J.W. Design of experiments on neural network's training for nonlinear time series forecasting. **Neurocomputing**, v. 72, p. 1160-1178, 2009.
- BASHEER, I.A. AND HAJMEER, M. Artificial Neural Networks: Fundamentals, Computing, Design, and Application. **Journal of Microbiological Methods**, v. 43, p.3-31, 2000.
- BATISTA, A.L. **Modelos de séries temporais e redes neurais artificiais na previsão de vazão**. Dissertação (Mestrado em Engenharia de Sistemas), Universidade Federal de Lavras, Minas Gerais, 2009.
- BERRY, M.; LINOFF G., **Data Mining Techniques**, John Wiley & Sons, Inc. 3., 1997.
- BERTRAND, J. W. M., FRANSOO, J. C. Modelling and Simulation: operations management research methodologies using quantitative modeling. **International Journal of Operations & Production Management**, v.22, p.241-264, 2002.
- BIANCHESI, N. et al. A Design of Experiments Comparative Study on Clustering Methods. **IEEE Access**, v.7, p.167726-167738, 2019.
- BONSALL, M. B.; WALLACE-HADRILL, S. M. A.; GEDDES, J. R.; GOODWIN, G. M.; HOLMES, E. A. Nonlinear time-series approaches in characterizing mood stability and mood instability in bipolar disorder. **Proceedings of the Royal Society B**, v. 279, p. 916-924, 2012.
- BRENCE J.; MASTRANGELO, C. Parameter Selection for a Robust Tracking Signal. **Quality and Reliability Engineering International**, v.22, p.493-502, 2006.
- BROWN, G. **Statistical Forecasting for Inventory Control**. New York: McGraw-Hill, 1959.
- CANDIOTI, L. et al. Experimental design and multiple response optimization. Using the desirability function in analytical methods development. **Talanta**, v.124, p.123-138, 2014.

- CHAN, K.S; TONG, H. On estimating thresholds in autoregressive models. **Journal of Time Series Analysis**, v.7, pp. 179-190, 1986.
- CHANG, J.; TSENG, C. Analysis of Correlation between Secondary PM2.5 and Factory Pollution Sources by Using ANN and the Correlation Coefficient. **IEEE Access**, v.5, p.22812-22822, 2017.
- CHEN, T.; CHEN H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, **IEEE Trans. Neural Netw.** v. 6, p.911–917, 1995.
- COHEN, J. **Statistical Power Analysis for the Behavioral Sciences**, 2nd ed. Hillsdale: Lawrence Erlbaum Associates, 1988.
- CONAMA Resolution No. 393/2007. Disponível em <http://www.brazilianr.com/brazilian-environmentallegislation/conama-resolution-39307/>. Acesso em 04 de fevereiro de 2022.
- CUI, L.; WANG, Z.; ZHOU, X. Optimization of elicitors and precursors to enhance valtrate production in adventitious roots of *Valeriana amurensis* Smir. ex Kom. **Plant Cell Tiss Organ Cult**, v.108, p.411-420, 2012.
- DE GOOIJER, J.G.; HYNDMAN, R.J. 25 Years of Time Series Forecasting. **International Journal of Forecasting**, v.22, p.443-473, 2006
- DEBOECK, G. J. **Trading on the Edge: Neural, Genetic, and Fuzzy Systems for Chaotic Financial Markets**, John Wiley & Sons, Inc., 1994.
- DENG, Y.; JARAIEDI, M.; ISKANDER, W. Tracking Signal Test to Monitor an Intelligent Time Series Forecasting Model. **Intelligent Manufacturing**, v.5263, p.149-160, 2004.
- DERRINGER, G.; SUICH, R. Simultaneous optimization of several response variables. **J Quality Technol**, v.12, p.214-219, 1980.
- DRAPER, N. R.; LIN, D. K. J. **Response Surface Designs**, In: Ghosh, S., Rao, C. R., “Handbook of Statistics: Design and Analysis Experiments”, Elsevier Science, New York, v. 13, p. 343-375, 1996.
- EBENEZER, T. I.; GEORGEM, Z. C. Produced water treatment technologies. **Int. J. Low-Carbon Techn.**, v.9, p.157–177, 2014.
- GARDNER, E. CUSUM vs. smoothed error forecast monitoring schemes: Some simulation results. **Journal of the Operational Research Society**, v.36, p.43-47, 1985.
- GERALD C.; DIMITRI, S. Knowledge-based modularization and global optimization of artificial neural network models in hydrological forecasting. **Neural networks**, v.20, p.528-536, 2007.
- GOMES, J.H.F. **Análise e otimização da soldagem de revestimento de chapas de aço ABNT 1020 com utilização de arame tubular inoxidável austenítico**. Dissertação no Programa de Pós Graduação em Engenharia de Produção, Universidade Federal de Itajubá, Itajubá, 2010.
- GUAN, Y.; LIU, P; JIN, B.; GUPTA, V.; LI, L. K. B. Nonlinear time-series analysis of thermoacoustic oscillations in a solid rocket motor. **Experimental Thermal and Fluid Science**, v. 98, p. 217-226, 2018.
- GRANGER, C.W.J.; ANDERSON, A.P. **An Introduction to Bilinear Time Series**

Models, Vandenhoeck & Ruprecht, Gottingen, 1978.

HAMILTON, J. A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. **Econometrica Journal of Economic Society**, v.57, p.357-384, 1989.

HARRINGTON, E. The desirability function. **Ind Quality Control**, v.21, p.494-498, 1965.

HAYKIN, S. **Neural Networks and Learning Machines**. 3. ed. New Jersey: Prentice Hall, 2009.

HIPPERT H.; TAYLOR, J. An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting. **Neural networks**, v.23, p.386-395, 2010.

HORNIK, K. Some new results on neural network approximation, **Neural Netw.** v.6, p.1069–1072, 1993.

HU, J.; WANG, X.; ZHANG, Y.; ZHANG, D.; ZHANG, M.; XUE, J. Time Series Prediction Method Based on Variant LSTM Recurrent Neural Network. **Neural Processing Letters**, v.2, 2020.

KIALASHAKI, A.; REISEL, J. Modeling of the energy demand of the residential sector in the United States using regression models and artificial neural networks. **Applied Energy**, v.108, p.271–280, 2013.

KRISHNAMURTHY, B. **A comparison of the relative efficiency of tracking signals in forecast control**. Graduate Theses, Dissertations, and Problem Reports. West Virginia University, 2006.

KUMAR D.; MURUGAN, S. A Novel Fuzzy Time Series Model for Stock Market Index Analysis using Neural Network with Tracking Signal Approach. **Indian Journal of Science and Technology**, v.10, 2017.

LIU, Z. et al. Forecast Methods for Time Series Data: A Survey. **IEEE Access**, v.9, p.91896-91912, 2021.

LUCAS, J.M.; SACCUCCI, M.S. Exponentially Weighted Moving Average Control Schemes: Properties and Enhancements. **Technometrics**, v.32, p.1-12, 1990.

LORSCHIED, I.; HEINE, B.O.; MEYER, M. Opening the ‘black box’ of simulations: Increased transparency and effective communication through the systematic design of experiments. **Comput. Math. Org. Theory**, v.18, p.22–62, 2012.

MAKRIDAKIS, S.; WHEELWRIGHT, S. **Forecasting Methods for Management**, 5th ed., New York: John Wiley, 1989.

MAO S.; XIAO, F. Time Series Forecasting Based on Complex Network Analysis. **IEEE Access**, v.7, p.40220-40229, 2019.

MASON, R. L.; GUNST, R. F.; HESS, J. L.; **Statistical Design and Analysis of Experiments with Applications to Engineering and Science**. John Wiley & Sons, 2003.

MATTA, C.; BIANCHESI, N.; OLIVEIRA, M.; BALESTRASSI, P.; LEAL, F. A comparative study of forecasting methods using real-life econometric series data. **Production**, v.31, 2021.

MCCLAIN, J. Dominant tracking signals. **International Journal of Forecasting**, v.4, p.563-572, 1988.

- MCCULLOCH, W.S., PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v.5, n.1, p. 115–133, 1943.
- MIGUEL, P. A. C.; FLEURY, A.; MELLO, C. H. P.; NAKANO, D. N.; TURRIONI, J. B.; LEE HO, L.; MORABITO, R.; MARTINS, R. A.; PUREZA, V. **Metodologia de pesquisa em engenharia de produção e gestão de operações**. 2ª. Ed. Rio de Janeiro: Elsevier, 2014.
- MIRCETIC, D.; ROSTAMI-TABAR, B.; NIKOLICIC, S.; MASLARIC, M. Forecasting hierarchical time series in supply chains: an empirical investigation. **International Journal of Production Research**, 2021.
- MO, F. et al. Statistical Analysis of the Influence of Imperfect Texture Shape and Dimensional Uncertainty on Surface Texture Performance. **IEEE Access**, v.5, p.27023-27035, 2017.
- MONTGOMERY, C.D. **Introduction to Statistical Quality Control**, 6th ed. New York: John Wiley & Sons, Inc., 2009.
- MONTGOMERY, D. C.; JENNINGS, C. L.; KULAHCI, M.; **Introduction to time series analysis and forecasting**. John Wiley & Sons, 2011.
- MORETTIN, P.; TOLOI, C. **Análise de séries temporais**. 2. Ed. São Paulo. E. Blucher, p.564, 2006.
- MYERS, R.H.; MONTGOMERY, D.C.; ANDERSON-COOK, C.M.; **Response Surface Methodology: Process and Product Optimization using Designed Experiments**. John Wiley & Sons, 2009.
- OLSON, O.; DELEN, D.; MENG, Y. Comparative analysis of data mining methods for bankruptcy prediction. **Decision Support Systems**, v.52, p.464-473. 2012.
- PANT M.; S. KUMAR. Particle swarm optimization and intuitionistic fuzzy set-based novel method for fuzzy time series forecasting. **Granular Computing**, 2021.
- PRIESTLEY, M. State-dependent models: a general approach to nonlinear time series analysis. **Journal of Time Series Analysis**, v.1, p.47–71, 1980.
- QIAN, B.; XIAO, Y.; ZHENG, Z.; ZHOU, M.; ZHUANG, W.; LI, S.; MA, Q. Dynamic Multi-Scale Convolutional Neural Network for Time Series Classification. **IEEE Access**, v.8, 2020.
- RAVI, P. An analysis of a widely used version of the CUSUM tracking signal. **Journal of the Operational Research Society**, v.65, p.1189-1192, 2014.
- RAY, J.; ENGELHARDT, F.R. **Produced Water: Technological Environmental Issues and Solutions**. New York: Plenum Press, 1992.
- ROSENTHAL, R. **Parametric measures of effect size**. **The Handbook of Research Synthesis**. In H. Cooper and L. V. Hedges, Eds. New York: Russell Sage Foundation, p. 231–244, 1994.
- SABETI, H.; AL-SHEBEEB, O.; JARIDI, M. Forecasting System Monitoring under Non-normal Input Noise Distributions. **Journal of Industrial Engineering and Management**, 2016.
- STAIUCULESCU, D.; BUSHYAGER, N.; OBATOYINBO, A.; MARTIN, L. J.; TENTZERIS, M. M., Design and Optimization of 3-D Compact Stripline and Microstrip Bluetooth/WLAN Balun Architectures Using the Design of Experiments

- Technique, **IEEE Transactions on Antennas and Propagation**, v.53, n.5, p. 1805–1812, May 2005.
- SUN, K. et al. Design and Application of a Variable Selection Method for Multilayer Perceptron Neural Network With LASSO. **IEEE Transactions on Neural Networks and Learning Systems**, v.28, p.1386-1396, 2007.
- SUPERVILLE, C. Tracking Signal Performance in Monitoring Manufacturing Processes. **Journal of Business and Management**, v.21, p.23-28, 2019.
- TONG H. **On a threshold model, Pattern Recognition and Signal Processing**, Sijhoff & Noordhoff, Amsterdam, 1978.
- TRIG, W. Monitoring a forecasting system. **Operations Research Quarterly**, v.15, p.271-274, 1964.
- TSAY R. **Analysis of Financial Time Series**, second ed., Wiley-Interscience, 2005.
- WANG Z.; LOU, Y. Hydrological time series forecast model based on wavelet de-noising and ARIMA-LSTM. **IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)**, p. 1697-1701, 2019.
- WEB OF SCIENCE. Disponível em [https://www-webofscience.ez38.periodicos.capes.gov.br/wos/woscc/basic-search](https://www.webofscience.ez38.periodicos.capes.gov.br/wos/woscc/basic-search). Acesso em 16 de agosto de 2021.
- WONG, W.; MIN, X.; CHU, W. Adaptive neural network model for time-series forecasting. **European Journal of Operational Research**, v.207, 2010.
- XIAO, D.; SHI, H.; WU, D. Short-term load forecasting using bayesian neural networks learned by Hybrid Monte Carlo algorithm. **Applied Soft Computing**, p.1822-1827, 2012.
- XIAO, H.; JIANG, X.; CHEN, C.; WANG, W.; WANG, C.; ALI, A.; BERTHE, A.; MOUSSA, R.; DIABY, V. Using time series analysis to forecast the health-related quality of life of post-menopausal women with non-metastatic ER+ breast cancer: A tutorial and case study. **Res Social Adm Pharm**, v.16, p. 1095-1099, 2020.
- YANG, M. Measurement of Oil in Produced Water. **Produced Water**, p.57–88, 2011.
- YU, L.; LAI, K. Adaptive Smoothing Neural Networks in Foreign Exchange Rate Forecasting. **Proceeding Computational Science ICCS 2005 5th International**, 2005.
- ZHAI, X. et al. MLP Neural Network Based Gas Classification System on Zynq SoC. **IEEE Access**, v.4, p.8138-8146, 2016.

APÊNDICE A – Pseudocódigo

Fase	Pseudocódigo
1	1: $y \leftarrow \text{dataset}$ 2: $t \leftarrow \text{time series size}$ 3: $\text{end} \leftarrow y(\text{lastPosition})$ 4: $n_{pred} \leftarrow \text{number of predictions}$
2	5: for $t \leftarrow 1$ to 50 6: $mlp_{net} \leftarrow \text{RNA_MLP_train}(t, y)$ 7: $y_{pred_1} \leftarrow \text{RNA_MLP_test}(mlp_{net}, y, n_{pred})$
3	8: $TS_1 \leftarrow \text{cts}(y_{pred_1}, y)$ 9: end for
	10: $l_s \leftarrow \text{upper limit}$ 11: $l_l \leftarrow \text{lower limit}$
4	12: for $t \leftarrow 51$ to end 13: $y_{pred_t} \leftarrow \text{RNA_MLP_test}(mlp_{net}, y, 1)$
5	14: $TS_t \leftarrow \text{cts}(y_{pred_t}, y)$ 15: if $(\text{cts}(t) > l_s \text{ or } \text{cts}(t) < l_l)$ 16: $mlp_{net} \leftarrow \text{RNA_MLP_train}(t, y)$ 17: else 18: end for
	19: // Functions 20: function $c \leftarrow \text{cts}(ypred, y)$ 21: $N \leftarrow ypred \text{ length}$ 22: $\text{end} \leftarrow y(\text{lastPosition})$ 23: $y \leftarrow y(\text{end} - N + 1: \text{end})$ 24: $\text{error} \leftarrow ypred - y$ 25: $\text{error}_{abs} \leftarrow \text{absolute value of error}$ 26: $\text{mad}(1) \leftarrow \text{error}_{abs}(1)$ 27: $\text{cusum}(1) \leftarrow 0$ 28: $\text{tsb}(1) \leftarrow 0$ 29: $\text{ad}(1) \leftarrow \text{error}_{abs}(1)$ 30: for $t \leftarrow 1$ to N 31: $\text{ad}(1) \leftarrow \text{error}_{abs}(t) + \text{ad}(t - 1)$ 32: $\text{mad}(1) \leftarrow \text{ad}(t)/t$ 33: $\text{cusum}(1) \leftarrow \text{error}(t) + \text{cusum}(t - 1)$ 34: $\text{cts}(1) \leftarrow \text{cusum}(t)/\text{mad}(t)$ 35: end for 36: end function 37: function $\text{net} \leftarrow \text{RNA_MLP_train}(N, y)$ 38: $t \leftarrow 1: (N - 1)$ 39: $x(t) \leftarrow y(t)$ 40: $y_p(t) \leftarrow y(t + 1)$ 41: $\text{net}_{param_{train}} \leftarrow 80/100$ 42: $\text{net}_{param_{valid}} \leftarrow 10/100$ 43: $\text{net}_{param_{test}} \leftarrow 10/100$ 44: $\text{net}_{type} \leftarrow \text{RNA MLP Backpropagation}$ 45: $\text{hiddenLayer}_{size} \leftarrow 10$ 46: $\text{net} \leftarrow \text{feedForwardNet}(\text{hiddenLayer}_{size}, \text{trainlm}')$ 47: $\text{net}_{param_{show}} \leftarrow 50$ 48: $\text{net}_{param_{lr}} \leftarrow 0.01$ 49: $\text{net}_{param_{epochs}} \leftarrow 1000$ 50: $\text{net}_{param_{goal}} \leftarrow 1e-06$ 51: $[\text{net}, \text{tr}] \leftarrow \text{train}(\text{net}, x, y_p)$ 52: end function 53: function $y_{pred} \leftarrow \text{RNA_MLP_test}(N, y, n_{pred})$ 54: $y_{pred} \leftarrow \text{net}(y)$ 55: end function

APÊNDICE B – Aplicação do Previsor nas Séries Temporais em MATLAB

STAR1

```
clear all; clc; close all;

% Inputs
tempo = 50; % número de t
n_pred = 50; % número de predições
limite = 100; % número de conjunto de dados

% Para reprodutibilidade
rng (1)

for i=1:limite; % Repetição das funções para todos conjuntos de dados

    y_01 = ExemploSimulado(tempo,0);
    rede_mlp = rna_mlp_train(tempo,y_01);
    y_pred_mlp_01 = rna_mlp_test(rede_mlp, y_01, n_pred);

    y_inic = y_01(end);
    y_02 = ExemploSimulado(tempo,1, y_inic);
    y_pred_mlp_02 = rna_mlp_test(rede_mlp, y_02, n_pred);

    qtd_fora_lim_mlp01_tsb(i,:) = [cont_tsb(y_pred_mlp_01,y_01,i)];
    qtd_fora_lim_mlp02_tsb(i,:) = [cont_tsb(y_pred_mlp_02,y_02,i)];

end

rng(2)
for i=1:limite; % Repetição das funções para todos conjuntos de dados

    y_011 = ExemploSimulado(tempo,0);
    y_pred_mlp_011 = rna_mlp_test(rede_mlp, y_011, n_pred);

    qtd_fora_lim_mlp011_tsb(i,:) = [cont_tsb(y_pred_mlp_011,y_01,i)];

end

% Armazenar resultados em Excel
xlswrite('Resultado_MLP01_STAR1', qtd_fora_lim_mlp01_tsb);
xlswrite('Resultado_MLP02_STAR1', qtd_fora_lim_mlp02_tsb);
xlswrite('Resultado_MLP011_STAR1', qtd_fora_lim_mlp011_tsb);

%Gerar gráficos das séries não lineares
y_real = [y_01 y_02];
% y_pred_nar = [y_pred_nar_01 y_pred_nar_02];
y_pred_mlp = [y_pred_mlp_01 y_pred_mlp_011 y_pred_mlp_02];
plot(y_real)
hold on
plot(y_pred_mlp)
hold off
title('Time Serie - STAR1')
legend(["Time Series" "ANN"],'Location', 'southwest')
exportgraphics(gcf,'STAR1.jpg','Resolution',300);
```

```

% Gerar gráficos dos erros para um conjunto de dado
grafico_erros(y_pred_mlp_01,y_01)
% exportgraphics(gcf,'Error_01.jpg','Resolution',300);
grafico_erros(y_pred_mlp_011,y_011)
% exportgraphics(gcf,'Error_02.jpg','Resolution',300);
grafico_erros(y_pred_mlp_02,y_02)
% exportgraphics(gcf,'Error_02.jpg','Resolution',300);

%-----
% Funções
%-----

function c = cont_tsb(ypred,y,i) % Função para contar qntd_fora_lim para
Tracking Signal TSB''

    N = length(ypred);
    y = y(end-N+1:end);
    erro = ypred-y;
    erro_abs = abs(erro);
    R0 = 0;
    ls = 4; % limite superior
    li = -4; % limite inferior

    mad(1) = erro_abs(1);
    eac(1) = 0;
    tsb(1) = 0;
    ad(1) = erro_abs(1);

    for t = 2:N

        ad(t) = erro_abs(t) + ad(t-1);
        mad(t) = ad(t)/t; % Mean absolute deviation
        eac(t) = erro(t) + eac(t-1);% Soma acumulada
        tsb(t) = eac(t)/mad(t); % Tracking signals

        % Contar primeiro ponto não conforme (fora dos limites)
        if (tsb(t) > ls || tsb(t) < li)
            R0=[R0+1;t];
            break
        else
            R0=[0;t];
        end

    end

    c= R0(2);

end

function grafico_erros(ypred,y) % Função para gerar os gráficos: séries
temporais, erros, tracking signals, CUSUM
    N = length(ypred);
    y = y(end-N+1:end);
    erro = ypred-y;

```

```

erro_abs = abs(erro);

mad(1)=erro_abs(1);
eac(1)=0;
ad(1)=erro_abs(1);
tsb(1)=0;

for t = 2:N
    eac(t) = erro(t) + eac(t-1);
    ad(t) = erro_abs(t) + ad(t-1);
    mad(t) = ad(t)/t;
    tsb(t) = eac(t)/mad(t); % Tracking signals
end

for t = 1:N
    ls(t) = 4; % limite superior
    li(t) = -4; % limite inferior
end

% Gráfico de séries temporais e previsão
figure
subplot(3,1,1)
plot(y)
axis('tight')
grid on
hold on
plot(ypred,'.-')
hold off
title('Time Series - STAR 1')
legend(['Observed' "Forecast"],'Location', 'bestoutside')
% ylabel("Values")

% Gráfico de erros
subplot(3,1,2)
stem(ypred - y)
grid on
% ylabel("Error")
title('Forecast Error')

% Gráfico Tracking signals
subplot(3,1,3)
stem(tsb)
grid on
hold on
plot(ls)
hold on
plot(li)
hold off
title('Tracking Signals')

end

function net = rna_mlp_train(N,y) % Função para treinar as redes neurais MLP

% Definir as variáveis entrada x(t) e saída y_p(t)
t = 1:N-1;
x(t) = y(t);
y_p(t) = y(t+1);

```

```

% Dividir os dados em treinamento, validação e teste
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;

% --- RNA MLP (Backpropagation) -----
titulo = 'MLP';
hiddenLayerSize = 10;
net = feedforwardnet(hiddenLayerSize,'trainlm');

% Definir parâmetros
net.trainParam.show = 50;
net.trainParam.lr = 0.01;
net.trainParam.epochs = 1000;
net.trainParam.goal = 1e-6;

%Treinar a rede neural MLP
[net,tr] = train(net,x,y_p);

end

function y_pred = rna_mlp_test(net, y, n_pred) % Função para previsão dos
dados através da rede neural MLP

% Calcular a saída Y da rede neural
y_pred = net(y);

% Calcular a performance da rede neural
perf = perform(net,y, y_pred);

end

function y = ExemploSimulado(N, parametro, y_inic) % Função para gerar as
séries temporais BL 1

% Parâmetros
a = 0.8; % inicial
b = 0.8; % inicial

if (parametro ~= 1) % série temporal inicial

    m = 0; % média
    d = 1;% desvio padrão

    er = m + d*rand(N,1);

    y(1) = er(1);
    for t=2:N
        y(t) = a*y(t-1)-b*y(t-1)*(1+exp(-10*y(t-1))).^(-1)+er(t);
    end

else % série temporal com parâmetros modificados

    m = 0.5; % média
    d = 1; % desvio padrão

    er = m + d*rand(N,1);

```

```

        y(1) = a*y_inic-b*y_inic*(1+exp(-10*y_inic)).^(-1)+er(1); % manter a
continuidade com a série temporal inicial

        for t=2:N
            y(t) = a*y(t-1)-b*y(t-1)*(1+exp(-10*y(t-1))).^(-1)+er(t);
        end
    end
end

```

BL 1

```
clear all; clc; close all;
```

```
% Inputs
```

```
tempo = 50; % número de t
```

```
n_pred = 50; % número de previsões
```

```
limite = 100; % número de conjunto de dados
```

```
% Para reprodutibilidade
```

```
rng (1)
```

```
for i=1:limite; % Repetição das funções para todos conjuntos de dados
```

```
    y_01 = ExemploSimulado(tempo,0);
```

```
    rede_mlp = rna_mlp_train(tempo,y_01);
```

```
    y_pred_mlp_01 = rna_mlp_test(rede_mlp, y_01, n_pred);
```

```
    y_inic = y_01(end);
```

```
    y_02 = ExemploSimulado(tempo,1, y_inic);
```

```
    y_pred_mlp_02 = rna_mlp_test(rede_mlp, y_02, n_pred);
```

```
    qtd_fora_lim_mlp01_tsb(i,:) = [cont_tsb(y_pred_mlp_01,y_01,i)];
```

```
    qtd_fora_lim_mlp02_tsb(i,:) = [cont_tsb(y_pred_mlp_02,y_02,i)];
```

```
end
```

```
rng(2)
```

```
for i=1:limite; % Repetição das funções para todos conjuntos de dados
```

```
    y_011 = ExemploSimulado(tempo,0);
```

```
    y_pred_mlp_011 = rna_mlp_test(rede_mlp, y_011, n_pred);
```

```
    qtd_fora_lim_mlp011_tsb(i,:) = [cont_tsb(y_pred_mlp_011,y_01,i)];
```

```
end
```

```
% Armazenar resultados em Excel
```

```
xlswrite('Resultado_MLP01_BL1', qtd_fora_lim_mlp01_tsb);
```

```
xlswrite('Resultado_MLP02_BL1', qtd_fora_lim_mlp02_tsb);
```

```
xlswrite('Resultado_MLP011_BL1', qtd_fora_lim_mlp011_tsb);
```

```
%Gerar gráficos das séries não lineares
```

```
y_real = [y_01 y_02];
```

```
% y_pred_nar = [y_pred_nar_01 y_pred_nar_02];
```

```
y_pred_mlp = [y_pred_mlp_01 y_pred_mlp_011 y_pred_mlp_02];
```

```
plot(y_real)
```



```

hold on
plot(y_pred_mlp)
hold off
title('Time Serie - BL1')
legend(["Time Series" "ANN"],'Location', 'southwest')
exportgraphics(gcf,'BL1.jpg','Resolution',300);

% Gerar gráficos dos erros para um conjunto de dado
grafico_erros(y_pred_mlp_01,y_01)
% exportgraphics(gcf,'Error_01.jpg','Resolution',300);
grafico_erros(y_pred_mlp_011,y_011)
% exportgraphics(gcf,'Error_02.jpg','Resolution',300);
grafico_erros(y_pred_mlp_02,y_02)
% exportgraphics(gcf,'Error_02.jpg','Resolution',300);

%-----
% Funções
%-----

function c = cont_tsb(ypred,y,i) % Função para contar qntd_fora_lim para
Tracking Signal TSB''

    N = length(ypred);
    y = y(end-N+1:end);
    erro = ypred-y;
    erro_abs = abs(erro);
    R0 = 0;
    ls = 4; % limite superior
    li = -4; % limite inferior

    mad(1) = erro_abs(1);
    eac(1) = 0;
    tsb(1) = 0;
    ad(1) = erro_abs(1);

    for t = 2:N

        ad(t) = erro_abs(t) + ad(t-1);
        mad(t) = ad(t)/t; % Mean absolute deviation
        eac(t) = erro(t) + eac(t-1); % Soma acumulada
        tsb(t) = eac(t)/mad(t); % Tracking signals

        % Contar primeiro ponto não conforme (fora dos limites)
        if (tsb(t) > ls || tsb(t) < li)
            R0=[R0+1;t];
            break
        else
            R0=[0;t];
        end
    end
    end
c= R0(2);

end

function grafico_erros(ypred,y) % Função para gerar os gráficos: séries
temporais, erros, tracking signals, CUSUM
    N = length(ypred);

```

```

y = y(end-N+1:end);
erro = ypred-y;
erro_abs = abs(erro);

mad(1)=erro_abs(1);
eac(1)=0;
ad(1)=erro_abs(1);
tsb(1)=0;
for t = 2:N
    eac(t) = erro(t) + eac(t-1);
    ad(t) = erro_abs(t) + ad(t-1);
    mad(t) = ad(t)/t;
    tsb(t) = eac(t)/mad(t); % Tracking signals
end

for t = 1:N
    ls(t) = 4; % limite superior
    li(t) = -4; % limite inferior
end

% Gráfico de séries temporais e previsão
figure
subplot(3,1,1)
plot(y)
axis('tight')
grid on
hold on
plot(ypred, '-.')
hold off
title('Time Series - BL 1')
legend(['Observed' "Forecast"], 'Location', 'bestoutside')
% ylabel("Values")

% Gráfico de erros
subplot(3,1,2)
stem(ypred - y)
grid on
% ylabel("Error")
title('Forecast Error')

% Gráfico Tracking signals
subplot(3,1,3)
stem(tsb)
grid on
hold on
plot(ls)
hold on
plot(li)
hold off
title('Tracking Signals')

end

function net = rna_mlp_train(N,y) % Função para treinar as redes neurais MLP

% Definir as variáveis entrada x(t) e saída y_p(t)
t = 1:N-1;
x(t) = y(t);
y_p(t) = y(t+1);

```

```

% Dividir os dados em treinamento, validação e teste
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;

% --- RNA MLP (Backpropagation) -----
titulo = 'MLP';
hiddenLayerSize = 10;
net = feedforwardnet(hiddenLayerSize, 'trainlm');

% Definir parâmetros
net.trainParam.show = 50;
net.trainParam.lr = 0.01;
net.trainParam.epochs = 1000;
net.trainParam.goal = 1e-6;

% Treinar a rede neural MLP
[net, tr] = train(net, x, y_p);

end

function y_pred = rna_mlp_test(net, y, n_pred) % Função para previsão dos
dados através da rede neural MLP

% Calcular a saída Y da rede neural
y_pred = net(y);

% Calcular a performance da rede neural
perf = perform(net, y, y_pred);

end

function y = ExemploSimulado(N, parametro, y_inic) % Função para gerar as
séries temporais BL 1

% Parâmetros
a = 0.7; % inicial

if (parametro ~= 1) % série temporal inicial

    m = 0; % média
    d = 1; % desvio padrão

    er = m + d*rand(N,1);

    y(1) = er(1);
    y(2) = er(2);

    for t=3:N
        y(t) = a*y(t-1)*er(t-2)+er(t);
    end

else % série temporal com parâmetros modificados

    m = 0.5; % média
    d = 1;% desvio padrão
    er = m + d*rand(N,1);

```

```

        y(1) = a*y_inic*er(49)+er(1); % manter a continuidade
        y(2) = a*y(1)*er(50)+er(2);

        for t=3:N
            y(t) = a*y(t-1)*er(t-2)+er(t);
        end
    end
end

```

NMA

```

clear all; clc; close all;

% Inputs
tempo = 50; % número de t
n_pred = 50; % número de predições
limite = 100; % número de conjunto de dados

% Para reprodutibilidade
rng (1)

for i=1:limite; % Repetição das funções para todos conjuntos de dados

    y_01 = ExemploSimulado(tempo,0);
    rede_mlp = rna_mlp_train(tempo,y_01);
    y_pred_mlp_01 = rna_mlp_test(rede_mlp, y_01, n_pred);

    y_inic = y_01(end);
    y_02 = ExemploSimulado(tempo,1, y_inic);
    y_pred_mlp_02 = rna_mlp_test(rede_mlp, y_02, n_pred);

    qtd_fora_lim_mlp01_tsb(i,:) = [cont_tsb(y_pred_mlp_01,y_01,i)];
    qtd_fora_lim_mlp02_tsb(i,:) = [cont_tsb(y_pred_mlp_02,y_02,i)];

end

rng(2)
for i=1:limite; % Repetição das funções para todos conjuntos de dados

    y_011 = ExemploSimulado(tempo,0);
    y_pred_mlp_011 = rna_mlp_test(rede_mlp, y_011, n_pred);

    qtd_fora_lim_mlp011_tsb(i,:) = [cont_tsb(y_pred_mlp_011,y_01,i)];

end

% Armazenar resultados em Excel
xlswrite('Resultado_MLP01_NMA', qtd_fora_lim_mlp01_tsb);
xlswrite('Resultado_MLP02_NMA', qtd_fora_lim_mlp02_tsb);
xlswrite('Resultado_MLP011_NMA', qtd_fora_lim_mlp011_tsb);

%Gerar gráficos das séries não lineares
y_real = [y_01 y_01 y_02];
% y_pred_nar = [y_pred_nar_01 y_pred_nar_02];
y_pred_mlp = [y_pred_mlp_01 y_pred_mlp_011 y_pred_mlp_02];
plot(y_real)

```

```

hold on
plot(y_pred_mlp)
hold off
title('Time Serie - NMA')
legend(["Time Series" "ANN"],'Location', 'southwest')
exportgraphics(gcf,'NMA.jpg','Resolution',300);

% Gerar gráficos dos erros para um conjunto de dado
grafico_erro(y_pred_mlp_01,y_01)
% exportgraphics(gcf,'Error_01.jpg','Resolution',300);
grafico_erro(y_pred_mlp_011,y_011)
% exportgraphics(gcf,'Error_02.jpg','Resolution',300);
grafico_erro(y_pred_mlp_02,y_02)
% exportgraphics(gcf,'Error_02.jpg','Resolution',300);

%-----
% Funções
%-----

function c = cont_tsb(ypred,y,i) % Função para contar qntd_fora_lim para
Tracking Signal TSB''

    N = length(ypred);
    y = y(end-N+1:end);
    erro = ypred-y;
    erro_abs = abs(erro);
    R0 = 0;
    ls = 4; % limite superior
    li = -4; % limite inferior

    mad(1) = erro_abs(1);
    eac(1) = 0;
    tsb(1) = 0;
    ad(1) = erro_abs(1);

    for t = 2:N

        ad(t) = erro_abs(t) + ad(t-1);
        mad(t) = ad(t)/t; % Mean absolute deviation
        eac(t) = erro(t) + eac(t-1); % Soma acumulada
        tsb(t) = eac(t)/mad(t); % Tracking signals

        % Contar primeiro ponto não conforme (fora dos limites)
        if (tsb(t) > ls || tsb(t) < li)
            R0=[R0+1;t];
            break
        else
            R0=[0;t];
        end

    end

    c= R0(2);

end

```

```

function grafico_erros(ypred,y) % Função para gerar os gráficos: séries
temporais, erros, tracking signals, CUSUM
    N = length(ypred);
    y = y(end-N+1:end);
    erro = ypred-y;
    erro_abs = abs(erro);

    mad(1)=erro_abs(1);
    eac(1)=0;
    ad(1)=erro_abs(1);
    tsb(1)=0;

    for t = 2:N
        eac(t) = erro(t) + eac(t-1);
        ad(t) = erro_abs(t) + ad(t-1);
        mad(t) = ad(t)/t;
        tsb(t) = eac(t)/mad(t); % Tracking signals
    end

    for t = 1:N
        ls(t) = 4; % limite superior
        li(t) = -4; % limite inferior
    end

    % Gráfico de séries temporais e previsão
    figure
    subplot(3,1,1)
    plot(y)
    axis('tight')
    grid on
    hold on
    plot(ypred,'.-')
    hold off
    title('Time Series - NMA')
    legend(['Observed' "Forecast"],'Location', 'bestoutside')
%     ylabel("Values")

    % Gráfico de erros
    subplot(3,1,2)
    stem(ypred - y)
    grid on
%     ylabel("Error")
    title('Forecast Error')

    % Gráfico Tracking signals
    subplot(3,1,3)
    stem(tsb)
    grid on
    hold on
    plot(ls)
    hold on
    plot(li)
    hold off
    title('Tracking Signals')

end

```

```

function net = rna_mlp_train(N,y) % Função para treinar as redes neurais MLP

    % Definir as variáveis entrada x(t) e saída y_p(t)
    t = 1:N-1;
    x(t) = y(t);
    y_p(t) = y(t+1);

    % Dividir os dados em treinamento, validação e teste
    net.divideParam.trainRatio = 80/100;
    net.divideParam.valRatio = 10/100;
    net.divideParam.testRatio = 10/100;

    % --- RNA MLP (Backpropagation) -----
    titulo = 'MLP';
    hiddenLayerSize = 10;
    net = feedforwardnet(hiddenLayerSize,'trainlm');

    % Definir parâmetros
    net.trainParam.show = 50;
    net.trainParam.lr = 0.01;
    net.trainParam.epochs = 1000;
    net.trainParam.goal = 1e-6;

    %Treinar a rede neural MLP
    [net,tr] = train(net,x,y_p);

end

function y_pred = rna_mlp_test(net, y, n_pred) % Função para previsão dos
dados através da rede neural MLP

    % Calcular a saída Y da rede neural
    y_pred = net(y);

    % Calcular a performance da rede neural
    perf = perform(net,y, y_pred);

end

function y = ExemploSimulado(N, parametro, y_inic) % Função para gerar as
séries temporais NMA

    % Parâmetros
    a = 1.0; % inicial
    b = 0.3; % inicial
    c = 0.2; % inicial
    d = 0.4; % inicial
    e = 0.25; % inicial

    if (parametro ~= 1) % série temporal inicial

        m = 0; % média
        d = 1;% desvio padrão

        er = m + d*rand(N,1);

        y(1) = er(1);
        y(2) = a*er(2)-b*er(1);
    end
end

```

```

    for t=3:N
        y(t) = a*er(t)-b*er(t-1)+c*er(t-2)+d*er(t-1)*er(t-2)-e*(er(t-
2).^2);
    end

    else % série temporal com parâmetros modificados

        m = 0.5; % média
        d = 1;% desvio padrão

        er = m + d*rand(N,1);

        y(1) = a*er(1)-b*er(50)+c*er(49)+d*er(50)*er(49)-e*(er(49).^2);
        y(2) = a*er(2)-b*er(1)+c*er(50)+d*er(1)*er(50)-e*(er(50).^2);

        for t=3:N
            y(t) = a*er(t)-b*er(t-1)+c*er(t-2)+d*er(t-1)*er(t-2)-
e*(er(t-2).^2);
        end
    end
end
end

```


APÊNDICE C – Aplicação do Previsor nos dados de TOG em MATLAB

```
% Aplicação TOG
clear all; clc; close all;

% Para reprodutibilidade
rng default

% Arquivo e diretório atual
dFolder = 'C \';
filename = 'TOG.xlsx';
A(:) = xlsread (filename, 'A1:A50');
B(:) = xlsread (filename, 'A55:A100');
%xlswrite('TOG',A)

%Inputs
tempo = 50;
n_pred_A = 50;
n_pred_B = 50;

%Treinar a rede neural
rede_mlp = rna_mlp_train(tempo,A);

%Prever a rede neural inicial
y_pred_A = rna_mlp_test(rede_mlp, A, n_pred_A);
y_pred_B = rna_mlp_test(rede_mlp, B, n_pred_B);

%Aplicar tracking signals
% qtd_fora_lim_togA(:) = [cont_tsb(y_pred_A,A)]
% qtd_fora_lim_togB(:) = [cont_tsb(y_pred_B,B)];

% Gerar gráficos das séries não lineares
% y_real = [A];
% y_pred = [y_pred_A];
y_real = [A B];
y_pred = [y_pred_A y_pred_B];
plot(y_real)
hold on
plot(y_pred)
hold off
title('Time Serie - TOG')
legend(["Time Series" "ANN"], 'Location', 'southwest')
% exportgraphics(gcf, 'STAR1.jpg', 'Resolution', 300);

% Gerar gráficos dos erros para um conjunto de dado
grafico_erros(y_pred_A,A)
grafico_erros(y_pred_B,B)
grafico_erros(y_pred,y_real)
% exportgraphics(gcf, 'Error_01.jpg', 'Resolution', 300);

%-----
% Funções
%-----
```

```

function grafico_erros(ypred,y) % Função para gerar os gráficos: séries
temporais, erros, tracking signals, CUSUM
    N = length(ypred);
    y = y(end-N+1:end);
    erro = ypred-y;
    erro_abs = abs(erro);

    mad(1)=erro_abs(1);
    eac(1)=0;
    ad(1)=erro_abs(1);
    tsb(1)=0;

    for t = 2:N
        eac(t) = erro(t) + eac(t-1);
        ad(t) = erro_abs(t) + ad(t-1);
        mad(t) = ad(t)/t;
        tsb(t) = eac(t)/mad(t); % Tracking signals
    end

    for t = 1:N
        ls(t) = 4; % limite superior
        li(t) = -4; % limite inferior
    end

    % Gráfico de séries temporais e previsão
    figure
    subplot(3,1,1)
    plot(y)
    axis('tight')
    grid on
    hold on
    plot(ypred,'.-')
    hold off
    title('Time Series - TOG')
    legend(['Observed' "Forecast"],'Location', 'bestoutside')
%     ylabel("Values")

    % Gráfico de erros
    subplot(3,1,2)
    stem(ypred - y)
    grid on
%     ylabel("Error")
    title('Forecast Error')

    % Gráfico Tracking signals
    subplot(3,1,3)
    stem(tsb)
    grid on
    hold on
    plot(ls)
    hold on
    plot(li)
    hold off
    title('Tracking Signals')
end

function net = rna_mlp_train(N,y) % Função para treinar as redes neurais MLP

    % Definir as variáveis entrada x(t) e saída y_p(t)

```

```

t = 1:N-1;
x(t) = y(t);
y_p(t) = y(t+1);

% Dividir os dados em treinamento, validação e teste
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;

% --- RNA MLP (Backpropagation) -----
titulo = 'MLP';
hiddenLayerSize = 10;
net = feedforwardnet(hiddenLayerSize,'trainlm');

% Definir parâmetros
net.trainParam.show = 50;
net.trainParam.lr = 0.01;
net.trainParam.epochs = 1000;
net.trainParam.goal = 1e-6;
net.trainParam.max_fail = 100;

%Treinar a rede neural MLP
[net,tr] = train(net,x,y_p);

end

function y_pred = rna_mlp_test(net, y, n_pred) % Função para previsão dos
dados através da rede neural MLP

% Calcular a saída Y da rede neural
y_pred = net(y);

% Calcular a performance da rede neural
perf = perform(net,y, y_pred);

end

```