



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Camila Ascendina Nunes Kamei

**Uma abordagem evolucionária para hibridização de preditores**

Recife

2024

Camila Ascendina Nunes Kamei

## **Uma abordagem evolucionária para hibridização de preditores**

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Doutora em Ciência da Computação.

**Área de Concentração:** Inteligência Computacional

**Orientador):** Paulo Salgado Gomes de Mattos Neto

**Coorientadores:** João Fausto Lorenzato de Oliveira e Péricles Barbosa Cunha de Miranda

Recife

2024

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Kamei, Camila Ascendina Nunes.

Uma abordagem evolucionária para hibridização de preditores /  
Camila Ascendina Nunes Kamei. - Recife, 2024.

90p.: il.

Tese (doutorado) - Universidade Federal de Pernambuco. Centro  
de Informática. Programa de Pós-Graduação em Ciência da  
Computação, 2024.

Orientação: Paulo Salgado Gomes de Mattos Neto.

Coorientação: João Fausto Lorenzato de Oliveira.

Coorientação: Péricles Barbosa Cunha de Miranda.

1. Séries Temporais; 2. Modelos híbridos; 3. Combinação de  
Modelos Lineares e Não Lineares; 4. Algoritmo Evolucionário. I.  
Mattos Neto, Paulo Salgado Gomes de - Orientador. II. Oliveira,  
João Fausto Lorenzato de - Coorientador. III. Miranda, Péricles  
Barbosa Cunha de - Coorientador. IV. Título.

UFPE-Biblioteca Central

CDD 006.31

**Folha de aprovação:** Inserir a folha de aprovação enviada pela Secretaria do curso de Pós-Graduação. A folha deve conter a **data de aprovação**, estar **sem assinaturas** e em formato **PDF**.

Dedico este trabalho aos meus pais, Joana e Nilton, que sempre me inspiraram, incentivaram a minha curiosidade com grande criatividade e me impulsionaram a estudar e permanecer estudando, apesar das dificuldades.

## AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, que me deu forças e sabedoria para prosseguir com os estudos até aqui, apesar de todos os desafios que tive que superar.

Agradeço a meu esposo, João Pedro, pelo incentivo ao longo do tempo e pela paciência em minhas ausências devido ao estudo e a rotina de trabalho.

Tenho uma grande gratidão pelos meus pais, Joana e Nilton Nunes, que sempre me incentivaram, me apoiaram, me deram colo e me financiaram em vários momentos e são exemplos de força, determinação e sabedoria. Aos meus irmãos: Pouline, Renato e Nilton filho, por sempre me auxiliarem a descansar e ver além das minhas longas horas de estudo. A minha sobrinha, Maria Eduarda, por não me deixar esquecer que ciência também é descoberta e se maravilhar. A todos os meus familiares pelo apoio e incentivo.

As minhas amigas de infância: Amanda Barros, Cataliny Duarte e Anna Carolina Silva, por estarem ao meu lado, mesmo nos momentos de maior angústia no desenvolvimento do projeto. A minha cunhada de coração, Cláudia Luna, por todo apoio, desabafo e terapia sobre a pós-graduação, que foi fundamental para renovar minhas energias.

Aos meus amigos e grandes pesquisadores: Eraylson Galdino e Domingos Silva, por suportarem minhas dúvidas, meus medos, me incentivarem a continuar pesquisando e por serem maravilhosos exemplos de amigos e pesquisadores.

Aos meus orientadores, Paulo Salgado, Fausto Lorrenzato e Péricles Miranda, minha eterna gratidão, por terem me aceitado como aluna e terem me ensinado como pesquisar. Além disso, por terem me apoiado, me incentivado, cobrado e guiado ao longo do processo com muita humanidade.

A todos que participaram de forma e indireta na produção deste trabalho, meu mais sincero agradecimento.

"Não são as espécies mais fortes que sobrevivem, nem as mais inteligentes, e sim as mais suscetíveis às mudanças."(Charles Darwin).

## RESUMO

A previsão de séries temporais é uma tarefa desafiadora em aplicações do mundo real, pois as séries temporais podem ser compostas por padrões lineares e não lineares, apresentar comportamento heterocedástico e ter ruídos aleatórios. Assim, a utilização de um modelo único para a modelagem e previsão de uma série temporal pode não ser suficiente para capturar todos os padrões. Para aumentar a acurácia da previsão, são utilizados sistemas híbridos, que combinam modelos estatísticos lineares com modelos não lineares de aprendizagem de máquina. O funcionamento destes sistemas pode ser dividido em três etapas: modelagem linear da série temporal, previsão do resíduo utilizando um modelo de aprendizagem de máquina e a geração da previsão final através da combinação das fases anteriores. Este trabalho propõe um sistema híbrido que busca o melhor conjunto de previsões dos modelos estatísticos e de aprendizagem de máquina com o objetivo de maximizar a sua acurácia. Para isso, foi proposta uma abordagem que busca a melhor combinação ponderada das componentes linear e não linear da série temporal, podendo utilizar a previsão atual e as estimativas passadas. O método proposto realiza: (i) modelagem linear da série temporal, (ii) modelagem não linear do resíduo e (iii) utiliza um algoritmo genético modificado, capaz de buscar qual previsão deve ser usada e o seu respectivo peso. Assim, a fase de combinação seleciona e pondera a contribuição de cada previsão na fase (iii). As simulações experimentais foram realizadas utilizando oito séries temporais do mundo real e o desempenho do sistema foi avaliado através da métrica erro quadrático médio. Os resultados mostram que a abordagem proposta obtém desempenho superior quando comparado com os modelos únicos e híbridos da literatura.

**Palavras-chaves:** Séries Temporais, Modelos híbridos, Combinação de Modelos Lineares e Não lineares, Algoritmo Evolucionário.

## ABSTRACT

Time series forecasting is a challenging task in real-world applications, as time series can be composed of linear and nonlinear patterns, present heteroscedastic behavior, and have random noise. Therefore, using a single model for modeling and forecasting a time series may not be enough to capture all patterns. To increase forecast accuracy, hybrid systems are used, which combine a linear model with a nonlinear model. The operation of these models can be divided into three stages: linear modeling of the time series, forecasting the residual using a machine learning model, and the final forecast is obtained by combining the previous stages. This work aims to identify the importance of each value and which values are part of the final forecast, aiming to improve the result. To this end, an approach was proposed that seeks the best weighted combination of the linear and nonlinear components of the time series, using both the current and past values of the series to compose the forecast. The proposed method performs: (i) linear modeling of the time series, (ii) nonlinear modeling of the residue and (iii) uses a modified genetic algorithm, with simple implementation and capable of searching for different types of numbers, to find the best weighting for the predictions made in the previous steps. The experimental simulations were performed using eight real-world time series and the performance of the model was evaluated through the mean squared error metric. The results show that the proposed approach obtains superior performance when compared to the single and hybrid models in the literature.

**Keywords:** Time Series, Hybrid Models, Model Combination, Evolutionary Algorithms.

## LISTA DE FIGURAS

Figura 1 – Etapa de Treinamento - Modelo Proposto. . . . .	37
Figura 2 – Etapa de Teste - Modelo Proposto. . . . .	38
Figura 3 – Algoritmo Genético Adaptativo Desenvolvido. . . . .	40
Figura 4 – Representação Cromossomo - exemplo. . . . .	41
Figura 5 – Cromossomo - exemplo. . . . .	42
Figura 6 – Mutação exclusiva - exemplo. . . . .	47
Figura 7 – Mutação inclusiva - exemplo. . . . .	47
Figura 8 – Busca local - Diagrama. . . . .	52
Figura 9 – Combinação de previsões . . . . .	57
Figura 10 – Comparação da melhor versão do Modelo Proposto com o modelo base (ARIMA/SARIMA) e as Séries Temporais (a) Carsales, (b) Electricity, (c) Lake, (d) Gas, (e) Nordic, (f) Pigs, (g) Pollution e (h) Redwine. . . . .	87
Figura 11 – Melhor Fitness durante a execução do Algoritmo Genético para as Séries Temporais (a) Carsales, (b) Electricity, (c) Lake, (d) Gas, (e) Nordic, (f) Pigs, (g) Pollution e (h) Redwine. . . . .	89

## LISTA DE TABELAS

Tabela 1 – Descrição e características das séries utilizadas. . . . .	55
Tabela 2 – Divisão do conjunto de dados. . . . .	56
Tabela 3 – Parâmetros utilizados no trabalho proposto. . . . .	58
Tabela 4 – Parâmetros <i>Gridsearch</i> - modelos não lineares. . . . .	59
Tabela 5 – Avaliação do MSE das versões do método proposto aplicado ao conjunto de teste. . . . .	60
Tabela 6 – Diferença percentual em comparação com o modelo linear inicial. . . . .	61
Tabela 7 – Valores de erro quadrático médio da melhor versão da abordagem proposta em comparação com modelos únicos e híbridos da literatura aplicados ao conjunto de teste. . . . .	62
Tabela 8 – Diferença percentual entre a melhor versão da abordagem proposta e outros modelos. . . . .	64
Tabela 9 – Equações de combinação para as melhores versões da abordagem proposta. . . . .	65
Tabela 10 – Média de MSE obtida nos experimentos. . . . .	69
Tabela 11 – Desvio padrão obtido no experimentos. . . . .	69
Tabela 12 – Parâmetros do PSO. . . . .	70
Tabela 13 – Avaliação das versões do PSO - com e sem valores binários. . . . .	71
Tabela 14 – Valores de erro quadrático médio da melhor versão do PSO, modelo proposto e literatura de sistemas únicos e híbridos. . . . .	72
Tabela 15 – Equação de combinação para as melhores versões do PSO sem binários. . . . .	73
Tabela 16 – Equação de combinação para a melhor versão do PSO com binários. . . . .	73
Tabela 17 – Valores de MSE do conjunto de teste - Algoritmo de Mínimos Quadrados . . . . .	75
Tabela 18 – Máscara Aplicada ao Algoritmo de Mínimos Quadrados . . . . .	75
Tabela 19 – Pesos encontrados pelo Algoritmo de Mínimos Quadrados . . . . .	76
Tabela 20 – Avaliação da melhor versão da AG aplicada ao algoritmo de mínimos quadrados . . . . .	77

## LISTA DE ABREVIATURAS E SIGLAS

<b>ADF</b>	Teste de Dickey-Fuller Aumentado
<b>AG</b>	<i>Algoritmo Genético</i>
<b>AM</b>	Aprendizagem de Máquina
<b>AR</b>	<i>Autoregressive</i>
<b>ARIMA</b>	<i>Autoregressive Integrated Moving Average</i>
<b>ARMA</b>	<i>Autoregressive Moving Average</i>
<b>CNN</b>	<i>Convolutional Neural Network</i>
<b>EMD</b>	<i>Empirical Mode Decomposition Algorithm</i>
<b>ETS</b>	Modelo de Suavização Exponencial
<b>FAC</b>	<i>Função de Autocorrelação</i>
<b>FACP</b>	<i>Função de Autocorrelação Parcial</i>
<b>GL</b>	<i>Generalization Loss Class</i>
<b>LSTM</b>	<i>Long Short-Term Memory Network</i>
<b>MA</b>	<i>Moving Average</i>
<b>MLP</b>	<i>Multilayer Perceptron</i>
<b>MPTM</b>	Mutação Makinen, Periaux e Toivanen
<b>MSE</b>	<i>Mean Squared Error</i>
<b>PSO</b>	<i>Particle Swarm Optimization</i>
<b>SARIMA</b>	<i>Seasonal Autoregressive Integrated Moving Average</i>
<b>SBX</b>	Crossover Binário Simulado
<b>SKU</b>	<i>Stock Keeping Unit</i>
<b>SVR</b>	<i>Support Vector Machine for Regression</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
1.1	CONTEXTO E MOTIVAÇÃO	14
1.2	OBJETIVO	16
1.3	ESTRUTURA DA TESE	16
<b>2</b>	<b>TRABALHOS RELACIONADOS</b>	<b>18</b>
2.1	SÉRIES TEMPORAIS	18
2.2	MODELAGEM E PREVISÃO DE SÉRIES TEMPORAIS	21
<b>2.2.1</b>	<b>Modelos Lineares</b>	<b>22</b>
<b>2.2.2</b>	<b>Modelos de Aprendizado de Máquina</b>	<b>25</b>
2.2.2.1	<i>Rede Neural Perceptron de Múltiplas Camadas</i>	25
2.2.2.2	<i>Regressão por Vetores de Suporte</i>	27
2.3	SISTEMAS HÍBRIDOS QUE UTILIZAM MODELAGEM DE RESÍDUOS	29
2.4	ALGORITMOS GENÉTICOS - <b>AG</b>	32
2.5	RESUMO DO CAPÍTULO	34
<b>3</b>	<b>MÉTODO PROPOSTO</b>	<b>36</b>
3.1	ABORDAGEM PROPOSTA	36
3.2	ALGORITMO GENÉTICO - DESENVOLVIDO	39
<b>3.2.1</b>	<b>Inicialização</b>	<b>40</b>
<b>3.2.2</b>	<b>Reprodução</b>	<b>43</b>
<b>3.2.3</b>	<b>Diversidade</b>	<b>50</b>
<b>3.2.4</b>	<b>Busca Local</b>	<b>51</b>
<b>3.2.5</b>	<b>Injeção Randômica</b>	<b>52</b>
<b>3.2.6</b>	<b>Crítérios de Parada</b>	<b>53</b>
<b>3.2.7</b>	<b>Métricas de Balanceamento</b>	<b>53</b>
3.3	RESUMO DO CAPÍTULO	54
<b>4</b>	<b>ANÁLISE EXPERIMENTAL</b>	<b>55</b>
4.1	PROTOCOLO EXPERIMENTAL	55
4.2	RESULTADOS EXPERIMENTAIS	60
<b>4.2.1</b>	<b>Análise das Versões da Abordagem Proposta</b>	<b>60</b>
<b>4.2.2</b>	<b>Comparação com a literatura</b>	<b>62</b>

4.3	EQUAÇÕES DE COMBINAÇÃO . . . . .	64
4.4	RESUMO DO CAPÍTULO . . . . .	67
<b>5</b>	<b>ARQUITETURA DO CROMOSSOMO, EFICÁCIA DO ALGORITMO GENÉTICO E AVALIAÇÃO ALGORITMO MÍNIMOS QUADRADOS</b>	<b>68</b>
5.1	EFICÁCIA DO ALGORITMO . . . . .	68
5.2	ARQUITETURA DO CROMOSSOMO . . . . .	70
5.3	AVALIAÇÃO ALGORITMO DE MÍNIMOS QUADRADOS . . . . .	74
5.4	RESUMO DO CAPÍTULO . . . . .	77
<b>6</b>	<b>CONCLUSÃO . . . . .</b>	<b>78</b>
6.1	CONSIDERAÇÕES FINAIS . . . . .	78
6.2	TRABALHOS FUTUROS . . . . .	79
	<b>REFERÊNCIAS . . . . .</b>	<b>80</b>
	<b>ANEXO A – GRÁFICOS COMPARATIVOS . . . . .</b>	<b>86</b>
	<b>ANEXO B – GRÁFICOS DE CONVERGÊNCIA – ALGORITMO GENÉTICO . . . . .</b>	<b>88</b>

# 1 INTRODUÇÃO

Este capítulo apresenta a introdução deste trabalho, compreendendo os seguintes tópicos: o contexto e a motivação da abordagem proposta na tarefa de previsão de séries temporais, objetivos de pesquisa, e a organização dos próximos capítulos.

## 1.1 CONTEXTO E MOTIVAÇÃO

Séries temporais têm um papel importante na tomada de decisões de várias áreas (BOX et al., 2016; HAJIRAHIMI; KHASHEI, 2019a), como finanças (IDREES; ALAM; AGARWAL, 2019), engenharia (YANG et al., 2017), agronomia (YIN et al., 2018), economia (KAN et al., 2020) e hidrologia (WIDIASARI; NUGROHO; WIDYAWAN, 2017). Desta forma, desenvolver uma abordagem que tenham alta acurácia em diferentes problemas de previsão do mundo real é uma tarefa desafiadora (ZHANG, 2003; de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019; L. et al., 2020; OLIVEIRA; SILVA; NETO, 2021; KHASHEI; BIJARI, 2010). A literatura de previsão de séries temporais tem destacado os sistemas híbridos que combinam modelos estatísticos e modelos de aprendizagem de máquina devido aos resultados da literatura e empíricos por eles obtidos (ZHANG, 2003; de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019; L. et al., 2020; OLIVEIRA; SILVA; NETO, 2021; KHASHEI; BIJARI, 2010; KONTOPOULOU et al., 2023; KAUR; PARMAR; SINGH, 2023).

O desenvolvimento destes sistemas híbridos são baseados em várias motivações. Primeiro que séries temporais do mundo real são geralmente compostas por padrões lineares e não lineares (ZHANG, 2003). Consequentemente, o uso de um modelo único pode levar a uma precisão inferior. Modelos lineares não são capazes de realizar a modelagem de padrões não lineares, modelos estatísticos paramétricos não lineares dependem de informações relacionadas ao problema e da correta configuração dos parâmetros para realizar a modelagem não linear (KURMI et al., 2023) e modelos de aprendizagem de máquina sozinho não são aptos a modelar os padrões lineares e não lineares. Segundo que, na prática, é difícil determinar qual o melhor modelo único para uma dada série temporal. O previsor pode ter um desempenho inferior na estimativa de padrões *out-of-sample* devido, por exemplo, à incerteza do modelo ou à mudança na estrutura temporal. Terceiro, um único modelo é dificilmente o mais apropriado para prever todos os fenômenos temporais (TASKAYA-TEMIZEL; CASEY, 2005), isso ocorre devido

---

à impossibilidade de realizar suposição prévia sobre o modelo de otimização assim como é apresentado no teorema *No Free Lunch* (HO; PEPYNE, 2001).

Os sistemas híbridos fazem uso da combinação de modelos lineares devido à segurança obtida pela metodologia Box & Jenkins, e de modelos de aprendizagem de máquina, porque eles podem ser vistos como modelos estatístico não paramétricos e não lineares, visto que são modelos capazes de modelar um rico conjunto de funções não lineares e são hábeis com um conjunto de dados vasto (WU et al., 2021). Estes sistemas que são compostos de três etapas: (a) modelagem linear da série temporal, (b) modelagem não linear dos resíduos da série, e (c) a combinação da previsão gerada nas etapas (a) e (b) para gerar o resultado. Os resíduos, ou série de erro, são calculados pela diferença entre a série atual e a previsão realizada pelo modelo linear. Na literatura, a adição simples é a função que mais é utilizada na etapa de combinação (c). Entretanto, alguns trabalhos apresentaram que considerar uma combinação tão simples pode piorar a previsão linear (de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019). Portanto, a etapa de combinação é crucial para obter um desempenho preciso.

Este trabalho propõe uma nova forma de realizar a combinação das componentes da série temporal - linear e não-linear. Este estudo desenvolveu uma abordagem para encontrar a melhor combinação ponderada entre a previsão da série temporal e os resíduos, utilizando para isto valores atuais e progressos. Para validar a proposta da abordagem, foram utilizados os modelos autorregressivo integrado de médias móveis (*Autoregressive Integrated Moving Average* (ARIMA)) ou sazonal autorregressivo integrado de médias móveis (*Seasonal Autoregressive Integrated Moving Average* (SARIMA)) para realizar a previsão linear e perceptron de multicamadas (*Multilayer Perceptron* (MLP)) ou regressão de vetores de suporte (*Support Vector Machine for Regression* (SVR)) para modelar a previsão não linear. Posteriormente, foi aplicado o *Algoritmo Genético* (AG) modificado desenvolvido para encontrar os melhores pesos na combinação linear dos valores provenientes dos modelos linear e não linear. Este algoritmo foi escolhido para encontrar a ponderação da abordagem devido à robustez, flexibilidade em relação à codificação em binários ou reais (GARCIA et al., 2022), escalabilidade apresentada por ele (SIQUEIRA et al., 2020) e além disto, é uma das meta-heurísticas mais utilizadas na literatura (PUCHTA; SIQUEIRA; KASTER, 2020).

A avaliação experimental é realizada utilizando oito séries temporais bem conhecidas do mundo real de diferentes domínios: Carsales, Electricity, Lake, Gas, Nordic, Pigs, Pollution e Redwine. Os resultados em termos do erro quadrático médio (*Mean Squared Error* (MSE)) apresentam que o modelo proposto (i) geralmente obtém maior precisão do que os sistemas

híbridos da literatura que usam modelagem residual, (ii) supera estatísticas tradicionais e modelos de previsão de aprendizagem de máquina na maioria das séries temporais, (iii) melhora a previsão de séries temporais realizada pelo modelo linear, e (iv) atinge resultados estatisticamente superiores em comparação aos modelos da literatura.

## 1.2 OBJETIVO

O objetivo deste trabalho é propor uma abordagem que busca o melhor conjunto de previsões dos modelos lineares e não lineares considerando estimativas atuais e passadas. A abordagem utiliza um algoritmo genético para selecionar quais previsões serão utilizadas na fase de combinação bem como seu respectivo peso. Dado este objetivo geral, os objetivos específicos são:

1. Investigar a combinação ponderada de preditores através do algoritmo genético;
2. Aplicar as técnicas desenvolvidas em séries temporais do mundo real;
3. Analisar o comportamento das técnicas propostas a partir da variação de seus parâmetros e de diferentes séries temporais;
4. Analisar o desempenho do método proposto juntamente as técnicas já existentes na literatura por meio de métrica.

## 1.3 ESTRUTURA DA TESE

A estrutura deste trabalho é composta por 5 capítulos conforme descritos a seguir:

**Capítulo 2 - Trabalhos Relacionados:** Neste capítulo são apresentados os conceitos sobre séries temporais e previsão de séries temporais. São apresentados os modelos lineares, os modelos de aprendizado de máquina que podem ser aplicados aos resíduos e o algoritmo evolucionário que pode ser aplicado na busca da combinação. Por fim, são exibidas as abordagens de combinação preditores da literatura que serviram de base para a construção do método proposto.

**Capítulo 3 - Metodologia Proposta:** Neste capítulo são apresentados os detalhes da estratégia para o treinamento e teste da abordagem proposta. Desta forma são apresentados diagramas e exemplos com o principal objetivo de facilitar o entendimento e a replicabilidade.

Além disto, é apresentado o algoritmo genético modificado para encontrar a melhor combinação dos modelos, bem como os parâmetros utilizados por ele, as operações envolvidas, condições de parada aplicados ao algoritmo e as otimizações desenvolvidas.

**Capítulo 4 – Análise Experimental:** Neste capítulo é apresentado o protocolo experimental utilizado no trabalho e são apresentados os resultados obtidos com a avaliação do o método proposto juntamente com as abordagens da literatura. Estes resultados são mensurados por meio da métrica de MSE (erro quadrático médio - *Mean Squared Error*). Ainda neste capítulo são apresentadas as equações de combinação que são resultados do modelo proposto.

**Capítulo 5 – Convergência e Arquitetura:** A convergência do algoritmo genético foi avaliada para a comprovação do uso de um algoritmo robusto no encontro da solução, bem como a arquitetura do cromossomo utilizado. Este capítulo apresenta a comprovação da convergência do algoritmo e da utilização da melhor arquitetura de cromossomo para obtenção de melhores resultados pelo algoritmo.

**Capítulo 6 - Conclusão:** Este capítulo apresenta a conclusão desta tese, e expõe pontos importantes deste trabalho e contribuições alcançadas levando em consideração os resultados obtidos. Por fim, são apresentadas novas pesquisas que podem ser desenvolvidas a partir desta pesquisa.

## 2 TRABALHOS RELACIONADOS

Este capítulo discorre sobre assuntos relacionados à previsão de séries temporais, explorando conceitos básicos e modelos amplamente aplicados nesta área. São exibidas estratégias para a combinação de preditores e por fim, são apresentados os conceitos de algoritmo evolucionário. Todos esses tópicos abordados servem de base para o método proposto nesta tese. Desta forma este capítulo é dividido nas seções: Séries Temporais (Seção 2.1), Modelagem e Previsão de Séries Temporais (Seção 2.2), Sistemas Híbridos que utilizam modelagem de resíduos (Seção 2.3), Algoritmos Genéticos (Seção 2.4) e Resumo (Seção 2.5).

### 2.1 SÉRIES TEMPORAIS

Uma série temporal consiste em um conjunto de observações geradas sequencialmente ao longo do tempo (BOX et al., 2016), (COWPERTWAIT; METCALFE, 2009), (BROCKWELL; DAVIS, 2006), podendo ser discreta ou contínua. A série temporal é discreta, quando as observações são realizadas em intervalos iguais em um determinado espaço de tempo (BROCKWELL; DAVIS, 2006), por exemplo, valores mensais de temperatura da cidade de Garanhuns - PE. Uma série temporal é contínua, quando as observações são registradas continuamente ao longo de algum intervalo de tempo (BROCKWELL; DAVIS, 2006), como, por exemplo, as ondas cardíacas obtidas no exame de eletrocardiograma (CONTINUOUS... , 2020). Todavia, é possível aproximar uma série contínua em discreta por meio de amostragem (MORETTIN; TOLOI, 2006). Neste procedimento, é necessário capturar intervalos de tempo iguais  $\Delta$ , utilizando a mesma unidade temporal (por exemplo: hora, dia ou mês), obtendo assim uma série discreta com  $N$  pontos observados no intervalo  $[0, T]$ , no qual  $N = \frac{T}{\Delta}$ .

Uma série temporal discreta pode ser representada pelo vetor  $\mathbf{Z}_t = \{z_t, t = 1, \dots, N\}$ , onde  $z_t$  corresponde ao valor obtido no tempo  $t$  e  $N$  corresponde ao número de observações (OLIVEIRA; LUDERMIR, 2015).

Além disto, as séries temporais também podem apresentar um ou mais componentes, como: sazonalidade, tendência, ciclos e irregularidades (COWPERTWAIT; METCALFE, 2009; ADHIKARI; AGRAWAL, 2013). Cada um destes componentes é apresentado em detalhes a seguir:

- Sazonalidade: são padrões que ocorrem num intervalo de tempo. A sazonalidade pode ser devido a fatores como estações do ano, feriados ou eventos regulares. Por exemplo,

as vendas de roupas de inverno podem ter uma sazonalidade anual, com picos durante os meses mais frios do ano;

- Tendência: indica a direção-geral dos dados em uma série temporal ao longo de um período de tempo. Pode ser ascendente, descendente ou estável. Por exemplo, se os dados de vendas de uma empresa mostram um aumento constante ao longo de vários anos, isso indica uma tendência de crescimento;
- Ciclos: são variações que não repetem de forma fixa (HYNDMAN; KHANDAKAR, 2008). Por exemplo, em séries temporais financeiras são exibidos os ciclos de prosperidade, declínio, depressão e recuperação;
- Irregularidades: também conhecidas como ruído, representam as variações aleatórias e não sistemáticas nos dados da série temporal.

Ademais das componentes da série temporal, existe um conceito importante para compreensão do comportamento da série, o qual é a estacionaridade. Esta refere-se a uma propriedade que descreve como as estatísticas resumidas dos dados (como média, variância e covariância) não mudam com o tempo (BOX et al., 2016). Duas formas de estacionaridade são consideradas: forte e fraca. A estacionaridade é forte quando a média, variância e covariância são constantes ao longo do tempo (SHUMWAY; STOFFER; STOFFER, 2000). Na estacionaridade fraca, a série temporal deve atender a constância em relação à média e variância, mas a covariância não precisa ser necessariamente constante.

Séries temporais do mundo real podem apresentar algum grau de não-estacionariedade, por exemplo, séries financeiras normalmente são não-estacionárias, porque apresentam, no caso mais simples, flutuações com inclinação positiva ou negativa (MORETTIN; TOLOI, 1981). Existem modelos que supõem que a série temporal seja estacionária, por exemplo, o *Autoregressive Moving Average* (ARMA). Desta forma, para utilizar esses modelos em séries temporais não-estacionárias, é necessário transformar a série em estacionária. Para isso, pode-se utilizar a função de diferenciação, que nada mais é do que diferenças sucessivas nos dados originais até obter uma série estacionária. A primeira diferença de  $z(t)$  é definida por:

$$Bz(t) = z(t - 1); \quad (2.1)$$

a segunda diferença:

$$B^2z(t) = B[Bz(t)] = B[z(t - 1)] = z(t - 2). \quad (2.2)$$

Geralmente são necessárias até duas diferenciações para obter-se uma série estacionária em relação a tendência (MORETTIN; TOLOI, 1981). Caso seja necessário, ao final do processo de modelagem, pode ser realizado o processo inverso da diferenciação para retornar a escala original.

Na análise de séries temporais, é comum separar esses quatro componentes (tendência, sazonalidade, ciclos e irregularidades) para entender melhor o comportamento dos dados ao longo do tempo. Esta análise pode ser realizada no domínio do tempo ou da frequência. Quando o estudo é realizado no domínio da frequência, é assumido que a série é formada por ondas seno e cosseno com diferentes frequências (BOX et al., 2016). Já quando a análise é feita no domínio do tempo, a série é observada através de seus componentes, com os seguintes objetivos: descrição, explicação, previsão e controle (CHATFIELD; XING, 2019).

- A análise descritiva corresponde a plotar as observações no tempo e então, obter medidas descritivas simples das principais propriedades da série;
- Análise explicativa ocorre quando as observações são feitas em duas ou mais variáveis, é possível usar a variação em uma série temporal para explicar a variação em outra série e isso pode levar a uma compreensão mais profunda do mecanismo que gerou uma determinada série temporal (CHATFIELD; XING, 2019);
- A análise preditiva consiste em prever os futuros valores da série analisada (CHATFIELD; XING, 2019). Isto pode ser realizado modelando dados anteriores através da definição de janela ou utilizando os lags específicos. Tradicionalmente usa-se a *Função de Autocorrelação* (FAC) ou a *Função de Autocorrelação Parcial* (FACP). A FAC realiza a correlação direta e indireta entre os intervalos de tempo, já a FACP realiza somente a correção direta, filtrando assim as componentes indiretas. Bem como são utilizadas as funções de autocorrelação, os algoritmos de busca podem ser utilizados para encontrar o tamanho da janela;
- Na análise de controle as séries temporais às vezes são coletadas ou analisadas para melhorar o controle sobre algum sistema físico ou econômico (CHATFIELD; XING, 2019). Esta análise está intimamente relacionada à análise de previsão em muitas situações (CHATFIELD; XING, 2019).

Neste trabalho é realizada a análise preditiva e mais detalhes sobre ela será dado na sessão a seguir.

## 2.2 MODELAGEM E PREVISÃO DE SÉRIES TEMPORAIS

A previsão de uma série temporal, que utilize o conceito de janela, consiste em encontrar o valor futuro da série, assim para uma série  $\mathbf{Z}$ , a estimativa de valor futuro da série  $\mathbf{Z}_{t+h}$ , onde  $h$  é o passo da previsão, é dada pela Equação 2.3.

$$\hat{z}_{t+h} = f(Z_t) + \epsilon_{t+1}. \quad (2.3)$$

Na equação 2.3,  $f(\cdot)$  equivale a um modelo de previsão que recebe como uma entrada a janela temporal  $(Z_t)$ ,  $\epsilon_{t+1}$  corresponde ao erro do modelo e retorna o valor predito  $\hat{z}_{t+h}$ . Quando se trabalha a série temporal com a subdivisão em janela, esta é formada por valores progressos da série e pode ser representado matematicamente pela Equação 2.4.

$$Z_t = (z_t, z_{t-1}, z_{t-2}, \dots, z_{t-m-1}), \quad (2.4)$$

em que,  $m$  corresponde aos retardos temporais, também conhecidos como *lags*, ao ponto a ser previsto. Uma das formas de realizar a escolha da quantidade de lags utilizadas na previsão de uma série temporal, é obter a correlação linear (ADHIKARI; AGRAWAL, 2013) através da FAC e FACP (BOX et al., 2016). Caso seja utilizado o modelo ARMA, o valor do tamanho da janela deverá ser definido de forma empírica.

A FAC é uma função  $\rho(\cdot)$  definida pela Equação 2.6 (BROCKWELL; DAVIS, 2002), na qual  $\gamma$  é a função de covariância para o *lag*  $h$ ,  $\mu$  a média da série e  $N$  o total de pontos da série.

$$\gamma(h) = N^{-1} \sum_{t=1}^{N-|h|} (z_{t+|h|} - \mu)(z_t - \mu), \quad (2.5)$$

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}. \quad (2.6)$$

A FACP é uma função  $\alpha(\cdot)$ , na qual  $\alpha(0) = 1$  e para  $h \geq 1$ ,  $\alpha(h) = \phi_{hh}$ ,  $\phi_{hh}$  é o último componente de:

$$\phi_h = \Gamma_h^{-1} \gamma_h, \quad (2.7)$$

$$\Gamma_h = [\gamma(i-j)]_{i,j=1}^h, \gamma_h = [\gamma(1), \gamma(2), \dots, \gamma(h)]', \quad (2.8)$$

em que  $\Gamma_h$  é a matriz de autocovariâncias e  $\gamma_h$  corresponde ao vetor de autocovariâncias.

Para a modelagem de séries temporais, na literatura são encontrados dois grupos de modelos popularmente utilizados para previsão de séries temporais: modelos lineares (BOX et al., 2016) e não lineares (ZHANG; PATUWO; HU, 1998).

Os modelos lineares são muito utilizados devido à implementação simples e capacidade de interpretação. Os modelos *Autoregressive* (AR), *Moving Average* (MA), ARIMA e SARIMA são exemplos de modelos lineares largamente aplicados na literatura (BOX et al., 2016), devido a metodologia de Box & Jenkins que assegura a modelagem linear da série temporal.

Os modelos não lineares tem sido explorados nas últimas décadas, principalmente as abordagens de Aprendizagem de Máquina (AM). A MLP e SVR são capazes de realizar uma modelagem não linear (ZHANG; PATUWO; HU, 1998; VAPNIK et al., 1997), não necessitam que a série seja estacionária e também são considerados como aproximadores universais (ZHANG; PATUWO; HU, 1998). Estes modelos foram escolhidos para a modelagem não linear da abordagem proposta por serem os modelos mais utilizados na literatura, facilitando assim a comparação com os modelos da literatura. Nas próximas subseções serão detalhados os modelos utilizados neste trabalho: ARIMA, SARIMA, MLP e SVR.

### 2.2.1 Modelos Lineares

Os modelos lineares escolhidos para compor este trabalho foram os modelos ARIMA (*Autoregressive Integrated Moving Average*) e SARIMA (*Seazonal Autoregressive Integrated Moving Average*). Cada um destes modelos serão explicados em detalhes a seguir.

O modelo **ARIMA** realiza a composição do modelo AR e MA, mas aplica a diferenciação da série, caso a série não seja estacionária. Assim, as séries que não são estacionárias passam pelo processo de diferenciação e se tornam estacionárias, podendo, portanto, serem analisadas pelos modelos anteriores.

O modelo ARIMA pode ser representado de forma matemática pela Equação 2.13, composto pelo modelo AR e modelo MA, inicialmente representados pelas Equações 2.9 e 2.10, respectivamente.

$$z_t = \phi_1 z_{t-1} + \phi_2 z_{t-2} + \dots + \phi_p z_{t-p} + \epsilon_t, \quad (2.9)$$

$$z_t = \theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} - \dots - \theta_q \epsilon_{t-q} + \epsilon_t, \quad (2.10)$$

aplicando a derivação, que foi apresentada na Equação 2.1 nos modelos, terá as Equações 2.11 e 2.12.

$$z_t = \phi_1 B z_t + \phi_2 B^2 z_t + \dots + \phi_p B^p z_t + \epsilon_t, \quad (2.11)$$

$$z_t = \theta_1 B \epsilon_t - \theta_2 B^2 \epsilon_t - \dots - \theta_q B^q \epsilon_t + 1, \quad (2.12)$$

simplicando as equações e adicionando a diferenciação da série, obterá a Equação 2.13.

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)(1 - B)^d z_t = c + (1 + \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q) \epsilon_t, \quad (2.13)$$

em que,  $z_t$  é o valor atual no tempo  $t$ ,  $B$  representa o operador de diferenciação,  $\phi_1, \dots, \phi_p$  são os parâmetros do modelo AR de ordem  $p$ ,  $\theta_1, \dots, \theta_q$  correspondem aos parâmetros do modelo MA de ordem  $q$ , a constante  $c$  representa o termo constante na equação que pode ser incluído dependendo da necessidade da modelagem. Para os erros aleatórios,  $(\epsilon_t)$  assume-se que são independente e identicamente distribuídos com média zero e variância constante  $\delta^2$ .

Já o modelo SARIMA é uma extensão do modelo ARIMA que leva em consideração a sazonalidade dos dados. Este modelo aplica a diferenciação nos dados até que a componente sazonal seja removida da série temporal (COWPERTWAIT; METCALFE, 2009). Este modelo é formado por quatro componentes, como apresentado a seguir.

- O termo polinomial AR não sazonal e sazonal de ordem  $p$  e  $P$ , como apresentam as Equações 2.14 e 2.15:

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \quad (2.14)$$

$$\Phi_p(B^s) = 1 - \Phi_1 B^s - \Phi_2 B^{s^2} - \dots - \Phi_p B^{s^P} \quad (2.15)$$

- Parte da MA não sazonal e sazonal da ordem  $q$  e  $Q$ , como mostra as Equações 2.16 e 2.17:

$$\theta_q(B) = 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q \quad (2.16)$$

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + \Theta_2 B^{s^2} + \dots + \Theta_Q B^{s^Q} \quad (2.17)$$

- O operador diferencial não sazonal com ordem  $d$  usado para eliminar tendências polinomiais, como apresenta a Equação 2.18:

$$(1 - B)^d \quad (2.18)$$

- O operador de diferenciação sazonal com ordem de  $D$  usada para eliminar padrões sazonais, assim como na Equação 2.19:

$$(1 - B^s)^D \quad (2.19)$$

Os parâmetros  $\phi$  e  $\theta$  são coeficientes do modelo ARMA,  $\Phi$  e  $\Theta$  são os coeficientes sazonais do ARMA,  $B$  corresponde ao operador de diferenciação, o qual afeta a série temporal  $z_t$ , como apresenta a Equação 2.20.

$$B^d z_t = z_{t-d} \quad (2.20)$$

Além disso, a forma geral do modelo  $SARIMA(p, d, q)(P, D, Q)_s$  para uma série  $z_t$  pode ser escrita como na Equação 2.21.

$$\phi_p(B)\Phi_P(B^s)(1 - B)^d(1 - B^s)^D z_t = \theta_q(B)\Theta_Q(B^s)\epsilon_t, \quad (2.21)$$

em que,  $s$  corresponde à duração da sazonalidade e  $\epsilon_t$  é ruído branco.

A modelagem dos modelos ARIMA e SARIMA segue a metodologia de Box e Jenkins (BOX et al., 2016), o que auxilia a obter uma modelagem mais acurada. Esta metodologia é composta pelas fases de: identificação, estimação e diagnóstico.

Na etapa de identificação são utilizados dois gráficos para mensurar a correlação entre as observações da série (PANKRATZ, 1983) e, assim são escolhidos os hiper-parâmetros do modelo  $(p, d, q)(P, D, Q)$ . A função de autocorrelação parcial (FACP) define os parâmetros  $p$  e  $P$  do modelo AR e a função de autocorrelação (FAC) define os parâmetros  $q$  e  $Q$  do modelo MA (MORETTIN; TOLOI, 2006). Já o parâmetro  $d$  é definido pelo número de diferenciações aplicadas a série até que ela se torne estacionária. O parâmetro  $D$  é definido pela quantidade de diferenciações necessárias para remover a componente de sazonalidade da série.

Na fase de estimação, os parâmetros são estimados via modelos matemáticos, como mínimos quadrados ou estimadores de máxima-verossimilhança. Finalmente, na etapa de diagnóstico, o modelo é avaliado para verificar se ele representa bem o comportamento da série

e se pode ser aplicado na previsão. Este diagnóstico é baseado na série de ruído (resíduo). Caso o resíduo da série corresponda a um ruído branco, então a modelagem foi realizada corretamente.

Modelos estatísticos possuem metodologias bem definidas para sua utilização, mas estes modelos assumem que a série temporal estudada é gerada por um processo linear, o que é muitas vezes inapropriado, pois a maioria das séries temporais do mundo real são não lineares (KHASHEI; BIJARI, 2010), (ZHANG, 2003).

## 2.2.2 Modelos de Aprendizado de Máquina

A aprendizagem de máquina é uma aplicação da Inteligência Artificial que proporciona aos sistemas a habilidade de aprender e melhorar com a experiência, sem ser explicitamente programado para a atividade. Duas técnicas de aprendizagem de máquina foram escolhidas para compor este trabalho: Rede Neural Perceptron de Múltiplas Camadas (MLP) e a Regressão por Vetores de Suporte (SVR). Cada um destes modelos serão descritos em detalhes nas seções seguir.

### 2.2.2.1 Rede Neural Perceptron de Múltiplas Camadas

Uma vantagem significativa da MLP sobre as outras classes de modelos não lineares é que MLP são aproximadores universais que podem aproximar uma larga classe de modelos de funções com um alto grau de acurácia (ZHANG, 2003). O poder destas redes vem do processamento paralelo das informações provenientes dos dados (KHASHEI; BIJARI, 2010). Nenhuma suposição anterior da forma do modelo é necessária para o processo de construção do modelo (KHASHEI; BIJARI, 2010), (HAYKIN, 2001).

Este modelo é caracterizado por uma rede de três camadas de unidades de processamento simples conectadas por ligações acíclicas (ZHANG, 2003). O relacionamento entre a saída ( $y_t$ ) e as entradas ( $y_{t-1}, y_{t-2}, \dots, y_{t-k}$ ) tem a representação matemática (KHASHEI; BIJARI, 2010) exposta na Equação 2.22.

$$y_t = w_0 + \sum_{j=1}^c w_j \cdot g \left( w_{0j} + \sum_{i=1}^k w_{ij} \cdot y_{t-i} \right) + \varepsilon_t, \quad (2.22)$$

em que,  $w_{ij}$  ( $i = 0, 1, 2, \dots, k, j = 1, 2, \dots, c$ ) e  $w_j$  ( $j = 0, 1, 2, \dots, c$ ) são parâmetros do

modelo, geralmente chamados de pesos. O parâmetro  $k$  corresponde ao número de nós de entrada. Já o parâmetro  $c$ , corresponde ao número de nós na camada oculta e  $g$  corresponde a função de ativação. Existem várias funções de ativação, cada uma indicada para a situação do neurônio na rede neural (KHASHEI; BIJARI, 2010). A camada de entrada não possui função de ativação, pois a sua atribuição é transferir as entradas para a camada oculta. As funções logística e tangente hiperbólica são as funções de ativação utilizadas neste trabalho como função de transferência da camada oculta, pois apresentam melhor desempenho (CHEN; WANG, 2020) e estão representadas nas Equações 2.23 e 2.24, respectivamente (KHASHEI; BIJARI, 2010).

$$Sig(x) = \frac{1}{1 + exp(-x)}, \quad (2.23)$$

$$Tanh(x) = \frac{1 - exp(-2x)}{1 + exp(-2x)}. \quad (2.24)$$

Uma vez que, o modelo da MLP, que foi apresentado na equação 2.22, realiza um mapeamento funcional não linear a partir das observações anteriores da série  $(y_{t-1}, y_{t-2}, \dots, y_{t-p})$  para gerar um valor futuro  $(y_t)$ , que é obtida a Equação 2.25.

$$\hat{y}_t = f(y_{t-1}, y_{t-2}, \dots, y_{t-k}, \mathbf{w}) + \varepsilon_t, \quad (2.25)$$

em que  $\mathbf{w}$  é um vetor com todos os parâmetros e  $f$  é a função determinada pela estrutura da rede neural e os pesos das conexões. Desta forma, a rede neural é equivalente um modelo autorregressivo não linear (ZHANG, 2003). Observe que a previsão apresentada é utilizada para previsões de um passo a frente.

A escolha do parâmetro  $c$  é dependente dos dados e não existe uma regra sistemática que auxilie nessa decisão (KHASHEI; BIJARI, 2010). A quantidade de lags utilizadas pela MLP é um dos parâmetros mais importantes a ser estimado, pois é ele o que determina a autocorrelação não linear na estrutura da série temporal (KHASHEI; BIJARI, 2010).

Uma vez que a estrutura da rede é especificada, a rede está pronta para o treinamento, que corresponde ao processo de estimação de parâmetros (ZHANG, 2003). Os parâmetros são estimados para reduzir o erro quadrático médio e isto é realizado por vários algoritmos de treinamento, como por exemplo, o algoritmo por retro propagação *backpropagation* (ZHANG, 2003) (HAYKIN, 2001).

O algoritmo de treinamento por retro propagação consiste em duas grandes etapas: propagação e retro propagação. Na etapa de propagação, um vetor de entrada é aplicado aos nós da rede e seu efeito se propaga através da rede, camada por camada e assim, é gerada uma saída como resposta real da rede (HAYKIN, 2001). Nesta etapa, os pesos sinápticos na rede são todos fixos. Já na etapa de retro propagação, os pesos da rede são todos ajustados de acordo com uma regra de correção do erro. A resposta real da rede é subtraída de uma resposta desejada (alvo) e isto resulta no sinal de erro. Este sinal é então propagado através da rede, na direção contrária das conexões da rede (HAYKIN, 2001). Os pesos sinápticos são ajustados para que a resposta real da rede se aproxime da resposta desejada (HAYKIN, 2001).

Para auxiliar na escolha dos pesos e evitar o sobre ajustamento e subajustamento, é utilizada a ferramenta estatística conhecida como validação cruzada. Neste método, o conjunto de dados disponível é dividido aleatoriamente em: conjunto de treinamento e conjunto de teste (HAYKIN, 2001). Posteriormente o conjunto de treinamento é dividido em dois subconjuntos distintos: conjunto de estimação e conjunto de validação. O conjunto de estimação é utilizado para selecionar o modelo, já o conjunto de validação é usado para testar ou validar o modelo (HAYKIN, 2001).

### 2.2.2.2 Regressão por Vetores de Suporte

A SVR é uma técnica de aprendizagem de máquina que apresenta estabilidade e robustez (DRUCKER et al., 1997), com capacidade de realizar previsões acuradas (OLIVEIRA; LUDERMIR, 2015). A característica de estabilidade do SVR está associada ao treinamento do modelo ser determinístico (DRUCKER et al., 1997).

A SVR é um modelo proposto por (DRUCKER et al., 1997) baseado na utilização do espaço de características em alta dimensão que penaliza a complexidade via um termo adicionado à função de erro (AHMED et al., 2010).

Para séries temporais um modelo de SVR poderia ser como o apresentado na Equação 2.26.

$$f(x) = (\mathbf{W} \times \phi(Z_t) + b), \quad (2.26)$$

em que,  $\mathbf{W}$  corresponde ao vetor de pesos,  $b$  corresponde ao limiar,  $\phi(x)$  corresponde a uma função kernel e  $x$  é a janela temporal. Janela esta que corresponde aos pontos da série temporal

utilizados na previsão. O treinamento do SVR é feito mediante programação quadrática e consiste em encontrar o conjunto de pesos  $\mathbf{W}$  e o limiar  $b$  que minimize a função de custo apresentada na Equação 2.27.

$$\frac{1}{2}\|w\|^2 + C \sum_{i=1}^l (\xi + \xi^*), \quad (2.27)$$

e que respeite as restrições apresentadas na Equação 2.28.

$$\begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i, \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0, \end{cases} \quad (2.28)$$

em que  $\xi$  e  $\xi^*$  correspondem as variáveis de folga, utilizadas para medir o custo dos erros de previsão. Já a variável  $C$  corresponde ao fator de regularização, usado para regularizar a complexidade e para definir a quantidade de desvios maiores que  $\xi$  que serão suportados. Desta forma, a variável  $\xi$  representa a função custo, contabilizada quando o valor predito cair dentro desta região e o valor de  $C$  está relacionado com a complexidade e suavização da função. Quanto maior o valor de  $C$ , maior é a complexidade da função, porque o modelo vai justar todos os pontos de dados mais precisamente, mesmo que isso resulte em um modelo mais complexo e possivelmente um *overfitting*, com uma captura maior da variação dos dados. Quanto menor o valor de  $C$ , mais suave será a função, pois modelo não vai tentar se ajustar a todas as variações de dados.

O uso de funções de *kernels* permite que a SVR realize um mapeamento não linear em espaços de dimensões maiores (OLIVEIRA; LUDERMIR, 2015). A função de regressão é na forma apresentada pela Equação 2.29.

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) k(x_i, x_j) + b, \quad (2.29)$$

em que  $\alpha$  e  $\alpha^*$  correspondem a multiplicadores de Lagrange e  $k(x_i, x_j)$  é uma função de kernel. Existem dois kernels não lineares na SVR: função de base radial (*radial basis function - RBF*) e polinomial (IZONIN et al., 2021). O kernel polinomial muitas vezes apresenta um tempo de operação melhor, mas com uma acurácia menor que o kernel RBF, por isso este é mais utilizado. O kernel RBF é amplamente utilizado em previsão de séries temporais e também é conhecido como kernel gaussiano. Este é apresentado na Equação 2.30.

$$k(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\gamma^2}\right), \quad (2.30)$$

em que  $\gamma$  corresponde ao desvio do kernel.

A SVR pode produzir uma solução única global e dependendo do tipo de kernel pode modelar séries temporais lineares e não lineares (OLIVEIRA; LUDERMIR, 2015).

Pode ser observado que o SVR utilizando o kernel RBF possui os seguintes parâmetros:  $C$ ,  $\xi$  e  $\gamma$ . Estes parâmetros precisam ser bem ajustados, a fim de evitar *underfitting* e *overfitting*.

### 2.3 SISTEMAS HÍBRIDOS QUE UTILIZAM MODELAGEM DE RESÍDUOS

Uma série temporal corresponde a uma sequência de observações no tempo (BOX et al., 2016). Estas observações podem ser representadas por  $Z_t$ . As observações no tempo podem ser discretas, com intervalos equidistantes de tempo.

Uma característica intrínseca das séries temporais, é que observações adjacentes são dependentes, geralmente (BOX et al., 2016). A análise de séries temporais consiste em estudar essa dependência. Para isto, são utilizados modelos para os dados da série temporal e estes modelos podem ser aplicados com diversas finalidades, entre elas, a previsão dos próximos valores da série temporal. Estes modelos podem ser lineares, não lineares ou a combinação destes modelos (modelos híbridos).

Zhang (2003) propôs um modelo híbrido de correção de erro, no qual é realizada uma combinação linear dos modelos linear e não linear de previsão. Assim, uma série  $Z_t$  seria composta pelo padrão linear ( $L_t$ ) e pelo padrão não linear ( $N_t$ ). Este modelo supõe que a combinação da série é dada pela adição da componente linear e não linear. Esta composição de modelos é definida pela Equação 2.31.

$$Z_t = L_t + N_t. \quad (2.31)$$

A obtenção de cada uma dessas componentes do modelo de Zhang é dada da seguinte forma: primeiro é obtida a previsão linear através do modelo ARIMA, depois é obtida a série de resíduos e é posteriormente realizada a previsão não linear do resíduo.

A previsão linear é realizada pelo modelo ARIMA, que utiliza a metodologia de Box & Jenkins, onde a série, quando a modelagem é feita corretamente, não apresenta correlação linear (de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019).

Com o modelo linear previsto é possível obter a série de erro ( $E_t$ ). Esta série é obtida através da diferença entre a série original ( $Z_t$ ) e a previsão linear ( $\hat{L}_t$ ), como apresenta a Equação 2.32.

$$E_t = Z_t - \hat{L}_t. \quad (2.32)$$

Como a componente linear foi prevista pelo modelo ARIMA, a série de resíduos não tem componentes lineares. Caso fossem aplicados testes como a função de correlação ou Box-Pierce, não seriam encontradas correlações lineares nos dados da série (de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019), não apresentando assim correlações lineares relevantes. Desta forma, o resíduo contém somente dados não lineares e, portanto, a previsão dele deverá ser realizada por um modelo não linear. Assim, um modelo não linear com  $n$  entradas da série de resíduo  $E_t$ , pode prever uma série não linear  $N_t$  como apresenta a Equação 2.33.

$$N_t = f(e_{t-1}, e_{t-2}, \dots, e_{t-n}) + \epsilon_t, \quad (2.33)$$

em que  $f(\cdot)$  representa a modelagem não linear realizada pela MLP e  $\epsilon_t$  representa o ruído branco, o qual não pode ser predito.

Por fim, a previsão da série  $\hat{Z}_t$  é realizada pela adição, que corresponde a uma composição linear da previsão linear  $\hat{L}_t$  e da previsão não linear  $\hat{N}_t$ , como apresentado na Equação 2.34.

$$\hat{Z}_t = \hat{L}_t + \hat{N}_t. \quad (2.34)$$

O trabalho de O. Santos Júnior, de Oliveira e de Mattos Neto (2019) propõe uma forma não linear de realizar a combinação dos padrões lineares e não lineares. No modelo de O. Santos Júnior, de Oliveira e de Mattos Neto (2019) a série é vista como uma composição linear e não linear de forma independente, em que primeiramente é realizada a previsão linear através do modelo ARIMA e representado por  $M_L(Z)_t$ . Posteriormente, é obtida a série de resíduo e assim é possível realizar a busca por padrões não lineares nestes dados, representado por  $M_{NL}(E)_t$ . Os padrões não lineares são obtidos através da modelagem da MLP ou da SVR. Para realizar a combinação dos modelos linear  $M_L(Z)_t$  e não linear  $M_{NL}(E)_t$ , é utilizada outro modelo MLP ou SVR. Vale ressaltar que o modelo de O. Santos Júnior, de Oliveira e de Mattos Neto (2019) utiliza a mesma quantidade de lags tanto no padrão linear, quanto no padrão não linear. Assim, a composição da série  $M_C$  é vista como:

$$M_C = f(S, R), \quad (2.35)$$

$$S = [M_L(Z)_{t+1}, M_L(Z)_t, \dots, M_L(Z)_{t+1-(lag-1)}],$$

$$R = [M_{NL}(E)_{t+1}, M_{NL}(E)_t, \dots, M_{NL}(E)_{t+1-(lag-1)}],$$

em que  $Z_t$  corresponde a série no tempo  $t$ .  $E_t$  representa o resíduo no tempo  $t$ .

A previsão da série é dada pela Equação 2.36.

$$\hat{Z}_{t+1} = M_C(S, R). \quad (2.36)$$

No trabalho de O. Santos Júnior, de Oliveira e de Mattos Neto (2019), apesar de ter sido oferecido maior flexibilidade de combinação dos modelos, todas as componentes da série possuem a mesma influência na previsão, pois todos os componentes da previsão possuem o mesmo peso. O trabalho Hajirahimi e Khashei (2019b) apresenta a composição ponderada de componentes da série.

Hajirahimi e Khashei (HAJIRAHIMI; KHASHEI, 2019b) apresentam a composição da série  $Z_t$  como uma combinação ponderada da componente linear e não linear, como apresenta a Equação 2.37.

$$Z_t = w_L L_t + w_{NL} N_t, \quad (2.37)$$

em que,  $w_L$  e  $w_{NL}$  representam os pesos aplicados ao modelo da composição.

Assim como nos trabalhos anteriores, é realizada primeiramente a modelagem da previsão linear  $\hat{L}_t$  com o modelo ARIMA, seguido da previsão não linear  $\hat{N}_t$  do resíduo  $E_t$ , empregando o modelo MLP para isto. A terceira etapa é encontrar o melhor conjunto de pesos para combinar as previsões lineares e não lineares. O algoritmo de mínimos quadrados é empregado neste trabalho para encontrar a ponderação de cada componente da série. Desta forma, a previsão final é apresentada na Equação 2.38.

$$\hat{Z}_t = w_L \hat{L}_t + w_{NL} \hat{N}_t. \quad (2.38)$$

Os modelos apresentados anteriormente apresentam quatro formas de realizar a combinação de previsões: uma combinação linear por meio de uma adição, uma combinação linear

empregando uma adição ponderada, uma combinação flexível (linear ou não linear) e uma combinação não linear mediante uma função não linear. Seriam estas as únicas formas de combinar os modelos? Será que uma combinação de modelos lineares e não lineares de forma ponderada e com observação de previsões anteriores não obteria uma melhor acurácia?

Pode-se observar que os trabalhos apresentados anteriormente visam obter resultados mais acurados, mas é desconhecido o sistema que gerou a série, uma vez que as técnicas aplicadas são *blackbox*, ou seja, o resultado é obtido, mas de que forma foi obtido não é transparente. Essa obscuridade é devida a utilização de modelos de combinação *blackbox*.

## 2.4 ALGORITMOS GENÉTICOS - **AG**

AG (*Genetic Algorithms* - **GA**) são algoritmos de busca baseados no mecanismo de seleção natural e genética natural (GOLDBERG, 1989). Este algoritmo se tornou popular porque é computacionalmente simples, de fácil implementação, mas poderoso na busca por melhores *fitness* em grandes espaços de busca (GOLDBERG, 1989), (FATYANOSA; ARITSUGI, 2021). Além disso, este algoritmo não é limitado por suposições restritivas sobre o espaço de busca (continuidade, existência de derivadas, unimodalidade, entre outros) (GOLDBERG, 1989). Este algoritmo foi selecionado para este trabalho, devido a capacidade de exploração dos grandes espaços de busca associado a facilidade de trabalhar com dois "universos" distintos (binários e reais).

Os algoritmos genéticos correspondem a um algoritmo evolucionário, desta forma, ele possui as etapas de um algoritmo evolucionário com algumas peculiaridades em relação aos operadores genéticos.

O primeiro AG desenvolvido foi o algoritmo genético canônico (*canonical genetic algorithm* - **CGA**). Ele foi proposto por Holland e possui as seguintes especificidades (ENGELBRECHT, 2007):

- Uma representação binária foi utilizada;
- A seleção proporcional foi utilizada para selecionar os pais para a recombinação;
- O cruzamento de um ponto foi empregado como principal método para produzir a prole;
- A mutação uniforme foi proposta como um operador, mas era um operador de pouca importância.

Posteriormente é que foi utilizado o poder de exploração do operador de mutação nos algoritmos genéticos (ENGELBRECHT, 2007).

Nos algoritmos genéticos atuais são utilizadas várias formas de representação além da representação binária, pode-se citar: vetores de números em ponto flutuante, vetores de números inteiros, matrizes de duas dimensões e vetores LISP (HERRERA; VERDEGAY, 1998). A representação em binário tem limitações quando se trabalha com espaços de busca contínuos, onde a representação em binário pode ficar muito grande, resultando em problemas de armazenamento e manipulação (PATHAN; PATSIAS; TAGARIELLI, 2018), além disto, a codificação em binário pode ocasionar o problema de *Hamming Cliffs* (alteração de mais de um bit para mudar para solução vizinha) (PATHAN; PATSIAS; TAGARIELLI, 2018). Essa representação é importante para a definição e aplicação de cada operador genético (*crossover* e mutação).

O operador de *crossover* nos algoritmos genéticos pode ser dividido em três categorias conforme a aridade do operador. Assim, três classes são possíveis: assexual, sexual e multi-recombinacional (ENGELBRECHT, 2007). *Crossover* assexual é quando a prole é gerada por meio de um único parente. *Crossover* sexual é quando dois parentes são utilizados para gerar um ou dois filhos. Já o *crossover* multi-recombinacional é quando mais de dois parentes são utilizados para gerar um ou mais filhos.

O operador de *crossover* de um ponto seleciona randomicamente um ponto de cruzamento, e os valores depois daquele ponto são trocados entre os dois parentes. O operador de *crossover* de dois pontos seleciona randomicamente duas posições e os valores entre estes pontos são trocados. No operador de *crossover* uniforme, os valores dos cromossomos são comparados individualmente entre ambos progenitores. Estes são trocados com uma probabilidade fixada.

Além do operador de *crossover*, o AG utiliza o operador de mutação para ampliar a exploração do espaço de busca, podendo ser aplicado a diferentes tipos de números, como binários e reais.

São exemplos de mutação em representações binárias (ENGELBRECHT, 2007): mutação uniforme ou aleatória (*uniform (random) mutation*) e mutação em ordem (*inorder mutation*). O operador de mutação uniforme seleciona posições aleatórias de bits e os valores dos bits correspondentes são negados. Já na mutação em ordem, são selecionados dois pontos de mutação aleatoriamente e apenas os bits entre esses pontos de mutação sofrem mutação aleatória.

A mutação uniforme realizada em representações de ponto flutuante é dada pela equação 2.39 (ENGELBRECHT, 2007).

$$x'_{ij}(t) = \begin{cases} \tilde{x}_{ij}(t) + \Delta(t, x_{max,j} - \tilde{x}_{ij}(t)) & \text{se o dígito aleatório é 0,} \\ \tilde{x}_{ij}(t) + \Delta(t, \tilde{x}_{ij}(t) - x_{min,j}(t)) & \text{se o dígito aleatório é 1,} \end{cases} \quad (2.39)$$

onde  $\Delta(t, x)$  retorna um valor randômico entre  $[0, x]$ .

A diversidade da população, dentro dos algoritmos genéticos, pode ser avaliada com métricas de distância, como, por exemplo: distância de Hamming (FATYANOSA; ARITSUGI, 2021) ou a distância euclidiana (SABAR et al., 2017). Essa métrica pode auxiliar na tomada de decisão no algoritmo e evitar a convergência prematura.

Os algoritmos genéticos são influenciados pela taxa de mutação e pela taxa de *crossover*. Usualmente estes valores são fixos, mas para se obter um melhor desempenho, pode-se controlar estes parâmetros dinamicamente (ENGELBRECHT, 2007).

Apesar do AG ser simples de implementar e poderoso na busca, mesmo em espaços de busca grandes, ele apresenta como desvantagem, a possibilidade de ficar preso em um ótimo local, resultando em soluções sub ótimas e convergência prematura (FATYANOSA; ARITSUGI, 2021). Algumas causas para estes problemas são: alta pressão de seleção, direções de busca limitadas ou perda de diversidade (FATYANOSA; ARITSUGI, 2021).

Para superar os problemas supracitados, pode-se utilizar várias estratégias, entre elas: métricas de diversidade e controle de parâmetros dinamicamente.

As métricas de diversidade são formas de vencer a falta de diversidade populacional que pode ocorrer no algoritmo, quando o espaço de busca possui vários ótimos locais e dessa forma o algoritmo pode ficar preso (TAKAHASHI, 2016). Um exemplo de métrica de diversidade pode ser encontrado no trabalho (SI; MIRANDA; BHATTACHARYA, 2021).

O controle de parâmetros dinamicamente corresponde a atualização dos parâmetros conforme a execução do algoritmo, por exemplo, pode ser utilizada a taxa de mutação e *crossover* com atualização dinâmica para que a exploração e exploração seja balanceada durante a execução do algoritmo.

## 2.5 RESUMO DO CAPÍTULO

Série temporal é um conjunto de variáveis expostas no tempo que podem ser afetadas por componentes chamados de sazonalidade, tendência, ciclos e irregularidades. A previsão de séries temporais pode utilizar modelos estatísticos e de aprendizagem de máquina que comu-

mente são caracterizados como modelos lineares e não lineares, respectivamente. Os modelos ARIMA e SARIMA são exemplos de modelos estatísticos lineares, já MLP e SVR são exemplos de abordagem de AM não lineares. Entretanto, ambos os modelos estatísticos e de AM possuem limitações. Os modelos estatísticos lineares podem encontrar dificuldade de modelar séries temporais reais, uma vez que estas normalmente apresentam padrões não lineares. Os modelos de AM, por sua vez, podem apresentar problemas, tais como má especificação de parâmetros, sobre ajuste e sub ajuste. Considerando esses problemas, surge a motivação de combinar modelos diferentes por meio da agregação de suas previsões, alcançando um resultado mais acurado que os resultados dos modelos separados. Desta forma, é comum utilizar métodos de combinação chamados de Sistemas Híbridos residuais. Os Sistemas Híbridos residuais criam um método sequencial que busca modelar os padrões lineares e não lineares da série temporal em etapas específicas. Para facilitar a busca pela combinação de modelos utilizada pelo sistema híbrido, podem ser utilizados algoritmos evolucionários.

### 3 MÉTODO PROPOSTO

Este capítulo apresenta a metodologia proposta para realizar os objetivos da pesquisa e está dividido nas seguintes seções: Seção 3.1, apresenta a arquitetura da abordagem estudada. Já a Seção 3.2, apresenta o algoritmo genético modificado para encontrar a melhor combinação de pesos. Por fim, na Seção 3.3 serão expostas considerações do capítulo.

#### 3.1 ABORDAGEM PROPOSTA

A abordagem proposta realiza a combinação dos modelos de previsão linear e não linear da série temporal. A combinação das previsões é executada em três etapas sequenciais:

1. Previsão da série temporal ( $Z_t$ ) usando um modelo linear ( $M_L$ );
2. Previsão do resíduo ( $E_t$ ) usando um modelo não linear ( $M_{NL}$ );
3. A combinação da previsão da série e do resíduo usando um modelo de combinação ( $M_C$ ), gerando assim, a previsão final.

O fluxo de funcionamento da abordagem proposta está dividida em duas etapas: treinamento e teste. Cada uma destas etapas são apresentadas nas Figuras 1 e 2, respectivamente.

Na fase de treinamento, a entrada do modelo corresponde ao conjunto de treinamento da série univariada ( $Z_t$ ). Primeiramente é realizado o treinamento do modelo linear ( $M_L$ ) utilizando a série temporal ( $Z_t$ ). Posteriormente, é encontrada a série de resíduos, através da diferença entre a série temporal ( $Z_t$ ) e a saída do ( $M_L$ ), como apresenta a Equação 3.1.

$$\mathbf{E}_t = \mathbf{Z}_t - \mathbf{M}_L(\mathbf{Z}_t). \quad (3.1)$$

A série de resíduo ( $E_t$ ), que foi obtida na etapa anterior, é utilizada no treinamento do modelo não linear ( $M_{NL}$ ). Com os modelos linear e não linear treinados, é possível realizar o treinamento do modelo de combinação ( $M_C$ ). Desta forma, a saída da etapa 1 corresponde a série  $M_L(Z_t)$  e o modelo linear ( $M_L$ ). Já a saída da etapa 2, corresponde a série de resíduo ( $E_t$ ) e o modelo não linear ( $M_{NL}$ ). Finalmente, a saída da etapa 3, equivale ao modelo de combinação ( $M_C$ ).

Após a fase de treinamento, onde os modelos foram treinados e gerados, entra-se na fase de teste, como apresenta a Figura 2.

Na fase de teste é empregado o conjunto de teste, que são dados da série que não foram aplicados na fase de treinamento, representado por  $(Z_q)$ . Nesta fase, aplica os modelos obtidos na etapa anterior para obter os valores de previsão. Primeiramente, é obtida a previsão linear  $(M_L(Z_q))$ . Com a série de resíduo, é conquistada a previsão não linear  $(M_{NL}(E_q))$ . Finalmente, partindo das previsões linear e não linear, é obtida a previsão final através do modelo de combinação  $(M_C)$ .

Como a previsão da série temporal se torna mais precisa quando combinada (HAJIRAHIMI; KHASHEI, 2019b), o modelo híbrido reúne as previsões dos modelos  $(M_L)$  e  $(M_{NL})$  de forma ponderada. A ponderação aplicada a cada um dos modelos é representada por:  $\mathbf{W}_L$  e  $\mathbf{W}_{NL}$ , respectivamente. A combinação escolhida foi a combinação linear (adição), porque métodos baseados em adição conseguem melhores resultados em vários conjuntos de dados (L. et al., 2020).

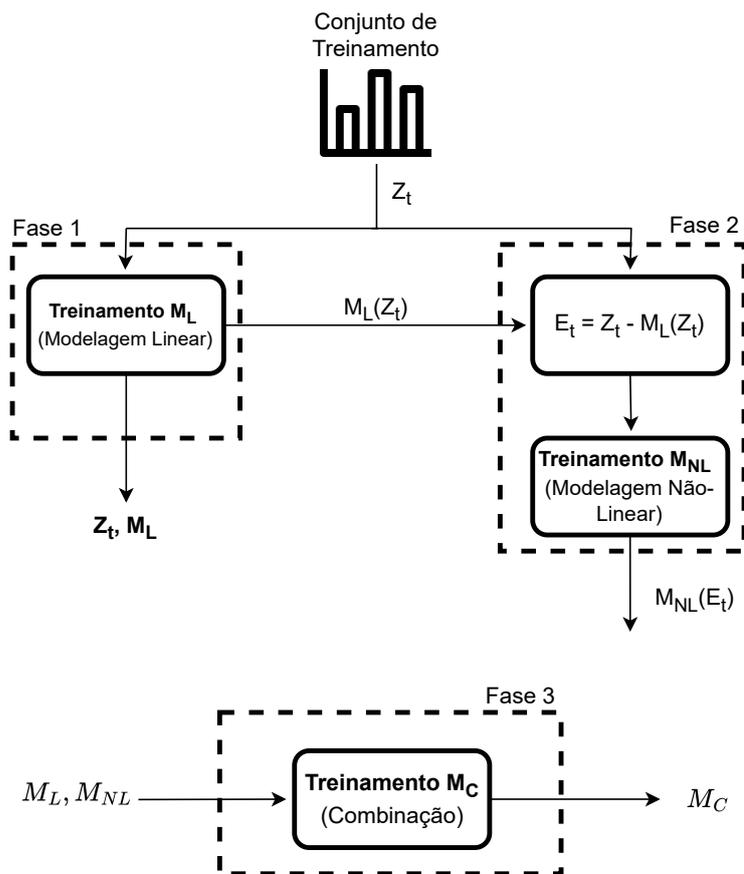


Figura 1 – Etapa de Treinamento - Modelo Proposto.

Fonte: Elaborada pela autora (2024).

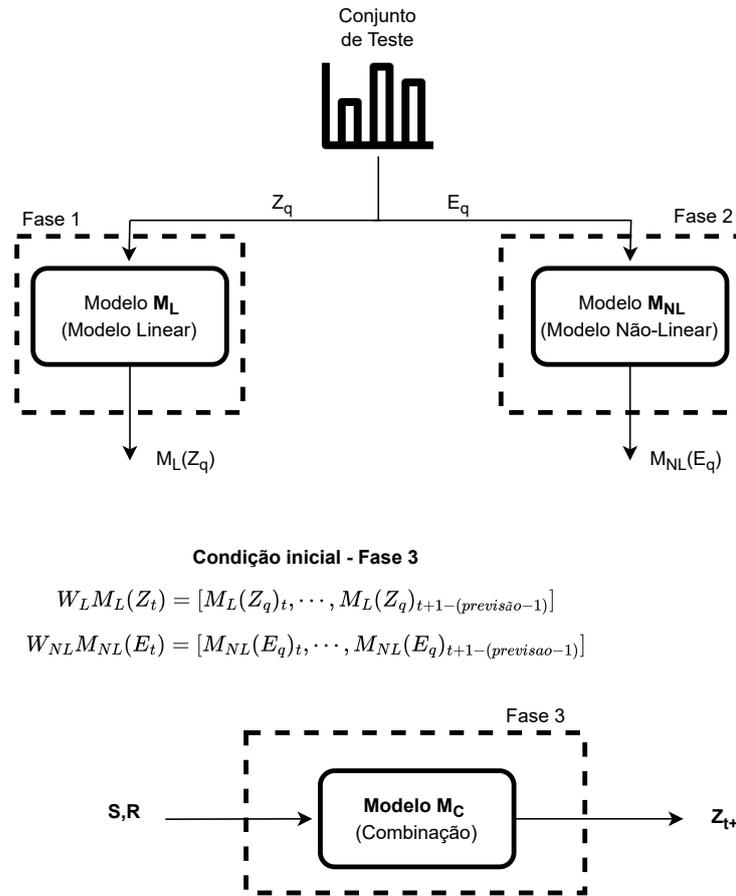


Figura 2 – Etapa de Teste - Modelo Proposto.

**Fonte:** Elaborada pela autora (2024).

Desta forma, o modelo  $M_C$ , descrito na Equação 3.2, procura pela melhor combinação linear entre a previsão linear da série temporal ponderada ( $W_L M_L(Z_t)$ ) e a previsão não linear do resíduo ponderado ( $W_{NL} M_{NL}(E_t)$ ) obtidos pelos modelos  $M_L$  e  $M_{NL}$  usando a previsão atual e um dado número de previsões anteriores ponderados ( $w_L$  - ponderação linear,  $w_{NL}$  - ponderação não linear).

$$M_C = \mathbf{W}_L \mathbf{M}_L(\mathbf{Z}_t) + \mathbf{W}_{NL} \mathbf{M}_{NL}(\mathbf{E}_t), \quad (3.2)$$

onde

$$\mathbf{W}_L \mathbf{M}_L(\mathbf{Z}_t) = [w_{L1} M_L(Z)_t, w_{L2} M_L(Z)_{t-1}, \dots, w_{Ln} M_L(Z)_{t+1-(previsão-1)}] \quad (3.3)$$

e

$$\mathbf{W}_{NL}\mathbf{M}_{NL}(\mathbf{E}_t) = [w_{NL1}M_{NL}(E)_t, w_{NL2}M_{NL}(E)_{t-1}, \dots, w_{NLn}M_{NL}(E)_{t+1-(previs\tilde{a}o-1)}]. \quad (3.4)$$

O objetivo dessa fase é a obtenção da previsão ( $Z_{t+1}$ ) e do modelo de combinação ( $M_C$ ), como apresentado na equação 3.5.

$$Z_{t+1} = M_C(\mathbf{W}_L\mathbf{M}_L(\mathbf{Z}_t), \mathbf{W}_{NL}\mathbf{M}_{NL}(\mathbf{E}_t)), \quad (3.5)$$

onde  $W_L M_L(Z_t)$  é um vetor com a previsão do tamanho da quantidade de previsões oriundo do modelo  $M_L$ ,  $W_{NL} M_{NL}(E_t)$  é um vetor de previsão com o tamanho da quantidade de previsões oriundo do modelo  $M_{NL}$  e estes são empregados como entrada do modelo  $M_C$ .

A abordagem proposta é flexível, pois ela fornece uma estrutura para que a previsão da série seja construída (PARTELOW, 2023) e assim, vários modelos lineares e não lineares podem ser aplicados para a obtenção da previsão final. Mas para validar o funcionamento deste trabalho os modelos ARIMA e SARIMA foram escolhidos para a previsão do padrão linear e para o padrão não linear foram utilizados os modelos MLP e SVR, pois estes são os modelos não lineares mais utilizados na literatura (de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019), o que facilita a comparação. A ponderação utilizada no modelo de combinação é descoberta através do algoritmo genético modificado.

O algoritmo genético será responsável por encontrar a melhor ponderação para as previsões anteriores, quais previsões anteriores que contribuem para a previsão e a melhor combinação dos modelos. Outra vantagem desta abordagem, é que o algoritmo genético apresenta as componentes da solução encontrada, o que facilita o entendimento do padrão gerador da série e fornece transparência a combinação (modelo *whitebox*).

## 3.2 ALGORITMO GENÉTICO - DESENVOLVIDO

O algoritmo genético desenvolvido neste projeto é formado de várias etapas, como apresentado na Figura 3. Para facilitar o entendimento de cada uma das etapas do algoritmo, esta sessão foi subdividida nas subseções: inicialização, reprodução, diversidade, busca local, injeção randômica, métricas de balanceamento, seleção de sobreviventes e critérios de parada. Cada uma destas etapas será detalhada a seguir.

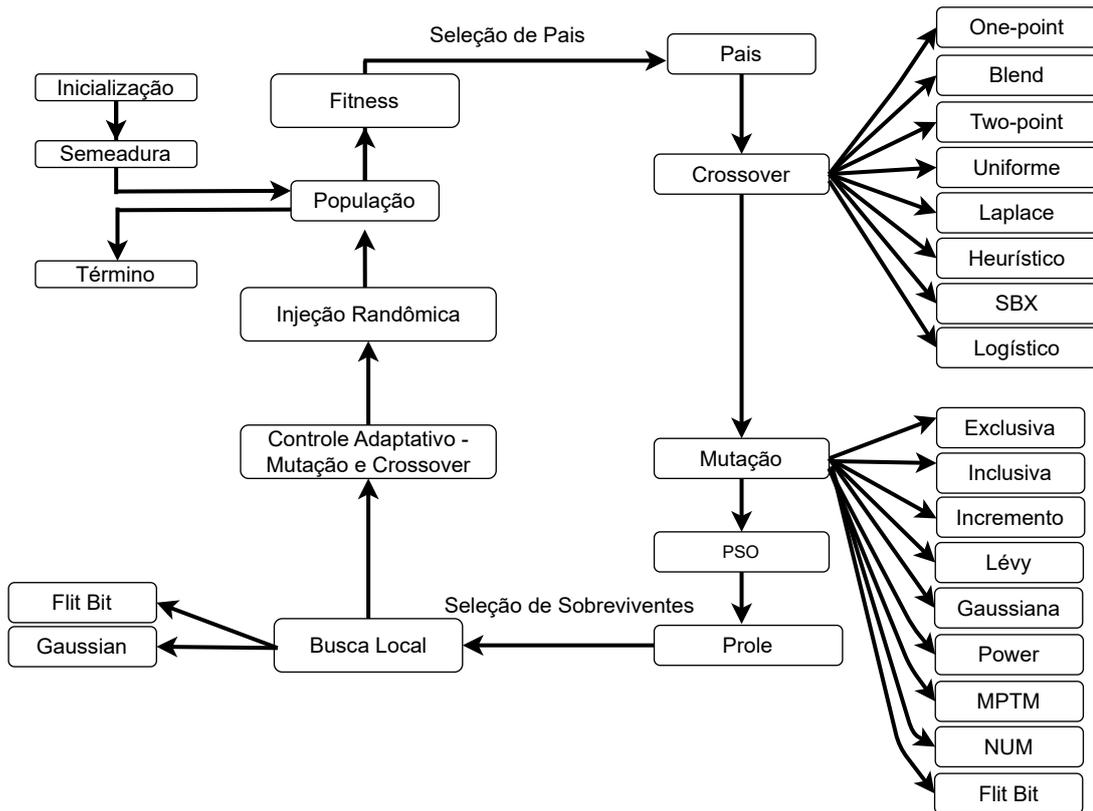


Figura 3 – Algoritmo Genético Adaptativo Desenvolvido.

Fonte: Elaborada pela autora (2024).

### 3.2.1 Inicialização

O algoritmo é iniciado com a criação dos indivíduos da população. Cada indivíduo é representado por um conjunto de genes (cromossomo) e um valor de *fitness* (aptidão).

Para compreender a composição dos genes utilizada no projeto, é necessário analisar a ponderação da abordagem proposta  $M_C$ . Para isto, será necessário decompor os pesos das Equações 3.7 e 3.8, como apresentado em 3.6.

$$M_C = \mathbf{W}_L \mathbf{M}_L(\mathbf{Z}_t) + \mathbf{W}_{NL} \mathbf{M}_{NL}(\mathbf{E}_t), \quad (3.6)$$

Em que,

$$\mathbf{W}_L \mathbf{M}_L(\mathbf{Z}_t) = (B_{L0} \cdot R_{L0} \cdot \hat{L}_0 + \dots + B_{Li} \cdot R_{Li} \cdot \hat{L}_i), \quad (3.7)$$

e

$$\mathbf{W}_{NL} \mathbf{M}_{NL}(\mathbf{E}_t) = (B_{NL0} \cdot R_{NL0} \cdot \hat{N}_0 + \dots + B_{NLi} \cdot R_{NLi} \cdot \hat{N}_i), \quad (3.8)$$

em que,  $B_{Li}$  e  $B_{NLi}$  ( $i = 0, 1, \dots$ , quantidade de previsões anteriores) correspondem ao peso binário, que indica a presença ou ausência do lag na previsão da série temporal para o modelo li-

near e não linear, respectivamente. Já  $R_{Li}$  e  $R_{NLi}$  ( $i = 0, 1, \dots$ , quantidade de previsões anteriores), são equivalentes à ponderação real desse lag na equação para o modelo linear e não linear, respectivamente.

A representação do cromossomo utilizado para cada valor de previsão linear e não linear no algoritmo genético, é apresentado na Figura 4.

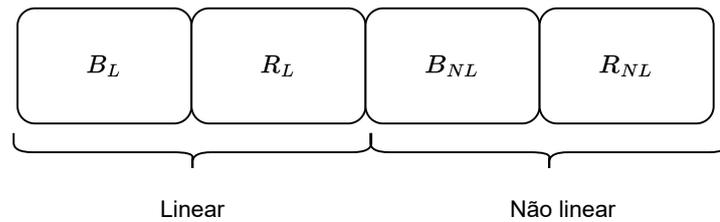


Figura 4 – Representação Cromossomo - exemplo.

**Fonte:** Elaborada pela autora (2024).

São utilizados 2 genes para o modelo linear e 2 genes para o modelo não linear para cada previsão utilizada no modelo de combinação.

Tomando como exemplo a combinação realizada com 4 pontos, em que são utilizados os pontos da previsão linear e não linear e os pontos da previsão anterior de cada modelo, a combinação é representada como a Equação 3.9.

$$L_t \cdot B_{L0} \cdot R_{L0} + L_{t-1} \cdot B_{L1} \cdot R_{L1} + N_t \cdot B_{NL0} \cdot R_{NL0} + N_{t-1} \cdot B_{NL1} \cdot R_{NL1}. \quad (3.9)$$

em que,  $B_{Li}$  ou  $B_{NLi}$  ( $i = 0, 1, \dots$ , quantidade de previsões anteriores) podem assumir os valores booleanos (0 ou 1), para representar a presença (valor igual à 1) ou ausência (valor igual à 0) do lag na previsão.  $R_{Li}$  ou  $R_{NLi}$  ( $i = 0, 1, \dots$ , quantidade de previsões anteriores) podem assumir valores reais, dentro do intervalo de  $[-2.0; 2.0]$ . O intervalo utilizado para atribuir os valores reais ainda se encontra em estudo, mas até o momento, este foi o intervalo que obteve melhores resultados.

O cromossomo que representa a combinação, tomada como exemplo, é apresentado na Figura 5.

Observa-se que para cada ponto da combinação são utilizadas duas ponderações: uma ponderação binária e outra real. Caso se deseje escalar o projeto e utilizar mais pontos, deve-se atentar para a ponderação está na quantidade correta.

Foram utilizados 20 genes para cada modelo utilizado no projeto, assim a quantidade máxima de previsões utilizada para cada modelo foi de 10 genes. Mas este valor pode ser

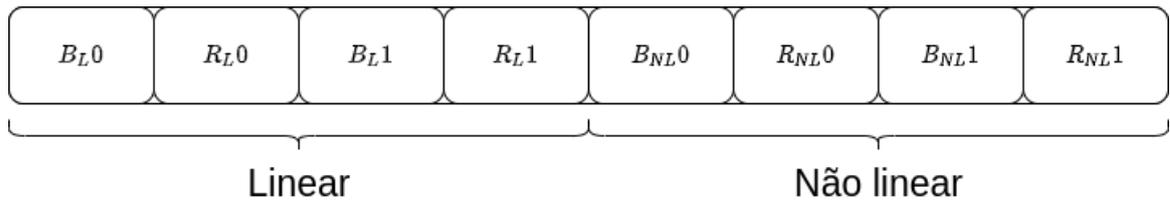


Figura 5 – Cromossomo - exemplo.

Fonte: Elaborada pela autora (2024).

obtido através da correlação cruzada, como no trabalho (de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019).

A geração dos indivíduos é realizada de duas formas: randômica e mediante sementeira. Na inicialização randômica os valores binários e reais são atribuídos aleatoriamente, respeitando os limites atribuídos como parâmetro. Já a sementeira, é baseada nos modelos de previsão de Zhang (ZHANG, 2003) e Hajirahimi (HAJIRAHIMI; KHASHEI, 2019b) e é aplicada somente a um indivíduo da população para cada modelo. Para a sementeira de Zhang, somente os valores da previsão atual de cada modelo é que possuem os valores binários iguais a 1 e a ponderação deles corresponde ao valor 1.0 (ou seja, sem ponderação), como apresenta a Equação 3.10 aplicada ao exemplo de 4 genes.

$$B_{L0} = B_{NL0} = 1 \quad (3.10)$$

$R_{L0} = R_{NL0} = 1$ , demais valores iguais a zero. Então:

$$(L_t \cdot 1 \cdot 1) + (L_{t-1} \cdot 0 \cdot 0) + (N_t \cdot 1 \cdot 1) + (N_{t-1} \cdot 0 \cdot 0).$$

Já na sementeira de Hajirahimi, somente os valores da previsão atual de cada modelo é que terá os valores binários iguais a 1 e a ponderação deles é obtida através do algoritmo de mínimos quadrados, como apresenta a Equação 3.11 aplicada ao exemplo.

$$B_{L0} = B_{NL0} = 1 \quad (3.11)$$

$R_{L0}$  = valor de peso obtido por mínimos quadrados

$R_{NL0}$  = outro valor de peso obtido por mínimos quadrados,

demais valores de pesos iguais a zero. Então:

$$(L_t \cdot 1 \cdot \text{minQuadrados}) + (L_{t-1} \cdot 0 \cdot 0) + (N_t \cdot 1 \cdot \text{minQuadrados}) + (N_{t-1} \cdot 0 \cdot 0).$$

Após a geração dos indivíduos da população, é realizado o cálculo do *fitness*. A função de *fitness* escolhida para o projeto foi o erro quadrático médio (*mean squared error* - MSE). A métrica de MSE foi escolhida, pois é uma métrica utilizada em vários trabalhos na literatura, como em: (L. et al., 2020), (de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019), (KHASHEI; BIJARI, 2011). O valor de MSE utilizado no algoritmo foi o valor de MSE de validação. Esta é uma métrica de acurácia, onde é mensurada a distância entre o valor desejado e o valor obtido (de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019). Esta métrica é apresentada na Equação 3.12.

$$MSE = \frac{1}{n} \sum_{t=0}^{n-1} (Z_t - \hat{Z}_t)^2, \quad (3.12)$$

onde  $n$  corresponde ao número de amostras,  $Z_t$  corresponde a série temporal e  $\hat{z}_t$  corresponde a previsão realizada pela combinação do modelo linear e do modelo não linear.

### 3.2.2 Reprodução

A prole é gerada através dos operadores de *crossover* e mutação de forma disjunta, onde, 90% dos filhos são gerados através do *crossover* (sem a ocorrência de mutação) e demais filhos (10% restante), através do *crossover* com a possibilidade de mutação, assim como no trabalho (SZPIRO, 1997).

O algoritmo genético é aplicado a números reais, fazendo isto o espaço de busca se torna enorme e contínuo. Como apresentada na argumentação da diagonal de Cantor (ROSEN, 2002): entre dois números reais existem incontáveis números. Desta forma, o espaço de busca que o algoritmo genético com codificação real precisa vasculhar, é enorme. Além disso, o algoritmo utiliza 10 valores de previsão (10 valores de previsão linear e 10 valores de previsão não linear), gerando assim um cromossomo com 40 genes e o algoritmo precisa encontrar o melhor valor para cada um deles. Portanto, o uso de múltiplos operadores de *crossover* e mutação permite que a AG procure os valores em diferentes regiões do espaço de busca e aumente as chances de encontrar o valor ótimo global (ZAINUDDIN; SAMAD; TUNGGAL, 2020; YI et al., 2015; HONG et al., 2002; HONG; WANG; CHEN, 2000).

O operador de *crossover* pode ocorrer de 8 formas: *one-point* (KORA; YADLAPALLI, 2017), *two-point* (KAYA; UYAR et al., 2011), *blend* (CERVANTES-CASTILLO; MEZURA-MONTES; COELLO, 2014a), uniforme (KORA; YADLAPALLI, 2017), Laplace (DEEP; THAKUR, 2007), heurístico

(ACKORA-PRAH; GYAMERAH; ANDAM, 2014), Crossover Binário Simulado (SBX) (*simulated binary crossover*) (VARGAS, 2018) e logístico (NAQVI; SHAD; KHAN, 2021). A escolha da operação de *crossover* que será aplicada é selecionada randomicamente, visando produzir uma população diversa. A seleção dos indivíduos é realizada mediante seleção de torneio em todos os tipos de *crossover* implementados, com o tamanho de 3 indivíduos. A seleção de torneio foi escolhida, pois ela previne que os melhores indivíduos dominem a operação, caso o tamanho do conjunto de seleção não seja muito grande (ENGELBRECHT, 2007). Todos os operadores de *crossover* respeitam a probabilidade de *crossover* (taxa de *crossover*).

- No *crossover one-point*, um ponto no cromossomo de ambos os pais é selecionado aleatoriamente como ponto de cruzamento. Os genes que estão à direita do ponto de cruzamento são trocados entre os pais, o que resulta em dois descendentes com material genético de ambos os pais (ENGELBRECHT, 2007) respeitando a probabilidade de *crossover*.
- O *crossover blend* (COLHERINHAS, 2016) funciona misturando os genes dos pais e extrapolando a região de cruzamento. Este cruzamento foi desenvolvido especialmente para genes com representação em ponto flutuante (ENGELBRECHT, 2007). Devido a isto, essa operação de cruzamento, é aplicada somente aos genes responsáveis pela ponderação da combinação, ficando os genes binários sendo gerados por um *crossover two-point*.

O *crossover blend* ocorre da seguinte forma: dado dois cromossomos  $x$  e  $y$ , um cromossomo  $z$  é produzido da seguinte forma:

$$z = x + \beta(y - x),$$

sendo  $\beta \sim U(-\alpha; 1 + \alpha)$ , onde  $U$  representa uma distribuição uniforme. Na literatura sugere-se  $\alpha = 0.5$ , para ser realizado exploração e exploração com a mesma probabilidade (CERVANTES-CASTILLO; MEZURA-MONTES; COELLO, 2014b). Este operador permite extrapolar a região de cruzamento dos pais. Por este motivo, ele atribui maior variabilidade genética durante a evolução das gerações.

- No *crossover two-point*, dois pontos de troca são escolhidos aleatoriamente dos cromossomos pais e o material genético entre estes pontos são trocados entre os pais (ENGELBRECHT, 2007).

- Já no *crossover* uniforme cada gene é escolhido de qualquer um dos pais com igual probabilidade e trocado entre os pais (ENGELBRECHT, 2007).
- O *crossover* Laplace é baseado na distribuição de Laplace e é um operador de cruzamento centrado no pai (NAQVI; SHAD, 2022). Dois filhos,  $x = (x_1, x_2, \dots, x_n)$  e  $y = (y_1, y_2, \dots, y_n)$  são gerados a partir dos pais  $x' = (x'_1, x'_2, \dots, x'_n)$  e  $y' = (y'_1, y'_2, \dots, y'_n)$  da seguinte forma (DEEP; THAKUR, 2007): primeiro são gerados números aleatórios uniformemente distribuídos  $u_i, r_i \in [0, 1]$ . Então, um número aleatório  $\beta_i$ , o qual satisfaz a distribuição de Laplace, é gerado pela Equação 3.13.

$$\beta_i = \begin{cases} a - b \log(u_i), & r_i \leq \frac{1}{2}; \\ a + b \log(u_i), & r_i > \frac{1}{2}, \end{cases} \quad (3.13)$$

em que  $a$  corresponde ao parâmetro de localização e  $b > 0$  é o parâmetro de escala (DEEP; THAKUR, 2007; NAQVI; SHAD, 2022). Com o valor de  $\beta_i$  calculado, então a prole é obtida através da Equação 3.14.

$$\begin{aligned} x &= x' + \beta_i |x' - y'|, \\ y &= y' + \beta_i |x' - y'|. \end{aligned} \quad (3.14)$$

Assim como apresentado no trabalho (DEEP; THAKUR, 2007), os parâmetros de localização deve ter o valor zero e o parâmetro  $b$  é indicado ter o valor 0.5.

- O *crossover* heurístico usa o valor de *fitness* do cromossomo dos pais como guia para a direção da busca (ACKORA-PRAH; GYAMERAH; ANDAM, 2014). Esta operação faz a mudança do pior parente (aquele indivíduo com o pior valor de *fitness*) para levemente o melhor parente (o indivíduo com valor de *fitness* melhor), como apresenta a Equação 3.15.

$$\begin{aligned} x &= \text{melhorParente} + \beta |\text{melhorParente} - \text{piorParente}|, \\ y &= \text{melhorParente}. \end{aligned} \quad (3.15)$$

Em que  $\beta$  é um número aleatório no intervalo  $[0,1]$ .

- O *crossover* SBX gera a prole da seguinte forma (VARGAS, 2018; NAQVI; SHAD, 2022): primeiro, um número  $\mu$  é gerado no intervalo de  $[0,1]$ . Então, o valor  $\beta_i$  é atribuído

como apresenta a Equação 3.16.

$$\beta_i = \begin{cases} (2\mu)^{\frac{1}{\eta_c+1}}, & \text{se } \mu \leq 0.5; \\ \left(\frac{1}{2(1-\mu)}\right)^{\frac{1}{\eta_c+1}}, & \text{caso contrário;} \end{cases}$$

em que,  $\eta_c$  é um número real não negativo definido pelo usuário, chamado de índice de distribuição. A prole é gerada como indica a Equação 3.16.

$$\begin{aligned} x &= 0.5[(1 + \beta_i)x' + (1 - \beta_i)y'], \\ x &= 0.5[(1 - \beta_i)x' + (1 + \beta_i)y']. \end{aligned} \quad (3.16)$$

Como apresentado em (VARGAS, 2018), o valor de  $\eta_c$  que melhora o *crossover* para problema mono objetivo é o valor 2. Desta forma, este projeto utiliza este valor para  $\eta_c$ .

- Finalmente, o *crossover* logístico é baseado na distribuição logística (NAQVI; SHAD; KHAN, 2021; NAQVI; SHAD, 2022). A partir dos parentes  $x' = (x'_1, x'_2, \dots, x'_n)$  e  $y' = (y'_1, y'_2, \dots, y'_n)$ , dois filhos são gerados como  $x = (x_1, x_2, \dots, x_n)$  e  $y = (y_1, y_2, \dots, y_n)$ , usando os seguintes passos:

1. Gere um número aleatório  $u$  entre 0 e 1;
2. O parâmetro  $\beta_i$  é criado invertendo a distribuição cumulativa da função de distribuição logística, como apresenta a Equação 3.17.

$$\beta_i = \mu - s \log \left( \frac{1-u}{u} \right) \quad (3.17)$$

3. A prole é gerada pela Equação 3.18.

$$\begin{aligned} x &= 0.5[(x' + y') + \beta_i|x' - y'|], \\ y &= 0.5[(x' + y') - \beta_i|x' - y'|]. \end{aligned} \quad (3.18)$$

Assim como apresentado no trabalho (NAQVI; SHAD; KHAN, 2021), o melhor valor para os parâmetros  $\mu$  e  $s$  são 0 e 4.5, respectivamente. Portanto, este projeto utiliza estes valores.

Posteriormente ao *crossover*, 10% filhos estão sujeitos da mutação (SZPIRO, 1997). Foram desenvolvidos oito tipos de mutação: exclusiva, inclusiva, com incremento, Lévy (ENGELBRECHT, 2007), Gaussiana (ENGELBRECHT, 2007), *power* (DEEP; THAKUR, 2007; NAQVI; SHAD, 2022), Mutação Makinen, Periaux e Toivanen (MPTM) (Makinen, Periaux, and Toivanen

*mutation* - (MPTM)) (DEEP; THAKUR, 2007; NAQVI; SHAD, 2022) e mutação não uniforme (*non-uniform mutation* - (NUM)) (DEEP; THAKUR, 2007; NAQVI; SHAD, 2022). A escolha da operação de mutação que será aplicada é selecionada randomicamente, para produzir uma população diversa, assim como no *crossover*. Todos os operadores de mutação respeitam a probabilidade de mutação (taxa de mutação).

- Na mutação exclusiva, os valores binários do cromossomo são atribuídos o valor zero conforme a taxa de mutação. Os valores dos genes de ponderação não são alterados nesta operação. Observe que essa operação corresponde a remover as previsões anteriores da série da modelagem. Aplicando a mutação exclusiva no exemplo com 4 pontos, pode-se ter como resultado a Figura 6.

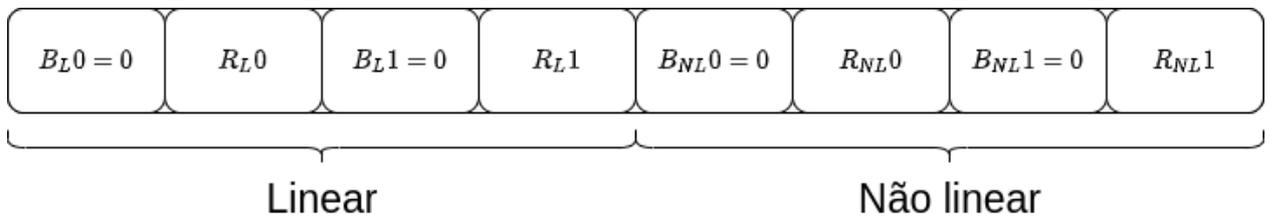


Figura 6 – Mutação exclusiva - exemplo.

**Fonte:** Elaborada pela autora (2024).

- Na mutação inclusiva, os valores binários do cromossomo são atribuídos o valor 1 conforme a taxa de mutação. Os genes de ponderação não são alterados nessa operação. Observe que essa operação corresponde a incluir as previsões anteriores da série na modelagem. Aplicando a mutação inclusiva no exemplo com 4 pontos, pode-se ter como resultado a Figura 7.

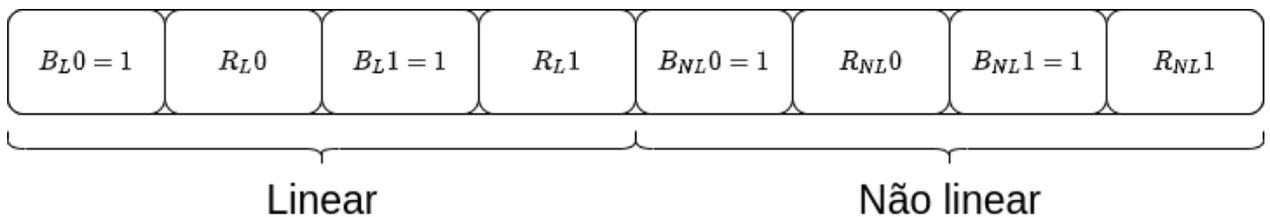


Figura 7 – Mutação inclusiva - exemplo.

**Fonte:** Elaborada pela autora (2024).

- Na mutação com contagem, os valores binários e reais são trocados conforme a quantidade passada como parâmetro na função ( $n_{max}$ ) sequencialmente. Na ponderação binária, é aplicado o *bit flip mutation*. Já na ponderação real, o peso é trocado por um valor randômico. Esta operação não utiliza a taxa de mutação.

- Na mutação com incremento, o número de mutações é escolhido aleatoriamente para cada indivíduo da prole usando uma distribuição uniforme de um determinado intervalo, o qual é determinado por  $[0, n_{max}]$ . O valor inicial para o número máximo de mutações ( $n_{max}$ ) é previamente definido na primeira iteração (IZIDIO et al., 2021). Para as demais iterações,  $n_{max}$  é acrescido de uma unidade até atingir o tamanho do cromossomo (totalizando 40 genes). Na mutação com incremento, a mutação é realizada trocando-se os valores dos booleanos (*bit flip mutation*) para os pesos binários e trocando-se os valores reais por valores randômicos na mutação uniforme de ponto flutuante (*float mutation uniform*).
- A mutação Lévy (ENGELBRECHT, 2007) os valores reais do cromossomo são substituídos de forma aleatória por um valor da distribuição Lévy com alfa igual a 1.5 e beta igual a -0.02, escolhidos por experimentação. A distribuição de probabilidade de Lévy é dada pela Equação 3.19.

$$L_{\alpha,\beta}(y) = \frac{1}{\pi} \int_0^{\infty} \exp(-\beta q^\alpha) \cos(qy) dq, \quad y \in \mathbb{R}. \quad (3.19)$$

A distribuição é simétrica com relação a  $y = 0$  e possui dois parâmetros:  $\alpha$  e  $\beta$  (LEE; YAO, 2004).  $\beta$  corresponde ao fator de escala satisfazendo  $\beta > 0$ . Já  $\alpha$ , controla a forma da distribuição e precisa estar no intervalo  $0 < \alpha < 2$ .

Os valores binários realizam uma mutação de *flip-bit*.

- A mutação gaussiana (ENGELBRECHT, 2007), os valores reais são substituídos por um valor da distribuição gaussiana com média 0 e desvio padrão 1 e os valores binários sofrem uma mutação *flip-bit*.
- A mutação *power* utiliza a distribuição *power* para modificar os valores dos pesos (DEEP; THAKUR, 2007; NAQVI; SHAD, 2022) e aplica a mutação *flip bit* nos valores de binários do cromossomo. Este operador funciona da seguinte forma:
  1. Primeiramente, um número aleatório  $s$ , o qual segue a distribuição  $s_l$  é uma número aleatório uniforme no intervalo de  $[0,1]$ , é criado; a variável  $p$  que corresponde ao índice da mutação.

2. Após determinar o valor de  $s$ , uma solução é criada como apresenta a Equação 3.20.

$$x = \begin{cases} x' - s(x' - (x')^l), & t < r; \\ x' + s((x')^u - x'), & t \geq r. \end{cases}$$

em que,  $t = \frac{(x' - (x')^l)}{(x')^u - x'}$ ,  $(x')^l$  e  $(x')^u$  corresponde ao limite superior e inferior da variável de decisão e  $r$  é um número uniformemente distribuído entre 0 e 1. Neste trabalho, o limite superior e inferior da mutação é de  $[-2, 2]$ .

- A mutação MPTM é usada para solucionar um largo conjunto de problemas de otimização (DEEP; THAKUR, 2007; NAQVI; SHAD, 2022). Neste projeto, esta mutação é aplicada nos valores de peso reais, já nos valores binários é aplicada o *flit bit*. A partir de um ponto  $x = (x_1, x_2, \dots, x_n)$  para mutar para o ponto  $x' = (x'_1, x'_2, \dots, x'_n)$ , são utilizados os seguintes passos:

1. Gere um número aleatório uniforme  $r$  no intervalo de  $[0,1]$ ;
2. A solução da mutação é dada pela Equação 3.20.

$$x'_i = (1 - t')x_i^l + t'x_i^u \quad (3.20)$$

em que,

$$t' = \begin{cases} t - t \left( \frac{t-r}{t} \right)^b, & \text{se } r < t, \\ t, & \text{se } r = t, \\ t + (1 - t) \left( \frac{r-t}{1-t} \right)^b, & \text{se } r > t, \end{cases}$$

e

$$t = \frac{x - x_i^l}{x_i^u - x}, \quad (3.21)$$

em que,  $x_i^l$  e  $x_i^u$  correspondem ao limite superior e inferior da  $i$ -ésima variável de decisão.

- A mutação não uniforme é uma mutação largamente aplicada em algoritmos genéticos com codificação real (DEEP; THAKUR, 2007; NAQVI; SHAD, 2022). Este operador é aplicado ao peso real, mas o valor binário do cromossomo utiliza a mutação *flit bit*. A partir de um ponto  $x = (x_1, x_2, \dots, x_n)$  o ponto mutado  $x' = (x'_1, x'_2, \dots, x'_n)$  é criado da seguinte forma:

$$x'_i = \begin{cases} x_i^t + \Delta(t, x_i^u - x_i^t), & \text{se } r \leq 0.5; \\ x_i^t - \Delta(t, x_i^t - x_i^l), & \text{caso contrário;} \end{cases}$$

em que  $t$  é o número atual da geração e  $r$  é um número aleatório uniformemente distribuído no intervalo  $[0,1]$ .  $x_i^l$  e  $x_i^u$  correspondem ao limite superior e inferior da  $i$ -ésima componente da variável de decisão. A função  $\Delta(t, y)$  é apresentada a seguir na Equação 3.22 e está limitada no intervalo de  $[0,y]$ .

$$\Delta(t, y) = y \left(1 - u^{(1-\frac{t}{T})}\right)^b, \quad (3.22)$$

em que  $u$  corresponde a um número aleatório uniformemente distribuído no intervalo  $[0,1]$ ,  $T$  é o número máximo de gerações e  $b$  corresponde ao parâmetro que determina a força do operador de mutação. No artigo (WANG et al., 2015) são apresentados os melhores valores para os parâmetros, onde para  $b$  o valor é 2. Desta forma, neste trabalho o valor 2 é utilizado para o parâmetro  $b$ .

Foi adicionado ao projeto o operador de convergência, chamado de operador *Particle Swarm Optimization* (PSO). Neste operador, os indivíduos são direcionados a realizar movimentos na direção do melhor indivíduo da população, muito parecido com a componente social do algoritmo *particle swarm optimization* - PSO (ENGELBRECHT, 2007). Assim, é realizada uma diferença entre os valores dos cromossomos do melhor indivíduo da população e indivíduo a sofrer mutação, o resultado dessa diferença é multiplicado por uma distribuição uniforme no intervalo  $[0,1]$  e adicionado ao valor do cromossomo, como apresentado na Equação 3.23.

$$\text{indivíduo} = \text{indivíduo} + \text{distribuição uniforme} \cdot (\text{melhor indivíduo} - \text{indivíduo}). \quad (3.23)$$

Este operador é aplicado somente a ponderação real, ficando a ponderação binária sendo atualizada através da mutação de *flit bit mutation*.

### 3.2.3 Diversidade

A taxa de mutação e *crossover* utilizada no projeto é adaptativa, ou seja, segundo a diversidade populacional, as taxas de *crossover* e mutação são incrementadas ou decrementadas,

esta operação foi baseada no trabalho (FATYANOSA; ARITSUGI, 2021). Para a adaptação das taxas de mutação e *crossover*, foi utilizada a diversidade média da população, apresentada na Equação 3.24.

$$div_{pop} = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^n d_e(x_i, x_j)}{\binom{n}{2}}, \quad (3.24)$$

onde,  $x$  é um indivíduo,  $d_e(x_i, x_j)$  corresponde a distância euclidiana entre os pares de indivíduos e  $n$  é o número de indivíduos. Caso a diversidade populacional tenha atingido o limiar  $\theta$ , isso implica que a diversidade reduziu e o algoritmo precisa explorar mais para que a população seja mais diversa. Neste trabalho, o limiar de diversidade utilizado foi de 5, este valor foi encontrado por experimentação. Dessa forma, a modificação das taxas de mutação e *crossover* obedeceram aos seguintes tópicos:

1. A **taxa de *crossover*** ( $C_r$ ) - é incrementada caso a diversidade populacional seja maior que o limiar. Caso contrário, é decrementada.
2. A **taxa de mutação** ( $M_r$ ) - é decrementada caso a diversidade populacional seja maior que o limiar. Caso contrário, é incrementada.

O incremento e decremento das taxas de mutação e *crossover* é baseado na taxa de *update*, parâmetro do algoritmo. Este parâmetro foi baseado no trabalho (FATYANOSA; ARITSUGI, 2021). Foi adicionado ao projeto a taxa mínima e máxima para *crossover* e mutação, a fim de evitar erros na execução, assim como no trabalho (FATYANOSA; ARITSUGI, 2021).

### 3.2.4 Busca Local

Com o intuito de balancear a exploração e exploração, foi adicionada uma etapa de busca local que realiza a seleção dos filhos. Essa etapa do algoritmo foi baseado no trabalho (SABAR et al., 2017). É verificado se o *fitness* da prole é melhor do que o pior indivíduo da população. Caso o *fitness* da prole seja melhor, a prole é adicionada à população. Senão, é realizada a busca local.

Para todos os indivíduos da população que não obtiveram uma melhoria no *fitness* (não obtiveram *fitness* melhor que o pior indivíduo da prole) é aplicada a busca local. Se após a busca local, a prole for melhor que o pior indivíduo da população, a prole é adicionada a

população. Senão, essa solução é descartada. A Figura 8 representa do fluxo de funcionamento do algoritmo de busca local.

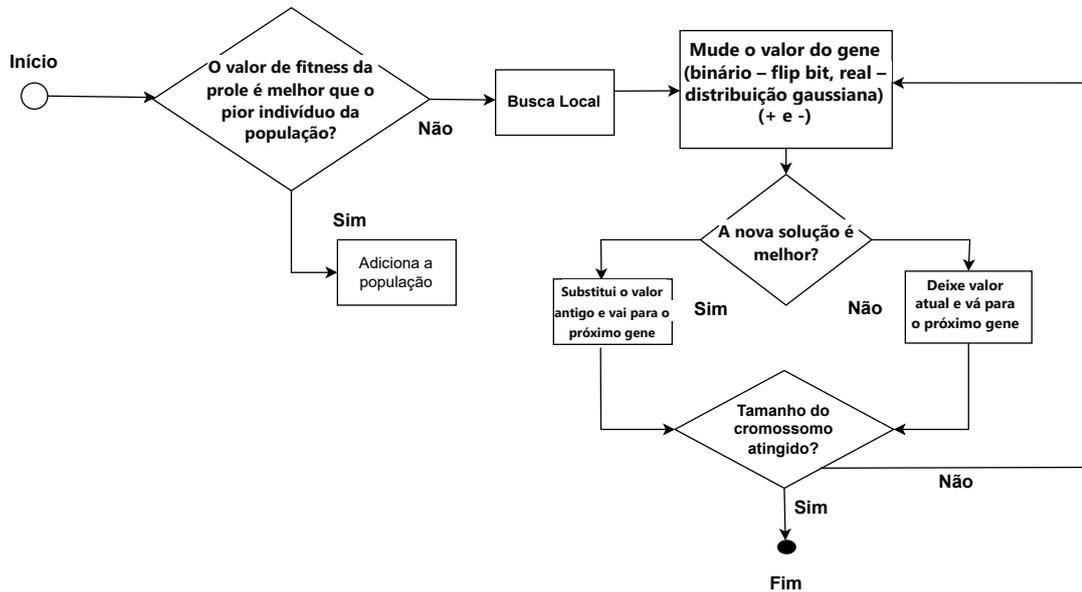


Figura 8 – Busca local - Diagrama.

Fonte: Elaborada pela autora (2024).

A busca local consiste em um operador de vizinhança, onde o operador varre os genes da solução, um por um, aplicando as ações:

1. Muda o valor do gene binário por uma operação de *flip-bit*. Se for a ponderação real, o valor é trocado por um valor dentro da distribuição Gaussiana e aplicado em todas as direções: distribuição Gaussiana positiva (+) e distribuição Gaussiana negativa (-). Para esta operação o limite da distribuição Gaussiana foi de  $[-2,2]$ . Este limite foi descoberto por experimentação.
2. Se alguns dos valores descobertos levarem a uma melhor solução em termos de *fitness*, então a solução substitui a solução anterior. Caso contrário, o novo valor é descartado e a busca local vai para o próximo gene. A busca local termina quando muda todos os valores do gene e não consegue uma melhoria.

### 3.2.5 Injeção Randômica

Buscando melhorar a diversidade populacional e evitar os mínimos locais, foi adicionado ao algoritmo a injeção randômica, como no trabalho (FATYANOSA; ARITSUGI, 2021). Quando a

diversidade populacional média fica abaixo do limiar  $\theta$ , a injeção randômica é inicializada. Os indivíduos são gerados randomicamente pela injeção, assim como acontece na inicialização. A quantidade de indivíduos gerados corresponde a 10% da população e estes substituem os 10% piores indivíduos da população.

### 3.2.6 Critérios de Parada

Os critérios de parada utilizados no projeto foram: número de gerações, número de gerações sem melhoria no *fitness* do melhor indivíduo (estagnação) e parada antecipada (*early stop*), não é utilizado limite na melhoria do *fitness*. A parada antecipada utilizada neste trabalho é baseado no trabalho (PIOTROWSKI; NAPIORKOWSKI, 2013), em que é utilizada a classe de perda de generalização (*Generalization Loss Class (GL)*) aplicada ao conjunto de validação e o algoritmo para quando  $GL > 20\%$ , como indica o trabalho (PIOTROWSKI; NAPIORKOWSKI, 2013). A Equação 3.25 descreve a métrica de Prechelt.

$$GL(t) = 100. \left( \frac{E_{va}}{E_{opt}} - 1 \right), \quad (3.25)$$

em que  $E_{va}$  corresponde ao erro do conjunto de validação e  $E_{opt}$  é o menor valor de erro do conjunto de validação obtido em épocas até  $t$ . A Equação 3.26 apresenta  $E_{opt}$ .

$$E_{opt}(t) := \min_{t' \leq t} E_{va}(t'). \quad (3.26)$$

### 3.2.7 Métricas de Balanceamento

Um bom desempenho de um algoritmo é alcançado com o balanceamento das fases de exploração e exploração. Desta forma, foi preciso avaliar estas métricas neste trabalho. Para avaliar a exploração e exploração do algoritmo, foram adicionadas métricas de diversidade, como apresentam os trabalhos (SI; MIRANDA; BHATTACHARYA, 2021), (SALLEH et al., 2018). As métricas implementadas foram: diversidade, exploração e exploração.

As métricas apresentadas nos trabalhos (SI; MIRANDA; BHATTACHARYA, 2021), (SALLEH et al., 2018) são aplicadas ao projeto.

A diversidade populacional é mensurada pelo incremento e decremento da distância entre os indivíduos da população no tempo  $t$ , como apresenta a Equação 3.27.

$$div(t) = \frac{1}{N} \sum_{i=1}^N |median(pop(t)) - ind_i(t)|, \quad (3.27)$$

em que  $median(pop(t))$  corresponde a mediana de toda a população,  $ind_i$  corresponde ao  $i$ -ésimo indivíduo da população. O valor  $N$  corresponde ao tamanho da população.

O cálculo da métrica de exploração e de exploração são apresentados nas Equações 3.28 e 3.29, respectivamente.

$$X_{pl}\% = \left( \frac{div}{div_{max}} \right) \times 100, \quad (3.28)$$

$$X_{pt}\% = \left( \frac{div - div_{max}}{div_{max}} \right) \times 100, \quad (3.29)$$

em que  $X_{pl}\%$  e  $X_{pt}\%$  correspondem as percentagens de exploração e exploração, respectivamente. Enquanto,  $div_{max}$  é o maior valor de diversidade encontrado a cada iteração na população. Esta métrica serve para avaliação da exploração e exploração do algoritmo.

### 3.3 RESUMO DO CAPÍTULO

Neste capítulo foi apresentada a metodologia proposta. A pesquisa está organizada em duas etapas: modelagem dos padrões lineares e não lineares e modelagem da combinação de previsões. A modelagem da combinação de previsões engloba o estudo dos parâmetros e técnicas de melhoria aplicados ao algoritmo genético e a análise de quantidade de previsões anteriores utilizadas no modelo de composição.

Foi apresentado também o modelo híbrido com a composição de três etapas sequenciais. As equações de cada etapa do processo de geração dos modelos que compõem a abordagem proposta são expostas em detalhes, bem como o diagrama explicativo.

Para o desenvolvimento da etapa de composição de previsões, foi desenvolvido um algoritmo genético adaptativo, onde neste capítulo foram apresentadas as técnicas desenvolvidas até o momento, como: semeadura, injeção randômica, busca local, taxa de mutação e *crossover* adaptativo baseado em diversidade populacional e novos operadores de mutação.

## 4 ANÁLISE EXPERIMENTAL

Este capítulo apresenta os resultados obtidos com a pesquisa. O capítulo está organizado em quatro seções: a Seção 4.1 apresenta o protocolo experimental e o conjunto de dados utilizado, a Seção 4.2 relata os resultados obtidos na pesquisa comparando os modelos do sistema híbrido e também comparando com a literatura, a Seção 4.3 traz a explicação da equação de combinação que é um dos resultados da abordagem e a Seção 4.4 apresenta as considerações deste capítulo.

### 4.1 PROTOCOLO EXPERIMENTAL

Para a avaliação da abordagem proposta foram utilizadas 8 séries temporais amplamente utilizadas na literatura de sistemas híbridos (ZHANG, 2003; KHASHEI; BIJARI, 2011; PANIGRAHI; BEHERA, 2017; de O. Santos Júnior; de Oliveira; de Mattos Neto, 2019) que possuem diferentes características (como tendência e sazonalidade). A Tabela 1 apresenta a descrição e comportamentos das séries utilizadas no projeto.

Tabela 1 – Descrição e características das séries utilizadas.

Série	Descrição	Características
Carsales	Número mensal de vendas de carros em Quebec	Não estacionária e sazonal
Electricity	Eletricidade na Austrália: eletricidade no usuário final	Não estacionária, sazonal e com tendência
Gas	Demanda mensal de gasolina em Ontario	Não estacionária, sazonal e com tendência
Lake	Nível mensal do lago Erie	Não estacionária e sazonal
Nordic	Total de energia consumida nos países Nórdicos: Noruega, Suécia, Finlândia e Dinamarca, no período de 01/01/2020 até 05/10/2020	Não estacionária e sazonal
Pigs	Número de porcos abatidos mensalmente em Victoria	Não estacionário
Pollution	Número de remessas de equipamentos de poluição mensalmente	Não estacionário e com tendência
Redwine	Número de vendas mensais de vinhos na Austrália	Não estacionário e sazonal

**Fonte:** Elaborada pela autora (2024).

Estas séries foram divididas em três conjuntos de dados: treinamento, validação e teste. O detalhamento desta divisão dos dados é apresentada na Tabela 2 e foi realizado baseado no

trabalho (OLIVEIRA; SILVA; NETO, 2021).

Tabela 2 – Divisão do conjunto de dados.

Série	Tamanho do Conjunto			
	Amostra	Treinamento	Validação	Teste
Carsales	108	31	36	20
Electricity	486	231	169	65
Gas	192	85	72	14
Lake	600	279	200	100
Nordic	3121	1479	1009	600
Pigs	188	70	62	35
Pollution	130	49	48	12
Redwine	187	78	66	22

**Fonte:** Elaborada pela autora (2024).

O conjunto de treinamento foi aplicado a parametrização do modelo, o conjunto de validação foi usado para escolher o melhor modelo gerado e o conjunto de teste foi utilizado para avaliar a acurácia do modelo. O conjunto de validação foi usado com o intuito de evitar o sobre ajuste e associado a ele foi aplicada a métrica de classe de perda de generalização de Prechelt (PRECHELT, 1998) para encontrar o ponto de sobre ajuste. Esta divisão de dados foi aplicada a todos os modelos utilizados na comparação neste projeto. Vale ressaltar que o somatório dos conjuntos de treinamento, validação e teste não resulta no tamanho da amostra, porque durante o processo de janelamento dos modelos de previsão linear e não linear, alguns dados são perdidos.

Para comprovar a eficácia da abordagem foram escolhidos os modelos ARIMA e SARIMA como modelo de previsão e para a modelagem dos resíduos foram escolhidos os modelos MLP e SVR.

Os modelos ARIMA e SARIMA foram escolhidos para este trabalho, porque eles são modelos simples, flexíveis e podem ser utilizados para modelar séries temporais com diferentes características (HAJIRAHIMI; KHASHEI, 2019a; ZHANG, 2003). Além disso, a metodologia de Box & Jenkins utilizada para ajustar os modelos ARIMA e SARIMA garante a modelagem adequada dos padrões lineares da série em análise (OLIVEIRA; SILVA; NETO, 2021). Além disto, o modelo SARIMA foi adicionado ao projeto proposto visando a adequação do modelo a série, pois só foi realizado o tratamento de normalização dos dados das séries, mesmo algumas séries apresentando sazonalidade.

Já na modelagem não linear, o modelo MLP foi escolhido, porque ele é um aproximador universal que pode aproximar uma vasta classe de modelos de funções com alto grau de acurácia (ZHANG, 2003). O modelo SVR foi escolhido devido à estabilidade e robustez (DRUCKER et al., 1997), com a capacidade de realizar previsões acuradas (OLIVEIRA; LUDERMIR, 2015).

A abordagem proposta é capaz de realizar a combinação de um modelo previsão com um modelo de previsão de resíduos. Este usa 10 valores de previsão (previsão atual e 9 previsões anteriores) para cada padrão da série (linear e não linear), como apresenta a Figura 9.

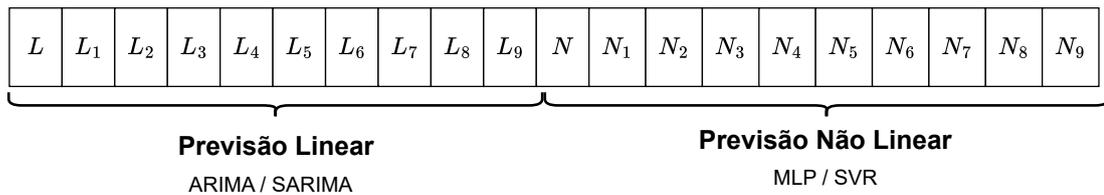


Figura 9 – Combinação de previsões

**Fonte:** Elaborada pela autora (2024).

O Teste de Dickey-Fuller Aumentado (ADF) foi utilizado para avaliar o comportamento de estacionaridade e tendência das séries (CAPORALE; PLASTUN, 2019). O teste de Kruskal-Wallis foi aplicado para analisar a sazonalidade do conjunto de dados (CAPORALE; PLASTUN, 2019).

Todo o conjunto de dados foi normalizado para o intervalo  $[0,1]$ , segundo a Equação 4.1.

$$Z_{norm} = \frac{Z_t - Z_{min}}{Z_{max} - Z_{min}}, \quad (4.1)$$

em que,  $Z_t$  corresponde ao valor da série temporal,  $Z_{min}$  e  $Z_{max}$  são os valores mínimo e máximo da série temporal, respectivamente.

Os parâmetros usados nos experimentos aplicados ao algoritmo genético são apresentados na Tabela 3.

A abordagem proposta foi comparada com modelos únicos e modelos híbridos. Os modelos híbridos utilizados na comparação seguem a decomposição em padrões lineares e não lineares. (PANIGRAHI; BEHERA, 2017) e (BABU; REDDY, 2014) consistem em sistemas Modelo de Suavização Exponencial (ETS) + MLP e ARIMA + MLP, respectivamente. (PANIGRAHI; BEHERA, 2017) aplica um modelo ETS na série temporal e um MLP nos resíduos. A previsão final é realizada pela adição da componente linear ETS com a componente não linear MLP. Bem como o modelo de (PANIGRAHI; BEHERA, 2017), são os modelos baseados no SARIMA, como SARIMA + MLP (RUIZ-AGUILAR; TURIAS; JIMÉNEZ-COME, 2014) e SARIMA + SVR (RUIZ-AGUILAR et al., 2014).

Tabela 3 – Parâmetros utilizados no trabalho proposto.

Parâmetro	Valor
População	60 indivíduos
Gerações	1000
Taxa de Cruzamento - inicial	0,8
Taxa de Mutação - inicial	0,2
Taxa Máx ( <i>crossover</i> e mutação)	1,0
Taxa Min ( <i>crossover</i> e mutação)	0,01
$\theta$ Limiar - Diversidade Populacional Média	5
Taxa de Atualização	0,01
$n_{max}$ (Incremento Mutação)	5
Limite da Busca Local	40 iterações
Percentual Injeção	10 % população
Espera Estagnação	100 gerações
Valor de Perda de Generalização (GL)	20 %

**Fonte:** Elaborada pela autora (2024).

(BABU; REDDY, 2014) desenvolveram um modelo, em que séries temporais são decompostas em padrões de alta e baixa volatilidade. Essa separação é realizada pelo filtro de médias móveis. A componente de baixa volatilidade é modelada pelo ARIMA e a componente de alta volatilidade é modelada pelo MLP. A ordem do filtro de médias móveis é incrementada até alcançar metade do tamanho do conjunto de treinamento. Este valor é baseado na minimização  $|\mathcal{J} - kurtosis(L_t)|$ . Conforme o teste de normalidade de Jarque-Bera, o valor de kurtosis igual a 3, indica que o dado segue uma distribuição Gaussiana e pode ser considerado de baixa volatilidade.

(BüYüKŞAHİN; ERTEKİN, 2019) desenvolveu um algoritmo de decomposição de modo empírico (*Empirical Mode Decomposition Algorithm* (EMD)), em que a série é dividida em componentes mais simples. O objetivo deste trabalho é facilitar o processo de modelagem e melhorar a acurácia da previsão final. Para cada componente, um sistema híbrido ARIMA + MLP foi usado e a previsão final foi composta pela adição dos componentes da previsão.

(OLIVEIRA; LUDERMIR, 2015) e (OLIVEIRA; LUDERMIR, 2014) propuseram um sistema ARIMA + SVR otimizado pelo algoritmo PSO. No trabalho (OLIVEIRA; LUDERMIR, 2015) um filtro ETS foi utilizado para decompor os dados em componentes de baixa e alta volatilidade. O algoritmo PSO foi aplicado para buscar os parâmetros: fator de suavização  $\alpha$ , os lags e os parâmetros da SVR. Já no trabalho (OLIVEIRA; LUDERMIR, 2014), a busca pela seleção do lag foi realizada

pelo PSO. Em ambos os trabalhos, o PSO foi executado por 100 iterações, com a população com 30 partículas, e os fatores pessoais ( $c_1$ ) e social ( $c_2$ ) iguais a 2.

(HAJIRAHIMI; KHASHEI, 2020) propôs um sistema híbrido composto da análise de padrões não lineares primeiro e posteriormente da análise linear dos resíduos pelo modelo ARIMA (MLP + ARIMA). Adicionalmente, modelos únicos de *deep-learning* foram desenvolvidos para o experimento (*Long Short-Term Memory Network* (LSTM) e *Convolutional Neural Network* (CNN)).

Os modelos únicos não lineares (CNN e LSTM) e os modelos MLP e SVR tiveram os parâmetros obtidos através da busca *gridsearch*. O espaço de busca utilizado pelo algoritmo *gridsearch* é apresentado na Tabela 4. Estes valores são baseados no trabalho (OLIVEIRA; LUDERMIR, 2015). Já os modelos ARIMA, SARIMA e ETS foram implementados usando a biblioteca *Forecast* para R (HYNDMAN; KHANDAKAR, 2008). Para os modelos híbridos da literatura, que utilizam um modelos não linear na combinação, a camada de entrada do modelo não linear foi buscada no espaço de [1,12] entradas.

Tabela 4 – Parâmetros *Gridsearch* - modelos não lineares.

Modelo	Parâmetros	Valores
SVR	Kernel	Função de Base Radial, Sigmoid
	Gamma	$[10^0, 10^{-1}, 10^{-2}, 10^{-3}]$
	Cost	$[10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4]$
	Epsilon	$[10^{-1}, 10^{-2}, 10^{-3}]$
MLP	Unidades na Camada Oculta	[2, 5, 10, 15, 20]
	Algoritmo	Adam
	Função de Ativação	Logística
LSTM	Unidades na Camada Oculta	5, 10, 100, 500
	Algoritmo	Adam
	Função de Ativação	Relu
CNN	Unidades na Camada Oculta	5, 10, 100, 500
	Algoritmo	Adam
	Função de Ativação	Relu

Fonte: Elaborada pela autora (2024).

A análise dos resultados foi realizada através da métrica de erro quadrático médio (MSE) que está representado na Equação 3.12. Esta foi a métrica escolhida, porque ela é largamente utilizada na literatura de previsão de séries temporais (OLIVEIRA; LUDERMIR, 2015).

## 4.2 RESULTADOS EXPERIMENTAIS

Os resultados foram obtidos de dois diferentes tipos de combinação de modelos: ARIMA + ML (MLP ou SVR) e SARIMA + ML (MLP ou SVR). Além disso, foi realizada a comparação com outros modelos da literatura para verificar a eficácia da abordagem proposta. Desta forma, para melhorar a compreensão dos resultados, este capítulo será dividido em duas partes: a análise das versões da abordagem proposta e a comparação com a literatura.

### 4.2.1 Análise das Versões da Abordagem Proposta

A análise da abordagem proposta foi iniciada com a comparação da abordagem proposta com os modelos lineares únicos. Para a obtenção dos resultados apresentados a seguir, o algoritmo genético foi executado 10 vezes e é apresentado o melhor desempenho destas execuções. A Tabela 5 apresenta o desempenho dos modelos, pode-se observar que este trabalho obteve melhores valores de MSE que os modelos lineares para todas as séries analisadas. Para as séries: Pigs, Pollution e Redwine, a abordagem proposta conseguiu um MSE com uma ordem de grandeza menor.

Tabela 5 – Avaliação do MSE das versões do método proposto aplicado ao conjunto de teste.

Método	Carsales	Electricity	Gas	Lake	Nordic	Pigs	Pollution	Redwine
ARIMA <sup>1</sup>	1.78E-02	5.69E-02	6.67E-03	1.59E-03	4.37E-04	1.14E-02	1.53E-02	3.11E-02
SARIMA <sup>1</sup>	1.70E-02	5.69E-03	6.67E-03	1.48E-03	4.40E-04	1.14E-02	1.43E-02	3.11E-02
AG(ARIMA + MLP)	<b>1.26E-02</b>	1.93E-03	6.10E-03	1.40E-03	2.63E-04	9.90E-03	<b>8.65E-03</b>	8.29E-03
AG(ARIMA + SVR)	1.74E-02	1.96E-03	3.33E-03	<b>1.36E-03</b>	2.29E-04	<b>9.32E-03</b>	1.29E-02	9.66E-03
AG(SARIMA + MLP)	2.32E-02	<b>1.78E-03</b>	5.51E-03	1.44E-03	2.18E-04	9.88E-03	9.44E-03	9.48E-03
AG(SARIMA + SVR)	1.69E-02	1.87E-03	<b>2.70E-03</b>	1.38E-03	<b>2.18E-04</b>	9.36E-03	1.36E-02	<b>7.89E-03</b>

Fonte: Elaborada pela autora (2024).

Os melhores resultados foram obtidos pela abordagem proposta, utilizando os modelos da seguinte forma: Carsale e Pollution (AG(ARIMA + MLP)), Lake e Pigs (AG(ARIMA + SVR)), Electricity (AG(SARIMA + MLP)) e Gas, Nordic e Redwine (AG(SARIMA + SVR)).

Para as três séries (Carsales, Electricity e Pollution) o modelo que utiliza MLP como modelo não linear obteve melhor MSE. Para outras cinco séries (Gas, Nordic, Pigs e Redwine) foram obtidos melhores resultados usando o SVR como modelo não linear. Mas o uso de ARIMA e SARIMA como modelo linear obteve um empate, pois para quatro séries (Carsales, Lake, Pigs

<sup>1</sup>(BOX et al., 2016)

e Pollution) o modelo ARIMA conseguiu melhores resultados. Já para outras quatro séries (Electricity, Gas, Nordic e Redwine) o modelo SARIMA obteve o melhor resultado de MSE.

A abordagem proposta também melhorou a combinação mesmo quando os modelos únicos tinham pior MSE comparado com outros modelos únicos. Para a série Carsales, Lake e Pollution, o ARIMA tinha um valor pior de MSE que o modelo SARIMA, mas a combinação que usa o ARIMA foi a vencedora. O mesmo ocorre com a série Nordic e o modelo SARIMA, pois nesta série o valor de MSE do SARIMA era pior que o ARIMA, mas a combinação que obteve melhor desempenho foi a que utiliza o SARIMA. Estas descobertas mostram que esta proposta é capaz de encontrar pesos adequados para combinar as previsões lineares e não lineares.

A Tabela 6 compara a abordagem proposta com o modelo linear que serve de base para a combinação, por exemplo, o modelo proposto AG(ARIMA + MLP) é comparado com o modelo ARIMA, bem como o modelo AG(SARIMA + MLP) é comparado com o modelo SARIMA. Esta tabela apresenta que para todas as séries, a abordagem proposta obtém melhor resultado de MSE em percentual, exceto para a série Carsale com a combinação SARIMA + MLP. Mas para as demais combinações, a abordagem proposta atinge melhor valor de MSE. Esta observação indica a importância da escolha do modelo linear e não linear que vai compor a combinação (ARIMA ou SARIMA e MLP ou SVR).

Tabela 6 – Diferença percentual em comparação com o modelo linear inicial.

Método	Carsales	Electricity	Gas	Lake	Nordic	Pigs	Pollution	Redwine
AG(ARIMA + MLP)	29.21	96.61	8.55	11.95	39.82	13.16	43.46	73.34
AG(ARIMA + SVR)	2.25	96.56	50.07	14.47	47.60	18.25	15.69	68.94
AG(SARIMA + MLP)	-36.47	68.72	17.39	2.70	50.45	13.33	33.99	69.52
AG(SARIMA + SVR)	0.59	67.14	59.52	6.76	50.45	17.89	4.90	74.63

Fonte: Elaborada pela autora (2024).

Observando o ganho percentual médio para cada modelo de combinação, pode-se observar que este trabalho melhora a combinação. Para cada combinação, a melhoria foi de: 39,51% para AG(ARIMA + MLP), 39,23% para AG(ARIMA + SVR), 27,45% para AG(SARIMA + MLP) e 35,24% para AG(SARIMA + SVR). Estes valores apresentam uma melhoria de mais de 25% por modelo.

#### 4.2.2 Comparação com a literatura

Ampliando a análise foi realizada a comparação das melhores versões da abordagem proposta com os modelos da literatura. A comparação é apresentada na Tabela 7. Esta tabela contém os modelos únicos e híbridos da literatura que se sobrepõem ao trabalho proposto ao utilizar a função linear como função de combinação.

Tabela 7 – Valores de erro quadrático médio da melhor versão da abordagem proposta em comparação com modelos únicos e híbridos da literatura aplicados ao conjunto de teste.

Método	Carsales	Electricity	Gas	Lake	Nordic	Pigs	Pollution	Redwine
ARIMA <sup>1</sup>	1.78E-02	5.69E-02	6.67E-03	1.59E-03	4.37E-04	1.14E-02	1.53E-02	3.11E-02
SARIMA <sup>1</sup>	1.70E-02	5.69E-03	6.67E-03	1.48E-03	4.40E-04	1.14E-02	1.43E-02	3.11E-02
CNN <sup>2</sup>	5.64E-02	2.90E-02	1.79E-02	3.97E-03	3.39E-02	1.87E-02	6.63E-02	2.93E-02
LSTM <sup>2</sup>	2.74E-02	5.48E-03	1.18E-02	1.88E-03	9.51E-04	1.19E-02	2.85E-02	3.10E-02
Babu <sup>3</sup>	2.31E-02	3.63E-03	4.69E-03	1.50E-03	3.39E-04	1.35E-02	3.30E-02	1.72E-02
Oliveira <sup>2</sup>	3.84E-02	4.06E-03	5.86E-03	1.58E-03	2.62E-04	1.27E-02	2.61E-02	1.19E-02
Oliveira <sup>4</sup>	3.46E-02	1.18E-02	8.13E-03	1.47E-03	3.42E-04	1.24E-02	1.94E-02	1.44E-02
Panigrahi <sup>5</sup>	3.01E-02	<b>1.37E-03</b>	8.21E-03	<b>1.35E-03</b>	3.63E-04	1.06E-02	1.29E-02	1.84E-02
ARIMA + MLP <sup>6</sup>	4.97E-02	2.31E-03	9.61E-03	1.43E-03	2.58E-04	1.03E-02	1.28E-02	9.45E-03
ARIMA + SVR <sup>7</sup>	1.94E-02	2.15E-03	4.77E-03	1.45E-03	2.29E-04	9.93E-03	1.43E-02	1.03E-02
ARIMA + RBF <sup>8</sup>	1.89E-02	5.74E-03	6.63E-03	1.72E-03	5.88E-04	1.02E-02	1.43E-02	2.97E-02
ARIMA + LSTM <sup>9</sup>	1.75E-02	4.28E-03	7.49E-03	1.40E-03	2.32E-04	1.16E-02	1.25E-02	1.16E-02
ARIMA + CNN <sup>10</sup>	2.61E-02	3.52E-03	5.28E-03	1.45E-03	2.57E-04	1.13E-02	1.21E-02	1.81E-02
SARIMA + MLP <sup>11</sup>	7.76E-02	2.21E-03	8.35E-03	1.44E-03	2.68E-04	1.03E-02	1.25E-02	1.06E-02
SARIMA + SVR <sup>12</sup>	1.94E-02	2.16E-03	4.77E-03	1.45E-03	2.29E-04	9.93E-03	1.43E-02	1.03E-02
Hajirahimi and Khashei (ARIMA + MLP) <sup>13</sup>	1.50E-02	2.29E-03	8.20E-03	1.45E-03	2.63E-04	1.02E-02	<b>8.65E-03</b>	1.10E-02
Hajirahimi and Khashei (ARIMA + SVR) <sup>13</sup>	1.71E-02	2.10E-03	3.60E-03	1.43E-03	2.31E-04	1.00E-02	1.42E-02	1.03E-02
Hajirahimi and Khashei (SARIMA + MLP) <sup>13</sup>	1.78E-02	2.16E-03	7.04E-03	1.45E-03	2.67E-04	1.02E-02	9.11E-03	1.17E-02
Hajirahimi and Khashei (SARIMA + SVR) <sup>13</sup>	1.72E-02	2.10E-03	3.60E-03	1.43E-03	2.31E-04	9.91E-03	1.43E-02	1.03E-02
Abordagem Proposta	<b>1.26E-02</b>	1.78E-03	<b>2.70E-03</b>	1.36E-03	<b>2.18E-04</b>	<b>9.32E-03</b>	<b>8.65E-03</b>	<b>7.89E-03</b>

Fonte: Elaborada pela autora (2024).

Analisando a tabela a possível perceber que o trabalho proposto alcançou o melhor valor de MSE para seis séries dentre as oito séries analisadas (Carsales, Gas, Nordic, Pigs, Pollution e Redwine). Para algumas séries, como Pigs, Pollution e Redwine, o sistema proposto obteve valor de MSE uma ordem de grandeza menor que a maioria dos modelos únicos e híbridos da literatura.

<sup>1</sup>(BOX et al., 2016)

<sup>2</sup>(OLIVEIRA; LUDERMIR, 2015)

<sup>3</sup>(BABU; REDDY, 2014)

<sup>4</sup>(OLIVEIRA; LUDERMIR, 2014)

<sup>5</sup>(PANIGRAHI; BEHERA, 2017)

<sup>6</sup>(ZHANG, 2003)

<sup>7</sup>(PAI; LIN, 2005)

<sup>8</sup>(YAN; MA, 2016)

<sup>9</sup>(LIU; TAN; WANG, 2019)

<sup>10</sup>(OLIVEIRA; SILVA; NETO, 2021)

<sup>11</sup>(RUIZ-AGUILAR; TURIAS; JIMÉNEZ-COME, 2014)

<sup>12</sup>(RUIZ-AGUILAR et al., 2014)

<sup>13</sup>(HAJIRAHIMI; KHASHEI, 2019b)

De fato este trabalho encontrou modelos melhores, porque para casos onde a combinação com o modelo ML obtém pior valor de MSE, como nas séries Carsales e Gas, o trabalho proposto obteve resultado de MSE superior. Para a série Carsales, todas as combinações (ARIMA + MLP, ARIMA + SVR, SARIMA + MLP e SARIMA + SVR) tem pior valor de MSE que os modelos únicos (ARIMA e SARIMA). Para a série Gas, algumas combinações (ARIMA + MLP e SARIMA + MLP) tem pior valor de MSE que os modelos únicos (ARIMA e SARIMA). Esta descoberta reafirma que a abordagem proposta é capaz de encontrar pesos adequados para a combinação da previsão linear e não linear.

Além disso, todos os valores de desempenho encontrados por este trabalho superaram os resultados dos modelos de Zhang (ZHANG, 2003) e Hajirahimi (HAJIRAHIMI; KHASHEI, 2019b).

Apesar dos esforços, o trabalho proposto não conseguiu obter um melhor resultado para as séries Electricity e Lake. Na série Electricity, a abordagem proposta conseguiu o segundo, terceiro, quarto e quinto melhores resultados. Já na série Lake, a abordagem proposta conseguiu o segundo, terceiro e quarto melhor MSE. Estes resultados corroboram com a motivação do desenvolvimento desta pesquisa, porque não existe um único modelo que obtenha melhor resultados para todos os tipos de comportamentos encontrados em séries temporais do mundo real.

A Tabela 8 apresenta a diferença percentual entre a melhor versão do modelo proposto e os outros modelos da literatura.

Nesta comparação, o percentual máximo de melhoria pelo modelo proposto foi de 83.76% para a série Carsales em relação à combinação SARIMA + MLP, 96.87% para a série Electricity em relação com o modelo ARIMA, 84.92% para a série Gas em relação à CNN, 65.74% para a série Lake em relação à CNN, 99.36% para a série Nordic em relação à CNN, 50.16% para a série Pigs em relação à CNN, 86.95% para a série Pollution em relação à CNN, 74.63% para a série Redwine em relação ao SARIMA e ao ARIMA.

---

<sup>1</sup>(BOX et al., 2016)

<sup>2</sup>(OLIVEIRA; LUDERMIR, 2015)

<sup>3</sup>(BABU; REDDY, 2014)

<sup>4</sup>(OLIVEIRA; LUDERMIR, 2014)

<sup>5</sup>(PANIGRAHI; BEHERA, 2017)

<sup>6</sup>(ZHANG, 2003)

<sup>7</sup>(PAI; LIN, 2005)

<sup>8</sup>(YAN; MA, 2016)

<sup>9</sup>(LIU; TAN; WANG, 2019)

<sup>10</sup>(OLIVEIRA; SILVA; NETO, 2021)

<sup>11</sup>(RUIZ-AGUILAR; TURIAS; JIMÉNEZ-COME, 2014)

<sup>12</sup>(RUIZ-AGUILAR et al., 2014)

<sup>13</sup>(HAJIRAHIMI; KHASHEI, 2019b)

Tabela 8 – Diferença percentual entre a melhor versão da abordagem proposta e outros modelos.

Método	Carsales	Electricity	Gas	Lake	Nordic	Pigs	Pollution	Redwine
ARIMA <sup>1</sup>	29.21	96.87	59.52	14.47	50.11	18.25	43.46	74.63
SARIMA <sup>1</sup>	25.88	68.72	59.52	8.11	50.45	18.25	39.51	74.63
CNN <sup>2</sup>	77.66	93.86	84.92	65.74	99.36	50.16	86.95	73.07
LSTM <sup>2</sup>	54.01	67.52	77.12	27.66	77.08	21.68	69.65	74.55
Babu <sup>3</sup>	45.45	50.96	42.43	9.33	35.69	30.96	73.79	54.13
Oliveira <sup>2</sup>	67.19	56.16	53.92	13.92	16.79	26.61	66.86	33.70
Oliveira <sup>4</sup>	63.58	84.92	66.79	7.48	36.26	24.84	55.41	45.21
Panigrahi <sup>5</sup>	58.14	-29.93	67.11	-0.74	39.94	12.08	32.95	57.12
ARIMA + MLP <sup>6</sup>	74.65	22.94	71.90	4.90	15.50	9.51	32.42	16.51
ARIMA + SVR <sup>7</sup>	35.05	17.21	43.40	6.21	4.80	6.14	39.51	23.40
ARIMA + RBF <sup>8</sup>	33.33	68.99	59.28	20.93	62.93	8.63	39.51	73.43
ARIMA + LSTM <sup>9</sup>	28.00	58.41	63.95	2.86	6.03	19.66	30.80	31.98
ARIMA + CNN <sup>10</sup>	51.72	49.43	48.86	6.21	15.18	17.52	28.51	56.41
SARIMA + MLP <sup>11</sup>	83.76	19.46	67.66	5.56	18.66	9.51	30.80	25.57
SARIMA + SVR <sup>12</sup>	35.05	17.59	43.40	6.21	4.80	6.14	39.51	23.40
Hajirahimi and Khashei (ARIMA + MLP) <sup>13</sup>	16.00	22.27	67.07	6.21	17.11	8.63	0.00	28.27
Hajirahimi and Khashei (ARIMA + SVR) <sup>13</sup>	26.32	15.24	25.00	4.90	5.63	6.80	39.08	23.40
Hajirahimi and Khashei (SARIMA + MLP) <sup>13</sup>	29.21	17.59	61.65	6.21	18.35	8.63	5.05	32.56
Hajirahimi and Khashei (SARIMA + SVR) <sup>13</sup>	26.74	15.24	25.00	4.90	5.63	5.95	39.51	23.40

Fonte: Elaborada pela autora (2024).

O ganho percentual médio para cada série alcançado pela abordagem proposta foi de: 45.31% para a série Carsales, 42.81% para a série Electricity, 57.29% para a série Gas, 11.63% para a série Lake, 30.54% para a série Nordic, 16.31% para a série Pigs, 41.75% para a série Pollution e 44.49% para a série Redwine. Estes resultados confirmam que a abordagem proposta obteve melhor resultado que outros modelos da literatura para as séries analisadas.

No Anexo A é possível ver os gráficos comparativos da série, a abordagem proposta e Modelo base (ARIMA ou SARIMA), onde se verifica que a abordagem proposta consegue se aproximar mais da série original que os modelos base.

#### 4.3 EQUAÇÕES DE COMBINAÇÃO

A abordagem proposta realiza a combinação das previsões lineares e não lineares de forma ponderada por meio de uma função linear e uma das vantagens deste modelo é que a combinação dos modelos é transparente, porque é possível ver a ponderação aplicada em cada valor da previsão.

A Tabela 9 apresenta as equações obtidas para as melhores versões da abordagem proposta. Nesta tabela  $\hat{L}$  representa a previsão linear (ARIMA ou SARIMA) e  $\hat{N}$  corresponde a previsão não linear (MLP ou SVR), já o número subscrito representa o índice da previsão anterior.

Tabela 9 – Equações de combinação para as melhores versões da abordagem proposta.

Série Temporal	Linear									Não Linear											
	$\hat{L}$	$\hat{L}_1$	$\hat{L}_2$	$\hat{L}_3$	$\hat{L}_4$	$\hat{L}_5$	$\hat{L}_6$	$\hat{L}_7$	$\hat{L}_8$	$\hat{L}_9$	$\hat{N}$	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_3$	$\hat{N}_4$	$\hat{N}_5$	$\hat{N}_6$	$\hat{N}_7$	$\hat{N}_8$	$\hat{N}_9$	
Carsales	0.959	0.051	-	-	-	-	-	-	-	-	0.256	-	0.184	-0.103	-	-	-	-	-	-	-
Electricity	1.003	-	-	-	-	-	-	-	-	-	1.01	0.017	-0.082	-0.212	-0.138	-0.046	-0.071	-	-	-	-
Gas	0.957	-	-	-	-	-	-	-	0.008	0.044	0.776	-0.012	-0.029	-	-0.05	-	-0.0142	-0.046	-	-	-
Lake	0.995	-	-	-	-	-0.036	-	0.006	0.03	0.005	0.63	-	-	-	-	-	-	-	-	-	-0.164
Nordic	0.995	-	0.003	-	-	-	-	0.0004	-	-	0.651	0.087	-	-	0.005	-0.022	-0.061	-0.017	-0.048	-	-
Pigs	1.028	-	-	-	-	-	-	-	-	-	0.944	-	-	-	-	0.518	-	-	0.518	-	-
Pollution	1.016	-	-	-	-	-	-	-	-	-	0.581	-	-	-	-	-	-	-	-	-	-
Redwine	1	-	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-0.313	-	0.056	-	-

Fonte: Elaborada pela autora (2024).

Estas equações podem representar a relação entre as previsões (linear e não linear) e as ponderações podem indicar a contribuição de cada previsão para a previsão final, podendo ser uma contribuição positiva ou corretiva (negativa).

Para todas as séries analisadas, foi observado que a maior ponderação ocorre na previsão atual ( $\hat{L}$  ou  $\hat{N}$ ), indicando que quanto mais próximo do valor atual, maior relevância tem para a previsão. É possível verificar também o comportamento de correção da previsão com a utilização de ponderação negativa nas séries: Carsales, Electricity, Gas, Lake, Nordic e Redwine.

Para algumas séries, somente três valores anteriores foram utilizados na previsão. Já para outras séries, foi necessário utilizar o número máximo de valores anteriores. A arquitetura do modelo facilita a flexibilidade e auxilia o algoritmo a encontrar a melhor quantidade de previsões que maximiza a acurácia da combinação.

Outro ponto interessante da Tabela 9, é observado na série Pollution. Para esta série, o MSE obtido foi o mesmo do modelo de Hajirahimi (HAJIRAHIMI; KHASHEI, 2019b) na combinação ARIMA + MLP, o qual utiliza pesos somente na previsão atual (um peso para previsão atual linear e outro peso para a previsão não linear), a equação obtida pela abordagem proposta apresenta que realmente este foi o melhor modelo encontrado.

Para a série Redwine, o melhor resultado de MSE foi composto por: previsão linear atual ponderada, previsão não linear ponderada e duas previsões não lineares anteriores ( $-\hat{N}_6$  e  $\hat{N}_8$ ). Este resultado mostra que uma pequena modificação no modelo de Zhang (ZHANG, 2003) ou no modelo de Hajirahimi (HAJIRAHIMI; KHASHEI, 2019b) pode levar a melhores resultados de

MSE.

#### 4.4 RESUMO DO CAPÍTULO

Neste capítulo foram exibidos os resultados experimentais obtidos com a abordagem proposta. Foram analisadas as combinações dos modelos lineares (ARIMA e SARIMA) com os modelos não lineares (MLP e SVR) em comparação com modelos únicos e híbridos. Inicialmente foram analisados a abordagem proposta com os modelos lineares que serviram como base (ARIMA e SARIMA). Foi observado que para todas as séries analisadas a abordagem proposta conseguiu um melhor resultado que os modelos-base, obteve melhores resultados até quando um dos modelos da combinação tinha pior valor de MSE. Além dessa análise, também foi realizada uma comparação com os trabalhos que utilizam a função linear na combinação dentro da literatura e os resultados foram promissores. A abordagem proposta obteve os melhores resultados para seis das oito séries estudadas. Nesta análise foi possível também verificar: uma melhora percentual da abordagem proposta em relação aos demais modelos da literatura e a abordagem proposta superou o desempenho dos modelos de Zhang e Hajirahimi para todas as séries. Além disto, também foi apresentado o modelo *white box* da combinação, onde é gerada uma equação com os pesos dos modelos linear e não linear para cada valor utilizado na previsão.

## 5 ARQUITETURA DO CROMOSSOMO, EFICÁCIA DO ALGORITMO GENÉTICO E AVALIAÇÃO ALGORITMO MÍNIMOS QUADRADOS

Este capítulo apresenta a convergência do algoritmo genético utilizado neste trabalho, como a arquitetura do cromossomo usado pelo algoritmo genético influencia na obtenção dos resultados e a aplicação do algoritmo de mínimos quadrados na busca da ponderação. Desta forma, o capítulo está dividido nas sessões: Seção 5.1 que apresentará em detalhes como a convergência do algoritmo genético se comporta, a Seção 5.2 que mostra a arquitetura do cromossomo, já a Seção 5.3 apresenta a aplicação do algoritmo de mínimos quadrados a metodologia proposta e para finalizar, a Seção 5.4 que mostra as considerações do capítulo.

### 5.1 EFICÁCIA DO ALGORITMO

Segundo Arora (ARORA, 2017), um algoritmo que comprovadamente converge a partir de um ponto arbitrário é chamado de método robusto. Desta forma, a convergência corresponde à capacidade do algoritmo de chegar à solução do problema (encontrar o ponto mínimo ou máximo), mesmo que existam várias execuções, o algoritmo sempre tende a um valor ótimo.

Visando validar a convergência do algoritmo genético modificado e assim garantir que ele é capaz de encontrar a solução para o problema da ponderação, serão utilizadas as seguintes métricas indicativas da convergência: evolução do *fitness* do melhor indivíduo ao longo do tempo, critérios de parada e média e desvio padrão das execuções do algoritmo para cada série.

Para cada série, o algoritmo genético modificado foi executado 10 vezes, em cada execução foi modificada a semente de aleatoriedade do algoritmo (*random seed*). Portanto, é possível a obtenção das métricas de média e desvio padrão. A média de MSE obtida nos experimentos são apresentadas na Tabela 10. Já o desvio padrão do MSE é apresentado na Tabela 11.

Observe que a média de MSE das execuções teve como resultado no máximo valores com duas casas decimais, apontando uma diferença mínima entre os resultados. Já o desvio padrão ficou com no mínimo 1 casa decimal, um valor pequeno de desvio para 10 execuções, o que comprova que o algoritmo consegue alcançar o valor mínimo com uma pequena perturbação.

Outro fator importante na garantia da convergência do algoritmo é a utilização de vários critérios de parada, assim como indica (EIBEN; SMITH, 2015). Foram utilizados como critérios de parada: o número de gerações, o número de gerações sem melhoria e *early stop* baseado

Tabela 10 – Média de MSE obtida nos experimentos.

Conjunto de Dados	ARIMA + MLP	ARIMA + SVR	SARIMA + MLP	SARIMA + SVR
Carsales	0.024	0.0191	0.0314	0.0188
Electricity	0.0021	0.002	0.0019	0.002
Gas	0.007	0.0041	0.0068	0.0035
Lake	0.0014	0.0014	0.0015	0.0014
Nordic	0.0003	0.0002	0.0003	0.0002
Pigs	0.0102	0.0097	0.0106	0.0097
Pollution	0.0101	0.0203	0.0111	0.0176
Redwine	0.01	0.01	0.0113	0.0094

**Fonte:** Elaborada pela autora (2024).

Tabela 11 – Desvio padrão obtido no experimentos.

Conjunto de Dados	ARIMA + MLP	ARIMA + SVR	SARIMA + MLP	SARIMA + SVR
Carsales	0.0058	0.011	0.0065	0.0011
Electricity	0.0002	0.0	0.0001	0.0001
Gas	0.0008	0.0007	0.0007	0.0005
Lake	0.0	0.0	0.0	0.0
Nordic	0.0	0.0	0.0	0.0
Pigs	0.0002	0.0002	0.0004	0.0003
Pollution	0.0009	0.0067	0.0011	0.0041
Redwine	0.0009	0.0002	0.001	0.0007

**Fonte:** Elaborada pela autora (2024).

no *generalization loss class* GL (PIOTROWSKI; NAPIORKOWSKI, 2013).

Ainda visando auxiliar a convergência do algoritmo, durante a execução são utilizadas estratégias de ampliação da divergência populacional, são elas: várias operações de *crossover* e mutação e a injeção randômica de 10% dos piores indivíduos da população.

Para completar a análise da convergência do algoritmo, são apresentados no Anexo B os gráficos de *fitness* do melhor indivíduo ao longo do tempo para a melhor execução de cada experimento e para cada série.

Assim como pode ser visualizado nos gráficos do Anexo B, o *fitness* ao longo da execução vai sendo reduzido até chegar a um mínimo, com a exceção da série Pollution, onde o mínimo foi encontrado logo início do algoritmo.

Em vista disso, pode-se concluir que o algoritmo genético modificado consegue alcançar a convergência e, assim é capaz de encontrar a solução para o problema da ponderação das previsões deste trabalho.

## 5.2 ARQUITETURA DO CROMOSSOMO

Para verificar a importância da utilização do peso binário na arquitetura do cromossomo, foi desenvolvido um experimento com o algoritmo PSO, onde foi aplicada a modificação do modelo para conter somente os pesos reais.

O PSO desenvolvido para realizar a comparação com o AG modificado não possui a busca pelo peso binário, dessa forma a partícula é dada por:

$$R_{L0}\hat{L}_0 + \dots + R_{L9}\hat{L}_9 + R_{NL0}\hat{N}_0 + \dots + R_{NL9}\hat{N}_9. \quad (5.1)$$

O PSO realiza a busca somente do parâmetro real, que na Equação 5.1 é representado por  $R$ . Para ser possível realizar a comparação, também foi implementado o algoritmo PSO com a mesma arquitetura utilizada pelo AG modificado, com os valores de pesos binários e reais.

O algoritmo PSO foi baseado no trabalho (YOLCU; EGRIOGLU; ALADAG, 2013). Os parâmetros aplicados ao algoritmo são apresentados na Tabela 12.

Tabela 12 – Parâmetros do PSO.

Parâmetro	Valor
População	60
Gerações	1000
Espera estagnação	100
Valor de Perda de Generalização (GL)	20%
C1i	2
C1f	3
C2i	2
C2f	3
w1	1
w2	2

**Fonte:** Elaborada pela autora (2024).

onde  $C1$  e  $C2$  representam os coeficientes cognitivo e social do algoritmo, respectivamente. A letra  $i$  e  $f$  ao lado destes parâmetros indica o intervalo inicial e final deles. Já o parâmetro  $w$  representa a inércia e o símbolo 1 e 2 ao lado dele, indica o valor inicial e final da inércia.

O PSO possui os mesmos critérios de parada do AG: número de gerações, iterações em estagnação e o *early stop* baseado no GL.

O conjunto de dados e a divisão deste conjunto foi o mesmo utilizado pelo AG, assim como foi apresentado na Seção 4.1. O PSO faz a escolha da melhor partícula pelo conjunto de validação.

O experimento foi realizado 10 vezes, em que a cada execução foi modificada a semente de aleatoriedade (*random seed*), mas os resultados apresentados a seguir são os melhores obtidos em todas as execuções.

A Tabela 13 apresenta a comparação do algoritmo PSO sem binários (PSO), com binários (PSO - bin) e os modelos-base (ARIMA e SARIMA).

Tabela 13 – Avaliação das versões do PSO - com e sem valores binários.

Método	Carsales	Electricity	Gas	Lake	Nordic	Pigs	Pollution	Redwine
ARIMA <sup>1</sup>	1.78E-02	5.69E-02	<b>6.67E-03</b>	1.59E-03	<b>4.37E-04</b>	<b>1.14E-02</b>	1.53E-02	3.11E-02
SARIMA <sup>1</sup>	<b>1.70E-02</b>	<b>5.69E-03</b>	<b>6.67E-03</b>	<b>1.48E-03</b>	4.40E-04	<b>1.14E-02</b>	1.43E-02	3.11E-02
PSO (ARIMA + MLP)	1.26E-01	1.82E-02	5.40E-02	5.35E-03	3.33E-03	2.61E-02	7.73E-02	5.52E-02
PSO (ARIMA + SVR)	4.10E-02	2.14E-02	3.00E-02	3.85E-03	3.90E-03	1.48E-02	2.03E-02	5.50E-02
PSO (SARIMA + MLP)	1.35E-01	2.36E-02	6.34E-02	7.60E-03	5.72E-03	2.04E-02	4.51E-02	5.72E-02
PSO (SARIMA + SVR)	6.47E-02	2.30E-02	4.00E-02	5.40E-03	7.18E-03	1.48E-02	3.25E-02	3.52E-02
PSO (ARIMA + MLP) - bin	1.21E-01	1.01E-02	3.00E-02	3.46E-03	2.75E-03	1.53E-02	3.46E-02	4.28E-02
PSO (ARIMA + SVR) - bin	2.61E-02	1.27E-02	3.10E-02	2.20E-03	1.28E-03	1.39E-02	<b>1.10E-02</b>	3.62E-02
PSO (SARIMA + MLP) - bin	9.54E-02	6.50E-03	1.25E-02	6.72E-03	1.10E-03	1.34E-02	4.21E-02	2.59E-02
PSO (SARIMA + SVR) - bin	2.15E-02	1.11E-02	2.04E-02	2.17E-03	2.98E-03	1.31E-02	3.46E-02	<b>1.32E-02</b>

Fonte: Elaborada pela autora (2024).

Da Tabela 13, é possível observar que o PSO com binários obteve um desempenho superior que o PSO sem binários, dado que o PSO com binários obteve MSE melhor que os modelos-base para as séries Pollution e Redwine. Também é possível verificar que o PSO com binários obteve melhor desempenho em comparação com PSO sem binários para todas as séries estudadas. Além disso, o modelo PSO com binários que utiliza o modelo SVR como modelo não linear obteve melhor resultado para as séries: Carsales, Lake, Pigs, Pollution e Redwine. Já para as séries: Electricity, Gas e Nordic, o PSO com binários que utiliza a MLP obteve melhor desempenho.

Este resultado corrobora com a arquitetura utilizada pelo AG, dado que para todas as séries o algoritmo PSO com binários obteve melhores resultados de MSE.

Já na Tabela 14, são apresentados os resultados do PSO com e sem binários em relação aos modelos da literatura. É possível verificar que o algoritmo genético obteve melhor valor de MSE que o algoritmo PSO, mas que o PSO com binários obteve valores de MSE mais competitivos que o PSO sem binários.

<sup>1</sup>(BOX et al., 2016)

Tabela 14 – Valores de erro quadrático médio da melhor versão do PSO, modelo proposto e literatura de sistemas únicos e híbridos.

Método	Carsales	Electricity	Gas	Lake	Nordic	Pigs	Pollution	Redwine
ARIMA <sup>1</sup>	1.78E-02	5.69E-02	6.67E-03	1.59E-03	4.37E-04	1.14E-02	1.53E-02	3.11E-02
SARIMA <sup>1</sup>	1.70E-02	5.69E-03	6.67E-03	1.48E-03	4.40E-04	1.14E-02	1.43E-02	3.11E-02
CNN <sup>2</sup>	5.64E-02	2.90E-02	1.79E-02	3.97E-03	3.39E-02	1.87E-02	6.63E-02	2.93E-02
LSTM <sup>2</sup>	2.74E-02	5.48E-03	1.18E-02	1.88E-03	9.51E-04	1.19E-02	2.85E-02	3.10E-02
Babu <sup>3</sup>	2.31E-02	3.63E-03	4.69E-03	1.50E-03	3.39E-04	1.35E-02	3.30E-02	1.72E-02
Oliveira <sup>2</sup>	3.84E-02	4.06E-03	5.86E-03	1.58E-03	2.62E-04	1.27E-02	2.61E-02	1.19E-02
Oliveira <sup>4</sup>	3.46E-02	1.18E-02	8.13E-03	1.47E-03	3.42E-04	1.24E-02	1.94E-02	1.44E-02
Panigrahi <sup>5</sup>	3.01E-02	<b>1.37E-03</b>	8.21E-03	<b>1.35E-03</b>	3.63E-04	1.06E-02	1.29E-02	1.84E-02
ARIMA + MLP <sup>6</sup>	4.97E-02	2.31E-03	9.61E-03	1.43E-03	2.58E-04	1.03E-02	1.28E-02	9.45E-03
ARIMA + SVR <sup>7</sup>	1.94E-02	2.15E-03	4.77E-03	1.45E-03	2.29E-04	9.93E-03	1.43E-02	1.03E-02
ARIMA + RBF <sup>8</sup>	1.89E-02	5.74E-03	6.63E-03	1.72E-03	5.88E-04	1.02E-02	1.43E-02	2.97E-02
ARIMA + LSTM <sup>9</sup>	1.75E-02	4.28E-03	7.49E-03	1.40E-03	2.32E-04	1.16E-02	1.25E-02	1.16E-02
ARIMA + CNN <sup>10</sup>	2.61E-02	3.52E-03	5.28E-03	1.45E-03	2.57E-04	1.13E-02	1.21E-02	1.81E-02
SARIMA + MLP <sup>11</sup>	7.76E-02	2.21E-03	8.35E-03	1.44E-03	2.68E-04	1.03E-02	1.25E-02	1.06E-02
SARIMA + SVR <sup>12</sup>	1.94E-02	2.16E-03	4.77E-03	1.45E-03	2.29E-04	9.93E-03	1.43E-02	1.03E-02
Hajirahimi and Khashei (ARIMA + MLP) <sup>13</sup>	1.50E-02	2.29E-03	8.20E-03	1.45E-03	2.63E-04	1.02E-02	<b>8.65E-03</b>	1.10E-02
Hajirahimi and Khashei (ARIMA + SVR) <sup>13</sup>	1.71E-02	2.10E-03	3.60E-03	1.43E-03	2.31E-04	1.00E-02	1.42E-02	1.03E-02
Hajirahimi and Khashei (SARIMA + MLP) <sup>13</sup>	1.78E-02	2.16E-03	7.04E-03	1.45E-03	2.67E-04	1.02E-02	9.11E-03	1.17E-02
Hajirahimi and Khashei (SARIMA + SVR) <sup>13</sup>	1.72E-02	2.10E-03	3.60E-03	1.43E-03	2.31E-04	9.91E-03	1.43E-02	1.03E-02
PSO (ARIMA + MLP) <sup>14</sup>	1.26E-01	1.82E-02	5.40E-02	5.35E-03	3.33E-03	2.61E-02	7.73E-02	5.52E-02
PSO (ARIMA + SVR) <sup>14</sup>	4.10E-02	2.14E-02	3.00E-02	3.85E-03	3.90E-03	1.48E-02	2.03E-02	5.50E-02
PSO (SARIMA + MLP) <sup>14</sup>	1.35E-01	2.36E-02	6.34E-02	7.60E-03	5.72E-03	2.04E-02	4.51E-02	5.72E-02
PSO (SARIMA + SVR) <sup>14</sup>	6.47E-02	2.30E-02	4.00E-02	5.40E-03	7.18E-03	1.48E-02	3.25E-02	3.52E-02
PSO (ARIMA + MLP) - bin <sup>14</sup>	1.21E-01	1.01E-02	3.00E-02	3.46E-03	2.75E-03	1.53E-02	3.46E-02	4.28E-02
PSO (ARIMA + SVR) - bin <sup>14</sup>	2.61E-02	1.27E-02	3.10E-02	2.20E-03	1.28E-03	1.39E-02	1.10E-02	3.62E-02
PSO (SARIMA + MLP) - bin <sup>14</sup>	9.54E-02	6.50E-03	1.25E-02	6.72E-03	1.10E-03	1.34E-02	4.21E-02	2.59E-02
PSO (SARIMA + SVR) - bin <sup>14</sup>	2.15E-02	1.11E-02	2.04E-02	2.17E-03	2.98E-03	1.31E-02	3.46E-02	1.32E-02
<b>Abordagem Proposta</b>	<b>1.26E-02</b>	1.78E-03	<b>2.70E-03</b>	1.36E-03	<b>2.18E-04</b>	<b>9.32E-03</b>	<b>8.65E-03</b>	<b>7.89E-03</b>

Fonte: Elaborada pela autora (2024).

A arquitetura do modelo híbrido permite que a equação de combinação seja obtida como solução do algoritmo, desta forma as equações obtidas para a melhor versão do PSO sem binários, são apresentadas na Tabela 15 e para o PSO com binários é apresentado na Tabela 16.

<sup>1</sup>(BOX et al., 2016)

<sup>2</sup>(OLIVEIRA; LUDERMIR, 2015)

<sup>3</sup>(BABU; REDDY, 2014)

<sup>4</sup>(OLIVEIRA; LUDERMIR, 2014)

<sup>5</sup>(PANIGRAHI; BEHERA, 2017)

<sup>6</sup>(ZHANG, 2003)

<sup>7</sup>(PAI; LIN, 2005)

<sup>8</sup>(YAN; MA, 2016)

<sup>9</sup>(LIU; TAN; WANG, 2019)

<sup>10</sup>(OLIVEIRA; SILVA; NETO, 2021)

<sup>11</sup>(RUIZ-AGUILAR; TURIAS; JIMÉNEZ-COME, 2014)

<sup>12</sup>(RUIZ-AGUILAR et al., 2014)

<sup>13</sup>(HAJIRAHIMI; KHASHEI, 2019b)

<sup>14</sup>(YOLCU; EGRIOGLU; ALADAG, 2013)

Tabela 15 – Equação de combinação para as melhores versões do PSO sem binários.

Série Temporal	Linear									Não Linear								
	$\hat{L}_1$	$\hat{L}_2$	$\hat{L}_3$	$\hat{L}_4$	$\hat{L}_5$	$\hat{L}_6$	$\hat{L}_7$	$\hat{L}_8$	$\hat{L}_9$	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_3$	$\hat{N}_4$	$\hat{N}_5$	$\hat{N}_6$	$\hat{N}_7$	$\hat{N}_8$	$\hat{N}_9$
Carsales	0.2922	0.9537	-0.9774	0.1789	0.166	-0.2693	0.6959	-0.7251	0.2723	-0.0844	0.858	0.1158	0.9372	0.0174	-0.622	-0.1918	0.1572	0.7714
Electricity	0.6917	-0.1363	0.1463	0.4804	0.0763	0.1916	-0.4344	0.7496	-0.5874	-0.2335	-0.3828	0.7406	0.1836	-0.518	-0.5214	-0.2122	0.2277	0.621
Gas	0.9383	-0.555	0.3378	0.8678	0.2871	0.3818	-0.6468	-0.4004	0.3536	0.9711	0.3002	0.6816	0.2307	0.5189	-0.105	-0.2355	-0.3319	0.5792
Lake	0.7867	0.2558	-0.3179	0.2466	0.5958	-0.8061	0.5546	-0.6367	0.6287	-0.9554	-0.531	0.0539	-0.8012	-0.0754	-0.5398	-0.8934	0.8693	0.1411
Nordic	0.9368	-0.4498	0.3516	0.3268	0.3458	0.2254	-0.5785	0.3477	-0.2238	0.8131	0.9775	0.827	0.9833	-0.9303	0.7997	-0.5003	0.1976	0.279
Pigs	0.7397	0.618	-0.0056	0.2437	0.7614	-0.3958	-0.8224	0.0531	0.8063	0.8691	-0.5925	0.1976	-0.8793	0.3047	0.8537	-0.3199	0.1959	-0.3205
Pollution	0.9	0.017	-0.9633	0.4527	0.4448	0.5115	-0.3651	0.1043	-0.0977	0.184	-0.113	0.6743	-0.7912	0.1717	-0.85	0.0206	0.7956	0.9925
Redwine	-0.422	-0.8613	0.7805	0.0927	0.3797	0.6541	0.1987	0.0131	-0.4342	0.787	0.9834	-0.0229	-0.7614	0.0134	-0.7739	-0.3104	-0.665	-0.3757

Fonte: Elaborada pela autora (2024).

Tabela 16 – Equação de combinação para a melhor versão do PSO com binários.

Série Temporal	Linear									Não Linear								
	$\hat{L}_1$	$\hat{L}_2$	$\hat{L}_3$	$\hat{L}_4$	$\hat{L}_5$	$\hat{L}_6$	$\hat{L}_7$	$\hat{L}_8$	$\hat{L}_9$	$\hat{N}_1$	$\hat{N}_2$	$\hat{N}_3$	$\hat{N}_4$	$\hat{N}_5$	$\hat{N}_6$	$\hat{N}_7$	$\hat{N}_8$	$\hat{N}_9$
Carsales	0.9793	-	0.2237	-0.2976	-	-	-	-	-0.0066	-0.1721	0.3730	0.2774	-0.9720	-	-	0.7481	-	-0.0927
Electricity	0.7791	-	-	-	-0.2845	0.3968	-	-	0.0367	-0.5314	0.7118	-	-0.7161	-0.23	-	-	-	-
Gas	0.7522	0.0936	0.0177	-0.2427	-	-0.0223	-	0.3690	-	0.2970	-	0.0592	-	-	-0.7024	-0.1421	-	-
Lake	0.9768	-	-	-	-	-	-	-	-	-0.8377	-	-	-	-	-	-0.7526	0.6886	0.8828
Nordic	0.9793	-	0.2237	-0.2976	-	-	-	-	-0.0066	-0.1721	0.3730	0.2774	-0.9720	-	-	0.7481	-	-0.0927
Pigs	0.9007	-	-	0.4564	-0.6530	-0.3886	0.0534	0.6560	-	-0.0491	-	0.1010	0.6400	-0.2864	-	-	-0.0944	0.4471
Pollution	0.9976	-0.4121	-	-	-	0.3699	-0.1034	-	-	-0.1089	0.932	-	-	-0.65	0.4339	-0.2898	0.9723	-
Redwine	0.6100	-	0.0319	-0.0244	-	-0.0483	0.5476	-	-	0.7280	0.0398	-	-	-0.5569	-	-	-	-

Fonte: Elaborada pela autora (2024).

Das Tabelas 15 e 16 é possível observar pela quantidade de pesos utilizados na previsão atual e anteriores, que o algoritmo que realiza a busca pelos valores binários na combinação, foi capaz de encontrar quais as previsões que realmente importam para a previsão final, pois várias valores de previsões foram descartadas. Já no PSO sem binários, todas as previsões receberam uma ponderação diferente de zero.

### 5.3 AVALIAÇÃO ALGORITMO DE MÍNIMOS QUADRADOS

Dado que a abordagem proposta utiliza uma combinação aditiva simples para reunir os modelos de previsão, surge a ideia de utilizar o algoritmo de mínimos quadrados como algoritmo a selecionar os pesos para a combinação.

Com o intuito de verificar a eficácia do algoritmo de mínimos quadrados na combinação dos modelos de previsão, utilizamos a biblioteca da linguagem de programação Python (`np.linalg.lstsq`) e foi buscado a ponderação da previsão atual linear e não linear. Foram utilizados 10 valores (1 valor da previsão atual e 9 valores de previsão anterior para cada modelo), dessa forma serão obtidos 20 pesos no total, assim como utilizado na avaliação do algoritmo PSO.

Dois experimentos foram propostos. O primeiro experimento faz a busca pelos 20 valores de ponderação (10 lineares e 10 não-lineares). Já o segundo experimento utiliza a máscara da equação obtida pelo algoritmo genético, para que somente os valores de previsão que foram utilizados como solução pela AG estejam disponíveis para a ponderação do algoritmo de mínimos quadrados.

O conjunto de dados foi separado em treinamento, validação e teste, com a mesma quantidade de dados utilizada pela AG. O conjunto de validação foi utilizado para escolha do modelo e o conjunto de teste foi utilizado para obtenção do desempenho do modelo de combinação.

Para o primeiro experimento foram obtidos os seguintes valores de MSE aplicados ao conjunto de teste apresentados na Tabela 17. Os valores da combinação aplicada aos modelos de previsão linear e não lineares utilizando o algoritmo de mínimos quadrados é representado na tabela por *MQ*.

Pode-se observar da Tabela 17 que o algoritmo de mínimos quadrados não foi capaz de superar o desempenho dos modelos base (ARIMA e SARIMA) para nenhuma das séries. Além disto, o algoritmo de mínimos quadrados não foi capaz de superar o desempenho do algoritmo genético.

Tabela 17 – Valores de MSE do conjunto de teste - Algoritmo de Mínimos Quadrados

Método	Carsales	Electricity	Gas	Lake	Nordic	Pigs	Pollution	Redwine
ARIMA	1.78E-02	5.69E-02	<b>6.67E-03</b>	1.59E-03	<b>4.37E-04</b>	<b>1.14E-02</b>	1.53E-02	<b>3.11E-02</b>
SARIMA	<b>1.70E-02</b>	<b>5.69E-03</b>	<b>6.67E-03</b>	<b>1.48E-03</b>	4.40E-04	<b>1.14E-02</b>	<b>1.43E-02</b>	<b>3.11E-02</b>
MQ (ARIMA + MLP)	7.19E-02	5.14E-02	5.75E-02	2.00E-02	2.29E-02	1.93E-02	1.26E-01	1.45E-01
MQ (ARIMA + SVR)	5.02E-01	4.16E-02	4.07E-02	1.89E-02	2.52E-02	2.13E-02	1.81E06	7.81E-02
MQ (SARIMA + MLP)	1.75E-01	4.96E-02	4.34E-02	2.02E-02	2.32E-02	1.87E-02	1.19E-01	1.34E-01
MQ (SARIMA + SVR)	1.30E-00	3.92E-02	3.77E-02	1.81E-02	2.52E-02	2.12E-02	1.66E06	8.554E-02

Já os resultados obtidos com o segundo experimento, a Tabela 18 apresenta a máscara aplicada a cada uma das séries temporais.

Tabela 18 – Máscara Aplicada ao Algoritmo de Mínimos Quadrados

Série	Linear									Não Linear										
	L	L1	L2	L3	L4	L5	L6	L7	L8	L9	N	N1	N2	N3	N4	N5	N6	N7	N8	N9
Carsale	1	1	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0
Electricity	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0
Gas	1	0	0	0	0	0	0	0	1	1	1	1	1	0	1	0	1	1	0	0
Lake	1	0	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	1
Nordic	1	0	1	0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	1	0
Pigs	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0
Polution	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Redwine	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0



Observando os pesos encontrados, é possível verificar que os pesos encontrados pelo algoritmo de mínimos quadrados mesmo com a adição da máscara, não são os mesmos e nem próximos dos valores de pesos que foram encontrados pela AG.

Os valores de MSE de teste, aplicado a melhor versão da AG, são apresentados na Tabela 20.

Tabela 20 – Avaliação da melhor versão da AG aplicada ao algoritmo de mínimos quadrados

Method	Carsales	Electricity	Gas	Lake	Nordic	Pigs	Pollution	Redwine
ARIMA	1.78E-02	5.69E-02	<b>6.67E-03</b>	1.59E-03	<b>4.37E-04</b>	<b>1.14E-02</b>	1.53E-02	<b>3.11E-02</b>
SARIMA	<b>1.70E-02</b>	<b>5.69E-03</b>	<b>6.67E-03</b>	<b>1.48E-03</b>	4.40E-04	<b>1.14E-02</b>	<b>1.43E-02</b>	<b>3.11E-02</b>
LS	4.39E-01	4.78E-01	7.17E-02	1.93E-01	7.91E-02	5.79E-01	2.03E-01	4.36E-01

Ainda é possível verificar que o algoritmo de mínimos quadrados não foi capaz de superar o desempenho dos modelos base e nem o desempenho do algoritmo genético. Estes resultados só corroboram com a necessidade da aplicação do algoritmo genético na busca pela melhor ponderação da combinação das previsões.

#### 5.4 RESUMO DO CAPÍTULO

Neste capítulo foram exibidos os resultados experimentais obtidos com análise da convergência do algoritmo genético e com o estudo da arquitetura utilizado no trabalho. Neste estudo foi possível provar que o algoritmo genético desenvolvido é robusto, dado que ele consegue convergir e assim, é capaz de encontrar uma solução para o problema. Também foi possível verificar que a arquitetura utilizada no trabalho, com o auxílio do algoritmo PSO. Foi possível verificar que o PSO que utiliza a busca pelo valor binário obteve melhores resultados que o PSO que só realiza a busca pela ponderação real. Além disso, foi possível observar que o PSO com binários conseguiu equações de combinação dos modelos menores, pois foi capaz de escolher as previsões que realmente importavam para a previsão final. Portanto, foi apresentada a relevância da utilização da busca pelos pesos binários e reais pelo algoritmo. Adicionado a isto, foram apresentados experimentos utilizando o algoritmo de mínimos quadrados, onde foi possível verificar o desempenho superior do algoritmo genético desenvolvido.

## 6 CONCLUSÃO

Este capítulo apresenta as conclusões desta tese na Seção 6.1 e sugestões para próximas pesquisas, na Seção 6.2.

### 6.1 CONSIDERAÇÕES FINAIS

A proposta desta tese é uma abordagem para combinação das componentes lineares e não lineares da série temporal, aplicado a previsão atual e prévia. O diferencial da abordagem proposta e os trabalhos da literatura é a aplicação da ponderação aos valores da previsão atual e anteriores, além da escolha dos valores de previsão que integrarão a previsão final. Desta forma, inicialmente é aplicado o modelo linear, obtendo a previsão da série temporal e o respectivo resíduo. Posteriormente, um modelo não linear modela o resíduo, consistindo na previsão não linear. Finalmente, é aplicado o algoritmo genético para busca da ponderação de cada previsão e realização da combinação linear das previsões atuais e anteriores.

Esta proposta consiste em uma abordagem, onde é possível combinar um modelo de previsão linear com um modelo de previsão de resíduo, e para validar este trabalho foram utilizados os modelos lineares ARIMA e SARIMA e como modelos não lineares, os modelos MLP e SVR. Os modelos lineares foram escolhidos devido a metodologia de Box & Jenkins, onde existe a garantia da modelagem dos padrões lineares. Já os modelos não lineares, foram escolhidos devido a estabilidade e robustez dos modelos.

A análise experimental foi conduzida por meio de oito séries temporais do mundo real e foram realizadas comparações com modelos únicos e híbridos existentes na literatura (BOX et al., 2016; OLIVEIRA; LUDERMIR, 2015; BABU; REDDY, 2014; OLIVEIRA; LUDERMIR, 2014; PANIGRAHI; BEHERA, 2017; ZHANG, 2003; PAI; LIN, 2005; YAN; MA, 2016; LIU; TAN; WANG, 2019; OLIVEIRA; SILVA; NETO, 2021; RUIZ-AGUILAR; TURIAS; JIMÉNEZ-COME, 2014; RUIZ-AGUILAR et al., 2014; HAJIRAHIMI; KHASHEI, 2019b).

Seguindo o teorema *no free lunch* (SILVA et al., 2020), nenhum método é o melhor em todas as séries temporais. Contudo, em termos de MSE (10 execuções), os resultados obtidos na abordagem proposta estão entre os melhores resultados considerando as séries temporais, com melhoria de até 99% em relação a literatura. Como também tem um desempenho melhor que as abordagens da literatura na maioria dos casos.

Este trabalho também realizou a análise da convergência e arquitetura do algoritmo genético modificado e foi possível verificar a robustez do algoritmo.

Desta forma, é possível listar as contribuições desta tese como a seguir:

1. Estratégia genérica que pode ser utilizada com qualquer modelo de previsão linear e de previsão de resíduo, podendo também ser aplicada a qualquer contexto de previsão de séries temporais;
2. Sistema que corrige as previsões lineares e previsões de resíduo em diferentes séries temporais;
3. Apresentação de sistema *white box* de combinação ponderada de preditores.

## 6.2 TRABALHOS FUTUROS

Esta tese propõe uma abordagem capaz de combinar de forma linear e ponderada previsões lineares e não lineares atuais e progressas, exibindo resultados competitivos quando comparados aos métodos da literatura. Desta forma, são apresentadas a seguir algumas oportunidades para pesquisas futuras:

- Desenvolvimento de uma ensemble sequencial que seja capaz de escolher o melhor modelo sequencial (ARIMA + MLP, ARIMA + SVR, SARIMA + MLP ou SARIMA + SVR, porque a escolha do modelo não linear é crucial para obter resultados acurados;
- Investigar outros modelos de aprendizagem de máquina como *Deep Learning*, *CNN* e *LSTM*;
- Aplicação da abordagem proposta em outras séries temporais, tais como: velocidade do vento (de MATTOS NETO et al., 2020), energia solar (ALKHAYAT; MEHMOOD, 2021) e unidade de manutenção de estoque (*Stock Keeping Unit (SKU)*)(MAKRIDAKIS; SPILIOTIS; ASSIMAKOPOULOS, 2021).

## REFERÊNCIAS

- ACKORA-PRAH, J.; GYAMERAH, S. A.; ANDAM, P. S. A heuristic crossover for portfolio selection. *Applied Mathematical Sciences*, v. 8, n. 65, p. 3215–3227, 2014.
- ADHIKARI, R.; AGRAWAL, R. K. An introductory study on time series modeling and forecasting. *CoRR*, abs/1302.6613, 2013.
- AHMED, N. K.; ATIYA, A. F.; GAYAR, N. E.; EL-SHISHINY, H. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, p. 594–621, 2010.
- ALKHAYAT, G.; MEHMOOD, R. A review and taxonomy of wind and solar energy forecasting methods based on deep learning. *Energy and AI*, v. 4, p. 100060, 2021. ISSN 2666-5468.
- ARORA, J. S. *Introduction to Optimum Design (Fourth Edition)*. [S.l.]: Academic Press, 2017.
- BABU, C. N.; REDDY, B. E. A moving-average filter based hybrid arima–ann model for forecasting time series data. *Applied Soft Computing*, v. 23, p. 27–38, 2014.
- BOX, G. E. P.; JENKINS, G. M.; REINSEL, G. C.; LJUNG, G. M. *Time Series Analysis: Forecasting and control*. [S.l.]: Wiley, 2016.
- BROCKWELL, P. J.; DAVIS, R. A. *Introduction to Time Series and Forecasting*. [S.l.]: Springer, 2002.
- BROCKWELL, P. J.; DAVIS, R. A. *Time Series: Theory and Methods*. [S.l.]: Springer, 2006.
- BüYÜKŞAHİN Ümit Çavuş; ERTEKİN Şeyda. Improving forecasting accuracy of time series data using a new arima-ann hybrid method and empirical mode decomposition. *Neurocomputing*, v. 361, p. 151–163, 2019.
- CAPORALE, G. M.; PLASTUN, A. On stock price overreactions: frequency, seasonality and information content. *Journal of Applied Economics*, Routledge, v. 22, n. 1, p. 602–621, 2019.
- CERVANTES-CASTILLO, A.; MEZURA-MONTES, E.; COELLO, C. A. C. An empirical comparison of two crossover operators in real-coded genetic algorithms for constrained numerical optimization problems. In: IEEE. *2014 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. [S.l.], 2014. p. 1–5.
- CERVANTES-CASTILLO, A.; MEZURA-MONTES, E.; COELLO, C. A. C. An empirical comparison of two crossover operators in real-coded genetic algorithms for constrained numerical optimization problems. p. 1–5, 2014.
- CHATFIELD, C.; XING, H. *The Analysis of Time Series: An introduction with r*. [S.l.]: CRC Press, 2019.
- CHEN, J.-C.; WANG, Y.-M. Comparing activation functions in modeling shoreline variation using multilayer perceptron neural network. *Water*, v. 12, n. 5, 2020.
- COLHERINHAS, G. B. *Ferramenta de otimização via algoritmos genéticos com aplicações em engenharia*. 103 p. Dissertação (ENM - Mestrado em Ciências Mecânicas (Dissertações)) — Universidade Federal de Brasília, 2016.

CONTINUOUS blood pressure measurement from one-channel electrocardiogram signal using deep-learning techniques. *Artificial Intelligence in Medicine*, v. 108, p. 101919, 2020.

COWPERTWAIT, P. S.; METCALFE, A. V. *Introductory Time Series with R*. [S.l.]: Springer, 2009.

de MATTOS NETO, P. S. G.; de OLIVEIRA, J. F. L.; SANTOS JÚNIOR, D. S. de O.; SIQUEIRA, H. V.; da NÓBREGA MARINHO, M. H.; MADEIRO, F. A hybrid nonlinear combination system for monthly wind speed forecasting. *IEEE Access*, v. 8, p. 191365–191377, 2020.

de O. Santos Júnior, D. S.; de Oliveira, J. F. L.; de Mattos Neto, P. S. G. An intelligent hybridization of arima with machine learning models for time series forecasting. *Knowledge-Based Systems*, v. 175, p. 72–86, 2019.

DEEP, K.; THAKUR, M. A new crossover operator for real coded genetic algorithms. *Applied Mathematics and Computation*, v. 188, n. 1, p. 895–911, 2007.

DRUCKER, H.; BURGESS, C. J.; KAUFMAN, L.; SMOLA, A. J.; VAPNIK, V. Support vector regression machines. *Advances in neural information processing systems*, p. 155–161, 1997.

EIBEN, A. E.; SMITH, J. E. *Introduction to Evolutionary Computing*. [S.l.]: Springer, 2015.

ENGELBRECHT, A. P. *Computational Intelligence: An introduction*. [S.l.]: Wiley, 2007.

FATYANOSA, T. N.; ARITSUGI, M. An automatic convolutional neural network optimization using a diversity-guided genetic algorithm. *IEEE Access*, v. 9, p. 91410–91426, 2021.

GARCIA, J. C. S.; TANAKA, H.; GIANNETTI, N.; SEI, Y.; SAITO, K.; HOUFUKU, M.; TAKAFUJI, R. Multiobjective geometry optimization of microchannel heat exchanger using real-coded genetic algorithm. *Applied Thermal Engineering*, v. 202, p. 117821, 2022.

GOLDBERG, D. E. *Genetic Algorithms: in search, optimization and machine learning*. [S.l.]: Addison-Wesley Publishing Company, Inc, 1989.

HAJIRAHIMI, Z.; KHASHEI, M. Hybrid structures in time series modeling and forecasting: A review. *Engineering Applications of Artificial Intelligence*, v. 86, p. 83–106, 2019.

HAJIRAHIMI, Z.; KHASHEI, M. Weighted sequential hybrid approaches for time series forecasting. *Physica A: Statistical Mechanics and its Applications*, v. 531, p. 121717, 2019.

HAJIRAHIMI, Z.; KHASHEI, M. Sequence in hybridization of statistical and intelligent models in time series forecasting. *Neural Processing Letters*, 2020.

HAYKIN, S. *Redes Neurais: Princípios e prática*. [S.l.]: Bookman, 2001.

HERRERA, M. L. F.; VERDEGAY, J. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artificial Intelligence Review*, v. 12, p. 265–319, 1998.

HO, Y.-C.; PEPYNE, D. L. Simple explanation of the no free lunch theorem of optimization. In: IEEE. *Proceedings of the 40th IEEE conference on decision and control (Cat. No. 01CH37228)*. [S.l.], 2001. v. 5, p. 4409–4414.

HONG, T.-P.; WANG, H.-S.; CHEN, W.-C. Simultaneously applying multiple mutation operators in genetic algorithms. *Journal of heuristics*, Springer, v. 6, p. 439–455, 2000.

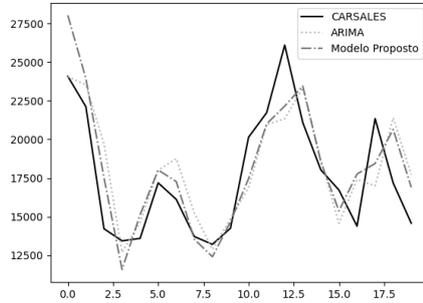
- HONG, T.-P.; WANG, H.-S.; LIN, W.-Y.; LEE, W.-Y. Evolution of appropriate crossover and mutation operators in a genetic process. *Applied intelligence*, Springer, v. 16, p. 7–17, 2002.
- HYNDMAN, R. J.; KHANDAKAR, Y. Automatic time series forecasting: The forecast package for r. *Journal of Statistical Software*, v. 27, n. 3, p. 1–22, 2008.
- IDREES, S. M.; ALAM, M. A.; AGARWAL, P. A prediction approach for stock market volatility based on time series data. *IEEEAccess*, v. 7, p. 17287–17298, 2019.
- IZONIN, I.; TKACHENKO, R.; GREGUS, M.; ZUB, K.; LOTOSHYNSKA, N. Input doubling method based on svr with rbf kernel in clinical practice: Focus on small data. *Procedia Computer Science*, v. 184, p. 606–613, 2021. The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.
- IZÍDIO, D. M. F.; NETO, P. S. G. de M.; BARBOSA, L.; OLIVEIRA, J. F. L. de; MARINHO, M. H. da N.; RISSI, G. F. Evolutionary hybrid system for energy consumption forecasting for smart meters. *Energies*, v. 14, p. 1–19, 2021.
- KAN, S.; CHEN, B.; MENG, J.; CHEN, G. An extended overview of natural gas use embodied in world economy and supply chains: Policy implications from a time series analysis. *Remote Sensing of Environment*, v. 137, 2020.
- KAUR, J.; PARMAR, K. S.; SINGH, S. Autoregressive models in environmental forecasting time series: a theoretical and application review. *Environmental Science and Pollution Research*, Springer, v. 30, n. 8, p. 19617–19641, 2023.
- KAYA, Y.; UYAR, M. et al. A novel crossover operator for genetic algorithms: ring crossover. *arXiv preprint arXiv:1105.0355*, 2011.
- KHASHEI, M.; BIJARI, M. An artificial network (p,d,q) model for timeseries forecasting. *Expert Systems with Applications*, v. 37, p. 479–489, 2010.
- KHASHEI, M.; BIJARI, M. A novel hybridization of artificial neural networks and arima models for time series forecasting. *Applied Soft Computing*, p. 2664–2675, 2011.
- KONTOPOULOU, V. I.; PANAGOPOULOS, A. D.; KAKKOS, I.; MATSOPOULOS, G. K. A review of arima vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet*, v. 15, n. 8, 2023.
- KORA, P.; YADLAPALLI, P. Crossover operators in genetic algorithms: A review. *International Journal of Computer Applications*, Foundation of Computer Science, v. 162, n. 10, 2017.
- KURMI, U.; SINGH, U.; SINGH, R.; SAYYED, M. Forecasting of area, production, and productivity of groundnut using nonlinear growth model in india. *International Journal of Statistics and Applied Mathematics 2023*, 2023.
- L. de O. J. F.; PACÍFICO, L. D. S.; G. de M. N. P. S.; BARREIROS, E. F. S.; RODRIGUES, C. M. d. O.; FILHO, A. T. d. A. A hybrid optimized error correction system for time series forecasting. *Applied Soft Computing*, v. 87, 2020.
- LEE, C.-Y.; YAO, X. Evolutionary programming using mutations based on the Lévy probability distribution. *Evolutionary Computation, IEEE Transactions on*, v. 8, p. 1 – 13, 2004.

- LIU, J.; TAN, X.; WANG, Y. Ccssap: software aging prediction for cloud services based on arima-lstm hybrid model. In: IEEE. *2019 IEEE International Conference on Web Services (ICWS)*. [S.l.], 2019. p. 283–290.
- MAKRIDAKIS, S.; SPILLOTIS, E.; ASSIMAKOPOULOS, V. The m5 competition: Background, organization, and implementation. *International Journal of Forecasting*, 2021. ISSN 0169-2070.
- MORETTIN, P.; TOLOI, C. M. C. *Análise de Séries Temporais*. [S.l.]: Editora Edgard Blücher, 2006.
- MORETTIN, P. A.; TOLOI, C. M. de C. *Modelos para previsão de séries temporais*. [S.l.]: Instituto de matemática pura e aplicada, 1981. v. 1.
- NAQVI, F. B.; SHAD, M. Y. Seeking a balance between population diversity and premature convergence for real-coded genetic algorithms with crossover operator. *Evolutionary Intelligence*, Springer, v. 15, p. 1864–5917, 2022.
- NAQVI, F. B.; SHAD, M. Y.; KHAN, S. A new logistic distribution based crossover operator for real-coded genetic algorithm. *Journal of Statistical Computation and Simulation*, Taylor & Francis, v. 91, n. 4, p. 817–835, 2021.
- OLIVEIRA, J. a. F. Lorenzato de; LUDERMIR, T. B. A hybrid evolutionary system for parameter optimization and lag selection in time series forecasting. In: *2014 Brazilian Conference on Intelligent Systems*. [S.l.: s.n.], 2014. p. 73–78.
- OLIVEIRA, J. F. de; LUDERMIR, T. B. A hybrid evolutionary decomposition system for time series forecasting. *Neurocomputing*, v. 180, p. 27–34, 2015.
- OLIVEIRA, J. F. L. de; SILVA, E. G.; NETO, P. S. G. de M. A hybrid system based on dynamic selection for time series forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–13, 2021.
- PAI, P.-F.; LIN, C.-S. A hybrid arima and support vector machines model in stock price forecasting. *Omega*, Elsevier, v. 33, n. 6, p. 497–505, 2005.
- PANIGRAHI, S.; BEHERA, H. A hybrid ets–ann model for time series forecasting. *Engineering Applications of Artificial Intelligence*, v. 66, 2017.
- PANKRATZ, A. *Forecasting with Univariate Box-Jenkins Models: Concepts and cases*. [S.l.]: John Wiley & Sons. Inc, 1983.
- PARTELOW, S. What is a framework? understanding their purpose, value, development and use. *Journal of Environmental Studies and Sciences*, v. 13, n. 3, p. 510–519, Sep 2023.
- PATHAN, M.; PATSIAS, S.; TAGARIELLI, V. A real-coded genetic algorithm for optimizing the damping response of composite laminates. *Computers & Structures*, v. 198, p. 51–60, 2018.
- PIOTROWSKI, A. P.; NAPIORKOWSKI, J. J. A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling. *Journal of Hydrology*, Elsevier, v. 476, p. 97–111, 2013.

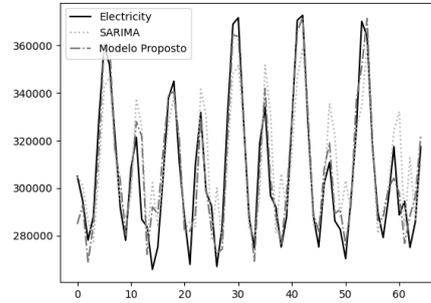
- PRECHELT, L. Early stopping-but when? In: *Neural Networks: Tricks of the trade*. [S.l.]: Springer, 1998. p. 55–69.
- PUCHTA, E. D. P.; SIQUEIRA, H. V.; KASTER, M. d. S. Optimization tools based on metaheuristics for performance enhancement in a gaussian adaptive pid controller. *IEEE Transactions on Cybernetics*, v. 50, n. 3, 2020.
- ROSEN, K. H. *Discrete Mathematics and Its Applications*. 5th. ed. [S.l.]: McGraw-Hill Higher Education, 2002.
- RUIZ-AGUILAR, J.; TURIAS, I.; JIMÉNEZ-COME, M. Hybrid approaches based on sarima and artificial neural networks for inspection time series forecasting. *Transportation Research Part E: Logistics and Transportation Review*, v. 67, p. 1–13, 2014.
- RUIZ-AGUILAR, J. J.; TURIAS, I. J.; JIMÉNEZ-COME, M. J.; CERBÁN, M. M. Hybrid approaches of support vector regression and sarima models to forecast the inspections volume. In: POLYCARPOU, M.; CARVALHO, A. C. P. L. F. de; PAN, J.-S.; WOŹNIAK, M.; QUINTIAN, H.; CORCHADO, E. (Ed.). *Hybrid Artificial Intelligence Systems*. [S.l.]: Springer International Publishing, 2014. p. 502–514.
- SABAR, N. R.; KIEU, L. M.; CHUNG, E.; TSUBOTA, T.; ALMEIDA, P. E. M. de. A memetic algorithm for real world multi-intersection traffic signal optimisation problems. *Engineering Applications of Artificial Intelligence*, v. 63, p. 45–63, 2017.
- SALLEH, M. N. M.; HUSSAIN, K.; CHENG, S.; SHI, Y.; MUHAMMAD, A.; ULLAH, G.; NASEEM, R. Exploration and exploitation measurement in swarm-based metaheuristic algorithms: An empirical analysis. *Recent Advances on Soft Computing and Data Mining*, p. 24–32, 2018.
- SHUMWAY, R. H.; STOFFER, D. S.; STOFFER, D. S. *Time series analysis and its applications*. [S.l.]: Springer, 2000. v. 3.
- SI, T.; MIRANDA, P.; BHATTACHARYA, D. Novel enhanced salp swarm algorithms using opposition-based learning schemes for global optimization problems. *Elsevier*, 2021.
- SILVA, E. G.; CAVALCANTI, G. D. C.; DE, J. F. L.; NETO, P. S. G. de M. On the evaluation of dynamic selection parameters for time series forecasting. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2020. p. 1–7.
- SIQUEIRA, H.; MACEDO, M.; TADANO, Y. de S.; ALVES, T. A.; STEVAN, S. L.; OLIVEIRA, D. S.; MARINHO, M. H. N.; NETO, P. S. de M.; OLIVEIRA, J. ao F. L. de; LUNA, I.; FILHO, M. de A. L.; SARUBBO, L. A.; CONVERTI, A. Selection of temporal lags for predicting riverflow series from hydroelectric plants using variable selection methods. *Energies*, v. 13, n. 16, 2020.
- SZPIRO, G. G. Forecasting chaotic time series with genetic algorithms. *Phys. Rev. E*, American Physical Society, v. 55, p. 2557–2568, 1997.
- TAKAHASHI, R. Verification of thermo-dynamical genetic algorithm to solve the function optimization problem through diversity measurement — diversity measurement and its application to selection strategies in genetic algorithms. In: *2016 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2016. p. 168–177.

- TASKAYA-TEMIZEL, T.; CASEY, M. C. A comparative study of autoregressive neural network hybrids. *Neural Networks*, v. 18, n. 5, p. 781–789, 2005. ISSN 0893-6080.
- VAPNIK, V.; GOLOWICH, S. E.; SMOLA, A. et al. Support vector method for function approximation, regression estimation, and signal processing. *Advances in neural information processing systems*, MORGAN KAUFMANN PUBLISHERS, p. 281–287, 1997.
- VARGAS, D. E. Um estudo dos parâmetros do algoritmo nsga-ii com o operador sbx em problemas de otimização estrutural multiobjetivo. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, v. 6, n. 2, 2018.
- WANG, J.; HUANG, W.; MA, G.; CHEN, S. An improved partheno genetic algorithm for multi-objective economic dispatch in cascaded hydropower systems. *International Journal of Electrical Power & Energy Systems*, Elsevier, v. 67, p. 591–597, 2015.
- WIDIASARI, I. R.; NUGROHO, L. E.; WIDYAWAN. Deep learning multilayer perceptron (mlp) for flood prediction model using wireless sensor network based hydrology time series data mining. *International Conference on Innovative and Creative Information Technology (ICITech)*, 2017.
- WU, Z.; RINCON, D.; GU, Q.; CHRISTOFIDES, P. D. Statistical machine learning in model predictive control of nonlinear processes. *Mathematics*, v. 9, n. 16, 2021.
- YAN, Q.; MA, C. Application of integrated arima and rbf network for groundwater level forecasting. *Environmental Earth Sciences*, Springer, v. 75, n. 5, p. 1–13, 2016.
- YANG, D.; DONG, Z.; LIM, L. H. I.; LIU, L. Analyzing big time series data in solar engineering using features and pca. *Solar Energy*, v. 153, p. 317–328, 2017.
- YI, W.; GAO, L.; LI, X.; ZHOU, Y. A new differential evolution algorithm with a hybrid mutation operator and self-adapting control parameters for global optimization problems. *Applied Intelligence*, Springer, v. 42, p. 642–660, 2015.
- YIN, H.; PRISHCHEPOV, A. V.; KUEMMERLE, T.; BLEYHL, B.; BUCHNER, J.; RADELOFF, V. C. Mapping agricultural land abandonment from spatial and temporal segmentation of landsat time series. *Energy Policy*, v. 210, p. 12–24, 2018.
- YOLCU, U.; EGRIOGLU, E.; ALADAG, C. H. A new linear & nonlinear artificial neural network model for time series forecasting. *Decision Support Systems*, v. 54, p. 1340–1347, 2013.
- ZAINUDDIN, F. A.; SAMAD, M. F. A.; TUNGGAL, D. A review of crossover methods and problem representation of genetic algorithm in recent engineering applications. *International Journal of Advanced Science and Technology*, v. 29, n. 6s, p. 759–769, 2020.
- ZHANG, G.; PATUWO, B. E.; HU, M. Y. Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, Elsevier, v. 14, n. 1, p. 35–62, 1998.
- ZHANG, G. P. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, v. 50, p. 159–175, 2003.

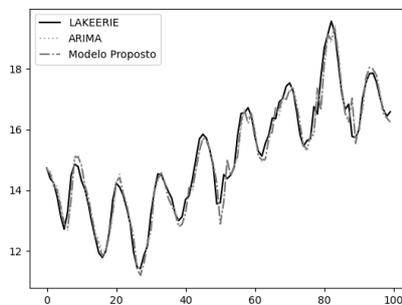
## ANEXO A – GRÁFICOS COMPARATIVOS



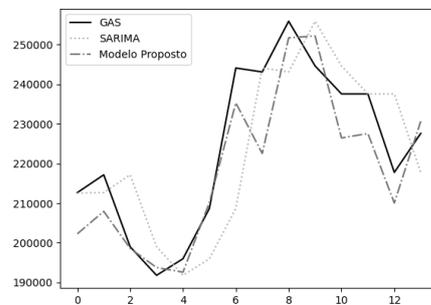
(a) Carsales



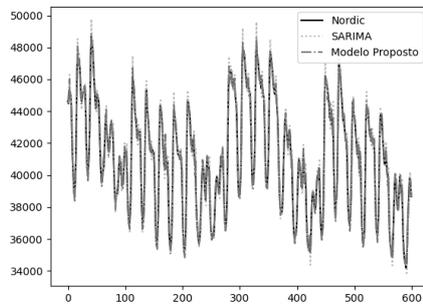
(b) Electricity



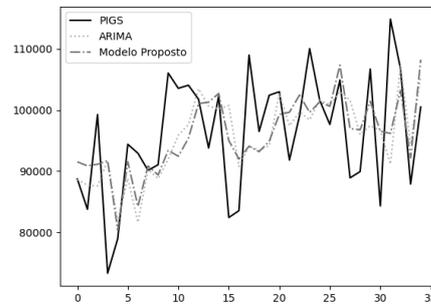
(c) Lake



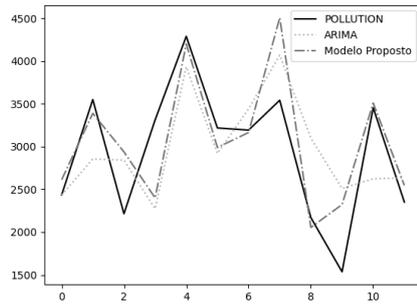
(d) Gas



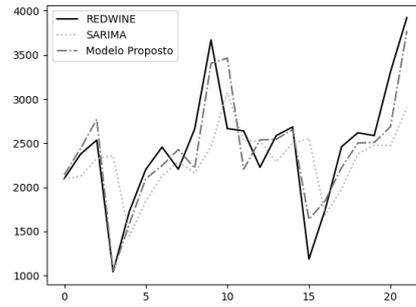
(e) Nordic



(f) Pigs



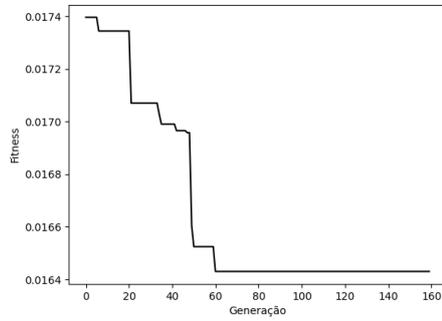
(g) Pollution



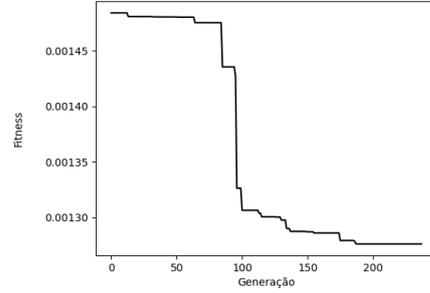
(h) Redwine

Figura 10 – Comparação da melhor versão do Modelo Proposto com o modelo base (ARIMA/SARIMA) e as Séries Temporais (a) Carsales, (b) Electricity, (c) Lake, (d) Gas, (e) Nordic, (f) Pigs, (g) Pollution e (h) Redwine.

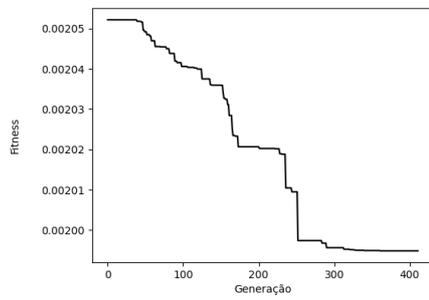
**Fonte:** Elaborada pela autora (2024).

**ANEXO B – GRÁFICOS DE CONVERGÊNCIA – ALGORITMO GENÉTICO**

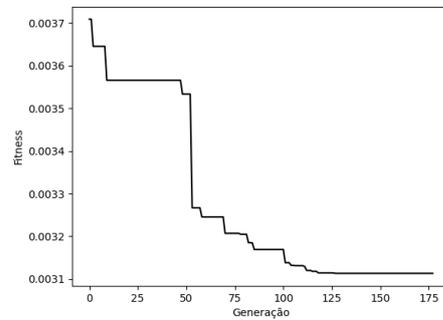
(a) Carsales



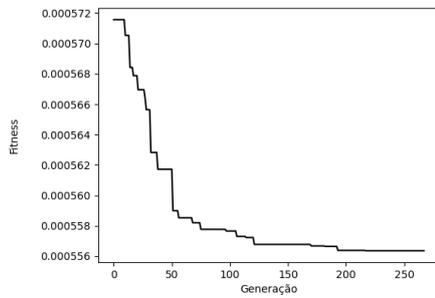
(b) Electricity



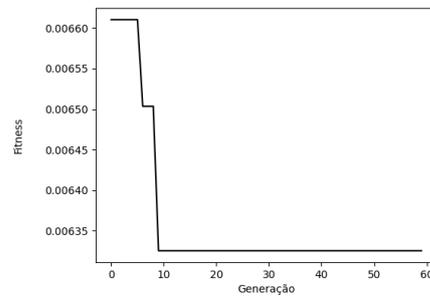
(c) Lake



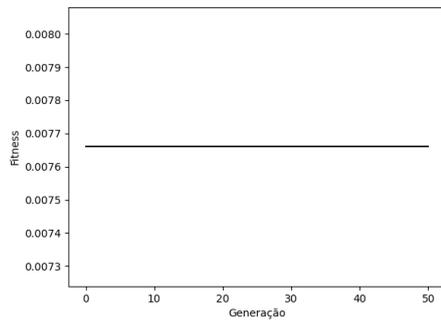
(d) Gas



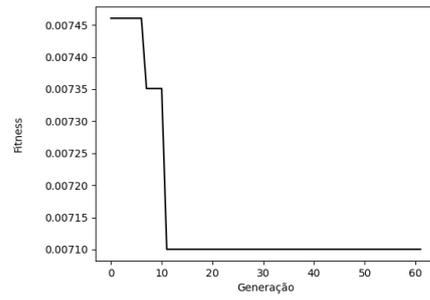
(e) Nordic



(f) Pigs



(g) Pollution



(h) Redwine

Figura 11 – Melhor Fitness durante a execução do Algoritmo Genético para as Séries Temporais (a) Carsales, (b) Electricity, (c) Lake, (d) Gas, (e) Nordic, (f) Pigs, (g) Pollution e (h) Redwine.

**Fonte:** Elaborada pela autora (2024).