

OTIMIZAÇÃO EVOLUTIVA DE HIPERPARÂMETROS PARA MODELOS DE SÉRIES TEMPORAIS NEBULOSAS

Patrícia de Oliveira e Lucas*
Petrônio Cândido de Lima e Silva**
Frederico Gadelha Guimarães***

* Instituto Federal do Norte de Minas Gerais - Campus Salinas, Brasil
(e-mail: patricia.lucas@ifnmg.edu.br).

** Instituto Federal do Norte de Minas Gerais - Campus Januária,
Brasil (e-mail: petronio.candido@ifnmg.edu.br)

*** Departamento de Engenharia Elétrica, Universidade Federal de
Minas Gerais, UFMG, Brasil (e-mail: fredericoguimaraes@ufmg.br)
Machine Intelligence and Data Science (MINDS) Laboratory,
Universidade Federal de Minas Gerais, UFMG, Belo Horizonte, Brasil

Abstract: Forecasting time series, an essential task in managing Smart Cities and Smart Grids, becomes a challenge when dealing with big data. The development of highly accurate machine learning models is made even more difficult by optimization of hyper-parameters, a costly computational task. To mitigate these challenges, the paper proposes the method of Time Series Nebulae of High Order Weighted, a versatile and customizable method through its hyperparameters. Altogether a hyperparameters optimization method is presented for Weighted High Order Nebulae Time Series that automates the generation of accurate and parsimonious models, using Genetic Algorithms adapted to multiobjective problems using the Dual Tour operator. The proposed methods were evaluated using a cluster with commodity hardware and a large volume of solar time series data, achieving good performance in viable processing time.

Resumo: A previsão de séries temporais, uma tarefa essencial no gerenciamento de *Smart Cities* e *Smart Grids*, torna-se um desafio quando precisa lidar com Big Data. O desenvolvimento de modelos de aprendizado de máquina altamente precisos é ainda mais difícil devido à otimização de hiperparâmetros, uma tarefa computacional dispendiosa. Para atenuar esses desafios, esse trabalho propõe o método de Séries Temporais Nebulosas de Alta Ordem Ponderada, um método versátil e customizável através de seus hiperparâmetros. Em conjunto é apresentado um método de otimização dos hiperparâmetros para Séries Temporais Nebulosas de Alta Ordem Ponderada que automatiza a geração de modelos acurados e parcimoniosos utilizando Algoritmos Genéticos. Os métodos propostos foram avaliados usando um cluster com hardware de commodity e um grande volume de dados de séries temporais de energia solar, alcançando bom desempenho em tempo de processamento viável.

Keywords: Fuzzy Time Series; Smart Grids; Big Data; Distributed Training; Hyperparameter Optimization.

Palavras-chaves: Séries Temporais Nebulosas; Smart Grids; Big Data; Treinamento Distribuído; Otimização de Hiperparâmetros.

1. INTRODUÇÃO

A demanda por modelos preditivos acurados e parcimoniosos, treinados a partir de grandes séries temporais, é onipresente em toda a ciência e engenharia. Mas otimização desses modelos - em especial aqueles baseados em aprendizado de máquina - é uma tarefa desafiadora tanto pelo conhecimento prévio da parametrização dos métodos quanto pelo seu alto custo computacional. Por outro lado, alguns métodos de previsão são caixas-pretas e encontram barreiras legais ou resistências devido à falta de transparência e auditabilidade.

O cenário exposto traz a oportunidade de explorar os métodos Séries Temporais Nebulosas (STN) Song and Chissom (1993), que têm atraído mais atenção e relevância nos últimos anos devido a muitos estudos relatando sua boa precisão em comparação com outros modelos Singh (2017). Os métodos de previsão do STN produzem modelos orientados a dados e não paramétricos e se tornaram atraentes devido à simplicidade, versatilidade, acurácia e desempenho computacional, além de produzir representações legíveis dos padrões de séries temporais, tornando seu conhecimento passível de ser auditado, reutilizável e atualizável. Exemplos de aplicações bem-sucedidas são mostrados em previsão de carga de energia Sadaei et al.

(2017); Severiano et al. (2017); Silva et al. (2017a, 2018); SAD (2019), previsão de preços do índice de estoque Sadaei et al. (2016); Silva et al. (2016); Talarposhti et al. (2016), séries temporais Silva et al. (2017b); Guney et al. (2018) entre outros.

O campo de pesquisa nas STN é amplo e ainda existem muitos aprimoramentos possíveis (e necessários) para esses métodos, incluindo uma alternativa que explicita todos os hiperparâmetros envolvidos no treinamento dos modelos. Para preencher essa lacuna o presente trabalho propõe o método de STN de Alta Ordem Ponderada (STNAOP), um método versátil e customizável através de seus hiperparâmetros. Em conjunto é apresentado um método de otimização dos hiperparâmetros para STNAOP que automatiza a geração de modelos acurados e parcimoniosos, utilizando Algoritmos Genéticos adaptados para problemas multiobjetivo usando o operador de Torneio Duplo.

O restante deste trabalho está organizado da seguinte forma: na seção 2, os principais conceitos de Séries Temporais Nebulosas são introduzidos, na seção 3, os procedimentos sequenciais do STNAOP são apresentados, na seção 4 o procedimento de otimização dos hiperparâmetros é discutido. Na seção 5 é avaliado o desempenho dos métodos. Finalmente, na seção 6, os principais achados da pesquisa são sintetizados.

2. PRELIMINARES

O método Séries Temporais Nebulosas (STN) foi introduzido em Song and Chissom (1993) para lidar com o conhecimento vago e impreciso em dados de séries temporais. Dada uma série temporal univariada $Y \in R^1$, e seus valores individuais $y(t) \in Y$, por $t = 0, 1, \dots, T$, o Universo do Discurso (UoD) U é delimitado pelos limites conhecidos de Y , tal que $U = [\min(Y), \max(Y)]$. Uma variável linguística $\tilde{A}$ é definida em U , onde os termos linguísticos são conjuntos fuzzy A_i , para $i = 1 \dots k$, cada um com sua própria função de pertinência (MF) μ_{A_i} . Essa etapa é chamada de particionamento de UoD e depende de três hiperparâmetros: o número de partições k , o método de particionamento Π e a função de pertinência μ . Comumente, as funções de pertinência são funções triangulares, trapezoidais, sigmóides ou gaussianas. O parâmetro k é um dos parâmetros mais importantes dos métodos do STF, pois impacta diretamente no viés/variância do modelo e o comprimento dos modelos.

O método Π que particiona U em k conjuntos fuzzy irão compor a variável linguística $\tilde{A}$ tal que $\Pi : Y, k, \mu \rightarrow \tilde{A}$. Existem muitos métodos de particionamento na literatura do STN, como o método de Grade mais simples Song and Chissom (1993) (todas as partições têm o mesmo tamanho), métodos heurísticos Huarng (2001); Chen and Phuong (2017); Xian et al. (2017), métodos meta-heurísticos Pal and Kar (2019); Cai et al. (2015), baseado em entropia Chen (2014) e *clusters* baseados em Dincer and Akkus (2018); Bose and Mali (2017); Zhang et al. (2018). Cada método tem seus próprios recursos e a melhor escolha depende da natureza dos dados e do contexto associado.

Existem várias categorias de métodos STN, variando principalmente por ordem e variância de tempo. A ordem é o número de *lags* (atrasos) usados na modelagem da

série temporal. Dados os dados da série temporal F , os modelos de Primeira Ordem precisam de $F(t-1)$ para prever $F(t)$, enquanto os modelos de Alta Ordem requerem $F(t-1), \dots, F(t-p)$ para prever $F(t)$.

Depois de Song and Chissom (1993), a STN Convencional (CSTN) foi proposto em Chen (1996) para simplificar os procedimentos de treinamento e previsão, gerando também regras mais legíveis. As STN ponderadas (STNP) adiciona pesos nas regras como em Yu (2005); Cheng et al. (2008); Ismail and Efendi (2011); Sadaei et al. (2014). Já os modelos de STN de Alta Ordem (STNAO) são capazes de reconhecer padrões passados mais complexos na série temporal, incluindo sazonalidades e tendências, e são discutidos em Severiano et al. (2017) e Zhang et al. (2018). Modelos ponderados de alta ordem são abordados em Jiang et al. (2017). A determinação dos melhores índices de defasagem para modelos de alta ordem é discutida em Guney et al. (2018), que usa ACF/PACF, veja Egrioglu (2014) e Carvalho Jr and Costa Jr (2017). Em Carvalho Jr and Costa Jr (2017) o parâmetro α -cut é discutido como um parâmetro de redução de ruído e regularização.

Alguns dos métodos citados acima incorporam a otimização do particionamento ou a indução de regras, mas eles não cobrem a otimização de todos os parâmetros. Este trabalho não propõe um novo esquema de particionamento, preferindo avaliar os existentes como um dos hiperparâmetros do modelo.

3. O MÉTODO DE SÉRIES TEMPORAIS NEBULOSAS DE ALTA ORDEM PONDERADA

O método de Séries Temporais nebulosas de Alta Ordem Ponderada (STNAOP) incorpora vários recursos presentes na literatura, representados pelos hiperparâmetros resumidos na Tabela 1. O método é composto de dois procedimentos: o procedimento de treinamento e o procedimento de previsão.

O procedimento de treinamento, explicado na subseção 3.1 e ilustrado na Figura 1, é um processo de três etapas responsável por criar um modelo ponderado de STN de alta ordem $\mathcal{M}$. O modelo final STNAOP $\mathcal{M}$ consiste em uma variável linguística fuzzy $\tilde{A}$ e um conjunto de regras difusas ponderadas sobre $\tilde{A}$. As entradas do procedimento de treinamento são os dados de treinamento de séries temporais nítidas Y e o conjunto de hiperparâmetros.

O procedimento de previsão, explicado na subseção 3.2, visa produzir uma estimativa pontual $\hat{y}(t+1)$ dada uma amostra de entrada, os últimos *lags* de dados Ω , usando as variáveis e as regras fuzzy induzidas no modelo $\mathcal{M}$.

3.1 Processo de treinamento

Estágio 1 - *Particionamento*:

- Definindo U*: O UoD define o espaço amostral, isto é, os limites conhecidos da série temporal Y , tal que $U = [\min(Y) - D_1, \max(Y) + D_2]$, em que $D_1 = \min(Y) \times 0.2$ e $D_2 = \max(Y) \times 0.2$ são usados para extrapolar os limites conhecidos como uma margem de segurança.
- Particionamento UoD*: Dividir U em k intervalos U_i com pontos médios c_i , por invocação do

Tabela 1. Hiperparâmetros STNAOP

Variável	Parâmetro	Tipo	Descrição
k	Número de partições	N^+	O número de conjuntos nebulosos que serão criados para descrever a série temporal
μ	Função de pertinência	$\mu : U \rightarrow [0, 1]$	Uma função que mede a pertinência de um valor $y \in U$ a um conjunto nebuloso
Π	Método de particionamento	$\Pi : Y, k, \mu \rightarrow \tilde{A}$	Um procedimento que recebe os dados da série temporal Y e o parâmetro k e retorna uma variável linguística $\tilde{A}$ com k conjuntos fuzzy $A_i \in \tilde{A}$, cada um com sua função de pertinência μ_{A_i}
α	α -cut	$[0, 1]$	A nota mínima de adesão no processo de fuzzyficação
Ω	Ordem	N^+	O número de atrasos usados no precedente de cada regra difusa
L	Lags		Um vetor dos índices dos lags, com comprimento Ω and $1 \leq L[i] < L[i+1]$ for $t = 0.. \Omega$

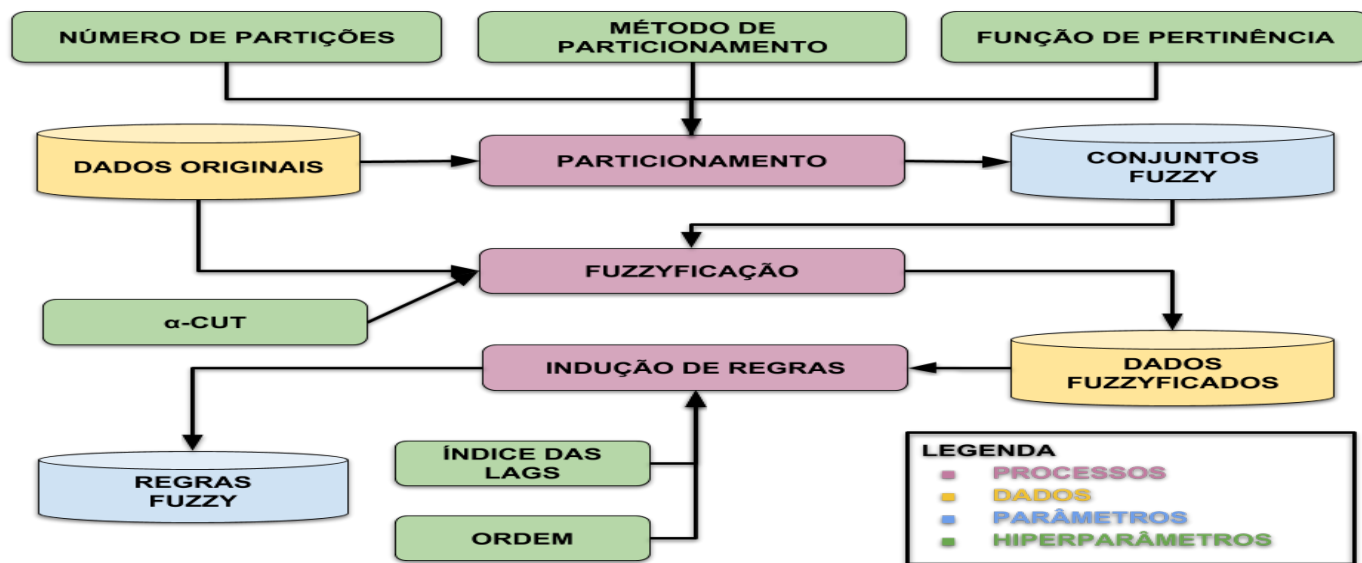


Figura 1. Processo de treinamento

método de particionamento Π , que definirá os comprimentos de todos os intervalos;

- (c) *Defina a variável linguística $\tilde{A}$* : Para cada intervalo U_i crie um conjunto fuzzy sobreposta A_i , com a função de pertinência μ_{A_i} . O ponto médio do conjunto fuzzy A_i será c_i , o limite inferior $l_i = c_{i-1}$ e o limite superior $u_i = c_{i+1} \forall i > 0$ e $i < k$ e $l_0 = \min U$, $l_k = \max U$. Cada conjunto fuzzy $A_i \in \tilde{A}$ é um termo linguístico da variável linguística $\tilde{A}$;

Estágio 2 - *Fuzzyficação*:

Transforme a série numérica original Y em uma série temporal fuzzy F , onde cada ponto de dados $f(t) \in F$ é uma matriz $1 \times k$ com os valores fuzzyfificados de $y(t) \in Y$ com relação aos termos linguísticos $A_i \in \tilde{A}$, onde a participação difusa é maior que a α -cut predefinida, isto é, $f(t) = \{A_i \mid \mu_{A_i}(y(t)) \geq \alpha \forall A_i \in \tilde{A}\}$;

Estágio 3 - *Indução de regra*:

- (a) *Gerar os padrões temporais de alta ordem*: Os padrões temporais difusos possuem o formato $A_{i0}, \dots, A_{i\Omega} \rightarrow A_j$, onde o precedente (ou LHS) é $f(t - L(\Omega)) = A_{i0}$, $f(t - L(\Omega - 1)) = A_{i1}$, ..., $f(t - L(0)) = A_{i\Omega}$, e o consequente (ou RHS) é $f(t + 1) = A_j$.
- (b) *Gerar a base de regra*: Selecione todos os padrões temporais com o mesmo precedente e agrupe seus

conjuntos subsequentes criando uma regra com o formato $A_{i0}, \dots, A_{i\Omega} \rightarrow w_k \cdot A_k, w_j \cdot A_j, \dots$, em que o LHS é $f(t - L(\Omega)) = A_{i0}$, $f(t - L(\Omega - 1)) = A_{i1}$, ..., $f(t - L(0)) = A_{i\Omega}$ e o RHS é $f(t + 1)$ em $A_k, A_j, \dots$ e os pesos $w_j, w_k, \dots$ são as frequências normalizadas de cada padrão temporal, de tal forma que:

$$w_i = \frac{\#A_i}{\#RHS} \forall A_i \in RHS \quad (1)$$

onde $\#A_i$ é o número de ocorrências de A_i em padrões temporais com o mesmo precedente LHS e $\#RHS$ é o número total de padrões temporais com o mesmo precedente LHS. Cada regra pode ser entendida como o conjunto ponderado de possibilidades que pode acontecer no tempo $t + 1$ (o consequente) quando um certo precedente $A_{i0}, \dots, A_{i\Omega}$ é identificado na anterior L lags (o precedente).

3.2 Processo de previsão

Passo 1 - *Fuzzyficação*:

Compute o grau de pertinência μ_{ti} para cada $y(t) \in Y$ onde $t \in L$ em cada conjunto fuzzy $A_i \in \tilde{A}$, de modo que $\mu_{ti} = \mu_{A_i}(y(t))$.

Passo 2 - *Correspondências de regras*:

Selecione K regras onde todos os conjuntos fuzzy A_i no LHS tem $\mu_{ti} > \alpha$; O grau de pertinência a regra fuzzy é mostrado abaixo, usando a função mínima como T-norma.

$$\mu_j = \bigcap_{t \in L, i \in \tilde{A}} \mu_{ti} \quad (2)$$

Passo 3 - *Pontos médios da regra*:

Para cada regra selecionada j , calcule o ponto médio mp_j como abaixo, onde c_{A_i} é o parâmetro c da função μ do conjunto fuzzy A_i :

$$mp_j = \sum_{i \in RHS} w_i \cdot c_i \quad (3)$$

Passo 4 - *Defuzzificação*:

Calcule a previsão como a soma ponderada do ponto médio da regra mp_j por suas classificações de associação μ_j para cada regra selecionada j :

$$\hat{y}(t+1) = \sum_{j \in K} \mu_j \cdot mp_j \quad (4)$$

4. OTIMIZAÇÃO DE HIPERPARÂMETROS

A otimização de hiperparâmetros é uma tarefa computacional onerosa e, em um contexto de Big Data, seu custo pode ser proibitivo. O procedimento geral de treinamento do STNAOP não incorpora qualquer tipo de otimização, deixando ao usuário a tarefa de buscar empiricamente os melhores valores para os hiperparâmetros. Por outro lado esses parâmetros têm grande impacto no desempenho do modelo final e sua otimização é altamente recomendada.

Dois objetivos conflitantes são testados durante a otimização do hiperparâmetros, a precisão e a parcimônia (a complexidade estrutural geralmente medida pelo número de parâmetros do modelo). Para aumentar a precisão de um modelo STN é necessário aumentar o número de conjuntos fuzzy e/ou a ordem do modelo, o que aumenta automaticamente o número de regras fuzzy. À medida que o número de regras aumenta, a complexidade computacional do modelo também aumenta e a abordagem de STN pode se tornar menos interessante do que outras alternativas tradicionais. Este problema de otimização visa então encontrar um equilíbrio entre esses dois objetivos, mantendo o modelo pequeno, rápido e preciso.

Antes de entrar na tarefa de otimização, é necessário discutir uma metodologia para avaliar a precisão dos previsores de séries temporais. A exatidão é comumente medida em uma metodologia de janela deslizante (*sliding window*), onde um conjunto de dados Y de tamanho T é fatiado em sequências sobrepostas de W - chamadas janelas de dados - com tamanho fixo e cada uma deslocada da anterior por um valor constante D . Cada janela de dados tem suas próprias divisões de treino e teste (geralmente uma divisão 70%/30%) e, em seguida, o método é treinado e testado T/D vezes com dados diferentes. Os melhores valores para o tamanho da janela de dados W e o deslocamento da janela D dependem do conjunto de dados. É aconselhável que, para um conjunto de dados Y com tamanho T , a janela de dados e o deslocamento de dados forneçam pelo menos 10 experiências, bem distribuídas sobre o conjunto de dados total, de modo que $W = T/2$ e $D = .2 \cdot W$.

O problema de otimização de hiperparâmetros do STNAOP é formulado abaixo, onde a função objetivo f_1 (5) controla a precisão (representada pela soma ponderada do erro médio $\bar{\epsilon}$ (8) e o desvio padrão do erro σ_ϵ (9)) do modelo e a função objetivo f_2 (6) controla a parcimônia (representado pela soma ponderada do comprimento do modelo $|\mathcal{M}|$ (10) e a soma dos *lags* $|L|$ (11)). Algumas restrições adicionais são impostas à ordem Ω (12), número de partições k (13), α -cut (14) e os índices de *lags* L (15).

Otimize:

$$\text{minimize } f_1 = 0.6\bar{\epsilon} + 0.4\sigma_\epsilon \quad (5)$$

$$\text{minimize } f_2 = 0.6|\mathcal{M}| + 0.4|L| \quad (6)$$

Onde:

$$RMSE = \sqrt{\sum_{t=0}^n (y(t) - \hat{y}(t))^2} \quad (7)$$

$$\bar{\epsilon} = W^{-1} \sum_{i=0}^W RMSE(i) \quad (8)$$

$$\sigma_\epsilon = W^{-1} \sum_{i=0}^W \bar{\epsilon} - RMSE(i) \quad (9)$$

$$|\mathcal{M}| = \sum_{i=0}^{rules} 1 \quad (10)$$

$$|L| = \sum_{i=0}^{\Omega-1} L(i) \quad (11)$$

Sujeito a:

$$\Omega \geq 1 \quad (12)$$

$$k \geq 3 \quad (13)$$

$$\alpha \in [0, 1) \quad (14)$$

$$1 \leq L(0) < \dots < L(\Omega) \quad (15)$$

A abordagem do Algoritmo Genético (AG) foi escolhida para resolver o problema de otimização. Os algoritmos genéticos, em sua forma original, são destinados à otimização mono-objetiva, mas com poucas adaptações também é possível lidar com problemas multiobjetivos mais simples. Neste trabalho foram feitas as adaptações no operador de seleção, que implementa a estratégia do Duplo Torneio, de modo a compor os dois objetivos no equilíbrio da população.

Cada indivíduo da população é representado por um vetor - o genótipo - com todos os hiperparâmetros da Tabela 1. Esse vetor contém valores reais, categóricos e de matriz, de acordo com o tipo de hiperparâmetro. A população inicial é gerada aleatoriamente, exceto pelo hiperparâmetro L , cujo tamanho é limitado por Ω e os índices de defasagem inicialmente consideram os defasagens mais significativas de ACF/PACF. O AG itera as etapas listadas abaixo até que os critérios de parada sejam atingidos:

- (1) **Avaliação:** Cada genótipo é transformado em um modelo treinado - o fenótipo - e então avaliado. A avaliação do fenótipo usa as métricas de acordo com a função objetivo f_1 e f_2 e após o procedimento

de avaliação, a população é classificada em ordem crescente.

- (2) **Seleção:** O operador de seleção é responsável por escolher uma parte dos indivíduos que sobreviverão para a próxima geração. Como o problema é multi-objetivo, uma estratégia de Torneio Duplo foi implementada para equilibrar a seleção entre os dois objetivos. No torneio duplo, a primeira rodada escolhe aleatoriamente dois pares de indivíduos na população, e cada dupla competirá entre si com base no objetivo f_1 . No segundo turno, os vencedores do primeiro turno competirão entre si com base no objetivo f_2 . Este processo é repetido por uma taxa SR da população.
- (3) **Elitismo:** Como o operador de seleção é aleatório, existe a possibilidade de que o melhor indivíduo da população seja descartado. A estratégia elitista manterá o melhor indivíduo da geração atual na próxima geração e descartará o pior.
- (4) **Cruzamento:** O operador de cruzamento combina os genótipos de dois indivíduos (i_1 e i_2) para gerar um indivíduo descendente (i_N). No cruzamento, dois indivíduos são selecionados aleatoriamente na população e ordenados como i_1 e i_2 de acordo com seus objetivos f_1 e f_2 . Para todos os genes, o processo de mixagem dará uma grande contribuição para o indivíduo mais bem classificado (com taxas de .7 e .3). Para os genes codificados reais, uma combinação linear como $i_N = .7i_1 + .3i_2$ será executada. Para os genes categóricos, o valor de i_N será i_1 com probabilidade .7 ou i_2 , caso contrário. Para o lag L , os $lags$ individuais também serão uma combinação linear de cada lag . Este processo é repetido por uma taxa CR da população.
- (5) **Mutação:** O operador de mutação visa introduzir diversidade na população, então um indivíduo é escolhido aleatoriamente e perturbações aleatórias são aplicadas em seus genes, tomando cuidado para manter os valores genéticos factíveis de acordo com as restrições do problema. Este processo é repetido por uma taxa MR da população.
- (6) **Critério de parada:** As gerações irão parar quando um destes critérios for alcançado: um número fixo de gerações sem melhoria ou número máximo de gerações.

Esse método de otimização de hiperparâmetros requer a escolha de alguns parâmetros, que são apresentados em sequência com seus valores recomendados: tamanho da população (valor padrão 30), número máximo de gerações (valor padrão 50), número máximo de gerações sem melhoria (valor 10), taxa de seleção $SR \in [0, 1]$ (valor padrão .5), taxa de cruzamento $CR \in [0, 1]$ (valor padrão .5), taxa de mutação $MR \in [0, 1]$ (valor padrão .05).

5. EXPERIMENTOS COMPUTACIONAIS

O objetivo desta seção é avaliar empiricamente o desempenho da otimização dos hiperparâmetros evolutiva na criação de um modelo de STNAOP parcimonioso e acurado, usando uma grande série temporal ambiental focada em energia solar renovável.

O conjunto de dados utilizado nesta pesquisa veio do SONDA - Sistema Nacional de Organização de Dados

Tabela 2. Parâmetros do algoritmo genético

Parâmetro	Valor
Tamanho da população	30
Número máximo de gerações	30
Número máximo de gerações sem melhoria	10
Taxa de seleção	.5
Taxa de cruzamento	.5
Taxa de mutação	.1

Tabela 3. Espaços de busca dos hiperparâmetros

Parâmetro	Espaço de Busca
k	[10, 100]
Π	Grid, Entropy, FCM
μ	triangular, trapezoidal, gaussiana
Ω	[1, 3]
α	[0, .5]
L	[1, 50]

Ambientais do Brasil ¹, um projeto governamental que agrupa dados ambientais (radiação solar, velocidade do vento, precipitação, etc.) do INPE - Instituto Nacional de Pesquisas Espaciais. A série temporal escolhida foi a radiação horizontal solar global da estação de telemetria de Brasília ², registrado entre 2012 e 2015, por minuto, somando 2 milhões de instâncias, onde 70 % foram utilizados em treinamento e validação dentro o algoritmo de otimização de hiperparâmetros e o restante apenas para testar os modelos gerados. Essa série temporal possui multi-sazonalidades e várias fontes de ruído, características que o STNAOP deve lidar.

Os métodos propostos foram implementados usando a linguagem Python usando a infra-estrutura da biblioteca pyFITS Silva (2018). Para avaliar a estabilidade do método, a otimização do hiper-parâmetro foi executada 30 vezes, com os parâmetros do algoritmo Genético apresentados na Tabela 2. Algumas restrições de viabilidade foram impostas aos espaços de busca de hiper-parâmetros, como apresentado na Tabela 3, para melhorar o processo de otimização. Os resultados do AG foi então comparados com uma *Grid Search* (Busca em Grade), um método guloso para encontrar a combinação ótima dos hiperparâmetros que testa todas as combinações possíveis. A *Grid Search* é um método de otimização exato, embora seja computacionalmente muito caro.

Uma amostra da convergência dos objetivos do AG é mostrada na Figura 2. Em média, o método converge rapidamente e termina por gerações máximas sem critérios de melhoria.

Os hiperparâmetros gerados pela média são apresentados na Tabela 4, que também apresenta o erro médio do RMSE e o comprimento do modelo (número de regras). Uma amostra do desempenho dos modelos gerados é mostrada na Figura 3. O design do algoritmo privilegiou o objetivo (precisão) de f_1 sobre o f_2 (parcimônia), e é observável pelo desvio padrão dessas métricas. Embora diferentes modelos com número de parâmetros muito diferentes obtivessem precisão aproximada, como regra geral, quanto maior o número de parâmetros melhor o desempenho.

¹ <http://sonda.ccst.inpe.br/>

² Código: BRB. Coordenadas: 15 36 '03 "S 47 42'47"O. Alt.: 1023m

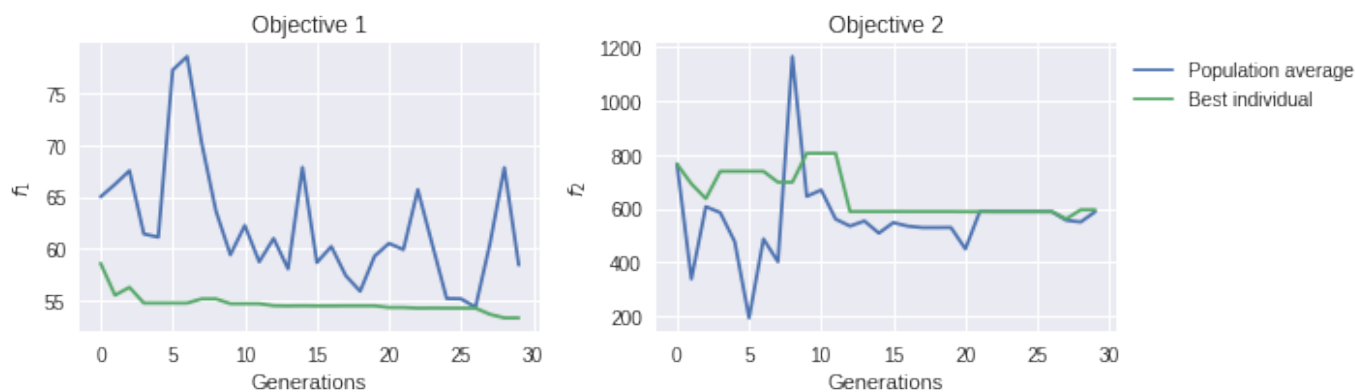


Figura 2. Convergência dos objetivos durante a otimização de hiperparâmetros

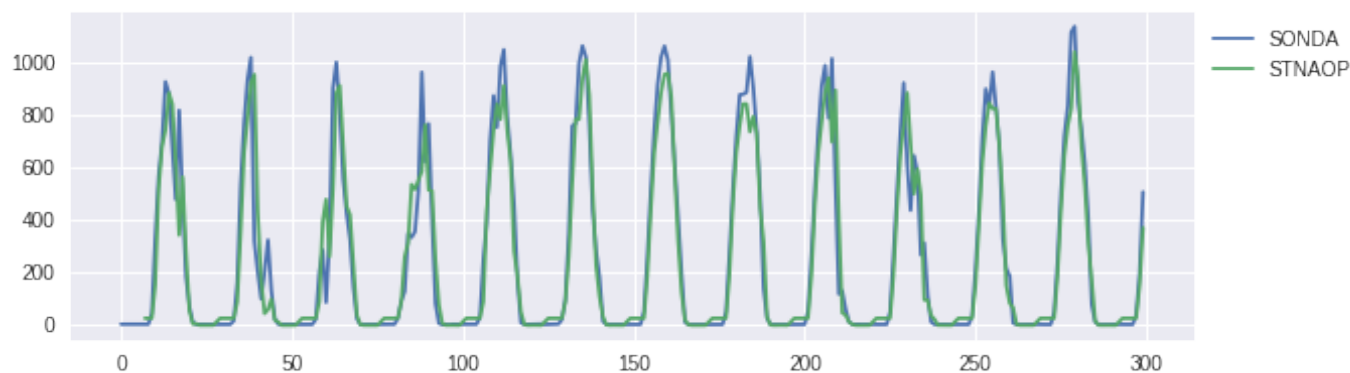


Figura 3. Exemplo de desempenho do melhor modelo gerado

Tabela 4. Resultados dos experimentos

Parâmetro	Grid Search	Alg. Genético
k	33	$21 \pm 7,88$
Π	Grid	Grid
μ	Gaussiana	Triangular
Ω	2	$2,1 \pm 0,8$
α	0,15	$0,3 \pm 0,1$
L	[1, 7]	[1, 6]
RMSE	76,97	$76,55 \pm 1,69$
$ \mathcal{M} $	649	$659,3 \pm 679,78$
Tempo (s)	42.230	$1.533,88 \pm 513,51$

6. CONCLUSÕES

As Séries Temporais Nebulosas são métodos de previsão orientados a dados/não paramétricos com legibilidade do modelo que se tornaram muito conhecidos nos últimos anos devido à sua simplicidade e precisão. Mas mesmo esses métodos apresentam desafios quando se busca otimizar a acurácia e a parcimônia no treinamento de seus modelos.

A otimização de modelos de aprendizado de máquina para grandes séries temporais é uma tarefa desafiadora a ser executada com procedimentos sequenciais ou mesmo paralelos executados em uma única máquina. Daí surge a necessidade da utilização de heurísticas que, embora relaxem o problema de otimização, reduzam significativamente o tempo necessário para encontrar bons parâmetros e hiperparâmetros para o modelo.

Este trabalho propõe uma família de métodos STN de Alta Ordem Ponderada que engloba procedimentos sequenciais

e distribuídos e um procedimento de otimização de hiperparâmetros baseado em Algoritmos Genéticos que explora a escalabilidade para melhorar a precisão e parcimônia do modelo.

Experimentos computacionais mostraram a efetividade do método, que realizou a otimização de hiperparâmetros usando uma série temporal de 2 milhões de registros, apenas possível em tempo factível com computação distribuída, gerando modelos de previsão precisos e parcimoniosos com baixo erro RMSE.

Pesquisas futuras estenderão o método proposto para séries temporais multivariadas e melhorarão o método de otimização, além de realizar comparações com outras abordagens na literatura.

AGRADECIMENTOS

O presente trabalho foi realizado com o apoio financeiro do IFNMG e das agências CAPES, FAPEMIG, CNPq.

REFERÊNCIAS

- (2019). Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy*, 175, 365 – 377. doi:<https://doi.org/10.1016/j.energy.2019.03.081>.
- Bose, M. and Mali, K. (2017). A novel data partitioning and rule selection technique for modeling high-order fuzzy time series. *Applied Soft Computing*, 63(17), 87–96. doi:10.1016/j.asoc.2017.11.011. URL <http://doi.org/10.1016/j.asoc.2017.11.011>.

- Cai, Q., Zhang, D., Zheng, W., and Leung, S.C.H. (2015). A new fuzzy time series forecasting model combined with ant colony optimization and auto-regression. doi:10.1016/j.knosys.2014.11.003. URL <http://doi.org/10.1016/j.knosys.2014.11.003>.
- Carvalho Jr, J. and Costa Jr, C. (2017). Identification method for fuzzy forecasting models of time series. *Applied Soft Computing*, 50, 166–182. doi:10.1016/j.asoc.2016.11.003.
- Chen, M.Y. (2014). A high-order fuzzy time series forecasting model for internet stock trading. *Future Generation Computer Systems*, 37, 461–467. doi:10.1016/j.future.2013.09.025. URL <http://dx.doi.org/10.1016/j.future.2013.09.025>.
- Chen, S.M. (1996). Forecasting enrollments based on fuzzy time series. *Fuzzy Sets and Systems*, 81(3), 311–319. doi:10.1016/0165-0114(95)00220-0. URL [https://doi.org/10.1016/0165-0114\(95\)00220-0](https://doi.org/10.1016/0165-0114(95)00220-0).
- Chen, S.M. and Phuong, B.D.H. (2017). Fuzzy time series forecasting based on optimal partitions of intervals and optimal weighting vectors. *Knowledge-Based Systems*. doi:10.1016/j.knosys.2016.11.019.
- Cheng, C.H., Chen, T.L., Teoh, H.J., and Chiang, C.H. (2008). Fuzzy time-series based on adaptive expectation model for TAIEX forecasting. *Expert Systems with Applications*, 34, 1126–1132. doi:10.1016/j.eswa.2006.12.021.
- Dincer, N.G. and Akkuş, (2018). A new fuzzy time series model based on robust clustering for forecasting of air pollution. *Ecological Informatics*, 43, 157–164. doi:10.1016/J.ECOINF.2017.12.001.
- Egrioglu, E. (2014). PSO-based high order time invariant fuzzy time series method: Application to stock exchange data. *Economic Modelling*. doi:10.1016/j.econmod.2014.02.017.
- Guney, H., Bakir, M.A., and Aladag, C.H. (2018). A Novel Stochastic Seasonal Fuzzy Time Series Forecasting Model. *International Journal of Fuzzy Systems*. doi:10.1007/s40815-017-0385-z.
- Huang, K. (2001). Effective lengths of intervals to improve forecasting in fuzzy time series. *Fuzzy Sets and Systems*, 123(3), 387–394. doi:10.1016/S0165-0114(00)00057-9.
- Ismail, Z. and Efendi, R. (2011). Enrollment forecasting based on modified weight fuzzy time series. *Journal of Artificial Intelligence*, 4(1), 110–118. doi:10.3923/jai.2011.110.118.
- Jiang, P., Dong, Q., Li, P., and Lian, L. (2017). A novel high-order weighted fuzzy time series model and its application in nonlinear time series prediction. *Applied Soft Computing*. doi:10.1016/j.asoc.2017.01.043.
- Pal, S.S. and Kar, S. (2019). Fuzzy Time Series Model for Unequal Interval Length Using Genetic Algorithm. *Information Technology and Applied Mathematics. Advances in Intelligent Systems and Computing*, 699. doi:10.1007/978-981-10-7590-2{_}15.
- Sadaei, H.J., Enayatifar, R., Abdullah, A.H., and Gani, A. (2014). Short-term load forecasting using a hybrid model with a refined exponentially weighted fuzzy time series and an improved harmony search. *International Journal of Electrical Power & Energy Systems*, 62(from 2005), 118–129. doi:10.1016/j.ijepes.2014.04.026.
- Sadaei, H.J., Enayatifar, R., Lee, M.H., and Mahmud, M. (2016). A hybrid model based on differential fuzzy logic relationships and imperialist competitive algorithm for stock market forecasting. *Applied Soft Computing Journal*, 40, 132–149. doi:10.1016/j.asoc.2015.11.026.
- Sadaei, H.J., Guimarães, F.G., Silva, C.J.d., Lee, M.H., and Eslami, T. (2017). Short-term load forecasting method based on fuzzy time series, seasonality and long memory process. *International Journal of Approximate Reasoning*, 83, 196–217. doi:10.1016/j.ijar.2017.01.006.
- Severiano, C.A.C., Silva, P.P.C.L., Sadaei, H.J.H., and Guimarães, F.F.G. (2017). Very Short-term Solar Forecasting using Fuzzy Time Series. In *2017 IEEE International Conference on Fuzzy Systems*. Naples, Italy. doi:10.1109/FUZZ-IEEE.2017.8015732.
- Silva, G.C., Alexandre, D., Vieira, G., Silva, J.L.R., Lisboa, A.C., Vieira, D.A.G., and Saldanha, R.R. (2018). Fuzzy time series applications and extensions: analysis of a short term load forecasting challenge. In *International conference on Time Series and Forecasting (ITISE 2018)*, volume 770. Granada, Spain.
- Silva, G.C., Silva, J.L.R., Lisboa, A.C., Vieira, D.A.G., and Saldanha, R.R. (2017a). Advanced Fuzzy Time Series Applied to Short Term Load Forecasting. In *4th IEEE Latin American Conference on Computational Intelligence*. IEE, Guadalajara.
- Silva, P., Sadaei, H.J., and Guimarães, F. (2016). Interval Forecasting with Fuzzy Time Series. In *IEEE Symposium Series on Computational Intelligence (IEEE SSCI 2016)*. Athens, Greece. doi:10.1109/SSCI.2016.7850010.
- Silva, P.C.d.L.e. (2018). pyFTS: Fuzzy Time Series for Python. doi:10.5281/zenodo.597359. URL <https://doi.org/10.5281/zenodo.597359>.
- Silva, P.C.d.L.e., Severiano Jr, C.A., Vieira, G.L., Guimarães, F.G., and Sadaei, H.J. (2017b). Probabilistic Forecasting with Seasonal Ensemble Fuzzy Time-Series. In *XIII Brazilian Congress on Computational Intelligence*. Rio de Janeiro.
- Singh, P. (2017). A brief review of modeling approaches based on fuzzy time series. *International Journal of Machine Learning and Cybernetics*, 8(2), 397–420. doi:10.1007/s13042-015-0332-y.
- Song, Q. and Chissom, B.S. (1993). Fuzzy time series and its models. *Fuzzy Sets and Systems*, 54(3), 269–277. doi:10.1016/0165-0114(93)90372-O.
- Talarposhti, F.M., Sadaei, H.J., Enayatifar, R., Guimarães, F.G., Mahmud, M., and Eslami, T. (2016). Stock market forecasting by using a hybrid model of exponential fuzzy time series. *International Journal of Approximate Reasoning*, 70, 79–98. doi:10.1016/j.ijar.2015.12.011.
- Xian, S., Zhang, J., Xiao, Y., and Pang, J. (2017). A novel fuzzy time series forecasting method based on the improved artificial fish swarm optimization algorithm. *Soft Computing*. doi:10.1007/s00500-017-2601-z.
- Yu, H.K. (2005). Weighted fuzzy time series models for TAIEX forecasting. *Physica A: Statistical Mechanics and its Applications*, 349(3), 609–624. doi:10.1016/j.physa.2004.11.006.
- Zhang, S., Zhang, S., Dyn, N., Zhang, W., Zhang, W., Zhang, S., and Zhang, S. (2018). Two-factor high-order fuzzy-trend FTS model based on BSO-FCM and improved KA for TAIEX stock forecastin. *Nonlinear Dynamics*. doi:10.1007/s11071-018-4433-5.