

MODELAGEM NEURO-FUZZY EVOLUTIVA NO ESPAÇO DE ESTADOS BASEADA EM COMPONENTES NÃO-OBSERVÁVEIS PARA PREVISÃO DE SÉRIES TEMPORAIS SAZONAIS

SELMO EDUARDO RODRIGUES JÚNIOR*, GINALBER LUIZ DE OLIVEIRA SERRA†

*Av. dos Portugueses, s/n, Bacanga, CEP: 65001-970.
Universidade Federal do Maranhão
São Luís, Maranhão, Brasil

†Av. Getúlio Vargas, 04, Monte Castelo, CEP: 65030-005
Instituto Federal de Educação, Ciência e Tecnologia
São Luís, Maranhão, Brasil

selmoeduardo@gmail.com, ginalber@ifma.edu.br

Abstract— In this paper, an evolving neuro-fuzzy Takagi-Sugeno Network (NF-TSN) for seasonal time series forecasting is proposed. This NF-TSN uses data from unobservable components extracted from the series to perform the forecast and modify its fuzzy rule base structure, adapting these data. The method that NF-TSN uses to identify these unobservable components is the Singular Spectrum Analysis. The experiments consider four time series to evaluate the NF-TSN performance and to compare with other methods in the literature. The proposed approach showed competitive results and it is a promising alternative for seasonal series forecasting and characterization of their unobservable components.

Keywords— Time Series Forecasting, Evolving, Neuro-Fuzzy Takagi-Sugeno, Singular Spectrum Analysis.

Resumo— Neste artigo, propõe-se uma Rede *Neuro-Fuzzy Takagi-Sugeno* (RNF-TS) evolutiva para a previsão de séries temporais sazonais. Essa RNF-TS usa dados das componentes não-observáveis extraídas da série para realizar a previsão e modificar sua estrutura de base de regras *fuzzy*, adaptando-se a esses dados. O método que a RNF-TS utiliza para identificar essas componentes é o de Análise Espectral Singular. Os experimentos consideram quatro séries temporais para avaliar o desempenho da RNF-TS proposta e para comparar com outras metodologias na literatura. A abordagem proposta apresentou resultados competitivos e é uma alternativa promissora para previsão de séries sazonais e caracterização de suas componentes não-observáveis.

Palavras-chave— Previsão de Séries Temporais, Evolutivo, *Neuro-Fuzzy Takagi-Sugeno*, Análise Espectral Singular.

1 Introdução

A previsão de séries temporais é um tema relevante para diversas áreas do conhecimento, assim como para empresas e organizações governamentais. A partir dos resultados da previsão, o especialista pode tomar decisões adequadas e realizar o planejamento de ações preventivas. Além da previsão, o especialista pode também estar interessado na caracterização de séries temporais, ou seja, na identificação de suas componentes não-observáveis. Essas componentes possuem padrões característicos que são importantes para o entendimento do comportamento da série (Brockwell and Davis, 2002). Para lidar com séries temporais reais que apresentam não-estacionaridades e diversas complexidades, métodos de previsão baseados em técnicas de inteligência computacional vêm sendo amplamente investigados. Em Abdollahzade et al. (2015), um método híbrido para previsão de séries temporais não-lineares e caóticas baseado em um modelo *neuro-fuzzy* linear local e a otimização do método de Análise Espectral Singular são desenvolvidos. Em Gan et al. (2014), um modelo autorregressivo quasi-linear, baseado em redes de funções de base radial, é proposto para previsão de séries temporais sazonais. É uma ten-

dência na comunidade científica o uso de técnicas de agrupamento e de abordagens evolutivas. Um modelo para previsão a longo prazo de séries com base no algoritmo de agrupamento *fuzzy c-means* é proposto em Wang et al. (2015). Em Birek et al. (2014), um algoritmo modificado com estrutura *fuzzy Takagi-Sugeno* evolutiva (eTS) aplicado para o problema de previsão de uma série temporal experimental é descrito.

Neste trabalho, é proposta uma metodologia evolutiva para previsão e caracterização de séries temporais a partir de uma Rede *Neuro-Fuzzy Takagi-Sugeno* (RNF-TS). As entradas dessa RNF-TS são as componentes não-observáveis extraídas da série temporal usando o método de Análise Espectral Singular (*Singular Spectrum Analysis* - SSA) (Abdollahzade et al., 2015). Em seguida, a RNF-TS usa essas entradas para adaptar seus parâmetros, evoluir a sua estrutura e prever valores futuros para a série. As proposições do consequente das regras *fuzzy* são modelos no espaço de estados, onde os estados são formados pelas próprias componentes extraídas. A originalidade dessa metodologia se apresenta nos seguintes aspectos: dados das componentes não-observáveis são usados para realizar a previsão da série temporal; os consequentes das regras *fuzzy* são mode-

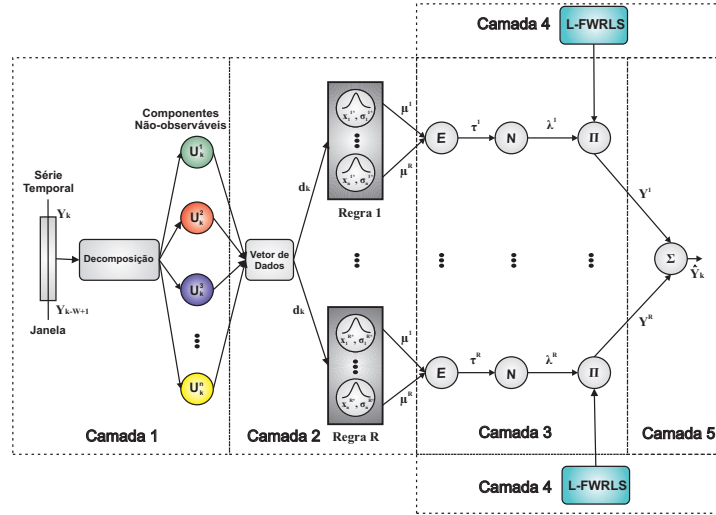


Figura 1: Estrutura da Rede *Neuro-Fuzzy Takagi-Sugeno* (RNF-TS).

los no espaço de estados, representando o comportamento dinâmico dessas componentes; estratégia “dividir para conquistar”, que divide um a série temporal complexa em um conjunto de componentes mais fáceis de lidar (Angelov, 2013).

2 Metodologia Proposta

A RNF-TS desenvolvida para previsão de séries temporais é exibida na Figura 1. Esta rede possui um ciclo de execução constituído por seis camadas: decomposição da série; aprendizado não-supervisionado do antecedente das regras *fuzzy*; agregação; estimação dos parâmetros dos consequentes das regras *fuzzy*; inferência da RNF-TS; e a reconstrução da série temporal. A descrição de cada uma dessas camadas é abordada nas subseções a seguir.

2.1 Decomposição de Séries Temporais

Na camada de decomposição, uma janela de dados ω_k com tamanho fixo e contendo W amostras da série temporal é denotada por:

$$\omega_k = [Y_{k-W+1}, Y_{k-W+2}, \dots, Y_k], \quad \omega_k \in \mathbb{R}^{1 \times W}, \quad (1)$$

onde Y_k é o valor da série temporal no instante k . Em cada instante, essa janela de dados se desloca, adicionando uma nova amostra Y_k . A partir dos dados de ω_k , n componentes não-observáveis são extraídas para todo k usando um método de decomposição: $U_k^1, U_k^2, \dots, U_k^n$. Neste trabalho, o método SSA é responsável por identificar essas componentes (Abdollahzade et al., 2015). O algoritmo básico do SSA é composto por quatro etapas: incorporação, decomposição em valores singulares, agrupamento e média diagonal. Seja ω_k com W amostras da série conforme Eq. (1). A

etapa de incorporação mapeia os dados de ω_k para uma sequência de vetores defasados com dimensão L , conforme a seguir (Abdollahzade et al., 2015):

$$\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_K], \quad \mathbf{M} \in \mathbb{R}^{L \times K}, \quad \text{onde} \\ \mathbf{m}_i = [\omega_k(i), \dots, \omega_k(i+L-1)]^T, \quad 1 \leq i \leq K, \quad \text{e} \\ K = W - L + 1, \quad \text{para } L < \frac{W}{2}, \quad (2)$$

sendo que $\mathbf{M}$ é conhecida como matriz trajetória e $i, \dots, i+L-1$ são índices que indicam os dados contidos em ω_k . Pode-se notar que $\mathbf{M}$ é uma matriz de Hankel, ou seja, os valores das diagonais da matriz que ascendem da esquerda para direita são iguais. A segunda etapa, que é a decomposição em valores singulares da matriz $\mathbf{M}$, é representada por:

$$\begin{aligned} \mathbf{M} &= \mathbf{S}_1 + \dots + \mathbf{S}_r, \quad \text{onde} \\ \mathbf{S}_i &= \sqrt{\rho_i} \mathbf{p}_i \mathbf{v}_i^T, \quad \mathbf{S}_i \in \mathbb{R}^{L \times K} \quad \text{e} \\ \mathbf{v}_i &= \mathbf{M}^T \mathbf{p}_i / \sqrt{\rho_i}, \quad \text{para } i = 1, \dots, r. \end{aligned} \quad (3)$$

sendo que $\rho_1, \rho_2, \dots, \rho_L$ são os autovalores de $\mathbf{M}\mathbf{M}^T$ em ordem decrescente; r é a quantidade de autovalores não-nulos ($r = \max\{i | \rho_i > 0\}$); $\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_r$ são vetores singulares à esquerda de $\mathbf{M}$ ou autovetores de $\mathbf{M}\mathbf{M}^T$; e $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r$ são vetores singulares à direita de $\mathbf{M}$ ou autovetores de $\mathbf{M}^T\mathbf{M}$. As matrizes $\mathbf{S}_i$ possuem posto igual a 1 (Abdollahzade et al., 2015). Em seguida, a etapa de agrupamento é iniciada, cuja função é distribuir o conjunto de índices $\{1, \dots, r\}$ em n subconjuntos disjuntos definidos como $I_1, \dots, I_n$. O valor de n corresponde a quantidade de componentes que devem ser extraídas pelo método SSA a partir dos dados da janela. Sendo assim, seja $I = \{i_1, \dots, i_q\}$ um grupo de índices qualquer. Logo, a matriz resultante $\mathbf{M}_I \in \mathbb{R}^{L \times K}$ associada com o grupo I é definida como $\mathbf{M}_I = \mathbf{M}_{i_1} + \dots + \mathbf{M}_{i_q}$.

Estas matrizes são calculadas para cada grupo $I = I_1, \dots, I_n$, implicando na seguinte relação:

$$\mathbf{M} = \mathbf{M}_{I_1} + \mathbf{M}_{I_2} + \dots + \mathbf{M}_{I_n}. \quad (4)$$

Por fim, a etapa de média diagonal transforma cada uma das n matrizes da Eq. (4) em uma componente não-observável. Nesse trabalho, uma adaptação dessa última etapa é realizada. Para a janela ω_k com W amostras da série temporal nos instantes $k - W + 1, k - W + 2, \dots, k$, o procedimento de média diagonal original resulta em W amostras associadas com esses instantes de tempo para cada uma das n componentes. A metodologia proposta necessita apenas dos valores das n componentes no instante k . Sendo assim, a alternativa proposta neste trabalho para esse procedimento é:

$$U_k^q = \mathbf{M}_I(j, K), \text{ onde } q = 1, \dots, n \quad (5)$$

$$j = W - K + 1,$$

sendo que W é o tamanho da janela, K é calculado conforme Eq. (2) e $\mathbf{M}_I(j, K)$ é o elemento da linha j e coluna K de uma das matrizes da Eq. (4).

2.2 Aprendizado Não-Supervisionado Evolutivo dos Antecedentes da RNF-TS

Após a camada de decomposição, os dados das n componentes não-observáveis são utilizados para compor um vetor de dados no instante k representado por:

$$\mathbf{d}_k = [U_{k-1}^1, U_{k-1}^2, \dots, U_{k-1}^n, Y_k], \mathbf{d}_k \in \mathbb{R}^{1 \times n+1}, \quad (6)$$

onde $U_{k-1}^1, U_{k-1}^2, \dots, U_{k-1}^n$ são as entradas da RNF-TS e Y_k é a saída desejada. Observe que os dados das componentes no instante $k - 1$ são usados para prever a observação da série no instante k . Depois da construção de $\mathbf{d}_k$, um método de agrupamento para extração de conhecimento a partir desse vetor, baseado na técnica *eClustering+* proposta em Angelov (2013), é aplicado. Esse método recursivo e evolutivo divide o espaço de dados em *clusters* que se sobrepõem, onde cada *cluster* está associado a uma regra *fuzzy*. Essa camada visa formar *clusters* com alta capacidade de generalização, determinar automaticamente a quantidade adequada de *clusters* e simplificar a base de regras, removendo os *clusters* menos significativos (Angelov, 2013). Essa técnica evolutiva de agrupamento usa o conceito de Estimativa Recursiva da Densidade (*Recursive Density Estimation* - RDE) como critério para criação de novos *clusters*. Um vetor $\mathbf{d}_k$ que satisfaz a Condição A a seguir possui alta capacidade de generalização, logo, os valores das componentes de $\mathbf{d}_k$ comporão o centro do novo *cluster* no espaço n -dimensional. Se essa condição não for satisfeita, os

pontos de $\mathbf{d}_k$ são associados ao *cluster* mais próximo (Angelov, 2013). A Condição A é denotada por:

$$D_k(\mathbf{d}_k) > \max_{i=1}^R D_k(\mathbf{d}_k^{i*}) \text{ OU } D_k(\mathbf{d}_k) < \min_{i=1}^R D_k(\mathbf{d}_k^{i*}) \quad (7)$$

onde R é o número de *clusters* existentes, $D_k(\mathbf{d}_k)$ é a densidade do vetor $\mathbf{d}_k$, $\mathbf{d}_k^{i*}$ corresponde ao centro do i -ésimo *cluster* e $D_k(\mathbf{d}_k^{i*})$ é a densidade desse ponto central. Para cada $\mathbf{d}_k$, sua densidade $D_k(\mathbf{d}_k)$ é recursivamente determinada por (Angelov, 2013):

$$D_k(\mathbf{d}_k) = \frac{k-1}{(k-1) \left(\sum_{j=1}^{n+1} d_{kj}^2 + 1 \right) + b_k - 2 \sum_{j=1}^{n+1} d_{kj} c_{kj}},$$

onde $D_1(\mathbf{d}_1) = 1$ (condição inicial),

$$b_k = b_{k-1} + \sum_{j=1}^{n+1} d_{(k-1)j}^2, \quad b_1 = 0,$$

$$c_{kj} = c_{(k-1)j} + d_{(k-1)j}, \quad c_{1j} = 0, \quad j = 1, \dots, n+1, \quad (8)$$

sendo que o primeiro vetor $\mathbf{d}_1$ que a RNF-TS recebe é definido como centro do primeiro *cluster* com densidade $D_1(\mathbf{d}_1) = 1$ (Angelov, 2013). Quando $\mathbf{d}_k$ é selecionado para ser o centro de um novo *cluster* pela Condição A, seu valor de densidade calculado é armazenado e passa a ser atualizado pela seguinte equação enquanto este *cluster* existir (Angelov, 2013):

$$D_k(\mathbf{d}_k^{i*}) = \frac{k-1}{k-1 + (k-2) \left(\frac{1}{D_{k-1}(\mathbf{d}_k^{i*})} - 1 \right) + \Lambda_k}$$

$$\Lambda_k = \sum_{j=1}^{n+1} (d_{kj} - d_{(k-1)j}), \quad (9)$$

pois todo novo $\mathbf{d}_k$ que chega a RNF-TS influencia na densidade de todos os *clusters* existentes. Depois que uma nova regra *fuzzy* é criada, a Condição B abaixo é avaliada para cada *cluster* existente:

$$\text{SE } \mu_i^j(d_{kj}) > e^{-1}, i = [1, R], j = [1, n] \quad (10)$$

$$\text{ENTÃO } R \leftarrow R - 1,$$

onde μ_i^j é a função de pertinência do i -ésimo *cluster* e da j -ésima entrada em $\mathbf{d}_k$, para $j = 1, \dots, n$. Essa condição é muito importante para evitar a redundância e controlar a elevada sobreposição entre os *clusters*. Todos os *clusters* que satisfazem essa condição são substituídos pelo novo *cluster* formado pela Condição A (Angelov, 2013). A função de pertinência gaussiana μ_i^j é calculada para um vetor $\mathbf{d}_k$ de acordo com a seguinte equação (Angelov, 2013):

$$\mu_j^i(d_{kj}) = e^{-\frac{(d_j^{*i} - d_{kj})^2}{2(\sigma_j^i)^2}}, \quad i = [1, R], j = [1, n], \quad (11)$$

sendo que σ_j^i é o raio do cluster i no eixo de entrada j . O valor de σ_j^i é atualizado recursivamente por (Angelov, 2013):

$$\sigma_{kj}^i = \sqrt{\zeta(\sigma_{(k-1)j}^i)^2 + (1 - \zeta)\frac{1}{S_k^i}(d_{kj} - d_j^{i*})^2} \quad (12)$$

onde ζ é uma constante e S_k^i é o suporte ou a quantidade de pontos associados ao i -ésimo *cluster*. Finalmente, as proposições do antecedente das regras *fuzzy* podem ser formadas usando o centro dos *clusters* que foram identificados pelo algoritmo de agrupamento evolutivo. Logo, os antecedentes possuem a seguinte composição, considerando o vetor $\mathbf{d}_k$ na Eq. (6):

$$\text{Regra}^i: \quad \mathbf{SE} (U_{k-1}^1 \sim U^{1i*}) \mathbf{E} (U_{k-1}^2 \sim U^{2i*}) \mathbf{E} (U_{k-1}^3 \sim U^{3i*}) \mathbf{E} \dots \mathbf{E} (U_{k-1}^n \sim U^{ni*}) \quad (13)$$

onde U_j^{i*} representa o centro do *cluster* i para a j -ésima entrada da RNF-TS, com $j = 1, \dots, n$.

2.3 Camada de Agregação

A terceira camada da RNF-TS usa a norma-t produto para calcular o grau de ativação de cada regra *fuzzy*, considerando as funções de pertinência obtidas na camada anterior, conforme a seguir (Angelov, 2013):

$$\tau_k^i(d_{kj}) = t\{\mu_j^i(d_{kj})\} = \prod_{j=1}^n \mu_j^i(d_{kj}), \quad i = [1, R]. \quad (14)$$

Os valores de τ_k^i devem ser normalizados tal que a soma de todos os graus de ativação seja igual a 1. Portanto o grau de ativação normalizado da regra i no instante k é (Angelov, 2013):

$$\lambda_k^i = \frac{\tau_k^i}{\sum_{l=1}^R \tau_k^l} \quad (15)$$

2.4 Estimativa Paramétrica Recursiva dos Consequentes no Espaço de Estados

A quarta camada envolve a proposição consequente das regras *fuzzy*. Neste trabalho, uma regra *fuzzy* completa da RNF-TS é representada por:

$$\mathbf{SE} (U_{k-1}^1 \sim U^{1i*}) \mathbf{E} \dots \mathbf{E} (U_{k-1}^n \sim U^{ni*}) \\ \text{ENTÃO } \mathbf{x}_k^i = \mathbf{A}_k^i \mathbf{x}_{k-1}^i + \mathbf{B}_k^i \mathbf{u}_{k-s}^i, \quad (16)$$

onde $\mathbf{A}_k^i \in \mathbb{R}^{n \times n}$, $\mathbf{B}_k^i \in \mathbb{R}^{n \times n}$ e:

$$\mathbf{x}_k^i = [U_k^1, U_k^2, \dots, U_k^n]^T, \mathbf{x}_k^i \in \mathbb{R}^{n \times 1}; \\ \mathbf{x}_{k-1}^i = [U_{k-1}^1, U_{k-1}^2, \dots, U_{k-1}^n]^T, \mathbf{x}_{k-1}^i \in \mathbb{R}^{n \times 1}; \\ \mathbf{u}_{k-s}^i = [U_{k-s}^1, U_{k-s}^2, \dots, U_{k-s}^n]^T, \mathbf{u}_{k-s}^i \in \mathbb{R}^{n \times 1}. \quad (17)$$

Em outras palavras, as componentes não-observáveis são os próprios estados e o modelo no espaço de estado representa o comportamento dinâmico dessas componentes. O vetor $\mathbf{u}_{k-s}^i$ considera a informação no instante $k-s$, onde s é o período sazonal da série temporal. O valor de s deve ser informado pelo especialista ou estimado a partir da função de autocovariância (Brockwell and Davis, 2002). Para séries sazonais, a informação no instante $k-s$ é relevante porque exibe um comportamento similar ao instante atual k . O método dos mínimos quadrados recursivos ponderado *fuzzy* e local (*Local Fuzzy Weighted Recursive Least Squares* - LFWRLS) estima as matrizes $\mathbf{A}_k^i$ e $\mathbf{B}_k^i$ das proposições consequentes para cada regra *fuzzy* (Angelov, 2013). Dessa forma, considerando as componentes ou estados extraídos na camada de decomposição, tem-se:

$$\mathbf{x}_k^i = \mathbf{A}_k^i \mathbf{x}_{k-1}^i + \mathbf{B}_k^i \mathbf{u}_{k-s}^i \\ \mathbf{x}_k^i = [\mathbf{A}_k^i \quad \mathbf{B}_k^i] \begin{bmatrix} \mathbf{x}_{k-1}^i \\ \mathbf{u}_{k-s}^i \end{bmatrix} \quad (18)$$

$$\mathbf{x}_k^i = \Delta_k^T \mathbf{v}^T.$$

Sendo assim, a estimação recursiva para cada regra *fuzzy* $i = [1, R]$ é dada por:

$$\mathbf{K}_k^i = \frac{\mathbf{P}_{k-1}^i \mathbf{v}^T}{\mathbf{v} \lambda_k^i \mathbf{P}_{k-1}^i \mathbf{v}^T + 1}, \quad \mathbf{K}_k^i \in \mathbb{R}^{2n \times 1}; \\ \Delta_k^i = \Delta_{k-1}^i + \lambda_k^i \mathbf{K}_k^i (\mathbf{x}_k^i - \mathbf{v} \Delta_{k-1}^i), \quad \Delta_k^i \in \mathbb{R}^{2n \times n}; \\ \mathbf{P}_k^i = \mathbf{P}_{k-1}^i - \lambda_k^i \mathbf{K}_k^i \mathbf{v}^T \mathbf{P}_{k-1}^i, \quad \mathbf{P}_k^i \in \mathbb{R}^{2n \times 2n}, \quad (19)$$

sendo que $\mathbf{K}_k^i$ é a matriz de ganho para cada regra e $\mathbf{P}_k^i$ é a matriz de covariância da regra i .

2.5 Camada de Inferência da RNF-TS

A quinta camada é a de inferência, cuja função é gerar o vetor de estado global $\hat{\mathbf{x}}_k$ da RNF-TS, realizando a média ponderada por λ_k^i e considerando a contribuição $\mathbf{x}_k^i$ de cada regra *fuzzy*. O vetor de estado global é calculado como:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^R \lambda_k^i \hat{\mathbf{x}}_k^i, \quad (20)$$

2.6 Camada de Reconstrução

A última camada usa o vetor de estado global da Eq. (20) e reconstrói a série temporal no instante k ao somar as componentes ou estados, isto é:

3 Resultados Experimentais

Nessa se  o, a metodologia proposta   avaliada e seus resultados s o comparados com outros trabalhos na literatura. Quatro s ries temporais s o consideradas: venda de livros, de roupas, de m veis e de *hardware* (Gan et al., 2014). Essas s ries temporais possuem dados mensais, ou seja, com per odo sazonal $s = 12$, e 120 amostras. Nos experimentos, a RNF-TS precisa realizar a previs o dos  ltimos doze valores da s rie, isto  , $h = 12$   o horizonte de predi  o. Logo, as primeiras 108 amostras da s rie temporal s o usadas para o treinamento e o restante   comparado com a sa da da RNF-TS para valida  o. O crit rio de erro RMSE   utilizado para avalia  o:

$$\text{RMSE} = \sqrt{\frac{1}{h} \sum_{k=1}^h (Y_k - \hat{Y}_k)^2} \quad (25)$$

onde h   horizonte de previs o, Y_k   o valor real da s rie temporal e $\hat{Y}_k$   o dado gerado pela RNF-TS no modo de treinamento ou no modo de previs o. Na Tabela 1, os resultados de acordo com o crit rio RMSE tanto para o treinamento quanto para a previs o das s ries s o exibidos, considerando diferentes modelos na literatura. A quantidade de regras geradas em cada caso est  listada na coluna R . As colunas dos modelos RBF-AR (Gan et al., 2014) e TDNN (Taskaya-Temizel and Casey, 2005) correspondem a m dia do RMSE para 30 execu  es dos respectivos modelos. No treinamento, a metodologia proposta foi superior em todos os casos. Em rela  o a previs o, a RNF-TS exibiu resultados melhores segundo o RMSE para as s ries de venda de livros, de roupas e de *hardware*, enquanto para a venda de m veis o modelo ANN+DSDT (Zhang and Qi, 2005) foi levemente superior. Pode-se observar na Figura 2 os resultados do treinamento e da previs o para cada s rie temporal. Em preto, os dados reais das s ries s o exibidos. A parte inicial em preto corresponde aos dados iniciais para a janela da Eq. (1). As 108 primeiras amostras para o treinamento e a previs o das 12  ltimas amostras geradas pela RNF-TS s o mostradas em azul e vermelho, respectivamente.

4 Conclus es

A metodologia proposta apresentou resultados competitivos em compara  o com outras t cnicas relevantes encontradas na literatura, tendo como vantagem a evolu  o da estrutura da rede neuro-fuzzy e a identifica  o das componentes n o-observ veis.

Agradecimentos

Os autores agradecem a CAPES e ao IFMA pelo suporte financeiro a este trabalho, e a PPGEE-

$$\begin{aligned} \hat{Y}_k &= [1 \ 1 \ \cdots \ 1] \hat{\mathbf{x}}_k, \\ \hat{Y}_k &= \sum_{j=1}^n \hat{x}_{kj}, \\ \hat{Y}_k &= \hat{U}_k^1 + \hat{U}_k^2 + \cdots + \hat{U}_k^n. \end{aligned} \quad (21)$$

2.7 Monitoramento da Qualidade dos Clusters

Durante a execu  o das camadas da RNF-TS, os *clusters* s o monitorados, anulando a influ ncia na sa da e em seguida removendo aqueles que apresentam baixa qualidade. A seguinte Condi  o C   usada para essa tarefa:

$$\begin{aligned} \text{SE } (\Omega_k^i < \bar{\Omega}_k - \hat{\Omega}_k) \text{ E } (S_k^i < 3) \text{ E } (k \geq I^{i*} + 10) \\ \text{ENT O } \lambda_k^i \rightarrow 0 \end{aligned} \quad (22)$$

sendo que S_k^i   o suporte do *cluster* i e I^{i*}   o instante k em que o *cluster* i foi gerado. O termo Ω_k^i corresponde a utilidade do *cluster* i no instante atual k , que   calculada como:

$$\Omega_k^i = \frac{\sum_{l=1}^k \lambda_l^i}{k - I^{i*}}, \quad i = [1, R], \quad (23)$$

Os termos $\bar{\Omega}_k$ e $\hat{\Omega}_k$ denotam respectivamente a utilidade m dia e o desvio padr o depois de k dados lidos. Ambos podem ser calculados recursivamente. Ap s a camada de infer ncia, os *clusters* que satisfazem a Condi  o C s o removidos.

2.8 Modos de Opera  o da RNF-TS

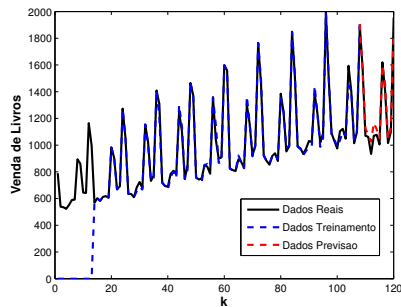
A RNF-TS proposta tem dois modos de opera  o: treinamento e previs o. No primeiro, h  informa  es dispon veis da s rie temporal original. Ent o, todo procedimento descrito nas se  es anteriores s o realizados para Y_k . Caso contr rio, quando n o h  Y_k dispon vel, a previs o ocorre. Nesta ocasi o, a RNF-TS n o evolui sua estrutura nem atualiza os par metros dos consequentes das regras *fuzzy*. Logo, as camadas de decomposi  o e de estima  o dos consequentes n o s o realizadas. Sendo assim, para obter os valores das componentes ou estados e construir o vetor $\mathbf{d}_k$, o vetor de estados $\hat{\mathbf{x}}_{k-1}$ calculado no instante anterior   realimentado para a entrada da RNF-TS, ent o:

$$\begin{aligned} \mathbf{d}_k &= [\hat{\mathbf{x}}_{k-1}^T \ \text{NULL}], \\ \mathbf{d}_k &= [\hat{U}_{k-1}^1, \hat{U}_{k-1}^2, \dots, \hat{U}_{k-1}^n \ \text{NULL}]. \end{aligned} \quad (24)$$

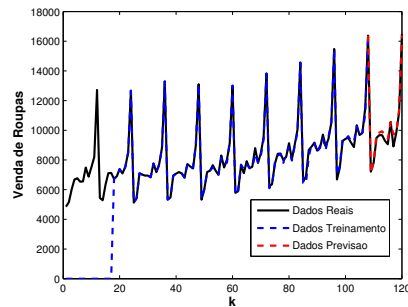
Em seguida, a camada de agrega  o executa normalmente, assim como as camadas de infer ncia e reconstru  o para prever o pr ximo valor da s rie.

Tabela 1: Comparação dos Resultados para as Séries Temporais Usando o RMSE (T - Treinamento e P - Previsão).

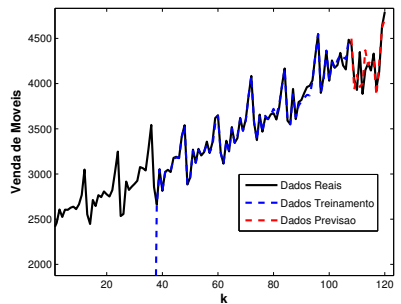
Série	R	RNF-TS		RBF-AR		ANN+DSDT		TDNN	ARIMA
		T	P	T	P	T	P	P	P
Livro	11	20.34	58.06	62.63	96.8	37.39	88.74	91.51	98.17
Roupas	4	143.8	312.02	244.89	337.72	174.07	315.43	372.5	519.6
Móveis	8	50.20	110.24	77.85	191.54	52.35	99.45	173.1	124.44
Hardware	4	25.79	22.80	40.31	23.53	26.16	49.17	35.7	100.71



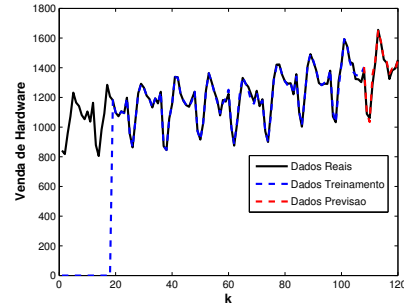
(a) Venda de Livros.



(b) Venda de Roupas.



(c) Venda de Móveis.



(d) Venda de Hardware.

Figura 2: Resultados da Execução da Rede RNF-TS.

UFMA por encorajar o desenvolvimento desta pesquisa.

Referências

- Abdollahzade, M., Miranian, A., Hassani, H. and Iranmanesh, H. (2015). A new hybrid enhanced local linear neuro-fuzzy model based on the optimized singular spectrum analysis and its application for nonlinear and chaotic time series forecasting, *Information Sciences, Elsevier* **295**: 107–125.
- Angelov, P. (2013). *Autonomous Learning Systems: From Data Streams to Knowledge in Real-time*, reprint edn, John Wiley & Sons, Lancaster University.
- Birek, L., Petrovic, D. and Boylan, J. (2014). Water leakage forecasting: the application of a modified fuzzy evolving algorithm, *Applied Soft Computing, Elsevier* **14**: 305–315.
- Brockwell, P. J. and Davis, R. A. (2002). *Introduction to Time Series and Forecasting*, 2 edn, Springer Texts in Statistics.
- Gan, M., Cheng, Y., Liu, K. and Zhang, G. (2014). Seasonal and trend time series forecasting based on a quasi-linear autoregressive model, *Applied Soft Computing, Elsevier* **24**: 13–18.
- Taskaya-Temizel, T. and Casey, M. (2005). A comparative study of autoregressive neural network hybrids, *Neural Networks* **18**: 781–789.
- Wang, W., Pedrycz, W. and Liu, X. (2015). Time series long-term forecasting model based on information granules and fuzzy clustering, *Engineering Applications of Artificial Intelligence, Elsevier* **41**: 17–24.
- Zhang, G. P. and Qi, M. (2005). Neural network forecasting for seasonal and trend time series, *European Journal of Operational Research, Elsevier* **160**: 501–514.