

PREVISÃO DE VENDAS POR SÉRIES TEMPORAIS EM UMA INDÚSTRIA DE ALIMENTOS: COMPARAÇÃO ENTRE TÉCNICAS ESTRUTURAIS E DE INTELIGÊNCIA COMPUTACIONAL

Thassyo Jorge Gonçalves Pereira

Laboratório de Logística, Universidade do Estado do Pará, Travessa Dr. Enéas Pinheiro, 2626, 66095-015, Marco, Belém, PA, Brasil.
thassyoy@ymail.com

Felipe Erdmann Oliveira

Universidade do Estado do Pará, Travessa Dr. Enéas Pinheiro, 2626, 66095-015, Marco, Belém, PA, Brasil.
felipeerdmann@hotmail.com

Rodrigo Rangel Ribeiro Bezerra

Universidade do Estado do Pará, Travessa Dr. Enéas Pinheiro, 2626, 66095-015, Marco, Belém, PA, Brasil.
rodrigorangell_rr@hotmail.com

Thiago Lima de Matos

Laboratório de Logística, Universidade do Estado do Pará, Travessa Dr. Enéas Pinheiro, 2626, 66095-015, Marco, Belém, PA, Brasil.
thiago_lp91@hotmail.com

André Clementino de Oliveira Santos

Departamento de Engenharia, Universidade do Estado do Pará, Travessa Dr. Enéas Pinheiro, 2626, 66095-015, Marco, Belém, PA, Brasil.
acos@superig.com.br

Resumo. O elo industrial do setor de alimentos apresenta alta variabilidade em sua demanda, dificultando a previsão de demanda por séries temporais. Desta forma, este artigo buscou utilizar técnicas avançadas de inteligência computacional para realizar a previsão de demanda de séries temporais, notadamente as Redes Neurais Artificiais e Regressão por Vetores de Suporte auxiliada por Otimização por Enxame de Partículas. Os resultados obtidos, entretanto, apontam para a performance inferior destas técnicas se comparadas aos modelos mais simples, demonstrando que, para o produto estudado, as técnicas de séries temporais não geraram resultados satisfatórios.

Palavras-chave: Previsão de demanda por séries temporais, Fábrica de alimentos, Regressão por Vetores de Suporte, Otimização por Enxame de Partículas, Redes Neurais Artificiais.

Abstract. The food industry, mainly the manufacturing link, presents a problem of great demand variability, which causes time series forecasting approaches to provide weak results to the well management of the company. Our paper uses computer intelligence techniques (Artificial Neural Networks and Support Vector Regression aided by Particle Swarm Optimization) to forecast a demand time series of a food manufacturing company. Their results, however, rank below the simple structural approaches used.

Key-words: Time Series Forecasting, Food Industry Support Vector Regression, Particle Swarm Optimization, Artificial Neural Networks.

1. INTRODUÇÃO

O setor de alimentos obteve em 2010 um faturamento correspondente a 9% do PIB brasileiro, dos quais 8% corresponde ao mercado de trigo e derivados^[2]. Este setor apresenta alta volatilidade de demanda, que piora quanto mais distante o elo está do consumidor final^[19].

Desta forma, este artigo busca estudar a previsão de demanda do produto mais vendido em uma fábrica de derivados de trigo (Belém – PA) empregando técnicas consagradas de previsão (média móvel simples e suavização exponencial dupla e simples) e inteligência computacional (Redes Neurais Artificiais e Regressão por Vetores de Suporte auxiliado por Otimização por Enxame de Partículas) visando comprovar a hipótese de estas últimas obterem melhores resultados.

Para tal, primeiramente realizamos um apanhado de literatura sobre previsão de demanda por média móvel (Seção 2.2), suavização exponencial simples (Seção 2.3), Método de Holt (Seção 2.4), Redes Neurais Artificiais (Seção 2.5), Regressão por Vetores de Suporte (Seção 2.6) e Otimização por Enxame de Partículas (Seção 2.7). Então, demonstramos a metodologia utilizada nos testes (Seção 3.2) da Seção 3.3 e concluímos com uma visão geral dos resultados obtidos, comprovação ou não da hipótese e recomendações para trabalhos complementares (Seção 4).

2. FUNDAMENTAÇÃO TEÓRICA

A previsão de demanda é vital na estratégia industrial, pois norteia as principais decisões financeiras, comerciais e operacionais^[8], como definição de políticas de serviço ao cliente, investimento em estoque, ordens de reposição, necessidade de aumento de capacidade produtiva e escolha de estratégias de operações^[12].

2.1 PROCESSO DE PREVISÃO DE DEMANDA

As técnicas de previsão são classificadas em quantitativas e qualitativas. As qualitativas (*e.g.* método Delphi, pesquisa de mercado e simulação de cenário) partem do consenso de opiniões. Já as quantitativas são métodos que tem como base dados quantitativos e técnicas estatísticas e são divididas em séries temporais e correlação. A primeira, também denominada estrutural possui como técnicas bastante utilizadas: média móvel, suavização exponencial, projeção de tendências e decomposição. A correlação é subdividida, principalmente, em regressão simples, regressão múltipla e métodos econométricos^[8]. As séries temporais partem do princípio de que a demanda futura será uma projeção de seus valores passados, não sofrendo influência de outras variáveis. Já a correlação busca desenvolver uma projeção da demanda de um determinado produto baseando-se na previsão de outra variável, logo é uma relação entre duas variáveis que possuem alguma correlação^[16].

A Figura 1 representa o processo de previsão de demanda. Na definição do objetivo do modelo tem-se a razão ou finalidade do modelo em questão, a sofisticação e o detalhamento do modelo dependem da importância do produto estudado. Na coleta e análise dos dados, a confiabilidade deve-se ao fator número de dados históricos, entretanto variações extraordinárias devem ser analisadas e substituídas por valores médios. O tamanho do período estudado tem influência direta na escolha da técnica mais adequada^[16].

Os gráficos de previsão podem conter tendência, sazonalidade, variações irregulares e variações randômicas. Após a obtenção das previsões utilizam-se algumas medidas de desempenho, objetivando a manutenção e monitoração do modelo, verificar a acurácia dos valores previstos, identificar, isolar e corrigir variações anormais e permitir a escolha de técnicas ou parâmetros mais eficientes^[16].

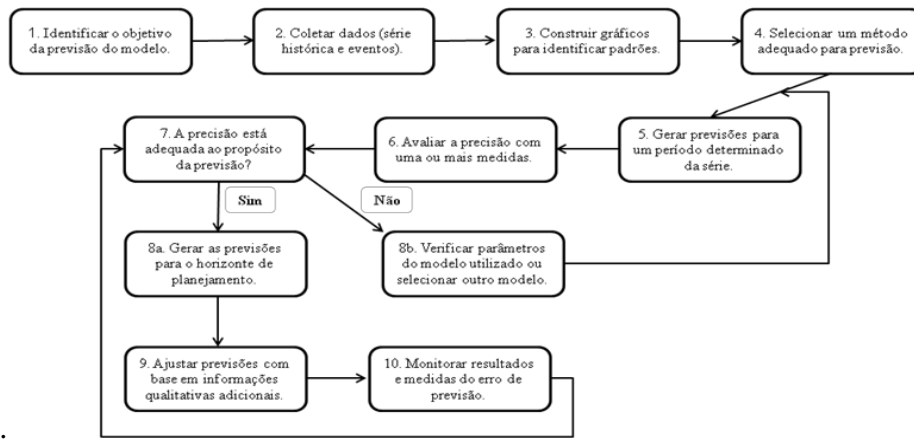


Figura 1 – Etapas do modelo de previsão da demanda (MESQUITA, 2008)

As medidas de desempenho são determinadas por um estimador, que é uma das muitas maneiras de se quantificar a diferença entre os valores implícitos em uma estimativa e os verdadeiros valores da quantidade a ser estimada, ou seja, o desvio entre o valor real da demanda e o valor previsto. Estimadores comuns são o Erro Percentual Médio Absoluto (MAPE - *Mean Absolute Percentual Error*), Erro Quadrado Médio (MSE – *Mean Squared Error*), a Raiz do Desvio Médio Quadrado (RMSE – *Root Mean Square Deviation*) e Desvio Médio Absoluto (MAD - *Mean Absolute Deviation*)^[16]. Os erros gerados pelo modelo devem ser objeto de controle, comumente via o gráfico de controle do 4 MAD, em que os erros gerados devem estar entre um limite superior de 4 MAD e um limite inferior de $-4MAD$. Esta faixa visa anular o erro acumulado pelo modelo^[16].

2.2 MÉDIA MÓVEL SIMPLES

É uma técnica de projeção que considera a média aritmética de “ n ” períodos anteriores, buscando suavizar as previsões^[8]. Valores historicamente baixos e valores historicamente altos se combinam, gerando assim uma previsão média com menor variabilidade do que os dados originais^[16].

A média móvel usa dados de um número predeterminado de períodos, normalmente os mais recentes, para gerar sua previsão. A cada novo período de previsão se substitui o dado mais antigo pelo mais recente^[16]. Podendo ser obtida a partir de (1).

$$Mm_n = \frac{\sum_{i=1}^n D_i}{n} \quad (1)$$

Onde Mm_n é a média móvel de n períodos, D_i é a demanda ocorrida no período i , n é o número de períodos e i é o índice do período ($i=1, 2, 3, \dots$).

Os valores previstos tendem a ser menos voláteis ou mais suaves do que os reais dados. A técnica tende a aplainar os picos e vales que ocorrem nos dados originais, podendo ser às vezes chamada de método de *ajuste*. Quanto maior o valor de n (quanto mais pontos de dados passados são nivelados juntos), mais suave será a previsão da média móvel^[10].

2.3 SUAVIZAÇÃO EXPONENCIAL SIMPLES

Esta técnica pressupõe que a demanda oscila em torno de uma demanda base constante. Partindo de um valor inicial, a “base” é corrigida a cada período, conforme no vos dados de demanda são incorporados à série histórica^[8].

Diferentemente da média móvel o método não atribui peso igual a todos os n períodos utilizados no cálculo e também não usa dados fora dos n períodos com os quais é utilizado na média móvel. Estes dois problemas são resolvidos com o *ajustamento exponencial*^[13].

Cada nova projeção é obtida com base na previsão anterior, acrescida do erro cometido na previsão anterior, corrigido por um coeficiente de ponderação, que determina a velocidade de resposta, logo se o coeficiente estiver entre [0, 0,5] a velocidade de resposta será suave, mas se estiver entre [0,5, 1] então o modelo terá uma resposta agressiva Podendo ser calculado por (2)^[16].

$$M_t = M_{t-1} + \alpha(D_{t-1} - M_{t-1}) \quad (2)$$

Onde M_t é a previsão para o período t , M_{t-1} é a previsão para o período $t-1$, α é o coeficiente de ponderação determinado por programação matemática visando minimizar dada medida de erro e D_{t-1} é a demanda do período $t-1$.

Embora se possa utilizar a técnica da média móvel ou de exponencial para prever um valor para qualquer período de tempo futuro, à medida que o horizonte de previsão se alarga, nossa confiança na exatidão da previsão diminui^[10].

2.4 SUAVIZAÇÃO EXPONENCIAL DUPLA (MÉTODO DE HOLT)

A suavização exponencial dupla consiste na adição de tendência ao modelo de suavização exponencial simples. Retorna valores $\hat{y}_t = a + bt$ em que a e b são atualizados para cada período conforme (3) e (4), respectivamente^[12].

$$\hat{a}_t = \alpha x_t + (1 - \alpha)(\hat{a}_{t-1} + \hat{b}_{t-1}) \quad (3)$$

$$\hat{b}_t = \beta(\hat{a}_{t-1} - \hat{b}_{t-1}) + (1 - \beta)\hat{b}_{t-1} \quad (4)$$

Onde α e β são fatores de ponderação entre 0 e 1 obtidos via programação matemática para minimização do erro da previsão.

2.5 REDES NEURAIS ARTIFICIAIS

Uma Rede Neural Artificial é um modelo computacional que mimetiza o cérebro humano, com unidades simples (neurônios) trabalhando em paralelo sem um controle central e conectadas pelas sinapses, que possuem pesos numéricos ajustados pelo elemento de aprendizado^[11]. Um neurônio artificial apresenta características similares ao neurônio natural em que as entradas e pesos sinápticos representam as sinapses e somatório e função de ativação representam excitação ou inibição do neurônio (analogamente ao processo de troca de substâncias neurotransmissoras). Cada sinapse possui sua entrada multiplicada pelo peso sináptico, os resultados destas multiplicações são somados e submetidos à função de ativação, que restringe a saída do neurônio a uma faixa entre 0 e 1 ou entre -1 e +1 (ou não, no caso da função linear). Estas características podem ser visualizadas na Figura 2 e (5) representa o neurônio artificial^[7].

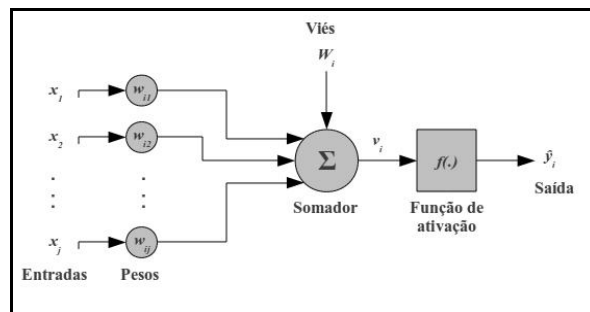


Figura 2 – Modelo de um neurônio artificial.

$f(.)$ é a função de ativação, x_j representa a entrada, w_{ij} representa o peso dado pelo neurônio em questão (i) à entrada j , W_i é o viés ou bias somado a todos os pares $w_{ij}x_j$, de modo que a saída do neurônio i é dada por (5).

$$\hat{y}_i = f(\sum_j w_{ij}x_j - W_i) \quad (5)$$

2.5.1 Função de ativação

A função de ativação visa mimetizar as características não lineares do neurônio real, de modo que funções comumente utilizadas são a função linear (6) e função tangente (ou hiperbólica, varia entre -1 e +1, conforme (7))^[1].

$$f(x) = ax \quad (6)$$

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (7)$$

2.5.2 Arquitetura da rede

A arquitetura da rede leva em consideração os seguintes parâmetros: número de camadas da rede (redes com apenas uma camada: só existe um neurônio entre qualquer entrada e qualquer saída da rede, redes com múltiplas camadas: existe mais de um neurônio entre uma entrada e uma saída), número de neurônios em cada camada, tipo de conexão entre os neurônios e topologia da rede (*feedforward* ou acíclica: a saída de um neurônio de uma camada não pode ser utilizada como entrada em neurônios de camadas anteriores, *feedback* ou cíclica: as saídas de camadas posteriores podem ser utilizadas nos neurônios de camadas anteriores)^[3].

2.5.3 Aprendizado

A aprendizagem de uma RNA consiste no processo iterativo de ajuste de parâmetros livres e pesos sinápticos, podendo ser supervisionado ou não^[7]. No aprendizado supervisionado são fornecidos pares (x, y) de entradas e saídas desejadas e a rede busca minimizar o erro entre o $\hat{y}(x)$ calculado e y via ajuste dos parâmetros. No aprendizado não supervisionado são fornecidas apenas entradas e a rede ajusta seus parâmetros com base em alguma regra de aprendizagem.

2.5.3.1 Algoritmo *backpropagation*

Considerando uma rede *feedforward* com uma camada intermediária e uma camada de saída, a camada intermediária repassa à camada de saída valores $v_{pj} = f(\sum_i w_{ij}x_{pi})$ com x_{pi} correspondendo ao valor da p -ésima observação da i -ésima entrada. O k -ésimo neurônio da camada de saída, então, gera valores conforme (8), em que W_{kj} é o vetor de viés^[6].

$$\hat{y}_{pk} = \sum_j W_{kj} f(\sum_i w_{ij}x_{pi}) \quad (8)$$

O algoritmo de aprendizado supervisionado *Backpropagation* (BP) visa encontrar na superfície de erro um mínimo global. É sintetizado na Tabela 1^[6].

<p>Inicialize aleatoriamente $w_{ij} = w_{ij}^I, W_{kj} = W_{kj}^I \in [-1, 1]$ Para todo $p = 1, 2, 3 \dots$: 1. Com w_{ij}^p e W_{kj}^p calcule v_{pj} e \hat{y}_{pk} 2. Calcule os erros da camada de saída: $\delta_{pk} = (y_{pk} - \hat{y}_{pk})$ 3. Idem para a camada intermediária: $\phi_{pj} = \sum_k \delta_{pk} W_{kj} v_{pj} (1 - v_{pj})$ 4. Atualize os pesos: $w_{ij}^{p+1} = w_{ij}^p + \eta \phi_{pj} x_{pi}$ $W_{kj}^{p+1} = W_{kj}^p + \eta \delta_{pk} v_{pj}$</p>

Tabela 1 – Algoritmo *Backpropagation*.

Em que η representa a taxa de aprendizado e varia entre $[0, 1]$. O algoritmo prossegue até que se tenha alcançado um δ_{pk} tão pequeno quanto se queira ou um determinado número de iterações seja atingido.

2.6 MÁQUINA DE VETORES DE SUPORTE

A Máquina de Vetores de Suporte (*Support Vector Machine*, SVM) baseia-se fortemente na teoria do aprendizado estatístico^[18] que consiste em características que máquinas de

aprendizado possuem para generalizar, de forma satisfatória, conjuntos de dados ainda não demonstrados^[14].

Se aplicada à classificação de dados, a SVM busca um hiperplano ótimo de separação entre as classes, minimizando a norma do vetor peso w (maximizando a margem entre as classes). Para regressão, a Regressão por Vetores de Suporte (*Support Vector Regression*, SVR), busca um hiperplano próximo o suficiente do maior número possível de pontos, cabendo a escolha de um hiperplano com uma norma baixa (evitando *underfitting*) aliado à minimização do somatório das distâncias dos pontos ao hiperplano (evitando *overfitting*)^[17]. Deste modo, SVR/SVM buscam a minimização de um risco estrutural, possuindo grande capacidade de generalização se comparadas à algoritmos de redes neurais, por exemplo, que buscam a minimização de um risco empírico advindo do conjunto de treinamento, possuindo, entretanto, alguns parâmetros livres, cuja determinação possui grande impacto na qualidade das soluções encontradas^[15].

Para uma melhor visualização, dado um conjunto de treinamento $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)\}$ (onde x é o vetor de entrada e y é o vetor com os valores desejados), ε -SVR aproximam uma função $f(x)$ (9) que tem um desvio não superior a ε de cada y_i , onde ε deve ser determinado previamente (w é o vetor de pesos e b é o viés ou *bias*).

$$f(x) = \langle w, x \rangle + b \quad (9)$$

Para generalização de $f(x)$, minimiza-se a norma euclidiana de w , gerando o problema de programação quadrática (10):

$$\min \frac{1}{2} \|w\|^2 \quad (10)$$

Sujeito a:

$$y_i - \langle w, x_i \rangle - b \leq \varepsilon \quad (11)$$

$$\langle w, x_i \rangle + b - y_i \leq \varepsilon \quad (12)$$

Difícilmente existirá uma resolução ao problema com precisão ε , sendo proposto por [4] a inclusão de variáveis de folga ξ_i' e ξ_i que permitirão alguns erros penalizados por um custo C :

$$\min \frac{1}{2} \|w\|^2 + C \sum_i (\xi_i' + \xi_i) \quad (13)$$

Sujeito a:

$$y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \quad (14)$$

$$\langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i' \quad (15)$$

$$\xi_i', \xi_i \geq 0 \quad (16)$$

Onde C é um parâmetro definido pelo usuário e $|\xi|$ equivale a 0 no caso de ser inferior a ε e a $|\xi| - \varepsilon$ caso contrário. Deste modo, restam apenas a definição das variáveis w e b , demonstrado a seguir.

2.6.1 Formulação dual e programação quadrática

Utilizando os multiplicadores de Lagrange α_i , α_i' , η_i e η_i' , pode-se escrever a função Lagrangeana L dada por (17):

$$\min \frac{1}{2} \|w\|^2 + C \sum_i (\xi_i' + \xi_i) - \sum_i \alpha_i (-y_i + \langle w, x_i \rangle + b + \varepsilon + \xi_i) - \sum_i \alpha_i' (y_i - \langle w, x_i \rangle - b + \varepsilon + \xi_i') - \sum_i (\eta_i \xi_i + \eta_i' \xi_i') \quad (17)$$

Onde as variáveis duais devem assumir valores positivos. Para obtenção do valor ótimo de L , as derivadas parciais das variáveis primais devem igualar zero, *i.e.*:

$$\frac{\partial L}{\partial b} = \sum_i (\alpha_i' - \alpha_i) \quad (18)$$

$$\frac{\partial L}{\partial w} = w - \sum_i (\alpha_i - \alpha_i') x_i \quad (19)$$

$$\frac{\partial L}{\partial \xi_i^{(\prime)}} = C - \alpha_i^{(\prime)} - \eta_i^{(\prime)} \quad (20)$$

De (19), é possível obter $w = \sum_i (\alpha_i - \alpha_i') x_i$ e $f(x) = (\alpha_i - \alpha_i') \langle x_i, x \rangle + b$, demonstrando a denominada expansão dos vetores suporte, em que w pode ser completamente descrito como uma combinação linear dos dados de entrada x_i . A substituição de (18), (19) e (20) em (17) retorna o seguinte modelo dual:

$$\max -\frac{1}{2} \sum_i (\alpha_i - \alpha_i') (\alpha_j - \alpha_j') \langle x_i, x_j \rangle - \varepsilon \sum_i (\alpha_i + \alpha_i') + \sum_i y_i (\alpha_i - \alpha_i') \quad (21)$$

Sujeito a:

$$\sum_i (\alpha_i - \alpha_i') \quad (22)$$

$$\alpha_i', \alpha_i \geq 0 \quad (23)$$

2.6.2 Condições de Karush-Kuhn-Tucker

Com w já representado, resta a determinação de b , encontrado por meio das condições de Karush-Kuhn-Tucker (KKT), que preconizam que, para a obtenção de um resultado ótimo, o produto entre as variáveis duais e as restrições deve ser nulo. Neste caso^[14]:

$$\alpha_i (-y_i + \langle w, x_i \rangle + b + \varepsilon + \xi_i) = 0 \quad (24)$$

$$\alpha_i' (y_i - \langle w, x_i \rangle - b + \varepsilon + \xi_i') = 0 \quad (25)$$

De onde pode-se notar, primeiramente, que apenas pares (x_i, y_i) com $\alpha_i^{(\prime)} = C$ ficam fora da zona de precisão ε . Segundo, qualquer $\alpha_i' \alpha_i = 0$, ou seja, nunca haverá um par de variáveis duais que são simultaneamente diferentes de 0, pois isto iria requerer folgas em ambas as direções. Finalmente, para $\alpha_i^{(\prime)} \in [0, C]$ temos $\xi_i^{(\prime)} = 0$, implicando que o segundo fator de (24) deve ser nulo. Assim, b pode ser encontrado por (26)^[14].

$$b = y_i - \langle w, x_i \rangle - \varepsilon \quad (26)$$

$$b = y_i - \langle w, x_i \rangle + \varepsilon$$

Finalmente, nota-se por (24) que apenas para os multiplicadores de Lagrange podem ser diferentes de zero, *i.e.*, para todos os pares dentro da zona de precisão ε α_i' e α_i são nulos: para o segundo fator em (24) é nulo, então α_i' e α_i têm que ser nulos para que as condições de KKT sejam satisfeitas. Desta forma, não são necessários todos os x_i para obter w : os x_i cujos coeficientes não são nulos são denominados vetores de suporte^[14].

2.6.3 Kernels

O modelo para SVR demonstrado até então, entretanto, não leva em consideração a não-linearidade da amostra, fenômeno que ocorre com frequência. Deste modo, os métodos SVR são adaptados via funções Kernel, que visam mapear esta não-linearidade. Conforme demonstrado, o algoritmo SVR depende apenas dos produtos-ponto entre as variáveis, de forma que [14] apresentam o seguinte modelo:

$$\max -\frac{1}{2} \sum_i (\alpha_i - \alpha_i') (\alpha_j - \alpha_j') k(x_i, x_j) - \varepsilon \sum_i (\alpha_i + \alpha_i') + \sum_i y_i (\alpha_i - \alpha_i') \quad (27)$$

Sujeito a:

$$\sum_i (\alpha_i - \alpha_i') \quad (28)$$

$$\alpha_i, \alpha_i \geq 0 \quad (29)$$

A função $k(\cdot)$ é denominada de função Kernel. Uma função Kernel comumente utilizada é denominada *Radial Basis Function* (RBF) e dada por $k(x, x) = e^{-\frac{\|x-x\|^2}{2\sigma^2}}$. Note-se a presença da variável σ , que introduz mais um parâmetro a ser definido pelo usuário^[14].

2.7 OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

A Otimização por Enxame de Partículas (*Particle Swarm Optimization*, PSO) consiste em uma técnica de inteligência computacional bioinspirada no comportamento de uma ninhada de pássaros voando em bando. Ela busca encontrar ótimos globais de funções complexas via busca guiada pelo espaço de solução em que se tem uma função objetivo que deve ser minimizada.

Primeiramente, os indivíduos são criados em posições e velocidades aleatórias e, a cada iteração t , estes parâmetros são atualizados via (30) e (31).

$$v(t+1)_i = Iv(t)_i + c_1A_1[pbest_i - s(t)_i] + c_2A_2[gbest - s(t)_i] \quad (30)$$

$$s(t+1)_i = s(t)_i + v(t+1)_i \quad (31)$$

Considerando como $v_i(t)$ e $s_i(t)$, respectivamente, como a velocidade e posição de uma partícula na iteração t , I como um valor positivo inferior a 1 equivalente à inércia, c_1 uma constante que mede a importância dada à melhor posição do bando e c_2 a importância dada à melhor posição de cada partícula, A_1 e A_2 números aleatórios entre 0 e 1, e $pbest_i(t)$ e $gbest(t)$ a melhor posição encontrada pela partícula e pelo bando até a iteração t , respectivamente.

A cada iteração, $pbest_i$ e $gbest$ são atualizados caso valores melhores sejam encontrados, de modo que o algoritmo segue até que um determinado número de iterações seja alcançado e os parâmetros I , c_1 e c_2 são definidos pelo usuário, bem como o número de partículas.

3. ESTUDO DE CASO

3.1 A EMPRESA

A empresa objeto de estudo é classificada como indústria alimentícia de fabricação de produtos prontos a consumir, especificamente inserida no mercado de massas alimentícias. Em uma perspectiva regional, a empresa se situa no Estado do Pará e ainda busca uma maior representatividade no mercado Paraense, pois ainda não completou 6 anos no Estado, tendo como concorrentes empresas que já possuem um bom posicionamento no mercado de Alimentos. Para concorrer neste mercado a empresa utiliza como estratégia o baixo custo dos seus produtos ao consumidor além da qualidade embutida, buscando atender principalmente às classes C e D na venda de seus produtos.

Tendo como principal objetivo atender a seus clientes de forma satisfatória e se tentando excluir as possibilidades de não atender a demanda pelos produtos, a previsão de demanda se torna uma importante ferramenta na gestão de estoques da empresa e deve, certamente, influenciar em importantes tomadas de decisão.

3.2 METODOLOGIA

O portfólio de produtos da fábrica engloba 35 produtos entre massas e biscoitos. Deste modo, para o estudo temporal da demanda foi realizada a classificação ABC (Figura 3) com a receita gerada em 2010 visando à identificação do produto que mais contribuiu para o faturamento e realização do estudo com seus dados. A técnica identificou um tipo de espagete econômico (doravante denominado “Produto X”) como responsável por 30% da receita de 2010.

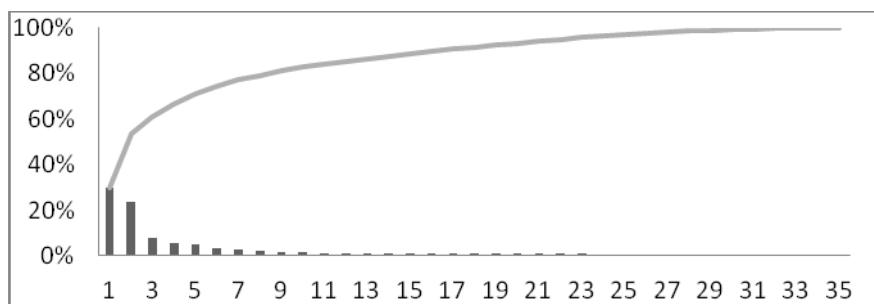


Figura 3 – Classificação ABC realizada na empresa

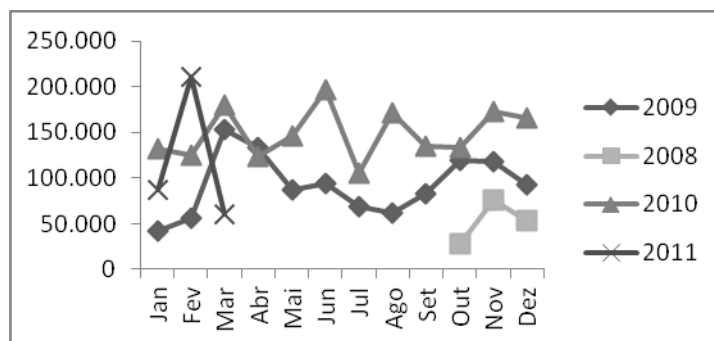


Figura 4 – Série empilhada da venda do Produto X

O estudo da demanda do Produto X será com base em seu histórico de caixas vendidas (dado no gráfico da Figura 4) desde o início de sua fabricação pela empresa (em Outubro de 2008) até séries mais recentes à época da coleta de dados (Março de 2011), compondo um total de 30 observações. A análise do gráfico não permite a visualização de componente de sazonalidade bem definido de modo que se sabe de antemão que técnicas específicas para este componente terão desempenho fraco. Percebe-se também uma leve tendência de alta.

Pela possibilidade de haver padrões não identificados, de modo que as técnicas RNA e SVR serão testadas juntamente com técnicas estruturais de previsão. Os resultados serão obtidos conforme os seguintes procedimentos:

- 1) Métodos Estruturais: média móvel, suavização exponencial e dupla.
- 2) Rede Neural Artificial: Será utilizada uma rede *feedforward* com 1 neurônio na camada de *input*, 20 neurônios na camada intermediária (com função tangencial de ativação) e 1 neurônio na camada de saída (com função linear), treinada com BP com taxa de aprendizado de 0,05. O código será desenvolvido na linguagem *matlab* na plataforma R2010a utilizando-se as 20 primeiras séries para treino da rede e as restantes para teste.
- 3) Regressão por Vetores de Suporte: será utilizado o algoritmo de Parrella (2007) com Kernel RBF e conjunto de treino e teste similar à RNA. Para a definição dos parâmetros C , ε e σ do modelo será utilizada a otimização por enxame de partículas utilizando $c_1 = c_2 = 2,05$, $I = 0,5$ e 1000 iterações possuindo como função objetivo o MAD, conforme Fei, Miao e Liu (2009). O código será desenvolvido na linguagem C++ e executado no interpretador G++ (Ubuntu).

Obtidos os resultados das 5 técnicas, estas terão seus MAD comparados visando a escolha da que menos desviou-se do observado (considerando apenas o conjunto de teste da RNA e da SVR). Então, esta técnica terá seus resultados submetidos ao gráfico de controle e, caso aprovada, recomendada para utilização pela empresa. Obviamente, após a escolha da melhor técnica, serão tecidos comentários sobre as performances gerais.

3.3 RESULTADOS E ANÁLISES

A Tabela 2 apresenta os valores reais de demanda, os resultados gerados pelas técnicas e seus respectivos MAD (“MMS” para média móvel simples, “HOLT” para o Método de Holt, “SES” para suavização exponencial simples, “SVR” para a regressão por vetores de suporte e “RNA” para a rede neural). Para a média móvel exponencial, foi encontrado $n^* = 7$ como ótimo, para a suavização exponencial simples, foi encontrado $\alpha^* = 0,317$ e para o Método de Holt foram encontrados os parâmetros $\alpha^* = 0,227$ e $\beta^* = 0,032$. O melhor resultado PSO após 10 testes encontrou para a regressão por vetores de suporte $C = 103.27,5$, $\varepsilon = 105,454$ e $\sigma = 0,00478$ equivalente a um MAD de treino igual a 24.778,3.

Ano	Mês	REAL	MMS	HOLT	SES	SVR	RNA
2008	Out	27.832	-	-	-	-	-
	Nov	75.749	-	-	27.832	-	-
	Dez	52.937	-	41.508	43.023	-	-
2009	Jan	42.627	-	45.190	46.166	-	-
	Fev	56.694	-	44.979	45.044	-	-
	Mar	153.236	-	48.820	48.737	-	-
	Abr	133.585	-	79.222	81.865	-	-
	Mai	86.749	77.523	96.254	98.261	-	-
	Jun	94.399	85.940	95.534	94.612	-	-
	Jul	68.959	88.604	97.120	94.544	-	-
	Ago	61.642	90.893	90.983	86.433	-	-
	Set	82.772	93.609	84.262	78.574	-	-
	Out	119.217	97.335	85.232	79.905	-	-
	Nov	118.022	92.475	96.314	92.368	-	-
	Dez	92.825	90.251	104.190	100.500	-	-
2010	Jan	131.450	91.119	102.817	98.067	-	-
	Fev	125.608	96.412	112.761	108.650	-	-
	Mar	179.355	104.505	118.450	114.026	-	-
	Abr	123.729	121.321	137.968	134.737	-	-
	Mai	146.588	127.172	136.573	131.247	-	-
	Jun	197.067	131.082	141.976	136.110	125.514	314.290
	Jul	105.654	142.375	160.331	155.435	126.862	351.180
	Ago	171.213	144.207	147.841	139.653	125.272	416.260
	Set	135.395	149.888	157.147	149.658	126.978	388.280
	Out	133.161	151.286	153.780	145.137	134.846	350.600
	Nov	173.034	144.687	150.545	141.340	132.325	280.980
	Dez	165.982	151.730	159.433	151.388	133.001	342.630
2011	Jan	86.910	154.501	163.968	156.014	132.904	398.620
	Fev	210.787	138.764	144.699	134.107	129.822	383.430
	Mar	59.856	153.783	165.609	158.416	130.612	379.230
MAD		-	31.830	32.188	33.426	42.021	216.644

Fonte: Resultados obtidos

Tabela 2 – Comparação entre as técnicas

Da tabela pode-se inferir que a média móvel simples gerou menores erros, seguida do Modelo de Holt. Ao contrário da premissa inicialmente adotada, os métodos SVR e RNA não geraram erros menores, muito pelo contrário no caso da RNA. Além da ausência de um padrão distinguível na série, podem ser especuladas várias razões para a

performance sensivelmente inferior de SVR e RNA. Para a RNA, talvez, a arquitetura de rede *feedforward* não seja a mais adequada, podendo a rede cíclica obter melhores resultados. Para a SVR a utilização do PSO para otimização dos parâmetros pode ter gerado um *overfitting* no conjunto de treinamento, ou mesmo o tipo de *Kernel* utilizado não seja o mais adequado.

Apesar de ter gerado os menores erros, a MMS gerou MAD relativamente alto e um MAPE de 28,9%, comprovando a alta variabilidade dos dados e a ineficiência de modelos de séries temporais, sendo mais recomendável a empresa investigar um modelo correlacional, onde poderia também utilizar uma RNA com um número maior de neurônios na camada de entrada. Diante de uma provável dificuldade de obtenção de dados para realização de previsões correlacionais, a Figura 5 apresenta o gráfico de controle do 4 MAD da MMS (LSC = 127320 e LIC = -127320), que demonstra que a técnica de previsão está sobre controle e pode ser utilizada pela empresa.

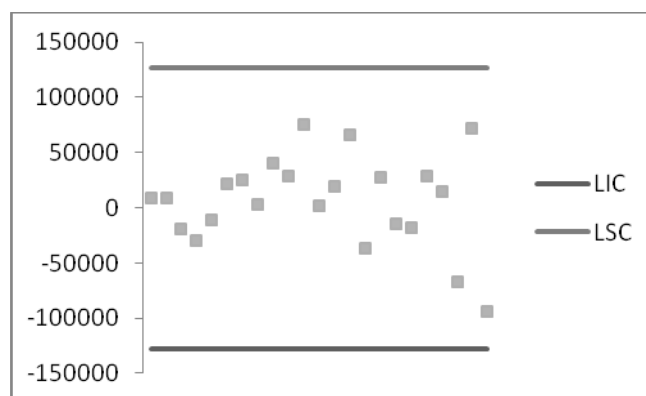


Figura 5 – Gráfico de controle 4 MAD

4. CONCLUSÃO

Comparação entre técnicas estatísticas e de inteligência computacional para a realização de previsão de demanda em um produto de alta variabilidade não comprovou a hipótese inicialmente suposta de que as técnicas de inteligência computacional obteriam melhores resultados, com a média móvel sendo a técnica testada de melhor *performance*.

Há de ser notado que os resultados gerados pela média móvel ainda deixam a desejar, visto que apresentam um MAPE próximo de 28,9%, atestando a ineficiência dos modelos de séries temporais para a previsão de vendas deste produto, devendo a empresa ou manter o resultado obtido, a custo de arcar com os erros gerados pelo método, ou realizar um estudo mais aprofundado sobre os fatores correlacionados com a demanda deste produto, visando um modelo mais preciso.

Trabalhos que venham a complementar o exposto neste artigo incluem a utilização de outros tipos de funções *Kernel* com a SVR, teste com RNA de arquitetura cíclica com mais camadas de *input*, para variáveis correlacionadas com a demanda do produto.

REFERÊNCIAS

- [1] ABELÉM, A. **Redes neurais artificiais na previsão de séries temporais**. Rio de Janeiro: PUC-Rio, 1994. Dissertação (Mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, Rio de Janeiro, 1994.
- [2] ASSOCIAÇÃO BRASILEIRA DAS INDÚSTRIAS DE ALIMENTAÇÃO (ABIA). **Banquete**. 2010. Disponível em: <<http://www.abia.org.br/anexos/a1baf22-9b65-4190-88d2-a0bcd8d322d1.pdf>>. Acesso em: 10 abr. 2011.

- [3] BRAGA, A.; LUDERMIR, T.; CARVALHO, A. **Redes neurais artificiais: teoria e prática**. Rio de Janeiro: LTC, 2000.
- [4] CORTES, C.; VAPNIK, V. Support vector networks. **Machine Learning**, v.20, p. 273-297, 1995.
- [5] FEI, S.; MIAO, Y.; LIU, C. Chinese grain production forecasting method based on particle swarm optimization-based support vector machine. **Recent Patents in Engineering**, v. 3, p. 8-12, 2009.
- [6] GOMES, D. **Modelos de redes neurais recorrentes para previsão de séries temporais de memórias curta e longa**. Campinas: UNICAMP, 2005. Dissertação (Mestrado) – Universidade Estadual de Campinas, Instituto de Matemática, Estatística e Computação Científica, Campinas, 2005.
- [7] HAYKIN, S. **Redes neurais: princípios e prática**. Porto Alegre: Bookman, 2001.
- [8] MESQUITA, M. **Previsão de demanda**. In: LUSTOSA, L. *et al.* Planejamento e controle da produção. Rio de Janeiro: Elsevier, 2008.
- [9] PARRELLA, F. **Online Support Vector Regression**. Genoa: University of Genoa, 2007. Dissertação (Mestrado) – University of Genoa, Department of Information Science, Genoa, 2007.
- [10] RAGSDALE, C. **Modelagem e análise de decisão**. São Paulo: Cengage Learning, 2009.
- [11] RUSSEL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. New York: Prentice-Hall, 1995.
- [12] SILVER, E.; PYKE, D.; PETERSON, R. **Inventory management and production planning and scheduling**. New York: John Wiley & Sons, 1998.
- [13] SLACK, N.; CHAMBERS, S.; JOHNSTON, R. **Administração da produção**. São Paulo: Atlas, 2009.
- [14] SMOLA, A.; SCHOLKOPF, B. **A tutorial on support vector regression**. NeuroCOLT2 Technical Report Series: London, 1998. Report no. NC2-TR-1998-030.
- [15] TAY, F.; CAO, L. Application of support vector machines in financial time series forecasting. **Omega**, v. 31, p. 309-317, 2001.
- [16] TUBINO, D. **Manual de Planejamento e controle da produção**. São Paulo: Atlas, 2000.
- [17] TRAFALIS, T.; INCE, H. Support vector machine for forecasting and applications to financial forecasting. In: IJCNN '00 Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN'00), 6, 2000, Washington. **Anais...** IJCNN '00 Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, 2000.
- [18] VAPNIK, V. **The nature of statistical learning theory**. New York: Springer-Verlag, 1995
- [19] VRIENS, A.; VERSTEIJNEN, E. **Forecasting and planning in the food industry: a recipe to make it light!**. EyeOn: Aarle-Rixtel, 2006.