

**Universidade Federal de Itajubá
Programa de Pós-Graduação em
Engenharia de Produção e Gestão**

**DELINEAMENTO DE EXPERIMENTOS NO
TREINAMENTO DE REDES NEURAIS ARTIFICIAIS
PARA O PROBLEMA DE PREVISÃO DE SÉRIES
TEMPORAIS NÃO LINEARES**

José Euclides Fernandes Giglio

Orientador: Pedro Paulo Balestrassi

Itajubá, 2021

Universidade Federal de Itajubá
Programa de Pós-Graduação em
Engenharia de Produção e Gestão

José Euclides Fernandes Giglio

DELINEAMENTO DE EXPERIMENTOS NO TREINAMENTO DE
REDES NEURAS ARTIFICIAIS PARA O PROBLEMA DE PREVISÃO
DE SÉRIES TEMPORAIS NÃO LINEARES

Dissertação submetida ao Programa de Pós-Graduação
em Engenharia de Produção e Gestão como parte dos
requisitos para obtenção do Título de Mestre em
Ciências em Engenharia de Produção

Área de concentração: Engenharia de Produção

Orientador: Pedro Paulo Balestrassi

Itajubá, 2021

DEDICATÓRIA

À minha esposa, Vera Lucia Pedroso Chinigo Giglio, por todo amor, apoio, incentivo, compreensão e inspiração durante o período de dedicação a este programa.

AGRADECIMENTOS

Aos meus pais, já em outra dimensão, Idelmo Giglio e Evani Fernandes Giglio, por sempre acreditarem em mim e terem proporcionado meu ingresso, graduação e pós-graduações nesta Universidade Federal de Itajubá desde a época de Escola Federal de Engenharia de Itajubá; onde estiverem recebam meu eterno amor.

Ao meu orientador, Pedro Paulo Balestrassi, não apenas por cumprir seu papel de orientador, mas também por compartilhar todo seu conhecimento, pela amizade, por acreditar em mim e me apoiar incondicionalmente e incansavelmente em todo o processo de formação atual e nesta pesquisa.

Aos demais professores que compartilharam comigo os seus conhecimentos neste programa, alguns desde a época de minha graduação, João Batista Turrioni, José Arnaldo Barra Montevechi e Edson de Oliveira Pamplona, outros contemporâneos da nossa graduação e que se tornaram meus professores, Carlos Henrique Pereira Mello e Carlos Eduardo Sanches da Silva e aos meus novos professores, que pude conhecer durante este programa, Anderson Paulo de Paiva, Jose Henrique de Freitas Gomes e André Luiz Medeiros.

Aos meus colegas de programa, especialmente aos amigos Carlos Henrique dos Santos, Natália Bianchesi, Marina Fernandes Branco Pitanga Lopes e Estevão Luiz Romão, indispensáveis para a motivação e evolução do trabalho.

À André Rodrigues Monticeli e Silvio Antonio Bueno Salgado, por compartilharem seus conhecimentos na confecção deste estudo e pesquisa.

A Juliana Helena Daroz Gaudêncio e Wesley Vieira, por comporem a banca de minha defesa e tanto colaborarem por mais conhecimento e um trabalho mais alinhado à academia.

À UNIFEI por fazer parte de minha vida desde 1988.

À Deus por me proporcionar a cada dia meu aperfeiçoamento e por me acompanhar todos os dias.

RESUMO

Uma série temporal é definida como uma coleção de observações de uma variável ao longo do tempo, cuja ordem dos dados é de fundamental importância devido a dependência entre estes valores consecutivos. A análise destes dados, e o entendimento desta correlação, é um importante instrumento no entendimento de fenômenos em diversas ciências, como Economia, Engenharias e Gestão de Operações, onde preços, demandas e valores são estas variáveis. A modelagem desta sequência de dados proporciona a sua utilização no objetivo de, com base nos dados históricos, realizar previsões para períodos futuros. Esta relação consecutiva pode ser considerada complexa e, não incomum, não lineares. O uso de Redes Neurais Artificiais tem se provado cada vez mais eficaz em estabelecer reconhecimento de padrões, modelagem e a previsão de valores futuros. Os programas estatísticos disponíveis no mercado disponibilizam ferramentas de uso amigável e de resultados demonstrados em diversos científicos disponíveis em publicações, porém o número de fatores e níveis que são disponibilizados para utilização durante o treinamento das Redes Neurais Artificiais, o que pode nos apontar a necessidade de centenas de anos para executarmos todas as combinações possíveis. Neste estudo a metodologia estatística de Delineamento de Experimentos (*Design of Experiments* - DOE) é aplicada com o propósito de determinar os melhores parâmetros de uma Rede Neural Artificial para a previsão de séries temporais não lineares e, assim, reduzir significativamente o tempo necessário para se apontar a escolha da melhor Rede Neural Artificial capaz de resolver nosso problema de previsão. Ao invés de utilizarmos a técnica mais comum de treinamento de uma Rede Neural Artificial, ou seja, o método empírico, o DOE é proposto para ser a melhor metodologia. A principal motivação para esta dissertação foi a previsão de séries temporais sazonais não lineares - que está relacionada com muitos problemas reais, tais como carga elétrica de curto prazo, preços diários e retornos, consumo de água, etc. Um estudo de caso é apresentado. O objetivo foi cumprido quando se comprovou atingir resultados de erros, entre previsão e valor real, menores para a Rede Neural Artificial do que o erro alcançado com o modelo.

Palavras-Chave: Previsão, delineamento de experimentos, Rede Neural Artificial, séries temporais não lineares, DOE.

ABSTRACT

A time series is defined as a collection of observations of a variable over time, whose data order has a fundamental importance due to the dependence between these consecutive values. The analysis of these data, and the understanding of this correlation, is an important tool in understanding phenomena in various sciences, such as Economics, Engineering and Operations Management, where prices, demands and values are these variables. The modeling of this data sequence provides its use in order to, based on historical data, make predictions for future periods. This consecutive relationship can be considered complex and, not uncommonly, non-linear. The use of Artificial Neural Networks has proven increasingly effective in establishing pattern recognition, modeling and predicting future values. The statistical programs available on the market provide user-friendly tools and results demonstrated in several scientific available in publications, but the number of factors and levels that are available for use during the training of Artificial Neural Networks, which may indicate the need for hundreds of years to execute every possible combination. In this study, the statistical methodology of Design of Experiments (DOE) is applied in order to determine the best parameters of an Artificial Neural Network for the prediction of non-linear time series and, thus, significantly reduce the time needed to point out the choice of the best Artificial Neural Network capable of solving our prediction problem. Instead of using the most common technique for training an Artificial Neural Network, that is, the empirical method, DOE is proposed to be the best methodology. The main motivation for this dissertation was the prediction of non-linear seasonal time series - which is related to many real problems, such as short-term electrical load, daily prices and returns, water consumption, etc. A case study is presented. The objective was fulfilled when it was proved to reach error results, between prediction and real value, smaller for the Artificial Neural Network than the error reached with the model.

Keywords: *Forecasting, design of experiments, artificial neural network, nonlinear time series, DOE.*

LISTA DE FIGURAS

| | | |
|-----------|--|----|
| Figura 1 | Publicações nas bases Scopus e Web of Science | 13 |
| Figura 2 | Publicações nas bases Scopus e Web of Science por década | 13 |
| Figura 3 | Uma série temporal típica sazonal de <i>Smoothing Transaction Autoregressive</i> (STAR2) | 23 |
| Figura 4 | Estrutura da rede neural artificial <i>feedforward</i> multicamadas | 26 |
| Figura 5 | Densidade espectral para o modelo STAR2 | 41 |
| Figura 6 | Média e SN para o delineamento de Taguchi (STAR2) | 52 |
| Figura 7 | Delineamento 2_{III}^{11-7} , Modelo STAR2 | 54 |
| Figura 8 | Delineamento mínimo 2_{IV}^{8-4} , Modelo STAR2 | 56 |
| Figura 9 | Delineamento mínimo 2^4 , Modelo STAR2 | 58 |
| Figura 10 | Previsão de RNA para o modelo STAR2 usando os parâmetros otimizados | 60 |
| Figura 11 | Séries temporais de cargas horárias (em Watts) de seis consumidores industriais | 62 |

LISTA DE QUADROS

| | | |
|----------|---|----|
| Quadro 1 | Uma coleção de modelos não lineares para séries temporais | 23 |
| Quadro 2 | Seleção de fatores para o treinamento da RNA para previsão de séries temporais não lineares | 39 |
| Quadro 3 | Escolha dos fatores da Rede Neural Artificial para séries temporais não lineares | 59 |
| Quadro 4 | Parâmetros da RNA para os seis consumidores industriais de carga de eletricidade | 63 |

LISTA DE TABELAS

| | | |
|----------|--------------------------------------|----|
| Tabela 1 | Arranjo cruzado de Taguchi | 51 |
| Tabela 2 | Delineamento 2_{III}^{11-7} | 53 |
| Tabela 3 | Delineamento mínimo 2_{IV}^{8-4} | 55 |
| Tabela 4 | Delineamento fatorial completo 2^4 | 57 |

SUMÁRIO

| | | |
|-------|--|----|
| 1 | Introdução | 11 |
| 1.1 | Contexto | 11 |
| 1.2 | Relevância | 12 |
| 1.3 | Justificativa | 13 |
| 1.4 | Objetivo geral | 14 |
| 1.5 | Estrutura do trabalho | 15 |
| 2 | Revisão da Literatura | 16 |
| 2.1 | Séries Temporais Não Lineares | 16 |
| 2.2 | Redes Neurais Artificiais e o Problema de Previsão de Séries Temporais | 24 |
| 2.3 | DOE para o Treinamento de Redes Neurais Artificiais para Previsão de Séries Temporais Não Lineares | 31 |
| 3 | Delineamento Experimental | 35 |
| 3.1 | Procedimentos Gerais | 35 |
| 3.2 | Planejamento Pré Experimental | 38 |
| 3.3 | Delineamentos e Resultados | 48 |
| 3.3.1 | Delineamento Taguchi de triagem para 19 fatores | 48 |
| 3.3.2 | Delineamento 2_{III}^{11-7} Fatorial Fracionado para 11 Fatores | 52 |
| 3.3.3 | Delineamento 2_{IV}^{8-4} Fatorial Fracionado para 8 Fatores | 54 |
| 3.3.4 | Delineamento 2^4 Fatorial Completo para 8 Fatores | 57 |
| 3.4 | Testes de Confirmação | 59 |
| 4 | Estudo de Caso: consumo de eletricidade de curto prazo | 61 |
| 5 | Conclusões e Trabalhos Futuros | 64 |
| | Referências Bibliográficas | 66 |

1 Introdução

1.1 Contexto

Um dos maiores problemas na ciência é realizar uma previsão (como prever o futuro a partir de conhecimentos passados). A análise de séries temporais é uma área de pesquisa fértil em diversos campos do conhecimento, principalmente nas áreas de engenharia e economia. É de grande interesse o estudo de métodos de previsão de séries temporais, ou seja, conseguir identificar e prever algumas características do processo num ponto futuro. As condições presentes determinam, em algum grau, resultados do futuro, possivelmente envolvendo relações complexas entre variáveis.

Uma situação ideal para a realização de previsões seria o conhecimento das equações que modelam os mecanismos responsáveis pela geração de séries temporais. No entanto, em muitos problemas reais essas informações não estão disponíveis ou não se tem condições ideais para construir equações que governem o comportamento das variáveis de interesse. Técnicas de programação elaboradas, como, por exemplo, inteligência artificial, vem sendo empregadas na resolução de novos e antigos problemas, pois fornecem soluções que apresentam melhores resultados na execução da tarefa, trazendo economia de tempo e de recursos computacionais. (Balestrassi *et al.*, 2009). Redes Neurais Artificiais vêm desempenhando um papel crescente nos últimos anos na abordagem e solução de problemas estatísticos importantes. Sua contribuição se dá principalmente em problemas de reconhecimento de padrões e problemas de previsão.

Em consonância, o *Design of Experiments* (DOE) é considerada uma importante metodologia para pesquisadores que lidam com aplicações práticas, com um vasto histórico de sucessos. Nos dias atuais, os recursos do DOE estão incorporados em muitos pacotes de programas estatísticos que facilitam os cálculos e as interpretações dos resultados. Similarmente, Redes Neurais Artificiais desempenham um papel importante no problema de previsão de séries temporais. Zhang, Patuwo e Hu (1998) afirmam que uma desvantagem frequente em usar Redes Neurais Artificiais em problemas de previsão é a escolha dos melhores parâmetros para a Rede Neural Artificial. Em geral, o processo de tentativa e erro é muito demorado e é o que a maioria dos praticantes usa para essa otimização, conforme. Shin *et al.* (2016) afirmam que o método empírico ainda é empregado nos dias atuais.

Em geral, os métodos de previsão partem de valores passados para prever os valores futuros. Até 1920, a previsão era efetuada através de extrapolação simples de um valor global, ajustado em função do tempo. Só em 1927, Yule criou o modelo Autorregressivo (AR), onde o valor previsto dependia de valores passados. Por duas décadas seguintes, os pesquisadores da área de estatística, acreditaram que o comportamento de uma série dependia dos modelos lineares e do ruído. Com o passar do tempo, porém, percebeu-se que modelos lineares não eram suficientes para a análise de séries temporais, já que a maioria das séries reais apresentam fortes tendências de não linearidade (KIM, 2003).

A partir de 1950, novos estudos foram realizados na área da previsão de séries temporais. Nesta época foi criada a técnica do Alisamento Exponencial, que exigia poucos cálculos e era de fácil utilização. Com o advento do computador nos anos 50, houve uma revolução nos métodos de previsão. Em 1970, surgiu a metodologia Box-Jenkins, que apresentava uma técnica completa de identificação do modelo que melhor se ajustava aos dados. Tal metodologia consiste em propor e ajustar modelos lineares estacionários (ou não-estacionários) a uma série de tempo observada.

Desde então, muitas técnicas têm sido testadas com o intuito de se efetuar previsões mais precisas e, preferencialmente, automatizadas. Muitos estudos na área da Inteligência Artificial (IA) foram dirigidos a estes modelos inspirados na natureza, como Redes Neurais Artificiais (RNA), Algoritmos Genéticos (AG) e Algoritmos Evolutivos (AE) e, dentre estes, a programação genética (KABOUDAN, 2001); todos apresentaram resultados promissores nesta área.

1.2 Relevância

Encontra-se na literatura científica diversos trabalhos abordando a análise de séries temporais com base em métodos estatísticos, tanto clássicos quanto aplicações de Redes Neurais Artificiais. Realizando-se uma análise bibliométrica nas bases científicas Scopus e *Web of Science* através das *palavras forecasting, design of experiments, artificial neural network, nonlinear time series*, DOE encontra-se milhares de artigos e *conference papers*. A Figura 1 nos ajuda a perceber claramente o crescente interesse pelo tema.

Nesta década já encontramos quantidade de publicações igual a 33% da última década e 25% se somarmos as décadas anteriores, como podemos evidenciar na Figura 2.

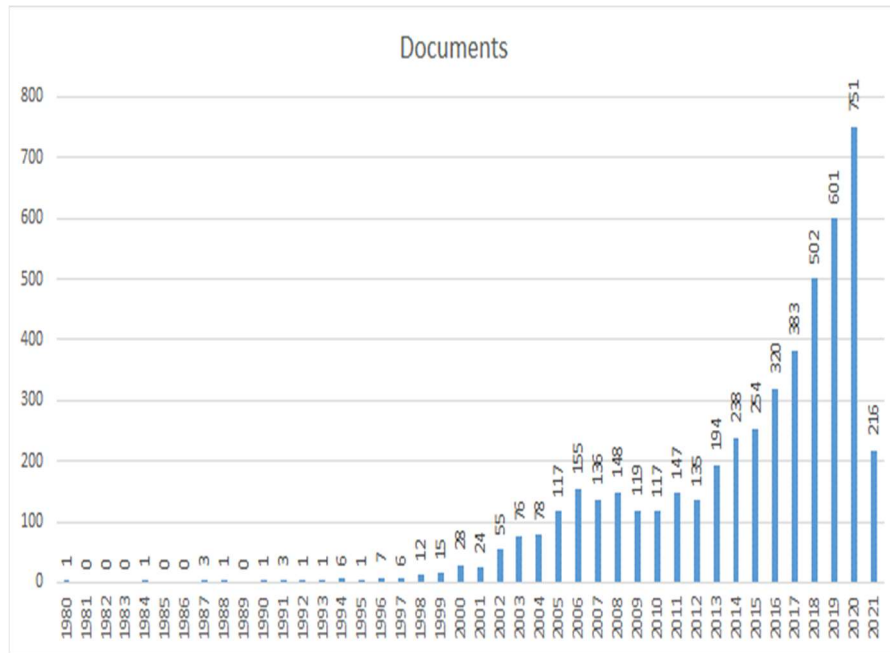


Figura 1 – Publicações nas bases Scopus e Web of Science

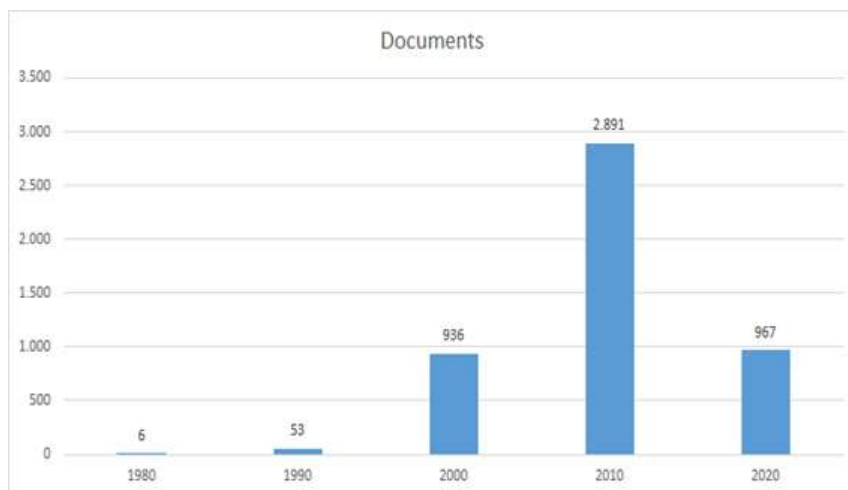


Figura 2 – Publicações nas bases Scopus e Web of Science por década

1.3 Justificativa

A previsão de demanda representa um grande impacto nas indústrias; o ramo aeronáutico possui diretorias dedicadas à área de previsão de mercado, envolvendo não apenas o seu próprio portfólio como o de concorrentes visando garantir o balanceamento ideal de estoques e a impossibilidade de desabastecimento provocado por falhas nas previsões.

O uso do tempo t de observações disponíveis de uma série temporal para prever seu valor em algum tempo futuro $t + 1$ pode fornecer uma base para planejamento econômico e de negócios, planejamento de produção, estoque e controle de produção, e controle e otimização de processos industriais. Conforme descrito originalmente por Holt *et al.* (1963), Brown (1962) e a monografia da *Imperial Chemical Industries* sobre previsão de curto prazo (OLIVER; BOYD, 1964), as previsões são geralmente necessárias durante um período conhecido como lead time, que varia com cada problema. Por exemplo, o lead time no problema de controle de estoque foi definido por Harrison (1965) como um período que começa quando um pedido de reposição de estoque é colocado na fábrica e dura até que o pedido seja entregue ao estoque (BOX; JENKINS; REINSEL, 2008).

O uso de Redes Neurais Artificiais, cada vez mais comum em nosso cotidiano, tem sempre como objetivo o de resolver problemas complexos em diversas situações. Elas podem aprender e modelar relações de entrada e saída de dados, além de complexos, não lineares. Não há métodos estruturados hoje para identificar qual estrutura de rede pode melhor aproximar a função, mapeando as entradas para as saídas. Conseqüentemente, os tediosos experimentos e procedimentos de tentativa e erro são frequentemente usados (ZHANG, 2003)

A integração de redes neurais e otimização fornece uma ferramenta para projetar parâmetros de rede e melhorar o seu desempenho. No artigo: *Optimum design for artificial neural network* (LIN; TSENG, 2000), o método Taguchi e a metodologia DOE são usados para otimizar os parâmetros da rede. Os usuários devem reconhecer os problemas da aplicação e escolher um modelo adequado de Rede Neural Artificial. Os problemas de otimização podem então ser definidos de acordo com o modelo. O método Taguchi é primeiro aplicado a um problema para descobrir os fatores mais importantes, então a metodologia DOE é usada para análises e previsões posteriores (KOHZADI; BOYD; KERMANSHAHI; KAASTRA, 1996)

Aplicar a metodologia Taguchi pode ser a forma de minimizar o procedimento de tentativa e erro e reduzir o tempo de duração dos experimentos.

1.4 Objetivos

1.4.1 Objetivo geral

O objetivo deste trabalho é comparar o desempenho de redes neurais artificiais treinadas por DOE, com uso do software Statistica, com o obtido pelos oito modelos de séries temporais não lineares apresentados, através do erro MAPE obtido por ambos.

1.4.2 Objetivos específicos

De modo a cumprir com o objetivo geral desse trabalho, surgem os seguintes objetivos específicos:

- Aplicar o DOE para triagem dos 24 fatores existentes para treinamento de Redes Neurais no software Statistica, estabelecendo quais são os fatores relevantes, e seus respectivos níveis, para o desempenho da rede;
- Calcular do erro MAPE, obtido para cada uma das oito séries temporais, utilizando-se os fatores e níveis triados anteriormente;
- Calcular o erro MAPE dos oito modelos de séries temporais;
- Comparar os erros calculados anteriormente para cada uma das oito séries temporais;
- Aplicar a mesma metodologia para um estudo de caso real com seis séries temporais de modo a reforçar a metodologia utilizada.

1.5 Estrutura do trabalho

Este trabalho está estruturado conforme segue.

Na seção 1 o trabalho apresenta a contextualização do assunto que será abordado, seguida pela justificativa e relevância.

Na seção 2, encontra-se a revisão da literatura e sua fundamentação teórica, onde são abordados os assuntos necessários para o desenvolvimento da pesquisa. Em seu início aborda-se o problema de previsão de séries temporais não lineares, seguido pelo tema de redes neurais artificiais e o problema de previsão de séries temporais, complementa-se com o DOE no treinamento das redes neurais artificiais para previsão de séries temporais não lineares.

Na seção 3 encontramos o delineamento experimental e os testes de confirmação.

Na seção 4 apresenta-se um estudo de caso exemplificando a aplicação da metodologia proposta.

E, por fim, a seção 5 apresenta as considerações finais e sugestões para trabalhos futuros.

2 Revisão da Literatura

2.1 Séries Temporais Não Lineares

Nesta seção, introduziremos algumas ideias básicas de séries temporais, enfatizando os chamados modelos não lineares.

Uma série temporal é uma coleção de observações feitas sequencialmente ao longo do tempo. A característica mais importante deste conjunto de dados é que as observações vizinhas são dependentes (BOX; JENKINS; REINSEL, 2008) e estamos interessados em analisar e modelar esta dependência.

Como a maior parte dos procedimentos estatísticos foram desenvolvidos para analisar observações independentes, o estudo de séries temporais requer o uso de técnicas específicas. Os modelos utilizados para descrever séries temporais são processos estocásticos, isto é, processos controlados por leis probabilísticas.

Uma série temporal é dita contínua quando as observações são feitas continuamente no tempo.

Definindo o conjunto $T = \{t; t_1 < t < t_2\}$ a série temporal será denotada por $\{y_t; t \in T\}$.

Uma série temporal é dita discreta quando as observações são feitas em tempos específicos, geralmente equi-espaçados. Definindo o conjunto $T = \{t_1, \dots, t_n\}$ a série temporal também será denotada por $\{y_t; t \in T\}$. Por simplicidade, podemos adotar $T = \{1, 2, \dots, n\}$. Muitas vezes uma série temporal discreta é obtida através da amostragem da série temporal contínua em intervalos de tempos iguais, Δt . Uma série temporal também pode ser multivariada. Se k variáveis são observadas a cada tempo (por exemplo discreto), denota-se por $\{y_{1t}, \dots, y_{kt}; t \in T\}$. Nesse caso várias séries correlacionadas devem ser analisadas conjuntamente, ou seja, em cada tempo tem-se um vetor de observações.

Existem, basicamente, dois enfoques usados na análise de séries temporais. Em ambos, o objetivo é construir modelos para as séries com propósitos determinados. Um dos enfoques é a análise no domínio temporal e os modelos propostos são modelos paramétricos (com um número finito de parâmetros). Já no outro enfoque, a análise é conduzida no domínio de frequências e os modelos propostos são modelos não paramétricos.

Quando lidamos com séries temporais, em alguns casos precisamos fazer previsões de valores futuros enquanto em outros, entender a estrutura da série ou sua relação com outras séries pode ser o interesse principal. Os principais objetivos no estudo de séries temporais são (ULLAH, 2014):

- Descrição. Descrever propriedades da série, por exemplo, o padrão de tendência, existência de variação sazonal ou cíclica, observações discrepantes (*outliers*), alterações estruturais (mudanças no padrão da tendência ou da sazonalidade) etc.;
- Explicação. Usar a variação em uma série para explicar a variação em outra série;
- Predição. Predizer valores futuros com base em valores passados. Aqui assume-se que o futuro envolve incerteza, ou seja, as previsões não são perfeitas. Porém devemos tentar minimizar os erros de previsão;
- Controle. Os valores da série temporal medem a qualidade de um processo de manufatura e o objetivo é o controle do processo. Um exemplo é o controle estatístico de qualidade onde as observações são representadas em cartas de controle.

O primeiro passo para análise de séries temporais é a escolha de um modelo matemático adequado para os dados. Os modelos utilizados para descrever séries temporais são processos estocásticos. Sendo T um conjunto arbitrário, um processo estocástico é uma família de variáveis aleatórias $\{X_t; t \in T\}$ definidas sobre o mesmo espaço de probabilidades $(\Omega; A; P)$.

Convém observar que para cada $t \in T$, X_t é uma variável aleatória definida sobre Ω , o que permite concluirmos que X_t é uma função de dois argumentos, $X(t; \omega)$, $t \in T$ e $\omega \in \Omega$.

Um processo estocástico $\{X_t; t \in T\}$ é chamado de processo contínuo (ou processo com parâmetro contínuo) se o conjunto T for um intervalo de números reais. Caso o conjunto T seja enumerável dizemos que o processo estocástico é um processo discreto (ou processo com parâmetro discreto).

As funções $\{X_t(\omega); \omega \in \Omega\}$ são chamadas de realizações ou trajetórias ou ainda de séries temporais.

Intuitivamente, uma série temporal $\{y_t; t = 0, \pm 1, \dots\}$ é dita estacionária se ela possui propriedades estatísticas similares às da série de "tempo deslocado" $\{y_{t+h}; t = 0, \pm 1, \dots\}$, para cada inteiro h . Restringiremos nossa atenção àquelas propriedades que dependem somente do primeiro e do segundo momento de $\{y_t\}$. Se $\{y_t\}$ é uma série temporal com $E(y_t^2) < \infty$, então definimos a função média de $\{y_t\}$ por

$$\mu(t) = E(y_t) \quad (1)$$

e a função de covariância de $\{y_t\}$ por

$$\gamma(r; s) = Cov(y_r; y_s) = E[(y_r - \mu(r))(y_s - \mu(s))] \quad (2)$$

para quaisquer que sejam os inteiros r e s .

A série temporal $\{y_t\}$ é dita fracamente estacionária se

- i) $\mu(t)$ é independente de t ;
- ii) $\gamma(t + h, t)$ é independente de t para cada h .

Dizer que a série temporal $\{y_t; t = 0; \pm 1; \pm 2; \dots\}$ é estritamente estacionária, equivale dizer que $(y_1; \dots; y_n)$ e $(y_{1+h}; \dots; y_{n+h})$ possuem as mesmas distribuições para qualquer inteiro h e $n > 0$. Sempre que usarmos o termo estacionária, estaremos nos referindo a uma série temporal fracamente estacionária.

Estacionaridade em uma série temporal implica um tipo de equilíbrio estatístico ou estabilidade nos dados da série. Conseqüentemente a série temporal apresenta uma média constante.

Em vista da condição (ii), sempre que usarmos o termo função de covariância para uma série temporal estacionária $\{y_t\}$, a função $\gamma(\cdot)$ pode ser definida por

$$\gamma(h) := \gamma(h, 0) = \gamma(t + h; t) \quad (3)$$

A função $\gamma(\cdot)$ será denominada função de autocovariância.

Seja $\{y_t\}$ uma série temporal estacionária. A função de autocovariância (ACVF) de $\{y_t\}$ é definida por:

$$\gamma(h) = \text{Cov}(y_{t+h}; y_t) \quad (4)$$

A função de Autocorrelação (ACF) de $\{y_t\}$ é definida por:

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \text{Cor}(y_{t+h}; y_t) \quad (5)$$

Se $\{y_t\}$ são ruídos independentes e identicamente distribuídos (i.i.d.) com $E(y_t^2) = \sigma^2 < \infty$ então $\mu(t) = E(y_t) = 0$ não depende de t . Com isso:

$$\gamma(t+h; t) = \begin{cases} \sigma^2, & h = 0 \\ 0, & h \neq 0 \end{cases} \quad (6)$$

ou seja, $\gamma(t+h, t)$ é independente de t .

Se $\{y_t\}$ é uma sequência de variáveis aleatórias cada uma com média zero e variância σ^2 , então $\{y_t\}$ é uma série temporal estacionária com a mesma função de covariância do exemplo anterior. Tal sequência é denominada ruído branco ou *white noise* (com média zero e variância σ^2) e denotada por:

$$\{y_t\} \sim WN(0, \sigma^2) \quad (7)$$

Se $\gamma(\cdot)$ é a função de autocovariância da série $\{y_t, t \in Z\}$, então:

1. $\gamma(0) \geq 0$
2. $|\gamma(h)| \leq \gamma(0)$ para todo $h \in Z$
3. $\gamma(-h) = \gamma(h)$ para todo $h \in Z$

Embora saiba-se calcular a função de Autocorrelação para modelos simples de séries temporais, em situações práticas, não se inicia com um modelo, mas sim com um conjunto de observações $\{y_1, \dots, y_T\}$. Para avaliarmos o grau de dependência dos dados e selecionarmos um modelo para os mesmos, nós precisaremos estimar a função (ACF).

Sendo y_1, \dots, y_T observações de uma série temporal. Definimos

i) A média da amostra y_1, \dots, y_T :

$$\bar{y} = \frac{1}{T} \sum_{i=1}^T y_i \quad (8)$$

ii) A função de autocovariância da amostra:

$$c_k = \hat{c}_k = \frac{1}{T} \sum_{t=1}^{T-k} (y_t - \bar{y})(y_{t+k} - \bar{y}) \quad (9)$$

$$k = 0, 1, 2, \dots, K$$

iii) A função de autocorrelação da amostra:

$$r_k = \hat{\rho}_k = \frac{c_k}{c_0} \quad (10)$$

$$k = 0, 1, 2, \dots, K$$

Métodos de séries temporais lineares têm sido amplamente usados nas duas últimas décadas. Recentemente, contudo, tem havido um crescente interesse em estender a estrutura clássica de Box-Jenkins para incorporar estruturas fora do padrão, como, por exemplo, não

linearidades, não gaussianidades e heterogeneidade. Desta forma, um grande número de modelos não lineares tem sido desenvolvido, como o modelo bi linear de Granger e Anderson (1978), o modelo de Tong (1978) *threshold autoregressive* (TAR), o modelo dependente do estado de Priestley (1980), o modelo comutador de Markov, de Hamilton (1989), o modelo Autoregressivo dos coeficientes funcionais de Chen e Tsay (1993), entre muitos outros. Embora as propriedades desses modelos tendam a sobrepor-se ligeiramente, cada um deles é capaz de capturar uma grande variedade de comportamentos não lineares. Na maioria das séries, no entanto, esse tipo de modelagem é muito complexo, devido algumas características frequentes, tais como, sazonalidade diária e semanal, efeito calendário nos finais de semana e feriados, alta volatilidade e presença de *outliers*. Em particular, o modelo de uma Rede Neural Artificial é capaz de aproximar qualquer não linearidade, com um grau de precisão arbitrária, da mesma forma que um modelo ARMA que possui uma boa aproximação para relações lineares (GOOIJER, 1998). Esta é a chamada propriedade universal das Redes Neurais Artificiais. Em suma, a Rede Neural Artificial *feedforward* com uma camada oculta pode ser vista como uma forma de parametrizar uma função contínua não linear qualquer.

Existem muitos modelos não lineares para séries temporais. Consideremos uma série temporal univariada y_t , o qual, por simplicidade, é observada em tempos espaçados. Denotemos por $\{x_t; t = 1, \dots, T\}$ e T e o tamanho da amostra.

Uma série temporal puramente estocástica diz-se linear se pode ser escrita na forma:

$$y_t = \mu \sum_{i=1}^{\infty} \psi_i a_{t-i} \quad (11)$$

em que μ é uma constante, ψ_i são números reais, como $\psi_0 = 1$ e a_t é uma sequência de variáveis aleatórias independentes e identicamente distribuídas (i.i.d), com função de distribuição bem definida.

Nós assumimos que a distribuição de a_t é contínua e $E(a_t) = 0$. Em muitos casos, nós ainda assumimos que $Var(a_t) = \sigma^2$ ou ainda, mais fortemente, que a_t segue uma distribuição normal.

Se

$$\sigma^2 \sum_{i=1}^{\infty} \psi^2 < \infty \quad (12)$$

então y_t é fracamente estacionária, ou seja, o primeiro e segundo momentos de y_t são invariantes no tempo. O processo ARMA é linear pois tem uma representação *MA* na equação mencionada. Qualquer série temporal que não satisfaz a equação anterior se diz não linear. Para mais detalhes veja Tsay (2005).

Uma abordagem natural na modelagem de séries temporais com modelos lineares parece definir regimes, ou estados diferentes, para permitir a possibilidade de que o comportamento dinâmico de variáveis dependa do regime que ocorre em qualquer ponto no tempo (PRIESTLEY, 1980).

Por comportamento dinâmico dependente do estado de uma série temporal entende-se que certas propriedades da série temporal, tais como a sua média, variância e / ou autocorrelação, são diferentes em diferentes regimes. Em particular, as autocorrelações tendem a ser maiores durante períodos de baixa volatilidade e menor durante os períodos de elevada volatilidade. Os períodos de baixa e alta volatilidade podem ser interpretados como diferentes regimes. Claro que, o nível de volatilidade no futuro não é conhecido com certeza. O melhor que se pode fazer é uma previsão sensata desse nível e, portanto, do regime que irá ocorrer no futuro (KRAGER; KUGLE, 1993). O estado-dependente, ou regime de alternância considera que o regime é estocástico e não determinístico. Existem duas classes principais de modelos de mudança de regime:

(i) os regimes determinados por variáveis observáveis que incluem o modelo Bilinear, o modelo TAR (*Threshold Autoregressive*) e o modelo SETAR (*Self-Exciting Threshold Autoregressive*) e

(ii) regimes determinados por variáveis não observáveis que incluem o modelo MSW (Markov-Switching). Nós restringimos a nossa atenção para os modelos que assumem que, em cada um dos regimes de comportamento dinâmico da série temporal é modelado por um modelo AR.

O quadro seguinte ilustra uma coleção de modelos não lineares para séries temporais.

Quadro 1: Uma coleção de modelos não lineares para séries temporais

| Modelo | Equação |
|---|--|
| <i>Sign Autoregressive (SAR)</i> | $y_t = \text{sign}(y_t - 12) + \epsilon_t$ |
| <i>Bilinear 1 (BL1)</i> | $y_t = 0,7y_{t-1} \epsilon_{t-2} + \epsilon_t$ |
| <i>Bilinear 2 (BL2)</i> | $y_t = 0,4y_{t-1} - 0,3y_{t-2} + 0,5y_{t-1}\epsilon_{t-1} + \epsilon_t$ |
| <i>Threshold Autoregressive (TAR)</i> | $y_t = \begin{cases} 0,9y_{t-1} + \epsilon_t & \text{se } y_{t-1} \leq 1 \\ -0,3y_{t-1} - \epsilon_t & \text{se } y_{t-1} > 1 \end{cases}$ |
| <i>Nonlinear Autoregressive (NAR)</i> | $y_t = \frac{0,7 y_{t-1} }{ y_{t-1} + 2} + \epsilon_t$ |
| <i>Nonlinear Moving Average (NMA)</i> | $y_t = \epsilon_t - 0,3\epsilon_{t-1} + 0,2\epsilon_{t-2} + 0,4\epsilon_{t-1}\epsilon_{t-2} - 0,25\epsilon_{t-2}$ |
| <i>Smooth Transition Autoregressive (STAR1)</i> | $y_t = 0,8 y_{t-1} - 0,8 y_{t-1} [1 + e^{-10y_{t-1}}]^{-1} + \epsilon_t$ |
| <i>Smooth Transition Autoregressive (STAR2)</i> | $y_t = 0,3 y_{t-1} + 0,6 y_{t-24} + (0,1 - 0,9y_{t-1} + 0,8y_{t-24}) [1 + e^{-10y_{t-1}}]^{-1} + \epsilon_t$ |

Fonte: Adaptado de Balestrassi *et al.* (2009)

Na Figura 3 mostramos um gráfico típico dos modelos mencionados, o modelo STAR2 com um componente Autorregressivo e uma sazonalidade de 24.

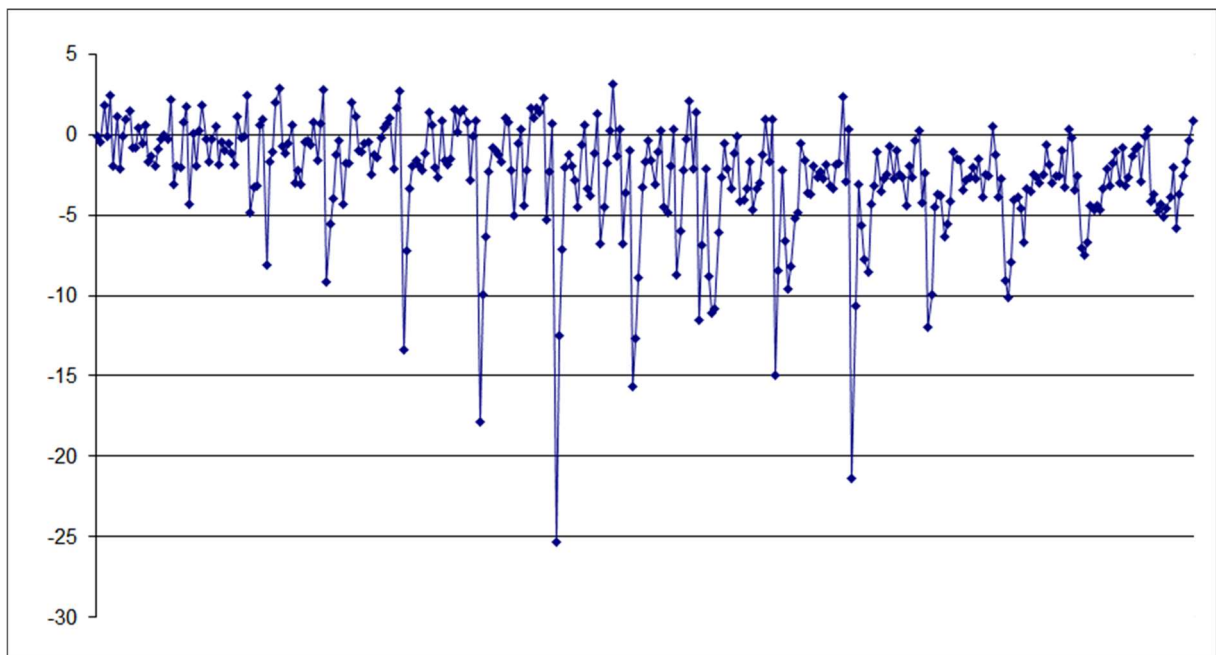


Figura 3: Uma série temporal típica sazonal de *Smoothing Transaction Autoregressive (STAR2)*

2.2 Redes Neurais Artificiais e o Problema de Previsão de Séries Temporais

As Redes Neurais Artificiais surgiram na década de 40, mais precisamente em 1943, quando o neurofisiologista Warren McCulloch e o matemático Walter Pitts, da universidade de Illinois, fizeram uma analogia entre células nervosas e o processo eletrônico num artigo publicado no *Bulletin of Mathematical Biophysics* com o título *A Logical Calculus of the Ideas Immanent in Nervous Activity*.

Em 1949 o biólogo e psicólogo Donald Hebb, que estudava o comportamento dos animais, escreveu um livro chamado *The organization of Behavior*, que reforçava as teorias de que o condicionamento psicológico estava presente em qualquer parte dos animais, pelo fato de que esta é uma propriedade de neurônios individuais. As ideias de Hebb não eram pioneiras, mas ele propôs um princípio de aprendizado em sistemas nervosos complexos, ou seja, uma lei que descreve o funcionamento quantitativo da sinapse e do processo de treinamento humano.

Desde então, vários outros pesquisadores, entusiasmados com as novas descobertas, voltaram-se para essa linha de pesquisa. Em 1951, Marvin Minsky, cofundador do laboratório de inteligência artificial do MIT, construiu o SNARC, o primeiro simulador de cadeia neural. Em 1956 na primeira conferência internacional de inteligência artificial, foi apresentado um modelo de Rede Neural Artificial pelo pesquisador de IBM Nathaniel Rochester.

Já em 1959, Frank Rosenblatt na universidade de Cornell, considerado como fundador da neuro-computação, criou uma rede de múltiplos neurônios do tipo discriminadores lineares e a batizou de Rede Perceptron.

Ainda no final da década de 50, Minsk e Seymour Papert lançaram uma obra chamada Perceptron a qual demonstrava que o modelo apresentado por Rosenblatt não era muito promissor, devido ao uso de técnicas empíricas, às grandes dificuldades da matemática envolvida e dos poucos recursos computacionais disponíveis na época.

Nos anos seguintes muitos artigos foram publicados e várias previsões exageradas e pouco confiáveis para a época foram anunciadas. A maioria dessas suposições falava de computadores com poder de raciocínio e/ou processamento igual ou superior ao do cérebro

humano. Desta forma a credibilidade de futuros estudos das redes neurais artificiais foram fortemente comprometidos.

O físico e biólogo, de reputação mundial, John Hopfield, também se interessou pela neuro-computação e escreveu vários artigos, em 1982, que levaram vários cientistas a se unirem nesta nova área emergente. Hopfield reacendeu as pesquisas em neuro-computação criticando fortemente as teorias apresentadas por Minsky e Papert na década de 50.

Este campo de pesquisa explodiu em 1986, quando o professor de psicologia de Stanford, David E. Rumelhart e seu colega James McClelland, professor de psicologia da Universidade de Carnegie Mellon, publicaram o livro *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Neste livro eles apresentaram um modelo matemático e computacional que propicia o treinamento supervisionado dos neurônios artificiais. Surgiu então o algoritmo *Backpropagation*, um algoritmo de otimização global sem restrições.

Em 1987 ocorreu a primeira conferência de redes neurais. Também foi formada a sociedade internacional de redes neurais (*International Neural Networks Society- INNS*) juntamente com o *INNS Journal* em 1989, do *Neural Computation* e do *IEEE Transactions on Neural Networks* em 1990.

Como citado por Tsay (2005), as Redes Neurais Artificiais são ferramentas gerais e flexíveis para aplicações em previsão.

As Redes Neurais Artificiais representam um tema popular na moderna análise de dados e podem ser classificadas como um método semi-paramétrico. Ao contrário dos métodos não-lineares baseados em modelos, as Redes Neurais Artificiais são abordagens baseadas em dados que podem capturar as estruturas não-lineares, sem hipótese prévia sobre a relação subjacente em um problema particular.

A Figura 4 mostra a estrutura da Rede Neural Artificial empregada no presente estudo: uma rede *feedforward* multicamadas, formada com o algoritmo *Backpropagation*. A Rede Neural Artificial possui três tipos de camadas, a saber, a camada de entrada, *input layer*, a camada de saída, *output layer*, e a camada oculta, *hidden layer*, a qual é intermediária entre a entrada e a saída. O número de camadas ocultas é geralmente um ou dois. Cada camada é constituída de neurônios, e os neurônios em duas camadas adjacentes estão completamente ligados com respectivos pesos, enquanto que os neurônios no interior da mesma camada não

estão ligados. Neste estudo, a camada de saída tem um único neurônio, que representa a previsão de um passo com base nos pontos anteriores.

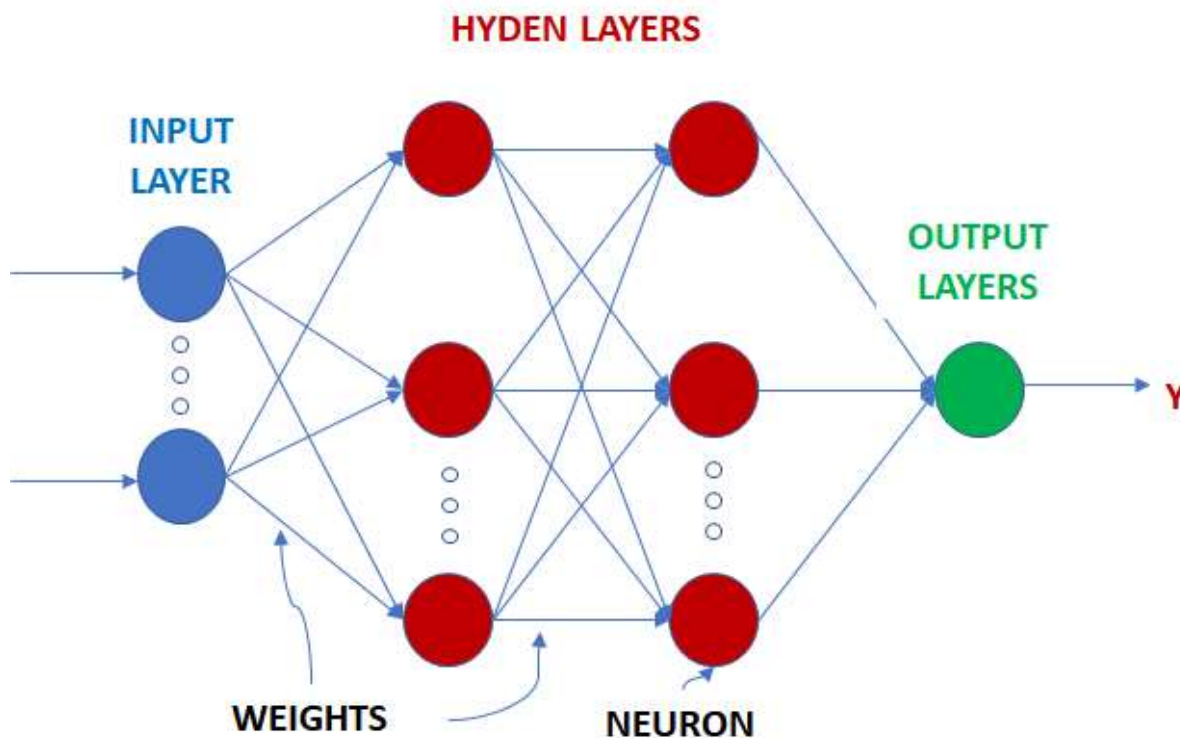


Figura 4: Estrutura da Rede Neural Artificial *feedforward* multicamadas

Os neurônios na camada de entrada são designados para um atributo nos dados, e produz uma saída que é igual ao valor (escala) de atributo correspondente. Por outro lado, para cada neurônio na camada escondida ou de saída, a seguinte transformação de entrada e saída é empregada:

$$v = f(\sum_{h=1}^H w_h u_h + w_0) \quad (13)$$

em que v é a saída, H é o número total de neurônios na camada anterior, u_h é a saída e h o número de neurônios na camada anterior, w_h é o peso de ligação correspondente, w_0 é a polarização (ou intercepção). A função f é a função de ativação também usada na camada de saída. A função de ativação, costuma ser denotada por $\varphi(v)$ e define a saída de um neurônio

em termos do campo local induzido v . Vejamos alguns casos clássicos de funções de ativação. A função $\varphi : R \rightarrow R$, dada por:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq 0 \\ 0 & \text{se } v < 0 \end{cases} \quad (14)$$

é denominada função de limiar.

Na literatura de engenharia, esta forma de função de limiar é normalmente referida como função de *Heaviside*. Correspondentemente, a saída do neurônio k que emprega essa função de limiar é expressa como:

$$y_k = \begin{cases} 1 & \text{se } v_k \geq 0 \\ 0 & \text{se } v_k < 0 \end{cases} \quad (15)$$

em que v_k é o campo local induzido do neurônio, isto é,

$$v_k = \sum_{i=1}^n w_{ki} x_i + b_k \quad (16)$$

A função $\varphi : R \rightarrow R$, dada por:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \geq \frac{1}{2} \\ v & \text{se } -\frac{1}{2} < v < \frac{1}{2} \\ 0 & \text{se } v \leq -\frac{1}{2} \end{cases} \quad (17)$$

é denominada função linear por partes.

Uma função $\varphi : R \rightarrow R$, tal que:

$$\varphi(v) = \begin{cases} 1 & \text{se } v \rightarrow +\infty \\ 0 & \text{se } v \rightarrow -\infty \end{cases} \quad (18)$$

é denominada função sigmoide.

As funções sigmóides são largamente as mais utilizadas em projetos de redes neurais e proporcionam um balanceamento entre comportamento linear e não linear. Outra vantagem é que as funções sigmóides são diferenciáveis (diferenciabilidade é uma característica importante da teoria das redes neurais). As funções sigmóides mais utilizadas são:

$$\varphi(v) = \frac{1}{1+e^v} \text{ e } \varphi(v) = \tanh(v) \quad (19)$$

Quando a rede neural é treinada com o algoritmo *Backpropagation* os pesos e os vies são otimizados.

Um tópico comum na análise de dados são as Redes Neurais Artificiais que podem ser classificadas como um método semi-paramétrico. Por oposição aos métodos baseados em modelos não lineares, Redes Neurais Artificiais são instrumentos poderosos para capturar estruturas não lineares de dados sem a suposição prévia sobre a relação subjacente em um problema particular.

Avaliação de Redes Neurais Artificiais a partir de perspectivas estatísticas e econométricas podem ser encontradas em Cheng e Titterton (1994). Atualmente, Redes Neurais Artificiais são usadas em uma grande variedade de modelos e problemas de previsão. Apesar de muitos modelos comumente usados em problemas reais serem lineares, a natureza da maioria do conjunto de dados reais sugere que modelos não lineares são mais apropriados para previsão e descrição do problema.

A literatura sobre Redes Neurais Artificiais é muito vasta e suas aplicações espalharam-se por vários domínios científicos com graus variados de sucesso. Na *M-Competition*, Lin e Tseng (1993), na *M2-Competition*, Makridakis *et al.* (1993) e na *M3-Competition*, Makridakis e Hibon (2000), muitos participantes utilizam Redes Neurais Artificiais. A principal razão para esse aumento da popularidade das Redes Neurais Artificiais é que elas têm se mostrado capazes de aproximar qualquer função não linear arbitrariamente dada. Assim, quando aplicada a uma série temporal, que é caracterizada por relações dinâmicas de natureza não linear, a Rede Neural Artificial irá detectar essas relações e proporcionar um encaixe superior quando comparado com os modelos lineares, sem a necessidade da construção de um modelo paramétrico específico para a série temporal não linear.

Para enfrentar o problema de previsão de séries temporais, alguns importantes trabalhos de pesquisa têm considerado as Redes Neurais Artificiais como uma metodologia promissora e que abordaram questões importantes

- Franses e Homelen (1998) exploraram a habilidade de uma Rede Neural Artificial capturar não linearidades como implicação dos modelos SETAR, MARKOV-SWITCHING e GARCH;
- Kaastra e Boyd (1996) fornecem um guia introdutório, em oito etapas, na concepção de redes neurais, para o procedimento de previsão de séries temporais econômicas;
- Bodyanskiy e Popov (2006) apresentam uma abordagem especial de Redes Neurais Artificiais para previsão de séries temporais financeiras com base na apresentação da série como uma combinação de componentes quase periódicos;
- Tseng *et al.* (2002) propõe um modelo de previsão híbrido, que combina séries temporais sazonais ARIMA (SARIMA) e Redes Neurais Artificiais treinadas com o algoritmo *Backpropagation*, modelos denominados SARIMABP;
- Karunasinghe e Liong (2005) investigaram o desempenho de uma Rede Neural Artificial como um modelo global sobre os modelos locais utilizados (técnica local das médias e técnica local dos polinômios) em séries temporais caóticas;
- No trabalho de pesquisa de Aitkenhead *et al.* (2003) óleo, água corrente e as variáveis climáticas foram mensurados durante vários meses em duas situações no nordeste da Escócia, utilizando coletores de dados e outros instrumentos de medida. Os conjuntos de dados foram utilizados para treinar redes neurais usando três diferentes métodos, incluindo um novo, sistema biologicamente plausível;
- BuHamra *et al.* (2003) combinam Box-Jenkins e a abordagem de Redes Neurais Artificiais para modelar séries temporais de dados de consumo de água no Kuwait;
- Shi *et al.* (1999) investigaram a modelagem de séries temporais não lineares usando o modelo *State-dependent Autoregressive*;
- Niska *et al.* (2004) estudaram o modelo de qualidade do ar, usando Redes Neurais Artificiais, uma tarefa difícil devido ao fenômeno caótico e não linear e um espaço amostral de alta dimensão;
- Zhang (2003) apresenta uma metodologia híbrida que combina o modelo ARIMA e Redes Neurais Artificiais para tirar vantagem do modelo ARIMA e das redes neurais na modelagem linear e não linear;

- Kim (2003) usa máquinas de vetores de suporte (SVMs) para previsão de séries temporais financeiras. Este estudo se aplica a previsão do índice de preços das ações;
- Ho *et al.* (2002) mostram um estudo comparativo de Redes Neurais Artificiais e modelos ARIMA na previsão de séries temporais *Backpropagation* e Redes Neurais recorrentes e apresentam um desempenho satisfatório quando comparados com ARIMA;
- Kohzadi *et al.* (1996) comparam ARIMA e redes neurais no desempenho de previsão de preços;
- Terasvirta *et al.* (2005) examinam a precisão das previsões do modelo *linear autoregressive, smooth transition autoregressive* (STAR) e redes neurais para 47 variáveis macroeconômicas mensais das economias do G7;
- Ghiassi *et al.* (2005) apresentam um modelo dinâmico de rede neural para previsão de eventos em séries temporais que utiliza uma arquitetura diferente dos modelos tradicionais;
- Balkin e Ord (2000) exploram um método chamado *Automated Artificial Neural Networks*, que é a tentativa de desenvolver um procedimento automático para seleção da arquitetura da Rede Neural Artificial para fins de previsão;
- Cubiles-de-la-Vega *et al.* (2002) propõem um procedimento para projetar uma perceptron multicamadas para previsão de séries temporais. Isto baseia-se na geração, de acordo com um conjunto de regras que emergem de um modelo ARIMA, previamente montada de um conjunto de modelos não lineares de previsão;
- Kalaitzakis *et al.* (2002) apresentam o desenvolvimento e aplicação de redes neurais avançadas para enfrentar com sucesso o problema da previsão da carga elétrica de curto prazo utilizando dados reais de carga a partir do sistema elétrico da ilha de Creta, na Grécia;
- Zhang e Qi (2001) investigaram o problema de sazonalidade e mostraram que limitados estudos empíricos sobre a previsão de séries temporais sazonais com redes neurais produzem resultados mistos;
- Em Chiang *et al.* (1996) foi relatado que a Rede Neural Artificial provou ser superior aos modelos de regressão quando a disponibilidade dos dados avaliados é limitada.

Este estudo é parcialmente motivado pelos resultados presentes no artigo de Zhang (2001), examinando a capacidade de uma Rede Neural Artificial para séries temporais lineares, usando tanto dados reais quanto simulados, onde afirma que redes neurais são muito competentes na previsão de séries temporais lineares, modelagem e em uma variedade de situações e estruturas neurais simples.

Hwang e Ang (2001) e Hwang (2001) motivados principalmente pela previsão de séries temporais lineares abordam algumas ideias que são relevantes para o presente trabalho: (i) Rede neurais artificiais *Backpropagation* geralmente apresentam um bom desempenho e consistência para a série de tempo correspondente a estruturas ARMA (p, q), principalmente quando o nível de ruído em particular foi considerado durante o treinamento da rede (ii) Dada a noção de que redes multicamadas *feedforward* podem atuar como aproximadores universais, é razoável esperar que redes *Backpropagation* podem executar previsões, pelo menos, comparáveis em dados lineares. Se assim for, pode ser conveniente aplicar redes *Backpropagation* independentemente da natureza de dados, especialmente quando a forma funcional de dados é desconhecida. Zhang *et al.* (1998) apresentam uma pesquisa estado da arte em aplicações das redes neurais artificiais na previsão de séries temporais e os seguintes pontos são considerados:

1. Em geral, Redes Neurais Artificiais apresentam um desempenho satisfatório em problemas de previsão;
2. Existem muitos fatores que podem afetar o desempenho das Redes Neurais Artificiais. No entanto, não há investigações sistemáticas destas questões. A metodologia de tentativa e erro é tipicamente adotada em problemas específicos por muitos pesquisadores, que é a razão primária para inconsistências na literatura;
3. Uma quantidade considerável de pesquisas tem sido realizada nesta área, dado o rápido crescimento da literatura.

2.3 DOE no Treinamento de Redes Neurais Artificiais para Previsão de Séries Temporais Não Lineares

O processo de treinamento de uma Rede Neural Artificial consiste na mudança da entrada dos parâmetros do algoritmo computacional, rodar o algoritmo e checar os resultados.

Isto pode ser tratado como um estudo de simulação para o problema de Redes Neurais Artificiais.

Apesar da quantidade de sucessos relacionados com aplicações industriais, DOE não é amplamente usado em simulação. Kleijnen *et al.* (2005) apontam que a falta do uso de DOE para simulação deve-se a:

- Os analistas de simulação não estão convencidos dos benefícios de DOE;
- A investigação em DOE é geralmente encontrada em revistas especializadas raramente lidas por analistas de simulação;
- A maioria dos DOE foram originalmente desenvolvidos para a experimentação do mundo real, em vez de desenvolvido especificamente para as configurações de simulação.

Esta pesquisa também aponta os principais benefícios do delineamento experimental no desenvolvimento do modelo e prevê que o uso de DOE se torne mais substancial nesta área.

Traduzindo para a terminologia de simulação pode-se dizer que uma entrada ou um parâmetro na simulação corresponde a um fator no DOE. Normalmente há muitos mais fatores na simulação do que em um experimento no mundo real. Um fator pode ser qualitativo ou quantitativo. Cada fator pode ser definido por dois ou mais valores, chamados de níveis de fator, normalmente codificados numericamente para fins de análise. Um delineamento experimental é uma combinação dos níveis de todos os fatores. Ao considerar simulações estocásticas, repetições significam que os diferentes números pseudoaleatórios (PRN), são utilizados para simular o mesmo experimento. A natureza da coleta de dados de cenários não é aleatória, mas sequencial. As saídas nas repetições são independentes e identicamente distribuídas (i.i.d.).

Desenvolver uma compreensão básica de simulação é equivalente a testar hipóteses sobre os efeitos do fator no DOE (KLEIJNEN *et al.*, 2005).

Outra questão importante quando se utiliza DOE para a simulação é que o objetivo principal não é a otimização. Ao utilizar DOE os esforços são dedicados a encontrar políticas ou decisões robustas, ao invés de ideais. É certamente verdade que encontrar a melhor política para um sistema simulado é um tema importante, e foram propostos vários métodos. Fu (2002) e Spal (2003) discutem a atual pesquisa e a prática de otimização para simulação. Estes métodos incluem técnicas heurísticas de busca, tais como algoritmos genéticos, a metodologia de superfície de resposta (RSM), *simulated-annealing*, *tabu-search* e métodos que analisam o modelo de simulação para estimar gradientes, tais como análise de perturbação e funções de

pontuação. O resultado da "otimização" está condicionado a pressupostos de distribuições específicas (normalmente assumido independente) e muitas variáveis de entrada. O termo "melhor" é problemático quando a probabilidade de todas estas suposições, mesmo que por um tempo limitado é efetivamente zero. Em contraste, uma abordagem de delineamento robusto trata todos estes pressupostos como fatores adicionais durante a execução do experimento. Estes são considerados fatores de ruído (em vez de fatores de decisão), porque eles são desconhecidos ou incontroláveis no ambiente do mundo real. Um sistema robusto funciona bem em uma variedade de condições de ruído que pode ser experimentado, por isso a implementação de uma solução robusta é muito menos provável de resultar em resultados inesperados. Esta filosofia de delineamento robusto é inspirada por Taguchi (1987), que usou o delineamento simples para identificar configurações de produtos robustos para a Toyota.

Embora os modelos não lineares do Quadro 1 possam ser úteis para um problema de dados em particular, eles não têm um apelo geral para outras aplicações. As pré-especificações de acordo com o modelo, limita a utilidade destes modelos paramétricos não lineares a muitos padrões lineares possíveis. De fato, a formulação de um modelo não linear apropriado para um determinado conjunto de dados é uma tarefa muito difícil, em comparação com a construção do modelo linear, porque existem mais possibilidades, muitos mais parâmetros e, portanto, mais erros podem ser cometidos (MAKRIDAKIS; HIBON, 2000). Além disso, uma especificação não linear particular pode não ser em geral suficiente para capturar todas as não linearidades nos dados. Como Diebold e Nason (1990) apontaram, a grande variedade de candidatos plausíveis a modelos não lineares determina uma boa aproximação para o processo de geração de dados, o que é uma tarefa difícil, e a grande variedade de modelos não lineares paramétricos é na verdade um pequeno subconjunto da classe plausível do processo de geração de dados não lineares.

Conforme já descrito, Redes Neurais Artificiais são candidatas naturais para previsão de séries temporais não lineares. No entanto, o grande número de parâmetros a serem selecionados para desenvolver um modelo de previsão de rede neural, significa que o processo de concepção ainda envolve muitas tentativas e erros. Os métodos tradicionais de estudar um fator em um tempo podem conduzir a resultados não confiáveis e enganosos, e as vezes pode produzir conclusões erradas. Estatisticamente, experiências concebidas num desempenho mais eficiente, que considerarem simultaneamente fatores múltiplos, podem detectar interações importantes. Usando DOE como um processo de planejamento de experimentos, é possível recolher dados apropriados, usando o número mínimo de experimentos, ao adquirir as

informações técnicas necessárias. A introspecção sobre os efeitos principais, bem como os efeitos de interação de fatores, incluindo o ruído, são úteis na tomada de decisão.

Alguns trabalhos têm estudado a aplicação de Redes Neurais Artificiais, simulando através de delineamento de experimentos, em um contexto diferente para as propostas deste trabalho e explorando diferentes fatores e níveis. Khaw et al. (1995) descrevem uma inovadora aplicação do método Taguchi para a determinação dos parâmetros do delineamento, que incluem tanto os aspectos micro e macroestrutural de uma rede neural. A viabilidade da utilização desta abordagem foi demonstrada por meio da otimização dos parâmetros do projeto de uma rede neural *Backpropagation* para determinar as políticas operacionais de um sistema de manufatura. Os resultados retirados desta pesquisa demonstraram que o método Taguchi proporciona um meio eficaz para melhorar o desempenho da rede neural, em termos de velocidade de aprendizagem e a precisão. Kim e Yum (2004) apresentam um estudo similar. Sukthomya e Tannock (2005) e Sukthomya e Tannock (2005) utilizam a mesma ideia do projeto experimental de Taguchi para definir os parâmetros de uma Rede Neural Artificial. Lin e Tseng (2000) usam a mesma abordagem Taguchi para a "Aprendizagem de Quantização Vetorial" em Redes Neurais Artificiais num aplicativo para sistemas de câmbio de bicicleta. Enemuoh e El-Gizawy (2003) descrevem um método para a criação de uma Rede Neural Artificial robusta para a predição de delaminação, a largura do dano, e a rugosidade da superfície do furo durante a perfuração em epóxi reforçado com fibra de carbono. Os efeitos do número de neurônios, camadas ocultas, a função de ativação e algoritmo de aprendizagem e o erro quadrático médio de previsão do modelo são quantificados. Zhang (2001) e Zhang *et al.* (2001) apresentam uma avaliação experimental de redes neurais para previsão de séries temporais lineares e não lineares. Três principais parâmetros da Rede Neural Artificial são examinados através de um experimento simulado através de um computador: nós de entrada, nós ocultos e tamanho da amostra.

Quando comparado com os trabalhos anteriores, o presente estudo pode ser considerado inovador pelos seguintes motivos:

- Em vez de utilizarmos a abordagem de DOE Taguchi, uma abordagem mista é examinada;
- O número de parâmetros da Rede Neural Artificial é aumentado;
- A sazonalidade está incluída para simular problemas não lineares reais;
- Interações entre parâmetros da Rede Neural Artificial são permitidas e avaliadas.

3 Delineamento Experimental

Nesta seção, o projeto experimental para o treinamento da RNA é examinado. Algumas diretrizes para experimentos industriais serão abordadas e algumas alterações são recomendadas para o contexto de treinamento da RNA. A implantação e os resultados das diretrizes são apresentados a seguir.

3.1 Procedimentos Gerais

Coleman e Montgomery (1993) apresentam algumas diretrizes para a concepção de um experimento que, apesar de ser focado em experimentos industriais, também pode ser utilizado para simulação computacional:

- a) Definição do problema
- b) Escolha de fatores, níveis e intervalos
- c) Seleção da variável de resposta
- d) Escolha do projeto experimental
- e) Realização do experimento
- f) Análise estatística dos dados
- g) Conclusões e Recomendações

Essas diretrizes são geralmente interativas e a estrutura não é rígida quando aplicada em experimentos reais. Algumas etapas são frequentemente realizadas simultaneamente, ou também na ordem inversa. As etapas a, b e c são chamadas planejamento pré-experimental. Alguns comentários relacionados à RNA e seu treinamento são seguidos.

- a) **Definição do problema.** Isto pode parecer um ponto obvio, mas em experimentos industriais, e também na simulação computacional, não é simples de obter toda a imagem de um problema e, geralmente, é necessária uma abordagem de equipe. Para esta aplicação, que envolve muitas áreas e conhecimentos, onde as opiniões diferentes são muitas vezes conflitantes, uma abordagem de equipe é apropriada. Além disso, uma declaração clara do problema contribui substancialmente para a solução do problema. O reconhecimento do problema e sua afirmação correta costumam dar foco para alavancar um objetivo.
- b) **Escolha dos fatores, níveis e intervalos.** Em experimentos do mundo real, apenas um pequeno número de fatores varia normalmente. Na verdade, é impraticável ou

impossível tentar controlar mais do que, digamos, 10 fatores; muitos experimentos publicados lidam com menos de 5. Em contraste, uma infinidade de fatores potenciais existe para modelos de simulação usados na prática. Uma boa programação evita fixar os fatores em valores numéricos específicos dentro do código; em vez disso, o código lê os valores dos fatores para que o programa possa ser executado para muitas combinações de valores. Obviamente, o código deve verificar se esses valores são admissíveis; isto é, essas combinações se enquadram no domínio experimental? Essa prática pode fornecer automaticamente uma lista de fatores potenciais. Em seguida, os usuários devem confirmar se realmente desejam experimentar todos esses fatores ou se desejam fixar alguns fatores em níveis nominais (ou básicos), a priori. Esse tipo de codificação ajuda a modificar a mentalidade dos usuários que, de outra forma, estariam inclinados a se concentrar em apenas alguns fatores. Em experimentos do mundo real, a mentalidade básica costuma ser que os dados devem ser obtidos simultaneamente, a menos que o projeto seja especificamente identificado como um projeto sequencial. Quando as amostras devem ser coletadas sequencialmente, o experimento é visto como sujeito a problemas de validade. Os analistas devem, portanto, randomizar a ordem de amostragem para evitar mudanças relacionadas ao tempo no ambiente experimental (como temperatura, umidade, confiança do consumidor e efeitos de aprendizagem) e realizar testes estatísticos apropriados para determinar se os resultados foram contaminados. A maioria dos experimentos de simulação são implementados sequencialmente, mesmo que não sejam formalmente analisados dessa forma. Se um pequeno número de pontos de projeto é explorado, esta implementação pode envolver os analistas alterando manualmente os níveis dos fatores (KLEIJNEN, SANCHEZ, LUCAS E CIOPPA, 2005). O aumento na velocidade dos computadores fez com que alguns analistas adicionassem mais detalhes aos seus modelos de simulação. Diferentes analistas podem usar diferentes conjuntos de fatores e níveis.

- c) **Seleção da Variável de Resposta.** Para experimentos industriais, a escolha de uma variável de resposta útil e prática, como rendimento, carga, custo, etc., envolve um estudo de capacidade do medidor e o erro é, em muitos casos, avaliado por estudo de repetibilidade e reprodutibilidade. Para simulação de computador, onde a condição inicial pode ser bloqueada (geralmente definindo a base de um número aleatório), a resposta é geralmente uma variável estimada que pode ser repetida sem erro estabelecendo as mesmas condições iniciais. A escolha experimental também é, em certa medida, muito mais complexa do que em uma simulação computacional e precisa

ser avaliada a priori. Média, desvio padrão, etc., são frequentemente usados como variáveis de resposta. Respostas múltiplas também são usuais em experimentos de projeto industrial e a otimização simultânea de várias variáveis de resposta envolve funções de desejabilidade (DERRINGER E SUICH, 1980).

- d) **Escolha do Delineamento Experimental.** O termo indica a criação de uma matriz, onde as colunas representam os elementos de entrada e cada linha representa uma combinação de níveis do fator. A escolha de concepção envolve o tamanho da amostra, a ordem de execução da experiência (para simulação de computador que é geralmente irrelevante), e várias restrições para compor a matriz final gerada como uma saída de folha de cálculo. Alguns pacotes de softwares estatísticos como o Minitab, Statistica, SPSS, JMP, Matlab, entre muitos outros, são bons programas que oferecem uma biblioteca de delineamentos clássicos. Estes delineamentos normalmente são gerados com os níveis codificados e são escolhidos com base no número de fatores e níveis, estrutura de alisamento, confundimento, resolução, a quantidade de tempo e recursos para executar os experimentos, etc. Um programa novo e interessante, o WebDOE, ajuda os usuários a projetar sua experiência através de um *easy-to-use interface* de web (CraryGroup, 2004). Kleijnen *et al.* (2005), usando DOE para simulação, recomendam um esquema baseado nos pressupostos da complexidade do sistema e afirmam que é melhor inicialmente se concentrar em um grande número de fatores (respeitando as limitações de tempo, custo, etc.). Desta forma, os analistas podem olhar amplamente através dos fatores no estudo de simulação. Às vezes, a intuição está errada e precisa ser desafiada. Com esta abordagem, a partir de pressupostos mínimos para uma experiência inicial, o analista terá a mente aberta para novos pressupostos e admite que pouco se sabe sobre a natureza da resposta. O analista tende a reduzir o esforço de coleta de dados iniciais, de fazer hipóteses simplificadoras. Os analistas podem reduzir o esforço computacional, fazendo suposições sobre a natureza do sistema. A melhor ideia é passar de suposições mínimas e concentrar-se na pequena lista de fatores selecionados após o experimento inicial, mantendo os demais fatores apenas em algumas configurações. Na prática, os resultados de experimentos, muitas vezes precisam ser modificados, ou seja, expandidos ou excluídos para obter informações mais detalhadas sobre o desempenho da simulação para uma região menor das combinações de fatores.
- e) **Realização do Experimento.** Executar o experimento em simulação de computador é geralmente mais fácil do que em experimentos industriais. Se o programa de simulação é de código aberto, é conveniente executar a simulação completa como um processo em

lotes, ou descontínuo. Infelizmente, isso não é o que acontece muitas vezes devido a habilidade de programação necessária e a restrição de software sempre imposta. Além disso, essa experimentação iterativa é considerada um erro em experiências industriais. Coleman e Montgomery (1993) salientam que a experimentação deve ser sequencial e, como regra geral, não mais do que cerca de 25% dos recursos disponíveis devem ser investidos no primeiro experimento.

- f) **Análise Estatística dos Dados.** Métodos estatísticos não terão muito sentido, se as orientações anteriores não forem seguidas. Métodos gráficos, análise residual e modelo de adequação desempenham um papel importante nesta fase. A análise estatística adiciona objetividade ao processo de tomada de decisão.
- g) **Conclusões e Recomendações.** Assumindo um experimento de design sequencialmente, as execuções de acompanhamento e o teste de confirmação também devem ser realizados.

3.2 Planejamento Pré Experimental

Nesta seção, o reconhecimento e declaração do problema, a seleção da variável de resposta e a escolha de fatores, níveis e intervalos serão discutidos. Relacionado com o treinamento de Redes Neurais Artificiais, o seguinte enunciado do problema foi considerado adequado, utilizando uma abordagem de trabalho em equipe:

Ao prever séries temporais não lineares usando RNA, tentativa e erro é o método mais comumente aplicado e o tempo consumido é a metodologia mais frequentemente usada. O problema aqui é estabelecer uma metodologia bem estruturada para estimar os parâmetros desta RNA.

Várias medidas de erro como *Root Mean Square Error* (RMSE), *Mean Absolute Percentage Error* (MAPE), *Median Absolute Percentage Error* (MdAPE), etc. têm sido usadas como medidas de desempenho para previsão de séries temporais. O leitor interessado pode verificar Armstrong e Collopy (1992), para uma melhor discussão sobre este tema. Deve notar-se que não há nenhuma medida de erro de previsão aceita uniformemente.

O erro médio absoluto percentual (MAPE) expressa precisão como uma porcentagem do erro. Como o MAPE é uma porcentagem, pode ser mais fácil de entender do que as outras estatísticas de medida de precisão. Por exemplo, se o MAPE é 5, em média, a previsão está incorreta em 5%.

Neste trabalho, o MAPE será utilizado como variável de resposta, definido como:

$$MAPE(\%) = \frac{1}{T} \sum_{t=1}^T \frac{|y_t - \hat{y}_t|}{y_t} 100 \quad (20)$$

em que y_t é a observação atual no tempo t , \hat{y}_t é o valor previsto e T é o número de previsões.

Vários fatores têm sido considerados na literatura para o treinamento de redes neurais artificiais. O Quadro 2, a seguir, apresenta os fatores considerados de triagem para problemas de previsão de séries temporais não lineares utilizando o software Statistica (StatSoftInc., version 7.1, 2005, www.statsoft.com).

Quadro 2: Seleção de fatores para o treinamento da RNA para previsão de séries temporais Não Lineares

| Fator | Símbolo | Níveis | Nº de Níveis |
|---|---------|--|--------------|
| Arquitetura da rede | - | MLP, RBF, GRNN, ARTMAP... | 1 |
| Número de camadas ocultas | HL | 0 (raramente), 1, 2 ou 3 (raramente) | 2 |
| Número de unidades por camada | UL | k(N+1), onde N é o número de entradas e k=1, 1.5, 2 | 3 |
| Função de regressão de saída | OF | Linear, Logistics Range | 2 |
| Problema tipo / modo de entrada | PT | Série temporal univariada / Regressão | 2 |
| Previsão X passos à frente | - | 1, 2, 3,... | 1 |
| Passos usados para prever | SP | 12, 24 | 2 |
| Algoritmo de treinamento - primeira fase | P1 | Backpropagation, Quick propagation, Delta bar delta | 3 |
| Algoritmo de treinamento - segunda fase | P2 | Gradiente descendente conjugado, Quasi-Newton, Levenberg-marquardt | 3 |
| Épocas | Ep | 100, 400 | 2 |
| Taxa de aprendizagem | LR | 0.1, 0.9 | 2 |
| Inicialização do método | IM | Unif(0,1), N(0,1) | 2 |
| Condições de parada | SC | 0, 0,1 | 2 |
| Melhoria do erro mínimo para treinamento | ET | 0, -0,1 | 2 |
| Melhoria do erro mínimo para o número de épocas | EE | 1, 25, 50 | 3 |
| Poda de unidades | PU | Não, com pequenos pesos <i>fan-out</i> (podando <i>threshold</i> =0.05) | 2 |
| Poda das variáveis de entrada | PI | Não, com pequenos pesos <i>fan-out</i> (podando <i>threshold</i> =0.05),(ratio=1). | 3 |
| Regularização no decaimento de pesos - Fase 1 | W1 | Não, FD=0.0001, FD=0.001, FD=0.01 | 3 |
| Regularização no decaimento de pesos - Fase 2 | W2 | Não, FD=0.0001, FD=0.001, FD=0.01 | 3 |
| Ajuste Backpropagation | BP | 4 experimentos para L4 Taguchi com fatores (A, B e C) | 4 |
| Ajuste Quick propagation | QP | 4 experimentos para L4 Taguchi com 3 fatores (A, B e C) | 4 |
| Ajuste delta-bar-delta (Fase condicional) | DD | 4 experimentos para L4 Taguchi com 2 fatores (A, B) | 4 |
| Tamanho da amostra | SS | k*24, em que k=7, 14, 21 | 3 |
| Método de amostragem | SM | Aleatória, <i>Bootstrapping</i> , Validação cruzada | 2 |

Detalhes sobre os fatores estão descritos a seguir.

- **Arquitetura da RNA.** Redes Neurais Artificiais são algoritmos de modelagem não linear. Exemplos de rede neurais para séries temporais não lineares são Perceptron multicamadas (MLP), função de base radial (RBF), *Generalized Regression Neural Network* (GRNN), *Support Vector Machine* (SVM), entre muitos outros. MLP é um dos tipos de rede mais populares, o único considerado neste trabalho e, em muitos domínios de problemas, parecem oferecer o melhor desempenho possível.
- **Número de Camadas Ocultas.** Uma camada oculta é um grupo de neurônios que têm uma função específica e são processados como um todo. Os resultados teóricos prescrevem que um MLP com uma camada oculta (Perceptron de três camadas) é capaz de se aproximar de qualquer função contínua. (HORNIK, STINCHCOMBE E WHITE, 1989).
- **Número de Unidades por Camada.** Nós ocultos são usados para capturar as estruturas não lineares em uma série temporal. A determinação de quantos nós ocultos usar na Rede Neural Artificial é outra questão difícil no processo de construção do modelo. Não existe uma teoria bem estabelecida ainda a dizer quantas unidades ocultas são necessárias para aproximar qualquer função dada. Na prática, o número de nós escondidos é muitas vezes escolhido através da experimentação ou por erro. Há uma série de "regras de ouro", mas o melhor número de unidades ocultas depende de uma forma complexa do número de entrada e de saída de unidades, o número de casos de treinamento, a quantidade de ruído nas metas, a complexidade da função ou classificação a ser aprendida, a arquitetura, o tipo de função de ativação, o treinamento do algoritmo, regularização, etc. Aqui vai ser usada uma quantidade entre o número de entrada e o número k ($N + 1$), onde N é o número de entradas e $k = 1, 1,5$ e 2 .
- **Função de Ativação.** Todas as redes neurais possuem entrada numérica e produzem uma saída numérica. A função de uma unidade de transferência é tipicamente escolhida de modo que ele possa aceitar a entrada em qualquer intervalo, e produz uma saída numa gama estritamente limitada (BISHOP, 1995).
- **Tipo do Problema (*Input mode*).** A forma como a série temporal é apresentada como entrada para a RNA é sempre considerado fundamental. Como um problema de séries temporais univariadas defasadas a informação para o processo autorregressivo é preservada. Deste modo, a entrada será uma amostra da própria série temporal, $[y_{t-1}, y_{t-2}, \dots, y_{t-24}]$ considerando um defasamento de 24 e a resposta será

supervisionada na observação um passo à frente $[y_t]$. Já como um problema de regressão serão usadas variáveis temporais explicativas (simuladas). Desta forma a entrada será constituída de um conjunto de valores fictícios (0s e 1s) mais as observações da série temporal $[y_{t-1}, y_{t-2}, \dots, y_{t-x}, d_1, d_2, \dots, d_k]$ em que x denota algumas últimas observações e d_k e uma transformação binária para a sazonalidade ou outras variáveis endógenas. Procedimentos de mineração de dados são geralmente úteis para pré processar a série temporal original e obter características especiais, como variância, autocovariância, média, etc., como entrada para as RNA. Neste estudo mostra-se o número de entradas como um problema de séries univariadas bem como no caso de um problema de regressão, como o mesmo modo que se pode comparar às metodologias. Devido a isso, devemos saber com antecedência a sazonalidade mais importante para definir as variáveis defasadas. Gráficos como função de autocorrelação (ACF) ou funções de densidade (gerado pelo *Fast Fourier Transform* - FFT) são recursos disponíveis em muitos pacotes que poderiam ajudar a definir a sazonalidade. Como uma série temporal nativa, o ACF e a função de densidade para o modelo STAR2 (como apresentado pela primeira vez na Figura 3) mostra claramente a sazonalidade na Figura 5.

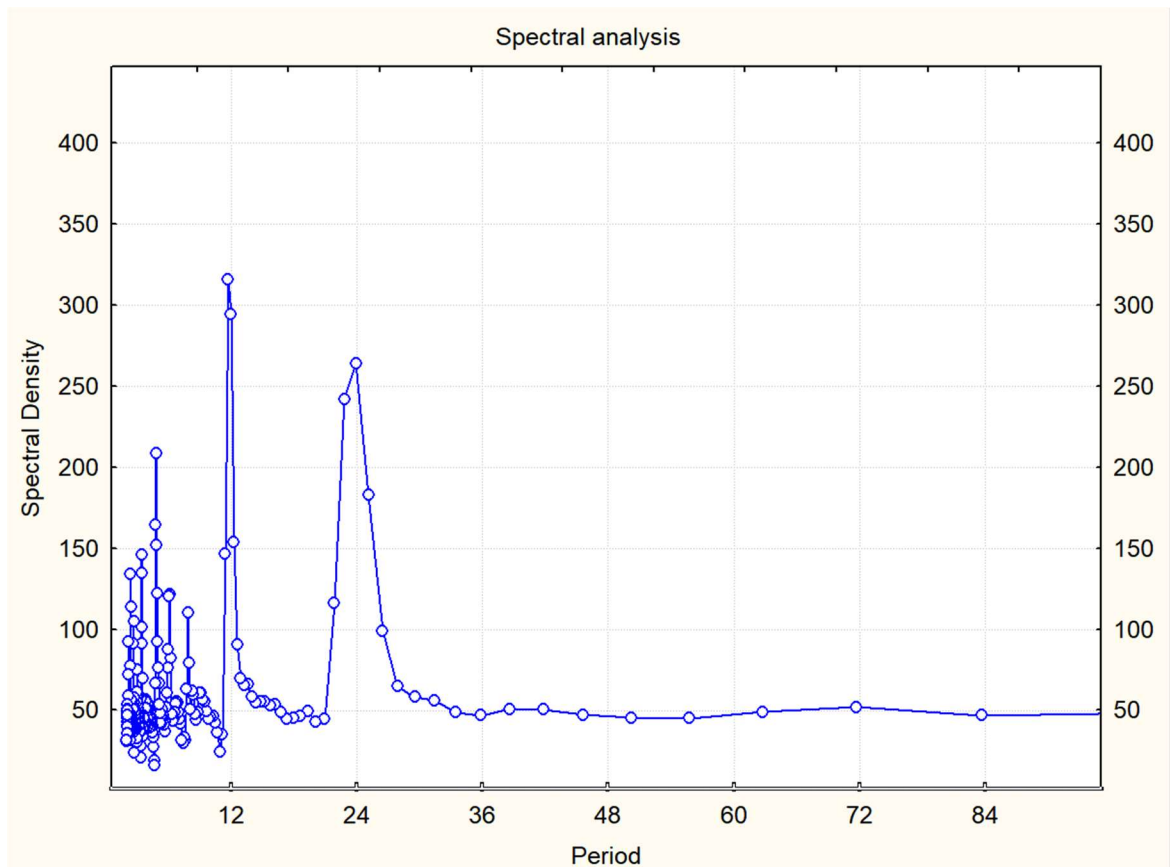


Figura 5: Densidade espectral para o modelo STAR2

- **Previsor de X passos a frente.** O número X denota o número de passos a frente dos valores de entrada desfasados que a saída está prevista. Neste caso, devido a pequena série temporal sintética é considerando o erro de propagação através dos passos previsores, apenas um passo à frente será usado. A saída da rede pode ser combinada com os valores de entradas anteriores, deslocados um passo no tempo com repetidas previsões.
- **Passos usados para prever.** O número de entradas da RNA é considerado aqui como 24 e 12, seja como um modo de série temporal univariada ou como um modo de regressão. Para o modo de entrada de séries temporais, o nível 24 corresponde a um período de série completa, considerando a sazonalidade. O nível de 12 corresponde a uma composição entre os últimos 8 pontos da série mais 4 estatísticas correspondentes da série de entrada (média, desvio padrão, autocorrelação para *lag* 1 e autocorrelação para *lag* 2). Já para o modo de entrada de regressão, o nível de 24 corresponde a uma composição de 20 valores da série de quatro variáveis binárias mais relacionadas com o último ponto. O nível de 12 corresponde a uma composição entre os 4 últimos pontos da série, além de estatísticas da série de entrada (média, desvio padrão, autocorrelações para *lag* 1 e autocorrelações para *lag* 2) mais quatro variáveis *dummy* referentes ao último ponto.
- **Algoritmo de treinamento, primeira fase.** Este fator é relacionado com o treinamento do algoritmo para a MLP no primeiro estágio e assume três níveis:
 - ***Backpropagation*.** Um algoritmo simples com um grande número de parâmetros de ajuste, convergência terminal muitas vezes lenta, porém uma boa convergência inicial.
 - ***Quickpropagation*.** Um algoritmo mais antigo com um desempenho comparável a *Backpropagation* na maioria das circunstâncias, embora pareça desempenhar visivelmente melhor em alguns problemas.
 - **Delta-Bar-Delta.** Outra variação do *Backpropagation*, que as vezes parece ter um melhor desempenho.
- **Algoritmo de treinamento segunda fase.** Este fator é relacionado com o treinamento do algoritmo para MLP no segundo estágio e assume três níveis.
 - **Gradiente descendente conjugado.** Um bom algoritmo genérico, geralmente, com uma convergência rápida.

- **Quasi-Newton.** Um poderoso algoritmo de treinamento de segunda ordem, com convergência muito rápida, mas que requer alta memória.
 - **Levenberg-Marquardt.** Um algoritmo extremamente rápido em certas circunstâncias.
- **Épocas.** Uma época é a apresentação de todo o conjunto na formação de uma rede neural em cada fase. Aumentando este número provavelmente irá melhorar a precisão do modelo, mas a custo do tempo, e diminuindo este número provavelmente diminuirá a precisão, mas com menos tempo. Diretamente relacionado com o tamanho da amostra de séries temporais este valor será adotado aqui nos níveis de 100 e 400.
- **Taxa de aprendizagem.** Um valor entre 0 e 1 que representa a variável de ajuste para o treinamento do algoritmo *Backpropagation*, *Quick propagation* e *Delta-Bar-Delta*. Taxas mais baixas de aprendizagem exigem mais iterações no treinamento. Uma taxa de aprendizado mais alta permite que a rede convirja mais rapidamente, porém as chances de uma solução não ótima são maiores. Os níveis escolhidos aqui foram 0,0 e 0,1.
- **Inicialização do método.** Este fator define como os pesos deverão ser inicializados para o treinamento e assume dois níveis:
 - **Variável Uniforme.** Os pesos são inicializados com um valor aleatório uniformemente distribuído dentro de um intervalo cujos valores mínimos e máximos são dados. Neste caso, mínimo e máximo são 0 e 1.
 - **Variável Gaussiana.** Os pesos são inicializados com um valor aleatório normalmente distribuído, dentro de um intervalo cujo média e desvio padrão são dados. Neste caso, $\mathcal{N}(0,1)$, foi aprovado.
- **Condições de parada.** Se o erro no treinamento cair abaixo dos valores alvo dados, a rede é considerada ter sido treinada suficientemente bem, e o treinamento é encerrado. O erro nunca cai para zero ou abaixo, de modo que o valor padrão de zero é equivalente a não ter um erro de alvo.
- **Melhoria do erro mínimo para treinamento.** Este fator representa a melhoria mínima (queda) no erro que deve ser feita, se a taxa de melhoria cair abaixo desse nível, o treinamento é encerrado. O valor padrão de zero implica que o treinamento será encerrado se o erro se deteriora. Pode-se também especificar uma taxa de melhoria negativa, o que é equivalente a dar uma taxa máxima de deterioração que será tolerada.

A melhoria é medida através de um número de épocas, a chamado "janela"; o que veremos mais à frente.

- **Melhoria do erro mínimo para o número de épocas.** Especifica-se o número de épocas através das quais a melhoria é medida. Alguns algoritmos, incluindo o algoritmo *Backpropagation*, demonstram ruído sobre os erros de treinamento e seleção, e todos os algoritmos podem mostrar ruído no erro de seleção. Portanto, não é geralmente uma boa ideia deter a formação na base de uma falha para atingir a desejada melhoria na taxa de erro ao longo de um único período. A janela especifica um número de épocas em que as taxas de erro são monitoradas para melhorias. O treinamento só é interrompido se o erro não melhorar para muitas épocas. Se a janela é nula, o limite mínimo de melhoramento não é utilizado em todos. Os níveis variáveis de 1, 25 e 50 definem o tamanho da janela para determinar a melhoria mínima. Os valores adotados foram considerados levando-se em consideração a sazonalidade das séries temporais.
- **Podar unidades com pequenos pesos *fan-out*.** Um neurônio com pequena magnitude de pesos *fan-out* (pesos que levam ao próximo nível) faz uma pequena contribuição para as ativações da camada seguinte e pode ser podado, levando a uma rede compacta, mais rápida, com desempenho equivalente. Esta opção é particularmente útil em conjunção com o peso Weigend que estimula o desenvolvimento de pequenos pesos, precisamente de modo que eles podem ser podados. Unidades ocultas com pequenos pesos de *fan-out* devem ser podadas. Se pesos *fan-out* de uma unidade tem magnitude menor do que um limiar, é um candidato para a poda.
- **Podar nas variáveis de entrada.**
 - **...com pesos pequenos *fan-out*.** Cada variável de entrada tem um ou mais neurônios da camada de entrada associados (mais de um por algumas variáveis nominais, e para redes de séries temporais), e o *fan-out* em todos esses neurônios de entrada deve ser menor do que o limite para a entrada a ser podada.
 - **... com baixa sensibilidade após o treino.** A análise de sensibilidade é executada depois que a rede é treinada, e as variáveis de entrada com treinamento e seleção de índices de sensibilidade abaixo de um valor limite são podadas. Uma variável de entrada com uma sensibilidade na proporção de 1,0 não contribui para a decisão da rede e pode ser podada sem qualquer prejuízo. Uma variável de entrada com sensibilidade inferior a 1,0, deve ser definitivamente podada (talvez surpreendentemente, entradas com sensibilidade

inferior a 1,0 sobre os dados de seleção não são uma ocorrência rara, um subproduto de um excesso de aprendizagem).

- **Regularização no decaimento de pesos- Fase 1.** O decaimento de peso pode ser aplicado separadamente às duas fases de um algoritmo de duas fases. Na primeira fase os algoritmos de *Backpropagation*, *Quickpropagation*, *Delta-Bar-Delta* foram considerados. Esta opção incentiva o desenvolvimento de pesos menores, o que tende a reduzir o problema de *over-fitting (sobre ajuste)*, assim potencialmente melhorar o desempenho e generalização da rede, e também permitindo podar a rede. O decaimento de peso funciona modificando a função de erro da rede para penalizar grandes pesos - o resultado é uma função de erro que compromete desempenho e tamanho do peso. Conseqüentemente, se for muito grande um termo de decaimento de peso este poderá prejudicar o desempenho da rede, e experimentação é geralmente necessária para determinar um fator de decaimento de peso adequado para um domínio de problema particular. Um estudo dos fatores de atenuação e de poda de peso indica que o número de entradas e as unidades podadas são aproximadamente proporcionais ao logaritmo do fator de deterioração. Por exemplo, se você usar um fator de decaimento de 0,001 e isso tiver um efeito insuficiente sobre os pesos, você pode tentar 0,01 no próximo, ao invés de 0,002, por outro lado, se os pesos foram mais ajustados, resultando em muita poda, tente 0,0001. Há também um fator secundário que normalmente é deixado no valor 1,0.
- **Regularização no decaimento de pesos -Fase 2.** Nesta fase, o algoritmo do gradiente descendente conjugado, Quasi-Newton e Levenberg-Marquardt são considerados.
- **Ajuste *Backpropagation***
 - **Ajuste da taxa de aprendizado e momento de cada época:** Normalmente o *Backpropagation* usa uma taxa de aprendizagem fixa e impulso ao longo do treinamento. Alguns autores, no entanto, recomendam alterar essas taxas em cada época (especificamente, através da redução da taxa de aprendizagem - esta é muitas vezes contrabalançada pelo aumento do momentum). Uma maior taxa de aprendizagem pode convergir mais rapidamente, mas também pode apresentar uma maior instabilidade. Os valores de 0,1 ou inferior são razoavelmente conservadores - taxas mais elevadas são toleráveis em alguns problemas, mas não em todas (em especial sobre os problemas de regressão, quando uma taxa maior pode realmente causar divergência catastrófica dos pesos). Momentum é usado para compensar a convergência lenta se ajustes de peso estão sempre em uma direção - o ajuste de "pega velocidade". Momentum

geralmente aumenta consideravelmente a velocidade de convergência do *Backpropagation*, e uma taxa mais elevada pode permitir que diminua a taxa de aprendizagem para aumentar a estabilidade, sem sacrifício em termos de velocidade de convergência.

- **Mistura da ordem de apresentação dos casos de cada época:** Neste caso, o algoritmo de *Backpropagation* ajusta os pesos da rede à medida que cada caso de treinamento é apresentado (ao invés da abordagem do lote, que calcula um ajustamento médio entre todos os casos de treinamento, e aplica-se a um único ajuste no final do período). Se a opção *shuffle* estiver marcada, a ordem de apresentação é ajustada em cada época, isso torna o algoritmo um pouco menos propenso a ficar em mínimos locais e é parcialmente responsável pela maior robustez do *Backpropagation* do que os algoritmos de treinamento de segunda ordem mais avançados a esse respeito.
- **Adicionar um ruído Gaussiano.** Um ruído gaussiano de um determinado desvio é adicionado ao valor de saída de destino em cada caso. Esta é uma outra técnica de regularização, o que pode reduzir a tendência da rede para *overfit*. O melhor nível de ruído depende do problema e deve ser determinado por experimentação. O desvio padrão do ruído gaussiano adicionado à saída alvo durante o treinamento, neste caso, é 0,10.
- **Ajuste *Quickpropagation***
 - **Taxa de aprendizagem.** Especifique a taxa de aprendizagem inicial, aplicada na primeira época e, posteriormente, o algoritmo *Quick propagation* determina alterações de peso independente para cada peso.
 - **Aceleração.** Especifica a taxa máxima de aumento geométrico na mudança de peso que é permitido. Por exemplo, uma aceleração de dois permitirá a mudança de peso a não mais do que duas vezes em cada época.
 - **Adicionar um ruído gaussiano.** O mesmo que o *Backpropagation*.
- **Ajuste Delta-Bar-Delta**
 - **Taxa de aprendizagem.** As seguintes opções estão disponíveis no *group-box* taxa de aprendizagem: inicial. Especifique a taxa de aprendizado inicial usada para todos os pesos na primeira época. Posteriormente, cada peso desenvolve seu próprio ritmo de aprendizagem. Incremento. Especifique o incremento linear adicionado à taxa de aprendizagem de um peso, se a inclinação permanecer em uma direção consistente. Decaimento. Especifique o fator de decaimento

geométrico utilizado para reduzir a taxa de aprendizagem de um peso, se a inclinação mudar de direção. Especifique o coeficiente de suavização usado para atualizar o gradiente suavizado bar-Delta. Ele deve estar no intervalo (0, 1). Se o fator de alisamento for alto, o valor bar-Delta é atualizado lentamente para levar em conta alterações no gradiente. Em uma superfície de erro ruidosa isto permite que o algoritmo mantenha alta taxa de aprendizagem de acordo com o gradiente de base.

- **Adicionar um ruído gaussiano.** O mesmo que o *Backpropagation*.
- **Tamanho da amostra.**
 - O conjunto de séries de temporais será representado tendo uma sazonalidade base horaria correspondente a 1 semana (168 pontos), 2 semanas e 4 semanas (que é de aproximadamente um mês). Esta é uma tentativa para se assemelhar a problemas práticos. A semente para o erro aleatório foi registrada para fins de aferições futuras.
- **Método de amostragem.**
 - **Reamostragem aleatória.** No método de reamostragem aleatória (Monte Carlo), os subconjuntos são amostrados aleatoriamente a partir de casos disponíveis. Cada caso disponível é atribuído a um dos três subconjuntos (treinamento, seleção ou teste) utilizando a proporção 2: 1: 1.
 - **Reamostragem por validação cruzada.** A validação cruzada é N vezes, onde N é o número de amostras tomadas. A informação disponível é dividida em N partes, e uma parte é atribuída ao conjunto de teste em cada amostra. O restante é dividido entre o treinamento e subconjuntos de seleção e pode-se especificar quantos são colocadas em cada um. Muitos autores, quando se utiliza validação cruzada, não utilizam um conjunto de seleção do todo, com base no fato de qualquer polarização constituída por uma rede particular; pode ser compensada por uma média de previsões em todas as redes em conjunto. No entanto, provavelmente é ainda aconselhável tomar alguns passos para aliviar o excesso de aprendizagem, o que pode incluir o uso de um conjunto de seleção, a decadencial de peso ou condições de parada determinadas experimentalmente para ser razoável no domínio do problema.

3.3 Delineamentos e Resultados

Nesta seção serão discutidas as diretrizes relacionadas com a escolha do delineamento experimental e seus resultados estatísticos.

Considerando-se os 24 fatores no Quadro 2, se torna fácil avaliar a complexidade de funcionamento de todas as possibilidades combinatórias, enquanto simula-se a Rede Neural Artificial. No caso de considerarmos apenas dois níveis para rodar a simulação completa, o número de execuções seria 2^{24} , ou 16.777.216 simulações. Levando-se em conta 20 minutos como tempo aproximado para cada simulação em um computador rápido, o tempo necessário seria de 233.017 dias ininterruptos de trabalho, ou 638 anos,

É improvável que o método de tentativa e erro, o mais utilizado, encontre a melhor combinação para a simulação da Rede Neural Artificial. Felizmente, um grande número de projetos está disponível hoje em dia para lidar com os procedimentos de triagem. Alguns potenciais delineamentos e estratégias são adequados para a primeira fase de triagem, onde os objetivos são identificar os fatores que podem mais afetar o desempenho, filtrar variáveis irrelevantes e estabelecer relações de causa e efeito. O método de rastreio utilizado depende do número de fatores que precisam ser rastreados. Fatoriais fracionários estão entre os tipos mais utilizados de delineamentos na indústria (MYERS E MONTGOMERY, 1995). No entanto não são práticos, quando o número de fatores for superior a 20 e não mais do que dois níveis. Outros métodos de triagem para um grande número de variáveis incluem i) *Design Group Screening* (KLEIJNEN, 1987) com alguma desvantagem relacionada com a formação de grupos, ii) Bifurcação Sequencial (TROCINE E MALONE, 2000), com o inconveniente de ser limitada as variáveis quantitativas iii) *Iterated Designs Fractional* (ANDRES E HAJAS, 1993), com uma desvantagem de não ser capaz de avaliar a interação, e iv) *Trocine Screening Procedure* (TROCINE, 2001) com uma desvantagem de apenas ser capaz de avaliar três ou quatro fatores críticos. Em geral, estes métodos são úteis e melhores do que a tradicional estratégia de tentativa e erro.

3.3.1 Delineamento Taguchi de triagem para 19 fatores

Neste trabalho uma abordagem Taguchi é considerada como delineamento de triagem, devido a estrutura dos fatores apresentados no Quadro 2, com 11 fatores de dois níveis, oito fatores de três níveis e três fatores com quatro níveis condicionais. Dos 24 fatores, dois deles

foram estabelecidos como um constrangimento e excluídos do delineamento devido as seguintes razões: i) A arquitetura da RNA tem de ser definida a priori para que os fatores possam ser os mesmos. Usando uma RNA diferente implicaria em diferentes fatores e níveis, que não faz sentido neste caso. Apenas MLP foi aqui avaliada. ii) O horizonte de previsão foi definido para um único passo à frente, o que simplifica consideravelmente os resultados da comparação. Apenas aumentar este intervalo não adiciona nenhuma grande contribuição para a otimização da rede e ao objetivo da pesquisa.

O delineamento Taguchi com sua estrutura interna / externa permite um bom ajuste para os fatores do Quadro 2, como apresentados na Tabela 1. Nesta fase de triagem, os dois fatores e três níveis são usados como uma matriz interna em um delineamento L_{36} Taguchi. Os fatores condicionais (BP, QP e DD) estão relacionados com a sintonia do fator P1 (Fase um do algoritmo de treinamento) para a MLP e será utilizado aqui como a estrutura de matriz externa. Isto significa que, quando o nível de *Backpropagation*, *Quick propagation* ou *Delta-Bar-Delta* para o fator P1 é escolhido, um ajuste específico será utilizado para cada nível. Um delineamento Taguchi L_4 para $y_1 \dots y_4$ foi então escolhido. Este delineamento baseia-se nos níveis (A, B e C) dos parâmetros de ajuste dos fatores BP, QP e DD de acordo com a Tabela 1. A variável de resposta, a métrica MAPE e a estrutura de matriz exterior é condicionada ao supramencionado fator P1. Quando P1 assumir, por exemplo, o nível 1 (o que significa *Backpropagation*) irá configurar A (Ajuste da taxa de aprendizado e ritmo de cada época), B (Ordem de apresentação aleatória dos casos em cada época) e C (Adicionar ruído Gaussiano) como sim (1) ou não (2) de acordo com o delineamento L_4 Taguchi. Cada ensaio de L_{36} será repetido 4 vezes através de y_1 para y_4 . A ideia é usar um delineamento com um número mínimo de ensaios. Neste caso, o delineamento L_4 Taguchi (com quatro ensaios) é semelhante a um delineamento clássico 2^{3-1} (com também quatro ensaios). Ambos os modelos podem ser considerados de resolução III. Para um delineamento de ordem elevada, tal como o delineamento fatorial, o número de execuções seria dobrado (e, portanto, o número de simulações). Neste caso, apesar do aumento da resolução de um delineamento de ordem mais elevada, o custo em tempo de execução é decisivo. Desde a criação L_4 Taguchi é usado somente na fase de triagem, e apenas por fatores condicionais, os efeitos de confusão (devido a resolução DOE) poderia ser resolvido nos próximos delineamentos se o delineamento experimental seguinte fosse obtido e o MAPE, considerando-se 24 ou 12 pontos de amostra de treinamento, fosse então calculado.

Taguchi recomenda que a média da resposta e, também, a razão sinal-para-ruído (SN), para cada ensaio na matriz interna, necessitam ser avaliados. O SN aqui pode ser calculado usando a seguinte função:

$$SN = -10 \log \left(\frac{1}{n} \sum_{i=1}^n y_i^2 \right) \text{ com } n = 4 \quad (21)$$

O SN é expresso numa escala de decibéis e, neste nível de seleção, a estratégia Taguchi de análise é apenas para escolher o vencedor. Os melhores valores de nível para a função SN computadorizada são sempre os maiores. Os valores para a média dependem do tipo de problema e aqui é desejável que MAPE seja tão pequeno quanto possível. Os defensores de Taguchi alegam que a utilização da razão SN elimina geralmente a necessidade de examinar as interações específicas entre a matriz interna e os fatores de matriz externa. Observando-se tanto a média como as relações SN, conforme mostrado na Figura 6, é fácil escolher o nível de fatores que resultam ao mesmo tempo em menores valores de médias e maiores de SNs. Valores limítrofes serão deixados para futuros delineamentos onde as interações poderão ser melhor avaliadas.

A partir desta análise de triagem, alguns resultados foram estabelecidos, considerando-se o MAPE MEAN e o SN como desvio:

- a) Os fatores IM, SC, PU, EE e PI foram considerados não significativos e mais delineamentos poderiam eliminar ou defini-los como fatores de ruído. Estes são os fatores que os efeitos eram constantes para o nível do fator;
- b) Os fatores PT, P1 e P2 foram considerados significativos e foram estabelecidos os níveis para delineamentos futuros. Estes são os fatores que resultaram em bastante efeitos com maior SN e menor média, o que significa P1 (nível 1 = Backpropagation), P2 (nível 3 = Levenberg-Marquardt) e PT (nível 1 = série temporal univariada);
- c) Os fatores de HL, SM, UL, W1, W2, SS, OF, SP, Ep, LR e ET foram consideradas fronteiras, porque não era tão claro para eliminar ou para selecionar o nível do fator. Aqui, é necessária uma investigação mais aprofundada.
- d) Devido as semelhanças em termos de SN, bem como em termos de média, os fatores de fronteira de três níveis (UL, W1, W2 e SS) foram, então, definidos como fatores de dois níveis.
- e) Para as variáveis de ruído, os testes t estatísticos não rejeitaram a hipótese nula de médias iguais. Desta forma, é bastante confortável negligenciar as variáveis de ruído em projetos posteriores.

f) Uma descoberta interessante está relacionada com o número de entradas podadas e o modo de entrada. Nos 24 pontos para um problema de séries temporais univariada, o número de entradas podadas é pequeno na maioria dos casos (média de 3). Isto significa que todos os pontos são importantes para estabelecer a previsão. Como um problema de regressão, o número de podas, foi estatisticamente maior do que para o modo de série temporal univariada (média de 7). Isto significa que algumas técnicas de pré-processamento não colocaram qualquer contribuição para a eficiência da rede neural.

g)

Tabela 1: Arranjo cruzado de Taguchi

| Matriz Externa (L₄) para BP/QP/DD | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|----|----|----|----|----|----|----|----|----|----|-----------------------|----|----|----|----|----|------|----|-------|-------|----------------|----------------|----------------|----------------|---|
| | | | | | | | | | | | | | | | | | | | | | Y ₁ | Y ₂ | Y ₃ | Y ₄ | |
| | | | | | | | | | | | | | | | | | | | | | BP | 1 | 1 | 2 | 2 |
| | | | | | | | | | | | | | | | | | | | | | QP | 1 | 2 | 1 | 2 |
| | | | | | | | | | | | | | | | | | | | | | DD | 1 | 2 | 2 | 1 |
| Matriz Interna (L₃₆) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 11 fatores de 2 níveis | | | | | | | | | | | 8 fatores de 3 níveis | | | | | | MAPE | | | | | | | | |
| HL | OF | PT | SP | Ep | LR | IM | SC | ET | PU | SM | UL | P1 | P2 | EE | PI | W1 | W2 | SS | MAPE1 | MAPE2 | MAPE3 | MAPE4 | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0,056 | 0,066 | 0,071 | 0,073 | | | |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 0,073 | 0,078 | 0,073 | 0,075 | | | |
| 3 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0,084 | 0,076 | 0,074 | 0,085 | | | |
| 4 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 0,093 | 0,100 | 0,102 | 0,096 | | | |
| 5 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 0,092 | 0,096 | 0,091 | 0,097 | | | |
| 6 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 1 | 1 | 1 | 0,103 | 0,095 | 0,103 | 0,102 | | | |
| 7 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 2 | 3 | 1 | 2 | 3 | 0,093 | 0,085 | 0,095 | 0,094 | | | |
| 8 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 1 | 2 | 3 | 1 | 1 | 0,088 | 0,098 | 0,103 | 0,101 | | | |
| 9 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 1 | 2 | 3 | 1 | 2 | 0,150 | 0,147 | 0,144 | 0,142 | | | |
| 10 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 1 | 1 | 3 | 2 | 1 | 3 | 2 | 0,041 | 0,039 | 0,046 | 0,044 | | | |
| 11 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 3 | 2 | 1 | 3 | 0,090 | 0,093 | 0,091 | 0,097 | | | |
| 12 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 3 | 3 | 2 | 1 | 3 | 2 | 1 | 0,091 | 0,093 | 0,092 | 0,097 | | | |
| 13 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 2 | 3 | 1 | 3 | 2 | 1 | 0,075 | 0,071 | 0,063 | 0,067 | | | |
| 14 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 3 | 1 | 2 | 1 | 3 | 2 | 0,131 | 0,123 | 0,121 | 0,123 | | | |
| 15 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 3 | 1 | 2 | 3 | 2 | 1 | 3 | 0,084 | 0,093 | 0,098 | 0,086 | | | |
| 16 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 3 | 2 | 1 | 1 | 3 | 0,107 | 0,095 | 0,097 | 0,093 | | | |
| 17 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 3 | 1 | 3 | 2 | 2 | 1 | 0,132 | 0,124 | 0,128 | 0,129 | | | |
| 18 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 3 | 1 | 2 | 1 | 3 | 3 | 2 | 0,117 | 0,105 | 0,103 | 0,107 | | | |
| 19 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 1 | 3 | 3 | 3 | 1 | 0,124 | 0,131 | 0,137 | 0,125 | | | |
| 20 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 3 | 2 | 1 | 1 | 1 | 2 | 0,114 | 0,120 | 0,116 | 0,115 | | | |
| 21 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 3 | 1 | 3 | 2 | 2 | 2 | 3 | 0,094 | 0,099 | 0,100 | 0,100 | | | |
| 22 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 1 | 2 | 2 | 3 | 3 | 1 | 2 | 0,103 | 0,106 | 0,112 | 0,119 | | | |
| 23 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 3 | 3 | 1 | 1 | 2 | 3 | 0,108 | 0,117 | 0,116 | 0,117 | | | |
| 24 | 2 | 1 | 2 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 2 | 2 | 3 | 1 | 0,115 | 0,107 | 0,105 | 0,110 | | | |
| 25 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 1 | 3 | 2 | 1 | 2 | 3 | 3 | 0,109 | 0,117 | 0,113 | 0,111 | | | |
| 26 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 2 | 1 | 3 | 2 | 3 | 1 | 1 | 0,063 | 0,065 | 0,066 | 0,062 | | | |
| 27 | 2 | 1 | 1 | 2 | 2 | 2 | 1 | 2 | 2 | 1 | 1 | 3 | 2 | 1 | 3 | 1 | 2 | 2 | 0,111 | 0,111 | 0,115 | 0,115 | | | |
| 28 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 3 | 2 | 2 | 2 | 1 | 1 | 0,102 | 0,111 | 0,113 | 0,118 | | | |
| 29 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 3 | 3 | 3 | 2 | 2 | 0,088 | 0,081 | 0,091 | 0,078 | | | |
| 30 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 1 | 2 | 3 | 2 | 1 | 1 | 1 | 3 | 3 | 0,149 | 0,136 | 0,135 | 0,129 | | | |
| 31 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 3 | 3 | 3 | 2 | 3 | 2 | 0,096 | 0,103 | 0,104 | 0,105 | | | |
| 32 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 2 | 1 | 1 | 1 | 3 | 1 | 3 | 0,087 | 0,088 | 0,076 | 0,085 | | | |
| 33 | 2 | 2 | 1 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 3 | 2 | 2 | 2 | 1 | 2 | 1 | 0,098 | 0,099 | 0,105 | 0,099 | | | |
| 34 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 1 | 3 | 1 | 2 | 3 | 2 | 3 | 0,099 | 0,107 | 0,107 | 0,090 | | | |
| 35 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 2 | 3 | 1 | 3 | 1 | 0,063 | 0,056 | 0,062 | 0,067 | | | |
| 36 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 1 | 2 | 2 | 1 | 3 | 2 | 3 | 1 | 2 | 1 | 2 | 0,067 | 0,070 | 0,055 | 0,064 | | | |

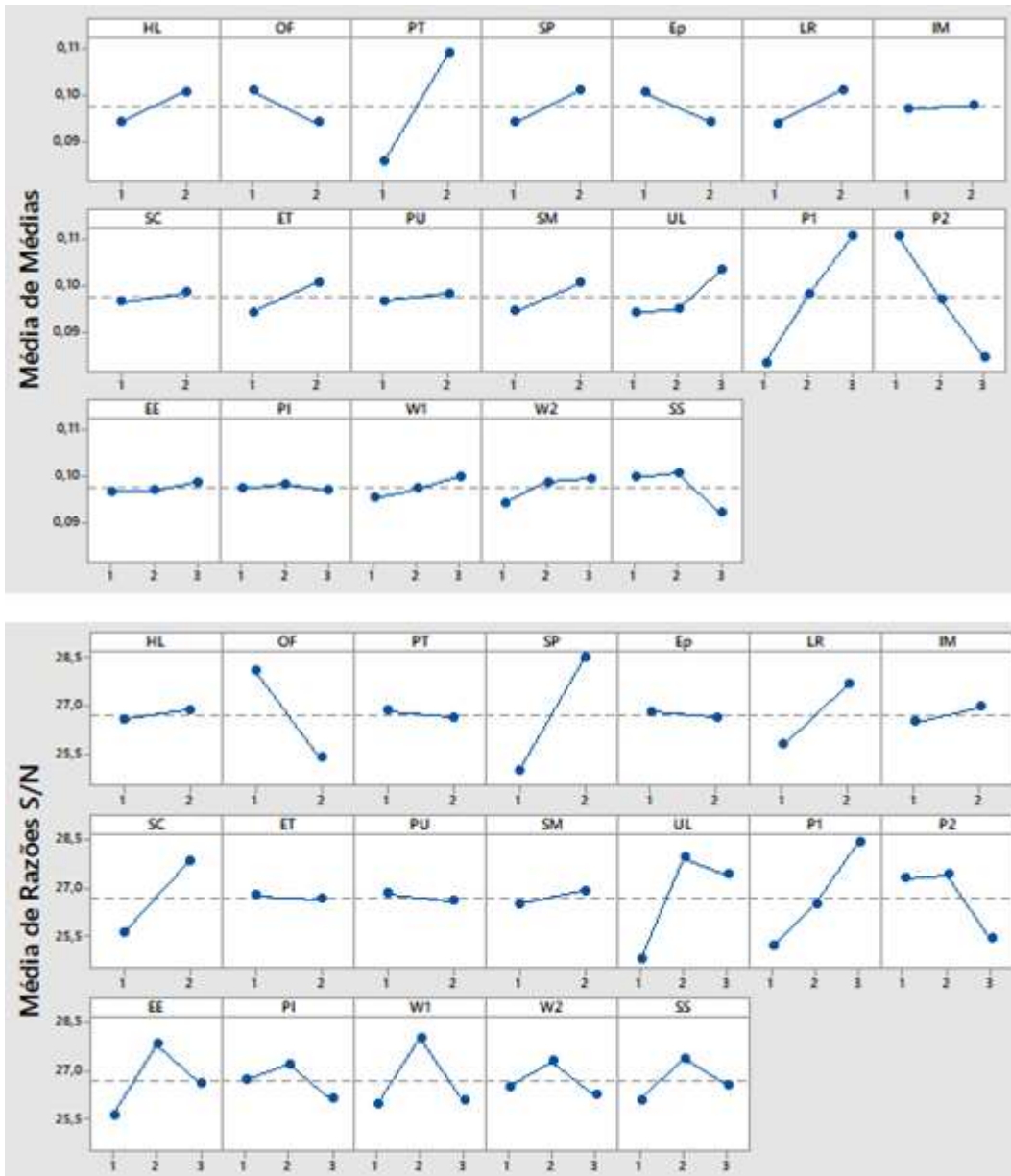


Figura 6: Média e SN para o delineamento de Taguchi (STAR2)

3.3.2 Delineamento 2_{III}^{11-7} Fatorial Fracionado para 11 Fatores

Para os 11 fatores restantes de dois níveis o delineamento Plackett-Burman fatorial fracionário de resolução III ou um delineamento L_{12} Taguchi poderiam ser usados potencialmente com os mesmos resultados e com a quantidade mínima de 12 repetições. Algumas opções próximas poderiam ser o delineamento Taguchi L_{16} ou o planejamento fatorial fracionário de resolução III, ambos com 16 repetições. Usando tanto Plackett-Burman e Taguchi, a análise da interação é muitas vezes difícil de se interpretar na prática. Se a escolha

é entre um delineamento geométrico com 16 repetições ou um delineamento Plackett-Burman com 12 repetições, que pode ter que ser dobrado (o que exige 24 repetições), o delineamento geométrico pode vir a ser uma escolha melhor (COLEMAN E MONTGOMERY, 1993). O delineamento fatorial fracionário geométrico (em unidades codificadas) foi então utilizado aqui e os resultados experimentais são apresentados na Tabela 2. MAPE 1, MAPE 2 e MAPE 3 consideraram as variáveis de ruído em seus níveis mais baixos, médios e superiores.

Tabela 2: Delineamento 2_{III}^{11-7}

| | HL | OF | SP | Ep | LR | ET | SM | UL | W1 | W2 | SS | MAPE 1 | MAPE 2 | MAPE 3 | Mean |
|----|----|----|----|----|----|----|----|----|----|----|----|--------|--------|--------|-------|
| 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 1 | 1 | 1 | 0,060 | 0,073 | 0,068 | 0,067 |
| 2 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | -1 | -1 | 0,060 | 0,054 | 0,054 | 0,056 |
| 3 | -1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | -1 | -1 | 1 | 0,012 | 0,004 | 0,003 | 0,006 |
| 4 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | 1 | 1 | -1 | 0,099 | 0,097 | 0,092 | 0,096 |
| 5 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | 0,072 | 0,074 | 0,074 | 0,074 |
| 6 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | 0,109 | 0,093 | 0,096 | 0,099 |
| 7 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 0,110 | 0,099 | 0,093 | 0,101 |
| 8 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | 1 | 1 | 0,056 | 0,055 | 0,053 | 0,055 |
| 9 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | 1 | 1 | 0,105 | 0,113 | 0,106 | 0,108 |
| 10 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | 0,041 | 0,038 | 0,046 | 0,042 |
| 11 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | 1 | -1 | 1 | 0,027 | 0,025 | 0,024 | 0,025 |
| 12 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | -1 | 0,105 | 0,112 | 0,114 | 0,110 |
| 13 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | 0,083 | 0,080 | 0,080 | 0,081 |
| 14 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | -1 | -1 | 1 | 0,118 | 0,118 | 0,121 | 0,119 |
| 15 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | -1 | -1 | 0,080 | 0,097 | 0,087 | 0,088 |
| 16 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0,093 | 0,095 | 0,105 | 0,098 |

Nesta segunda análise os resultados foram os seguintes:

- a) Os fatores ET e W1 foram considerados não significativos e mais delineamentos poderiam eliminar ou defini-los como um fator de ruído. Estes são fatores em que os efeitos foram constantes para o nível do fator.
- b) O fator LR foi considerado significativo e poderia ser estabelecido o seu nível para delineamentos futuros. Este é um fator cujo efeito resultou em erro médio menor.
- c) Os fatores HL, SM, UL, W2, SS, OF, SP, e Ep foram consideradas incertas, pois não era tão claro para eliminar ou para selecionar o nível do fator. Aqui, é necessária uma investigação mais aprofundada.

- d) Para as variáveis de ruído, o teste t não rejeitou a hipótese nula de médias iguais. Isso significa que a configuração obtida no passo anterior, usando o delineamento Taguchi, era coerente.

A Figura 7, seguinte, apresenta o gráfico de Pareto e os efeitos principais plotados para os fatores da Tabela 2.

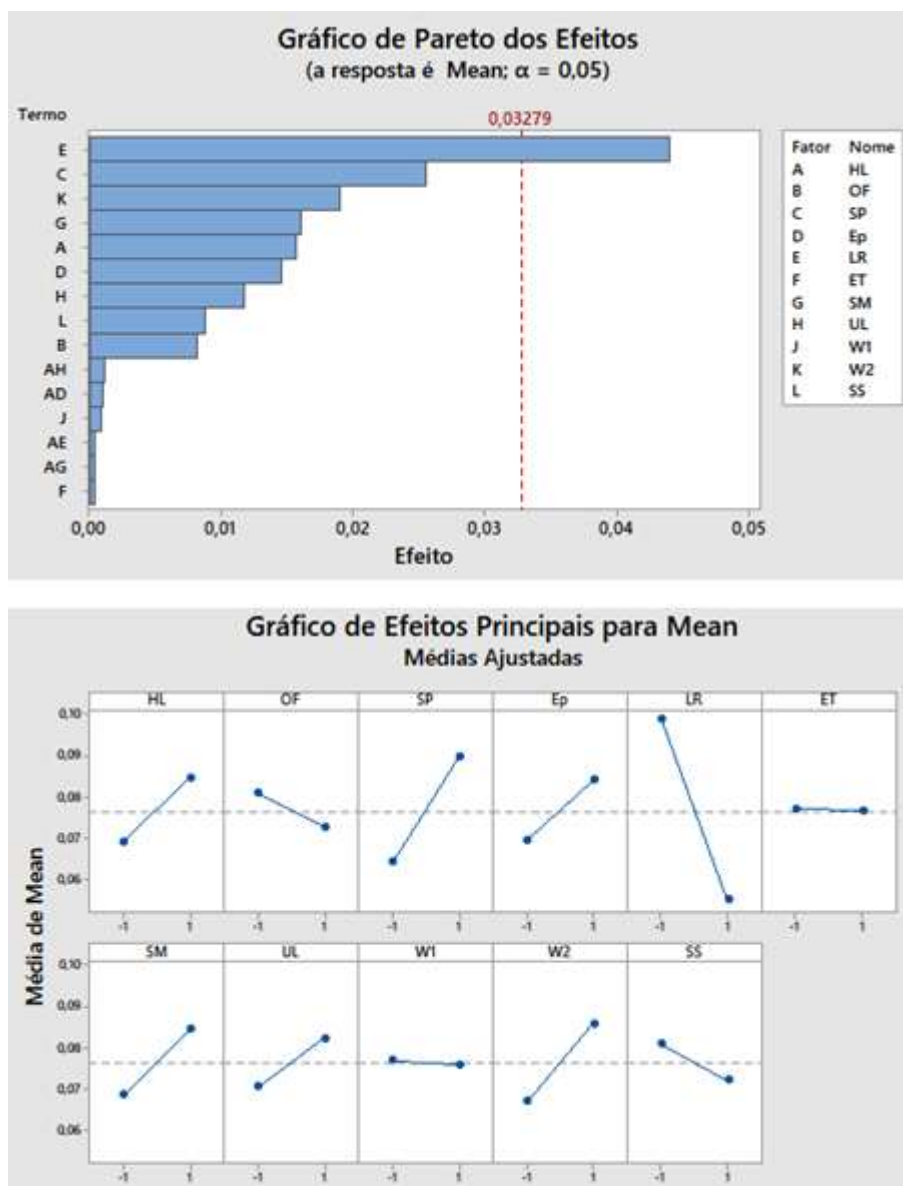


Figura 7: Delineamento 2^{11-7} , Modelo STAR2

3.3.3 Delineamento 2^{8-4}_{IV} Fatorial Fracionado para 8 Fatores

Para os oito fatores restantes com dois níveis o delineamento L_{12} Taguchi é a escolha com o número mínimo de repetições. Algumas opções próximas poderiam ser o delineamento

Taguchi L_{16} ou o planejamento fatorial fracionário de resolução IV, ambos com 16 repetições. Devido a análise da interação, a escolha de 2_{IV}^{8-4} é natural, considerando os mesmos critérios mencionados no projeto anterior.

Um delineamento fatorial fracionário de resolução IV tem os principais efeitos livres de interações de dois fatores. Algumas interações de dois fatores têm relações entre si. A resolução do delineamento IV que contém exatamente $2k$ repetições são chamados de delineamentos mínimos, como é no presente caso, em que $k = 8$.

Os resultados do delineamento mínimo 2_{IV}^{8-4} são apresentados na Tabela 3

A estrutura geradora para este delineamento mínimo 2_{IV}^{8-4} é representada por

$$A = BCD; F = ACD; G = ABC; H = ABD$$

Isto significa que um fator principal (A, F, G e H) é confundido com as interações de três fatores (BCD, ACD, ABC e ABD, respectivamente), que são geralmente de pequenos efeitos. Ao analisar os efeitos principais, maior a resolução é a melhor.

A Figura 8 apresenta os principais efeitos plotados para os fatores da Tabela 3. Algumas interações também são mostradas.

Tabela 3: Delineamento mínimo 2_{IV}^{8-4}

| | HL | SM | UL | W2 | ET | OF | SP | Ep | MAPE 1 | MAPE 2 | MAPE 3 | Mean |
|----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|---------------|---------------|---------------|-------------|
| 1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 0,045 | 0,058 | 0,053 | 0,052 |
| 2 | 1 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | 0,123 | 0,116 | 0,117 | 0,119 |
| 3 | -1 | 1 | -1 | -1 | 1 | 1 | -1 | 1 | 0,047 | 0,039 | 0,038 | 0,041 |
| 4 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | -1 | 0,060 | 0,058 | 0,053 | 0,057 |
| 5 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | -1 | 0,068 | 0,069 | 0,069 | 0,069 |
| 6 | 1 | -1 | 1 | -1 | -1 | 1 | -1 | 1 | 0,092 | 0,076 | 0,078 | 0,082 |
| 7 | -1 | 1 | 1 | -1 | -1 | -1 | 1 | 1 | 0,102 | 0,091 | 0,084 | 0,092 |
| 8 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 0,026 | 0,024 | 0,023 | 0,024 |
| 9 | -1 | -1 | -1 | 1 | -1 | 1 | 1 | 1 | 0,107 | 0,115 | 0,108 | 0,110 |
| 10 | 1 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 0,037 | 0,033 | 0,041 | 0,037 |
| 11 | -1 | 1 | -1 | 1 | 1 | -1 | 1 | -1 | 0,048 | 0,046 | 0,046 | 0,047 |
| 12 | 1 | 1 | -1 | 1 | -1 | -1 | -1 | 1 | 0,064 | 0,070 | 0,073 | 0,069 |
| 13 | -1 | -1 | 1 | 1 | 1 | -1 | -1 | 1 | 0,076 | 0,073 | 0,073 | 0,074 |
| 14 | 1 | -1 | 1 | 1 | -1 | -1 | 1 | -1 | 0,088 | 0,087 | 0,090 | 0,088 |
| 15 | -1 | 1 | 1 | 1 | -1 | 1 | -1 | -1 | 0,005 | 0,021 | 0,011 | 0,012 |
| 16 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0,079 | 0,082 | 0,091 | 0,084 |

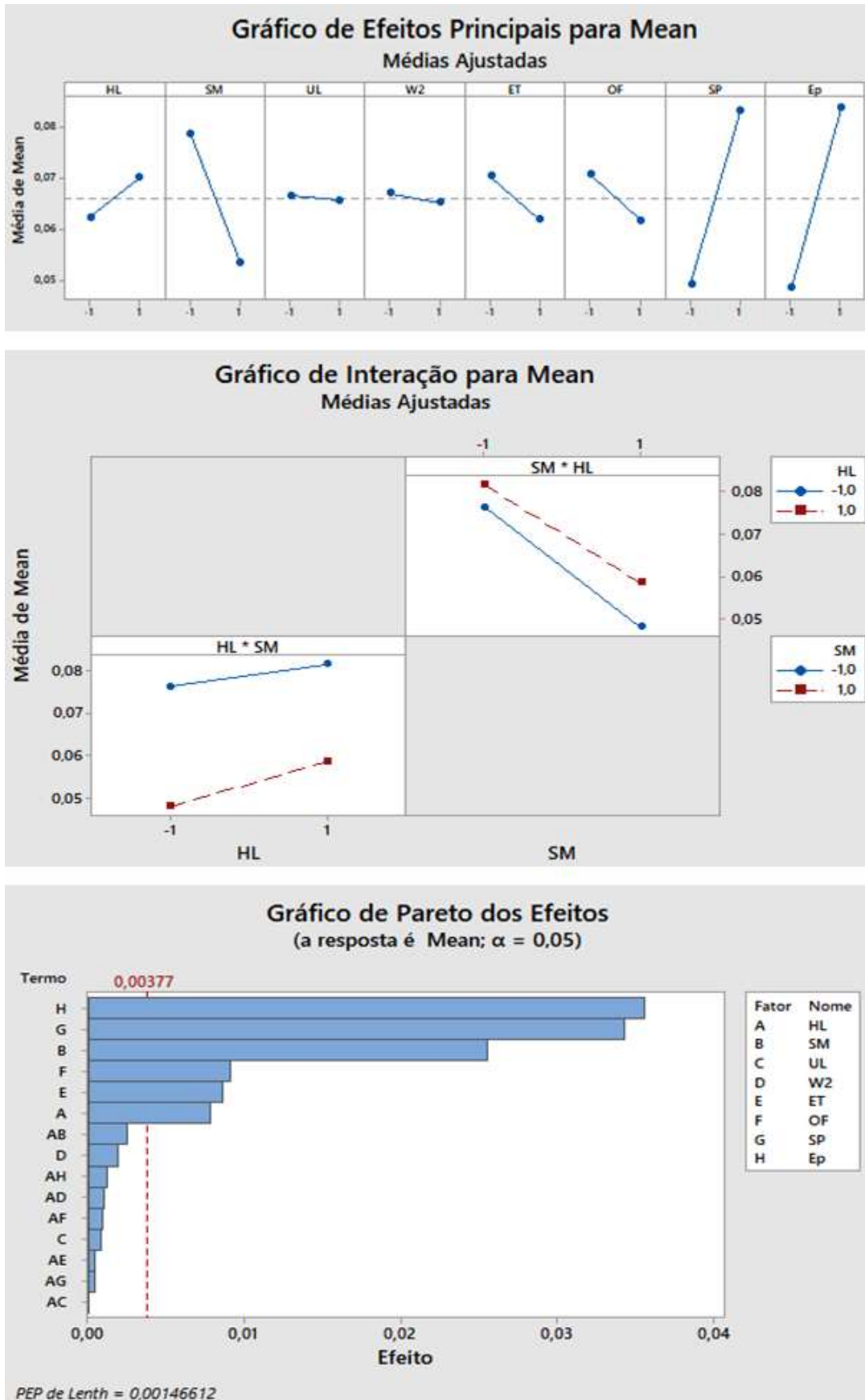


Figura 8: Delineamento mínimo 2^{8-4}_{IV} , Modelo STAR2

Similarmente ao delineamento anterior, este delineamento resultou em:

- a) O fator UL foi considerado não significativo. Seu efeito foi constante para o nível do fator.
- b) Os fatores SM, SP e Ep foram considerados significativos e seus níveis poderiam ser estabelecidos.
- c) Os fatores HL, UL, SS e SP foram considerados fronteiras. Uma interação de dois fatores foi maior do que um fator principal, mas não estatisticamente significativa.
- d) Para as variáveis de ruído, o teste t não rejeitou a hipótese nula de medias iguais.

3.3.4 Delineamento 2⁴ Fatorial Completo para 8 Fatores

Para os quatro fatores restantes de dois níveis de um delineamento fatorial completo com 16 repetições foi escolhido como mostra a Tabela 4.

| | OF | ET | HL | W2 | MAPE 1 | MAPE 2 | MAPE 3 | Mean |
|----|-----------|-----------|-----------|-----------|---------------|---------------|---------------|-------------|
| 1 | -1 | -1 | -1 | -1 | 0,10 | 0,12 | 0,11 | 0,11 |
| 2 | 1 | -1 | -1 | -1 | 0,07 | 0,06 | 0,06 | 0,07 |
| 3 | -1 | 1 | -1 | -1 | 0,07 | 0,06 | 0,06 | 0,06 |
| 4 | 1 | 1 | -1 | -1 | 0,05 | 0,05 | 0,05 | 0,05 |
| 5 | -1 | -1 | 1 | -1 | 0,07 | 0,07 | 0,07 | 0,07 |
| 6 | 1 | -1 | 1 | -1 | 0,04 | 0,02 | 0,02 | 0,03 |
| 7 | -1 | 1 | 1 | -1 | 0,03 | 0,02 | 0,01 | 0,02 |
| 8 | 1 | 1 | 1 | -1 | 0,02 | 0,01 | 0,01 | 0,01 |
| 9 | -1 | -1 | -1 | 1 | 0,12 | 0,12 | 0,12 | 0,12 |
| 10 | 1 | -1 | -1 | 1 | 0,07 | 0,07 | 0,08 | 0,07 |
| 11 | -1 | 1 | -1 | 1 | 0,07 | 0,07 | 0,07 | 0,07 |
| 12 | 1 | 1 | -1 | 1 | 0,06 | 0,06 | 0,07 | 0,06 |
| 13 | -1 | -1 | 1 | 1 | 0,08 | 0,08 | 0,08 | 0,08 |
| 14 | 1 | -1 | 1 | 1 | 0,03 | 0,03 | 0,04 | 0,03 |
| 15 | -1 | 1 | 1 | 1 | 0,02 | 0,04 | 0,03 | 0,03 |
| 16 | 1 | 1 | 1 | 1 | 0,02 | 0,02 | 0,03 | 0,02 |

Tabela 4: Delineamento fatorial completo

Os resultados experimentais estão apresentados na Figura 9

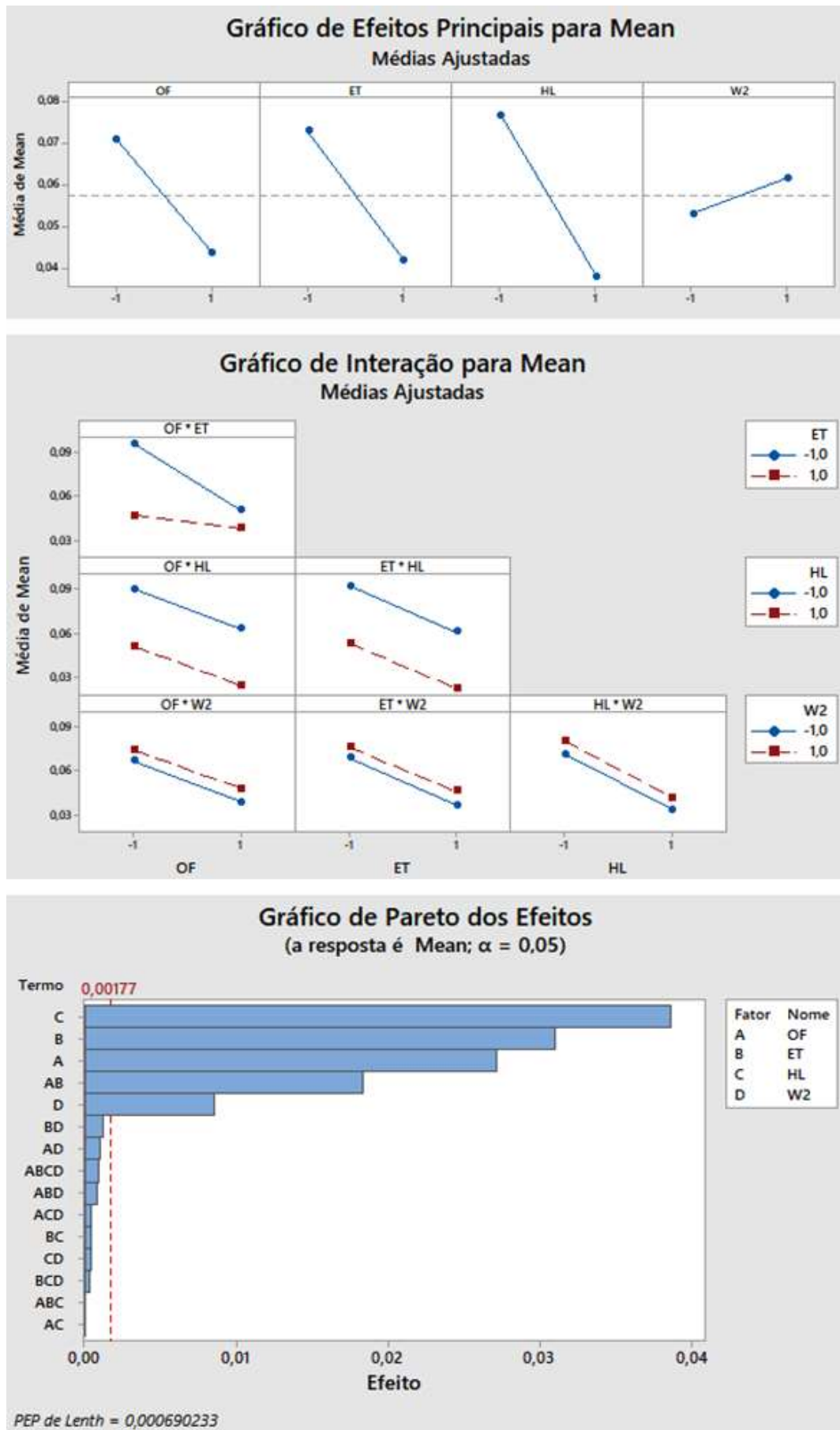


Figura 9: Delineamento mínimo 2^4 , Modelo STAR2

Para este delineamento fatorial completo os resultados foram:

Todos os fatores de OF, ET, HL e W2 foram considerados significativos e seus níveis poderiam ser estabelecidos.

- As interações de segunda ordem entre OF, ET, HL e W2 foram consideradas significativas.
- Para as variáveis de ruído, o teste t não rejeitou a hipótese nula de medias iguais

3.4 Testes de Confirmação

O Quadro 3 apresenta os resultados da simulação para as oito séries temporais não lineares. Os delineamentos utilizados em cada modelo não linear também são mencionados. Eles foram obtidos de acordo com o procedimento semelhante, como mostrado anteriormente. A MAPE final da RNA e a MAPE do modelo também foram calculados.

Quadro 3: Escolha dos fatores da Rede Neural Artificial para séries temporais não lineares

| Fator | Modelo não linear | | | | | | | |
|---------------|-------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | SAR | BL1 | BL2 | TAR | NAR | NMA | STAR1 | STAR2 |
| HL | 2* | 2* | 2* | 2* | 2* | 2* | 2* | 2* |
| SM | Random* | Aleatorio* | Aleatorio* | Aleatorio* | Aleatorio* | Aleatorio* | Aleatorio* | Aleatorio* |
| UL | 2(N+1)* | 2(N+1)* | 2(N+1)* | 2(N+1)* | 2(N+1)* | 2(N+1)* | 2(N+1)* | 2(N+1)* |
| W1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| W2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| SS | 21x24* | 21x24* | 21x24* | 21x24* | 21x24* | 21x24* | 21x24* | 21x24* |
| OF | Logistic* | Logistic* | Logistica* | Logistica* | Logistica* | Logistica* | Logistica* | Logistica* |
| SP | 12* | 12* | 12* | 12* | 12* | 12* | 12* | 12* |
| Ep | 400* | 400* | 400* | 400* | 400* | 400* | 400* | 400* |
| LR | 0.9 * | 0.9 * | 0.9 * | 0.9 * | 0.9 * | 0.9 * | 0.9 * | 0.9 * |
| ET | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P1 | BP* | BP* | BP* | BP* | BP* | BP* | BP* | BP* |
| P2 | Lev.Mar.* | Lev.Mar.* | Lev.Mar.* | Lev.Mar.* | Lev.Mar.* | Lev.Mar.* | Lev.Mar.* | Lev.Mar.* |
| PT | Univ.* | Univ.* | Univ.* | Univ.* | Univ.* | Univ.* | Univ.* | Univ.* |
| PU | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| EE | 25 | 25 | 25 | 25 | 25 | 25 | 25 | 25 |
| PI | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 | 0.05 |
| IM | N(0,1) | N(0,1) | N(0,1) | N(0,1) | N(0,1) | N(0,1) | N(0,1) | N(0,1) |
| SC | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| BP | Factors On | Factors On | Factors On | Factors On | Factors On | Factors On | Factors On | Factors On |
| QP | Factors On | Factors On | Factors On | Factors On | Factors On | Factors On | Factors On | Factors On |
| DD | Factors On | Factors On | Factors On | Factors On | Factors On | Factors On | Factors On | Factors On |
| Delineamentos | Tag $L_{36} L_4$ | $L_{36} L_4$ | $L_{36} L_4$ | $L_{36} L_4$ | $L_{36} L_4$ | $L_{36} L_4$ | $L_{36} L_4$ | $L_{36} L_4$ |
| MAPE (ANN) | 0.0435 | 0.0292 | 0.0194 | 0.0248 | 0.0366 | 0.0468 | 0.0246 | 0.0282 |
| MAPE (Model) | 0.0623 | 61 | 0.0656 | 0.0781 | 0.0664 | 0.0789 | 75 | 0.0788 |

* Efeito significante

As principais conclusões sobre estes resultados experimentais estão resumidas a seguir:

- a) Os fatores HL, SM, UL, SS, OF, P1, P2, PT foram considerados significativos para as oito séries temporais não lineares. As características de cada modelo não linear foram irrelevantes para se determinar os parâmetros da RNA.
- b) Para a maioria das séries temporais o Método de Amostragem (SM) interage significativamente com o Número de Unidades por camada (UL). Isto é teoricamente esperado, pois redes neurais complexas (como com quatro camadas, por exemplo) requerem mais amostras para o treinamento. O mesmo ocorre com a expectativa de HL e UL. Para a interação entre UL e HL os principais fatores atuam na maneira de diminuir a MAPE e os atos de interação no caminho oposto.

A Figura 10 mostra os resultados de previsão para o modelo SETAR2 usando os parâmetros otimizados da rede neural tal como apresentado no Quadro 3. A RNA apresenta um bom ajuste quando comparada ao modelo não linear específico.

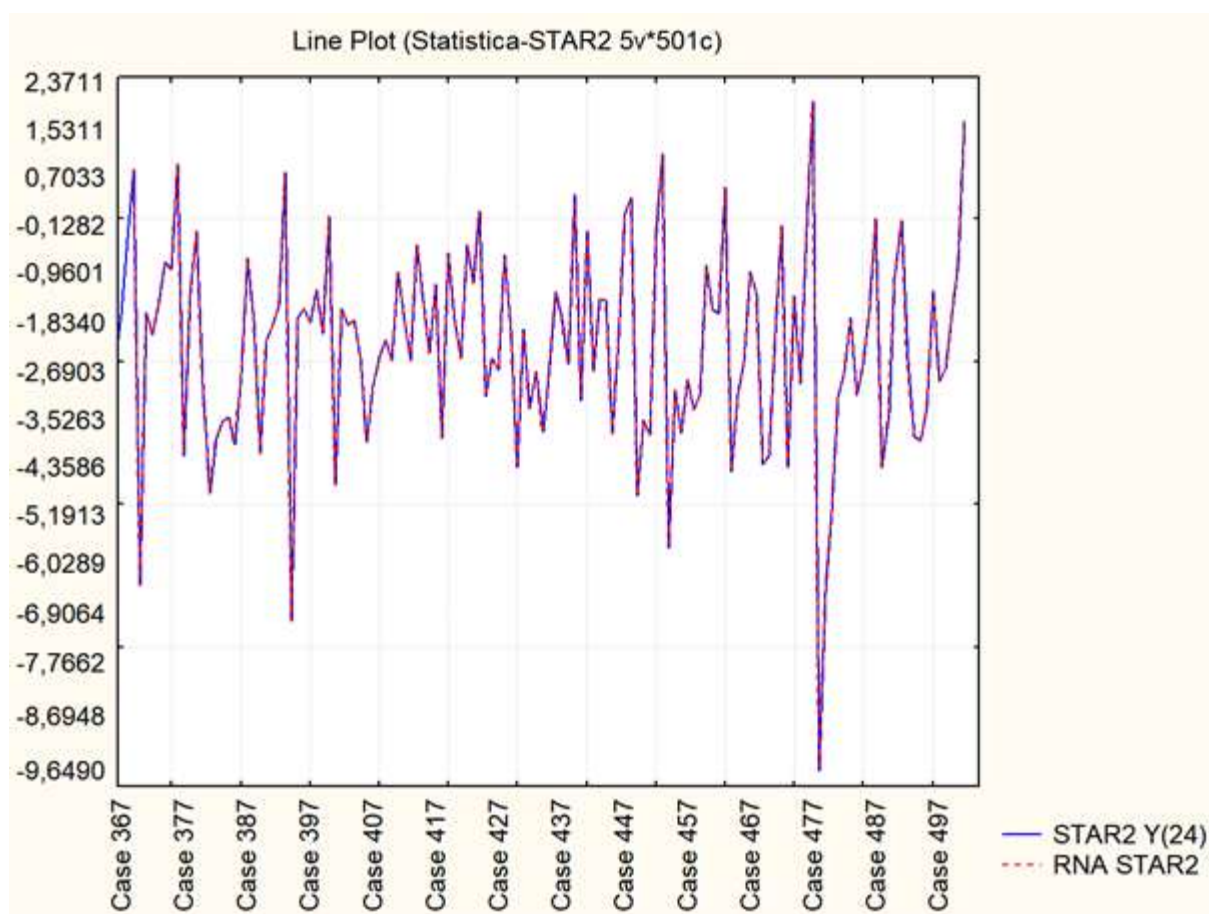


Figura 10: Previsão de RNA para o modelo STAR2 usando os parâmetros otimizados

4 Estudo de Caso: consumo de eletricidade de curto prazo

As empresas que comercializam eletricidade em mercados desregulamentados, como Estados Unidos, Brasil, Inglaterra e na maioria dos outros países, usam a previsão de séries temporais para calcular a demanda de eletricidade como parte das informações necessárias para definir contratos de compra e venda. Se forem empresas de produção ou comercialização, seus interesses residem principalmente em otimizar a carga de energia e os preços vendidos aos seus clientes. Este estudo de caso trata da modelagem e previsão de uma variável importante que afeta o alcance de um bom portfólio de contratos de eletricidade: a carga (ou consumo) de eletricidade. O conhecimento de seu valor futuro e de sua variação é essencial para calcular o risco e o retorno da carteira. Embora seja possível contratar uma determinada quantidade de energia, sem variação permitida para um consumidor, isto representa um alto risco. No processo industrial existem muitos fatores imprevisíveis que podem causar um aumento ou diminuição no consumo de energia. Se um produtor de eletricidade puder aceitar e gerenciar esse risco, o produtor terá mais oportunidades no mercado. Portanto, é importante gerenciar esses riscos.

Conforme considerado por Angelus (2001) e Mount (2001), prever os preços e cargas de eletricidade é uma tarefa árdua devido a um grande número de fatores, como demanda e fornecimento de eletricidade, número de gerações, restrições de transmissão e distribuição, etc. Para mercados desregulados, grande quantidade de energia elétrica é comprada e vendida tanto no mercado à vista quanto no mercado bilateral entre agentes de diferentes regiões geográficas. Uma simples mudança climática, imprevisível, pode adicionar volatilidade na demanda e, portanto, volatilidade nos preços. Pai e Hong (2006) também afirmam que a previsão da carga de eletricidade é complexa de conduzir devido à não linearidade de seus fatores influenciados. Eles afirmam que o fator mais importante na gestão da estratégia do sistema de energia regional ou nacional tem sido a previsão precisa da carga de demanda futura.

Este estudo de caso pretende usar a estrutura DOE / RNA para prever a carga de eletricidade de seis consumidores industriais no Brasil usando séries históricas horárias.

A Figura 11 representa o padrão da série temporal da carga horária de eletricidade para seis consumidores industriais. Procedimentos para melhorar a coleta de dados foram aplicados para tratar dados ausentes, *outliers*, erros de digitação, sazonalidade e erros. A sazonalidade da semana foi observada em todas as séries temporais (de defasagem 168). O conjunto de dados disponível compreende três anos de carga horária de eletricidade da Duke Energy, uma produtora que atua no Brasil.

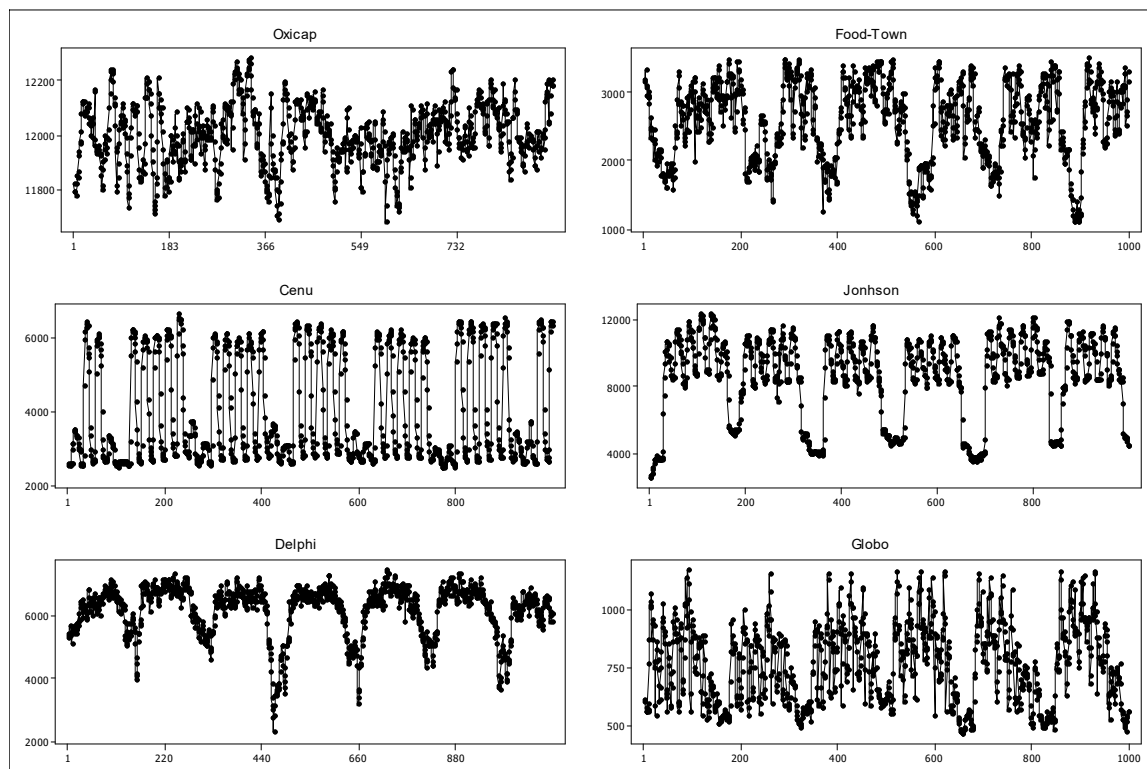


Figura 11: Séries temporais de cargas horárias (em Watts) de seis consumidores industriais

Um problema que surge ao lidar com várias séries temporais geralmente está relacionado à maneira como o método de previsão é tratado. A este respeito é importante referir que para todo este conjunto de carga elétrica o comportamento das seis séries temporais não pode ser considerado um processo multivariado. A correlação entre as variáveis não foi significativa em nenhum par das seis séries temporais. Além disso, alguns fatores, como dados ausentes, número diferente de amostras e intervalos diferentes, tornaram esse tipo de abordagem menos provável de ocorrer. Dessa forma, cada série precisa ser modelada de forma independente. Algumas séries temporais que vêm de clientes industriais com a mesma atividade podem ser explicadas pelas mesmas variáveis, mas este fato não foi considerado neste artigo. Apenas um método específico multivariado com os mesmos parâmetros, geralmente não é capaz de se ajustar a todo o conjunto de séries temporais.

O Quadro 4 apresenta os principais resultados do treinamento das RNA e previsão de carga elétrica para as seis séries temporais consideradas. Alguns modelos não lineares também foram estimados para fins comparativos e o erro MAPE foi obtido considerando $T = 24$ h (1 dia de antecedência) como intervalo de predição. Os resultados da MAPE são estatisticamente significativos quando comparados aos modelos não lineares adotados para cada série temporal.

É correto supor que as RNA foram bastante competentes em prever a carga de eletricidade de curto prazo para os consumidores industriais.

Quadro 4: Parâmetros da RNA para os seis consumidores industriais de carga de eletricidade

| Fator | Carga elétrica | | | | | |
|-------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| | Cenu | Food-Town | Oxicap | Delphi | Johnson | Globo |
| HL | 2 * | 2 ** | 2 * | 2 ** | 2 ** | 2 ** |
| SM | <i>Random**</i> | <i>Random**</i> | <i>Random*</i> | <i>Random*</i> | <i>Random*</i> | <i>Random**</i> |
| UL | 50 ** | 50 * | 50 * | 50 ** | 50 * | 50 * |
| W1 | Não | 0.001 | 0.001 | Não | 0.001 | 0.001 |
| W2 | Não | 0.001 | 0.01 | Não | 0.0001 | 0.01 |
| SS | 21 x 24** | 21 x 24* | 21 x 24** | 21 x 24* | 21 x 24* | 21 x 24** |
| OF | <i>Logistics**</i> | <i>Logistics**</i> | <i>Logistics**</i> | <i>Logistics*</i> | <i>Logistics*</i> | <i>Logistics**</i> |
| SP | 48 | 24* | 24 | 48* | 24 | 24* |
| Ep | 500 | 500 | 400* | 400 | 500 | 400 |
| LR | 0.9* | 0.9 | 0.9 | 0.9* | 0.9* | 0.9 |
| ET | -0.1 | -0.1 | 0 | 0 | 0 | 0 |
| P1 | BP** | BP* | BP* | BP** | BP* | BP** |
| P2 | Lev. Mar.* | Lev. Mar.** | Lev. Mar.** | Lev. Mar.* | Lev. Mar.* | Lev. Mar.* |
| PT | Univ.* | Univ.* | Univ.** | Univ.** | Univ.** | Univ.* |
| PU | <i>Fan-out</i> | <i>Fan-out</i> | <i>Fan-out</i> | <i>Sensitivity</i> | <i>Fan-out</i> | <i>Fan-out</i> |
| EE | 1 | 25 | 25 | 25 | 50 | 25 |
| PI | <i>Fan-out</i> | <i>Sensitivity</i> | <i>Fan-out</i> | <i>Sensitivity</i> | <i>Fan-out</i> | <i>Fan-out</i> |
| IM | <i>N(0.1)</i> | <i>U(0.1)</i> | <i>N(0.1)</i> | <i>N(0.1)</i> | <i>N(0.1)</i> | <i>U(0.1)</i> |
| SC | 0.1 | 0.1 | 0.1 | 0.1 | 0 | 0 |
| BP | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> |
| QP | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> |
| DD | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> | <i>Factors On</i> |
| Principais interações | HL.UL | SM.UL | HL.UL | HL.UL | HL.UL | HL.UL |
| | P2.PT | | OF.P2 | SM.UL | OF.P2 | OF.P2 |
| | | | SM.UL | | SM.UL | SM.UL |
| Delineamentos fatoriais | Tag L ₃₆ L ₄ | Tag L ₃₆ L ₄ | Tag L ₃₆ L ₄ | Tag L ₃₆ L ₄ | Tag L ₃₆ L ₄ | Tag L ₃₆ L ₄ |
| | 2_{III}^{10-6} | 2_{III}^{11-7} | 2_{III}^{9-5} | 2_{III}^{10-6} | 2_{III}^{10-6} | 2_{III}^{11-7} |
| | 2_{IV}^{7-3} | 2_{IV}^{8-4} | 2_{IV}^{6-2} | 2_{IV}^{7-3} | 2_{IV}^{6-2} | 2_{IV}^{7-3} |
| | 2 ⁴ | 2 ⁴ | 2 ³ | 2 ⁴ | 2 ³ | 3 |
| MAPE (ANN) | 0.102 | 0.053 | 0.113 | 0.095 | 0.872 | 0.105 |
| MAPE (Model) | 0.186 (STAR) | 0.101 (BILINEAR)) | 0.173 (BILINEAR) | 0.163 (STAR) | 0.107 (NAR) | 0.173 (BILINEAR) |

*P-value<0.05

**P-value<0.001

5 Conclusões e Trabalhos Futuros

Motivado pela falta de evidências de que uma RNA pode ser facilmente parametrizada, este estudo aplicou a metodologia do DOE para otimizar o treinamento da RNA para problemas de séries temporais não lineares. A pesquisa descreve uma abordagem baseada em DOE fatorial usando triagem, Taguchi, experimentos fatoriais fracionários e completos para definir os parâmetros de uma rede neural perceptron multicamadas *feedforward*. A abordagem usa experimentos fatoriais clássicos para definir sequencialmente os principais parâmetros da RNA para que um erro de predição mínimo possa ser alcançado.

Os principais fatores e interações foram identificados usando esta abordagem e os resultados sugerem que RNA usando DOE podem ter um desempenho melhor comparativamente aos modelos autorregressivos não lineares existentes.

Oito séries de tempo não lineares sintéticas foram estimadas usando RNA e os resultados mostraram que as RNA foram capazes de generalizar modelos autorregressivos realistas melhor do que o modelo específico proposto. Esta propriedade não linear geral torna as RNA uma alternativa promissora na previsão desses modelos. Seis séries temporais de carga elétrica industrial foram exploradas como um estudo de caso e os resultados confirmaram a estrutura DOE / RNA proposta.

Como uma pesquisa adicional, os seguintes pontos precisam ser melhor abordados:

1. A abordagem evolutiva (por exemplo, algoritmo genético) tem sido referida como uma exceção em termos de treinamento de uma RNA, em que níveis de evolução como pesos de conexão, estrutura de rede, regras de aprendizagem, etc., são otimizados usando um algoritmo evolutivo. No entanto, os estudos sobre a abordagem evolucionária geralmente estão preocupados com um certo nível de evolução, e a pesquisa sobre a evolução simultânea de vários níveis ainda está em seu estágio inicial (YAO, 1999).
2. Vários métodos diferentes foram propostos para construir a arquitetura ótima de uma RNA e consideram fatores topológicos como o número de entradas, o número de camadas ocultas e o número de neurônios. O algoritmo de poda (REED, 1993), o algoritmo de tempo polinomial (ROY, KIM, E MUKHOPADHYAY, 1993), a técnica de decomposição canônica (WANG, MASSIMO, THAM E MORRIS, 1994) e o critério de informação de rede (MURATA, YOSHIZAWA E AMARI, 1994) estão entre esses

métodos. Nenhum deles, entretanto, pode garantir a melhor solução de RNA para o problema de previsão de séries temporais não lineares. Uma pesquisa interessante é considerar esses métodos também como variáveis na presente estrutura DOE.

Referências Bibliográficas

- Aitkenhead, M.J.; McDonald, A.J.S.; Dawson, J.J.; Couper, G.; Smart, R.P., Billett, M., Hope, D., Palmer, S. A novel method for training neural networks for time- series prediction in environmental systems, *Ecol. Model.* 162 (2003) 87–95.
- Andres, T.H.; Hajas, W.C. Using iterated fractional factorial design to screen parameters in sensitivity analysis of a probabilistic risk assessment model, in: *Proceedings of the Joint International Conference on Mathematical Models and Supercomputing in Nuclear Applications*, Karlsruhe, Germany, 1993, pp.19–23.
- Angelus, A. Electricity price forecasting in deregulated markets, *Electricity J.* 4 (2001) 32–41.
- Armstrong, J.S.; Collopy, F. Error measures for generalizing about forecasting methods: empirical comparisons, *Int. J. Forecast* 8(1992)99–111.
- Balestrassi, P.P.; Popova, E.; Paiva, A.P.; Marangon Lima, J.W. Design of experiments on neural network's training for nonlinear time series forecasting, *Neurocomputing* 72 (2009)1160 – 1178
- Balkin, S.D.; Ord, J.K. Automatic neural network modeling for univariate time series, *Int. J. Forecast.* 16 (2000) 509–515.
- Bishop, C. *Neural Networks for Pattern Recognition*, University Press, Oxford, 1995.
- Bodyanskiy, Y.; Popov, S., Neural network approach to forecasting of quasiperiodic financial time series, *Eur. J. Oper.Res.*175(3) (2006) 1357–1366.
- Brown, R. G., *Smoothing, Forecasting and Prediction of Discrete Time Series*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
- Box, G.E.P.; Jenkins, G.M. *Time Series Analysis: Forecasting and Control*, revised ed., Holden-Day, San Francisco, 1976.
- BuHamra, S.; Smaoui, N.; Gabr, M., The Box–Jenkins analysis and neural networks: prediction and time series modeling, *Appl. Math. Model.* 27 (2003) 805–815.
- Chan, K.K.; Spedding, T.A., On-line optimization of quality in a manufacturing system, *Int. J. Prod. Res.* 39 (6) (2001) 1127–1145.
- Chen, R.; Tsay, R.S., Functional-coefficient autoregressive models, *J. Am. Stat. Assoc.* 88 (1993) 298–308.
- Chen, T.; Chen, H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems, *IEEE Trans. Neural Netw.* 6 (1995) 911–917.
- Cheng, B.; Titterington, D.M.; *Neural networks: a review from a statistical perspective*, *Stat. Sci.* 9 (1994) 2–54.
- Chiang, W.C.; Urban, T.L.; Baldrige, G.W. A neural network approach to mutual fund net asset value forecasting, *Omega*, *Int. J. Manage. Sci.* 24 (1996) 205–215.

- Coleman, D.E.; Montgomery, D.C. A systematic approach to planning for a designed industrial experiment *Technometrics* 35 (1993) 1–27.
- CraryGroup, 2004, <http://www.webdoe.cc/>.
- Cubiles-de-la-Vega, M.-D.; Mejias, R.P.; Acosta, A.P.; Garcia, J.M. *Intell.Data Anal.* 6(2002)53–65.
- De Gooijer, J.G. On threshold moving-average models, *J. Time Ser. Anal.* 19 (1998) 1–18.
- Derringer, G.; Suich, R. Simultaneous optimization of several response variables, *J. Qual. Technol.* 12 (4) (1980) 214–219.
- Diebold, F.X.; Nason, J.A. Nonparametric exchange rate prediction, *J. Int. Econ.* 28 (1990) 15–332.
- Enemuoh, E.U.; El-Gizawy, A.S. Optimal neural network model for characterization of process-induced damage in drilling carbon fiber reinforced epoxy composites, *Mach. Sci. Technol.* 7 (3) (2003) 389–400.
- Franses, P.H.; Van Homelen, P. On forecasting exchange rates using neural networks, *Appl. Financ. Econ.* 8 (1998) 689–696.
- Fu, M.C. Optimization for simulation: theory vs. practice, *INFORMS J. Comput.* 14 (2002) 192–215.
- Ghiassi, M.; Saidane, H.; Zimbra, D.K. A dynamic artificial neural network model for forecasting time series events, *Int. J. Forecast.* 21 (2005) 341–362.
- Granger, C.W.J.; Anderson, A.P. *An Introduction to Bilinear Time Series Models*, Vandenhoeck & Ruprecht, Gottingen, 1978.
- Granger, C.W.J.; Terasvirta, T. *Modelling Nonlinear Economic Relationships*, Oxford University Press, 1993.
- Hamilton, J.D. A new approach to the economic analysis of nonstationary time series and the business cycle, *Econometrica* 57 (1989) 357–384.
- Harrison, P. J., “Short-term sales forecasting,” *Appl. Statist.*, 14, 102–139, 1965.
- Ho, S.L.; Xie, M.; Goh, T.N. A comparative study of neural network and Box–Jenkins ARIMA modeling in time series prediction, *Comput. Ind. Eng.* 42 (2002) 371–375.
- Holt, C. C., F. Modigliani, J. F. Muth, and H. A. Simon, *Planning Production, Inventories and Work Force*, Prentice-Hall, Englewood Cliffs, NJ, 1963.
- Hornik, K. Some new results on neural network approximation, *Neural Netw.* 6 (1993) 1069–1072.
- Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators, *Neural Netw.* 2 (1989) 359–366.
- Hwang, H.B. Insights into neural-network forecasting of time series corresponding to ARMA (p; q) structures, *Omega* 29 (2001) 273–289.

- Hwang, H.B.; Ang, H.T. A simple neural network for ARMA (p; q) time series, *Omega* 29 (2001) 319–333.
- Kaasra, I.; Boyd, M. Designing a neural network for forecasting financial and economic time series, *Neurocomputing* 10 (1996) 215–236.
- Kaboudan, M. A. Genetically evolved models and normality of their fitted residuals. *Journal of Economic Dynamics & Control*, v. 25, n. 11, p. 1719-1749, 2001.
- Kalaitzakis, K.; Stavrakakis, G.S.; Anagnostakis, E.M. Short-term load forecasting based on artificial neural networks parallel implementation, *Electric Power Syst. Res.* 63 (2002) 185–196.
- Karunasinghe, D.S.K.; Liong, S.Y. Chaotic time series prediction with a global model: artificial neural network, *J. Hydrol.* 2 (2005) 1–14.
- Khaw, J.F.C.; Lim, B.S.; Lim, L.E.N. Optimal design of neural networks using the Taguchi method, *Neurocomputing* 7 (1995) 225–245.
- Kim, K.J. Financial time series forecasting using support vector machines, *Neurocomputing* 55 (2003) 307–319.
- Kim, Y.S.; Yum, B.J. Robust design of multilayer feedforward neural networks: an experimental approach, *Eng. Appl. Artif. Intell.* 17 (2004) 249–263.
- Kleijnen, J.P.C.; Sanchez, S.M.; Lucas, T.W.; Cioppa, T.M. State-of-the-art review: a user's guide to the brave new world of designing simulation experiments, *INFORMS J. Comput.* 7 (3) (2005) 263–289.
- Kleijnen, J.P.C., *Statistical Tools for Simulation Practitioners*, MarcelDekker Inc. Publ., NewYork,1987.
- Kohzadi, N.; Boyd, M.S.; Kermanshahi, B.; Kaasra, I. A comparison of artificial neural network and time series models for forecasting commodity prices, *Neurocomputing* 10 (1996) 169–181.
- Krager, H.; Kugle, P. Nonlinearities in foreign exchange markets: a different perspective, *J. Int. Money Finance* 12 (1993) 195–208.
- Lin, T.Y.; Tseng, C.H. Optimum design for artificial neural networks: an example in a bicycle derailleur system, *Eng. Appl. Artif. Intell.* 13 (2000) 3–14.
- Makridakis, S.; Andersen, A.; Carbone, R.; Fildes, R.; Hibon, M.; Lewandowski, R.; Newton, J.; Parzen, E.; Winkler, R. The accuracy of extrapolation (time series) methods: results of a forecasting competition, *J. Forecast.* 1 (1982) 111–153.
- Makridakis, S.; Chatfield, C.; Hibon, M.; Lawrence, M.; Mills, T.; Ord, K.; Simmons, L.F. The M-2 competition: a real-time judgmentally based forecasting study, *Int. J. Forecast.* 9 (1993) 5–23.
- Makridakis, S.; Hibon, M. The M3-competition: results, conclusions and implications, *Int. J. Forecast.* 16 (2000) 451–476.

- Mount, T. Market power and price volatility in restructured markets for electricity, *Decis. Support Syst.* 30 (2001) 311–325.
- Murata, N.; Yoshizawa, S.; Amari, S. Network information criterion-determining the number of hidden units for an artificial neural network model, *IEEE Trans. Neural Netw.* 5 (1994) 865–872.
- Myers, R.H.; Montgomery, D.C. *Response Surface Methodology: Process and Product Optimization Using Design Experiments*, Wiley, New York, 1995.
- Niska, H.; Hiltunen, T.; Karppinen, A.; Ruuskanen, J.; Kolehmainen, M. Evolving the neural network model for forecasting air pollution time series, *Eng. Appl. Artif. Intell.* 17 (2004) 159–167.
- Oliver & Boyd, Edinburgh, *Short Term Forecasting*, ICI Monograph 2, 1964.
- Pai, P.F.; Hong, W.C. Support vector machines with simulated annealing algorithms in electricity load forecasting, *Energy Convers. Manage.* 46 (2005) 2669–2688.
- Priestley, M.B. State-dependent models: a general approach to nonlinear time series analysis, *J. Time Ser. Anal.* 1 (1980) 47–71.
- Qi, M.; Zhang, G.P. An investigation of model selection criteria for neural network time series forecasting, *Eur. J. Operat. Res.* 132 (2001) 666–680.
- Reed, R. Pruning algorithms- a survey, *IEEE Trans. Neural Netw.* 4 (1993) 740–747.
- Roy, A.; Kim, L.S.; Mukhopadhyay, S. A polynomial time algorithm for the construction and training of a class of multilayer perceptrons, *Neural Netw.* 6 (1993) 535–545.
- Shi, Z.; Tamura, Y.; Ozaki, T. Nonlinear time series modelling with the radial basis function-based state-dependent autoregressive model, *Int. J. Syst. Sci.* 30 (7) (1999) 717–727.
- Shin, H.-C., Roth, H.R., Gao, M., Mollura, D., Summers, R.M. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning (2016) *IEEE Transactions on Medical Imaging* 35(5), 7404017, pp. 1285–1298
- Spall, J.C. *Introduction to Stochastic Search and Optimization; Estimation, Simulation, and Control*, Wiley, New York, 2003.
- StatSoft Inc., *Statistica (data analysis software system)*, version 7.1, 2005, www.statsoft.comS.
- Sukthomya, W.; Tannock, J. The optimization of neural network parameters using Taguchi's design of experiments approach: an application in manufacturing process modeling, *Neural Comput. Appl.* 14 (2005) 337–344.
- Sukthomya, W.; Tannock, J. The training of neural networks to model manufacturing processes, *J. Intell. Manuf.* 16 (2005) 39–51.
- Taguchi, G. *Systems of Experimental Design*, vols. 1 & 2, UNIPUB/Kraus International Publications, New York, 1987.

- Terasvirta, T.; VanDijk, D.; Medeiros, M.C. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: re-examination, *Int.J.Forecast.*21 (2005)755–774.
- Tong, H. On at threshold model, in: C.H. Chen (Ed.), *Pattern Recognition and Signal Processing*, Sijhoff&Noordhoff, Amsterdam,1978.
- Trocine, L. An efficient, effective, and robust procedure for screening more than 20 independent variables employing a genetic algorithm, Dissertation, University of Central Florida, Orlando,FL,USA,2001
- Trocine, A.L.; Malone, L.C. Finding important independent variables through screening designs: a comparison of methods, in: *Proceedings of the Winter Simulation Conference*,2000.
- Tsay, R. *Analysis of Financial Time Series*, seconded. Wiley-Interscience, 2005.
- Tseng, F.M.; Yu, H.C.; Tzeng, G.H., Combining neural network model with seasonal time series ARIMA model, *Technol.Forecast.Soc.Change*72 (69) (2002)71–87.
- Ullah, M. I., *Basic Statistics and Data Analysis*, <https://itfeature.com/time-series-analysis-and-forecasting/objectives-of-time-series-analysis> (2014)
- Wang, Z.; Massimo, C.D.; Tham, M.T.; Morris, A.J. A procedure for determining the topology of multilayer feedforward neural networks, *Neural Netw.*7 (1994)291–300.
- Yao, X. Evolving artificial neural networks, *Proc.IEEE*87(9) (1999)1423–1447.
- Zhang, G.P.; Patuwo, B.E.; Hu, M.Y. A simulation study of artificial neural networks for nonlinear time-series forecasting, *Comput.Operat.Res.*28 (2001)381–396.
- Zhang, G.P.; Patuwo, B.E.; Hu, M.Y., Forecasting with artificial neural networks: the state-of-the-art, *Int. J. Forecast.*14 (1998)35–62.
- Zhang, G.P.; Qi, M., Neural network forecasting for seasonal and trend time series, *Eur.J.Operat.Res.*160 (2005)501–514.
- Zhang, G.P., An investigation of neural networks for linear time-series forecasting, *Comput. Operat. Res.*28 (2001)1183–1202.
- Zhang, G.P. Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing*50(2003)159–175.