

PREVISÃO DOS RETORNOS DO IBOVESPA UTILIZANDO REDES NEURAIAS ARTIFICIAIS *FEEDFORWARD* EVOLUTIVAS

FORECASTING OF IBOVESPA RETURNS USING *FEEDFORWARD* EVOLUTIONARY ARTIFICIAL NEURAL NETWORKS

Edgar Leite dos Santos Filho* E-mail: edgarlsfilho@gmail.com

Wesley Vieira da Silva* E-mail: wesley.vieira@pucpr.br

Claudimar Pereira da Veiga* claudimar.veiga@gmail.com

Ubiratã Tortato* E-mail: ubirata.tortato@pucpr.br

*Pontifícia Universidade Católica do Paraná, PUC-PR, Curitiba, PR

Resumo: Diante dos desafios de se antecipar às incertezas e movimentos do mercado acionário e da necessidade de se tomar decisões de compra ou venda embasadas em métodos racionais, os operadores do mercado de ações encontram nos métodos estatísticos e econométricos o apoio para direcionar suas decisões. Em vários trabalhos científicos de previsão de séries financeiras, a escolha do método mais adequado permanece como uma preocupação central. Neste contexto, este artigo tem como objetivo a comparação dos desempenhos de uma rede neural artificial *feedforward* evolutiva (RNAE) e um modelo AR+GARCH, para a previsão um passo à frente de uma série temporal formada pelos retornos do IBOVESPA. A RNAE foi treinada com o algoritmo de evolução diferencial auto-adaptável e o modelo AR+GARCH ajustado para servir como referência de comparação. Como medidas de desempenho, foram utilizadas as métricas de análise da raiz do erro quadrático médio (REQM) e do índice de desigualdade de Theil (*U-Theil*). Neste estudo, a RNAE *Feedforward* obteve melhores resultados em relação ao coeficiente de desigualdade U-Theil, ajustando-se melhor e capturando o comportamento não linear dos retornos.

Palavras-chave: Previsão de Séries Temporais. Redes Neurais Artificiais Evolutivas. Evolução Diferencial. Modelagem GARCH.

Abstract: Facing the challenges of anticipating financial market uncertainties and movements, and the necessity of taking buy or sell decisions supported by rational methods, market traders found in statistics and econometrics methods, the base to support their decisions. In several scientific papers about forecasting financial time series, method selection keeps as central concern. This paper compares the performance of evolutionary feedforward artificial neural network (EANN) and an AR+GARCH model, for one step ahead forecasting of IBOVESPA returns. The EANN is trained by self-adapting differential evolution algorithm and AR+GARCH model is adjusted to be used as performance reference. The root mean square error (RMSE) and U-Theil inequality coefficient were used as performance metrics. Simulation results showed EANN feedforward achieved better results, fit better and captured the nonlinear behavior of returns.

Keywords: Time Series Forecasting. Evolutionary Artificial Neural Networks. Deferential Evolution. GARCH Modeling.

1 INTRODUÇÃO

Partindo-se do pressuposto que os fatores que influenciaram em padrões de comportamento do passado das séries temporais, continuarão a influenciar no comportamento futuro (DELURGIO, 1998; MORETTIN; TOLOI, 1987), torna-se necessária à utilização de métodos que capturem características do comportamento passado a fim de prever o próximo acontecimento com certo grau de confiança (BÓDIS, 2004). Muitos trabalhos na literatura sobre previsão em mercados financeiros utilizando séries temporais têm se centrado em modelos não-lineares univariados, uma vez que esses modelos capturam vários tipos de assimetria e outros fenômenos não-lineares (CHEN; GERLACH; TAI, 2008).

Este artigo considera a modelagem e previsão de séries temporais utilizando rede neural artificial *feedforward* evolutiva e modelo auto-regressivo heteroscedástico condicional generalizado. A modelagem abrange a especificação do modelo, estimação dos parâmetros e avaliação do modelo estimado.

Segundo Calôba e Saliby (2002) e Zhang (2003), as redes neurais representam uma metodologia eficaz para previsão de séries temporais não-lineares. Uma forma de estimação dos parâmetros, frequentemente utilizada em aplicações envolvendo RNAs, é o algoritmo *backpropagation*. Para Haykin (2001), a invenção do algoritmo de aprendizado por retropropagação para redes neurais *feedforward* é um marco no desenvolvimento histórico das redes neurais, pois somente após sua introdução, as propriedades das redes neurais foram reconhecidas, e as pesquisas conseguiram atenção de pesquisadores de diferentes áreas de conhecimento. Uma preocupação na aplicação desse algoritmo é a sua propensão em encontrar e prender-se a mínimos locais, frente a problemas que envolvem funções não-diferenciáveis ou multimodais (BERTSEKAS, 1995).

Em econometria, testes de especificação de modelos tem papel importante na avaliação do modelo. Usualmente esses testes preocupam-se com os resíduos do modelo estimado. Os resíduos são usados para testar propriedades de independência serial e normalidade do processo de erro. Em aplicações que envolvem RNAs, a previsão fora da amostra é a forma comum de avaliação de desempenho (RECH, 2002).

Devido à complexidade do processo de previsão e à sua aplicabilidade em diversas situações, em vários trabalhos científicos, a preocupação central é a escolha do método mais adequado. No caso da volatilidade de mercados financeiros, é difícil extrair um conjunto coerente de procedimentos mais adequados com base nos achados empíricos descritos na literatura existente (ANDERSEN; BOLLERSLEV; LANGE, 1999).

Neste contexto, o objetivo deste trabalho é comparar o desempenho de duas técnicas não-lineares para a previsão dos retornos do IBOVESPA: o modelo AR+GARCH e Redes Neurais Artificiais (RNA) *feedforward* evolutivas. Trata-se de uma pesquisa descritiva, predominantemente quantitativa e *ex-postfact*. Este trabalho encontra-se estruturado em cinco seções. A primeira seção refere-se à parte introdutória; a segunda traz uma breve revisão de literatura sobre o tema abordado; a terceira enfoca os aspectos metodológicos; a quarta mostra a apresentação e análise dos resultados empíricos e a quinta seção enfatiza as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICO-EMPÍRICA

Esta seção apresenta uma breve descrição sobre os modelos da família ARCH, as redes neurais artificiais *feedforward* evolutivas, o algoritmo de evolução diferencial e o IBOVESPA.

2.1 Retornos

Segundo Morettin e Tolo (2004), devido às características estatísticas das séries temporais financeiras é preferível trabalhar-se com séries de retornos em detrimento das séries de preços dos ativos. Se o preço P no tempo t for denotado por $p_t = \log P_t$, define-se o *retorno composto continuamente* ou simplesmente retorno como:

$$r_t = \log \frac{P_t}{P_{t-1}} = \log(1 + R_t) = p_t - p_{t-1} \quad (1)$$

Basicamente toma-se o logaritmo dos preços e em seguida obtêm-se os retornos com base no cálculo da primeira diferença da série de preços já

transformada. Nesta pesquisa utilizaremos a definição de retornos conforme equação (1).

2.2 Modelos da família ARCH

Enquanto uma regressão linear simples assume que o processo do erro da previsão é homocedástico e tem como foco a modelagem da média condicional, os modelos da família ARCH (Heterocedasticidade Condicional Auto-Regressiva, do inglês, *Autoregressive Conditional Heteroskedasticity*) introduzem uma segunda equação ao modelo de regressão padrão, que tem por objetivo a modelagem da variância condicional. Geralmente ajusta-se um modelo ARMA (p,q) para eliminar a autocorrelação da série temporal em análise e em seguida um modelo da família ARCH para os resíduos no caso dos quadrados dos resíduos apresentarem heterocedasticidade condicional.

Os resíduos são as diferenças entre o valor ajustado pela equação de média condicional e o valor observado na série temporal. Para que seja possível a aplicação de um modelo ARCH, esses desvios devem ser independentes e identicamente distribuídos com média zero e variância unitária. Com a intenção de verificar a presença de heterocedasticidade condicional na série de resíduos, examina-se a função de autocorrelação e autocorrelação parcial dos quadrados dos resíduos e determina-se a existência de heterocedasticidade pela função de autocorrelação e a quantidade de resíduos defasados necessários para modelar a heterocedasticidade pela função de autocorrelação parcial.

As alterações no comportamento da variância dos resíduos de uma série de dados é característica freqüentemente encontrada em séries temporais financeiras e é denominada heterocedasticidade. Neste tipo de série, a heterocedasticidade apresenta-se a partir de grandes (pequenos) retornos absolutos seguidos por outros grandes (pequenos) retornos. Isto significa que existem períodos que mostram agrupamentos de alta (baixa) volatilidade e estes podem ser observados em séries diárias e semanais, de ações, *commodities* e câmbio.

Na tentativa de trazer maior eficiência para a modelagem de séries temporais, especialmente as financeiras, Robert Engle (1982) apresentou o modelo de heterocedasticidade condicional auto-regressiva. O modelo ARCH apóia-se na

idéia de que o resíduo v_t do retorno de um ativo não é serialmente correlacionado, mas é dependente de α_t e esta dependência pode ser representada por uma função quadrática de seus valores defasados. Um modelo AR(1)+ARCH(1) pode ser representado pelas seguintes equações:

$$r_t = \phi_0 + \phi_1 r_{t-1} + v_t \quad (2)$$

$$v_t = \sqrt{\sigma_t^2} \varepsilon_t \quad (3)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 v_{t-1}^2, \quad (4)$$

onde $\{v_t\}$ é uma sequência de variáveis aleatórias independentes e identicamente distribuídas com média zero e variância 1, $\alpha_0 > 0$, e $\alpha_i \geq 0$ para todo $i > 0$. Na prática, pressupõe-se que v_t possui distribuição normal, t-Student ou distribuição generalizada de erro. Pela estrutura do modelo percebe-se que grandes variações do passado implicam em grande variância. Isto significa que grandes variações tendem a ser seguidas por outras grandes variações.

A equação (4) demonstra que a variância tem dois componentes: uma constante e a informação mais recente sobre a volatilidade, que neste caso modela-se com o resíduo quadrado do último período (o termo ARCH). Nesse modelo, v_t é heterocedástico condicional a v_{t-1} .

Como a variância de v_t na equação (2) depende apenas da volatilidade do último período, Pindyck e Rubinfeld (2004) referem-se a esse modelo com ARCH(1). De modo mais geral, a variância poderia depender de qualquer número de volatilidades defasadas. Em suma, pode-se descrever o modelo ARCH (m) como:

$$\sigma_t^2 = \alpha_0 + \alpha_1 v_{t-1}^2 + \dots + \alpha_m v_{t-m}^2 \quad (5)$$

Os modelos ARCH são usados para modelar a volatilidade do retorno de um ativo. A volatilidade tem sua utilidade em várias aplicações financeiras, como por exemplo, em precificação de opções e na alocação de recursos em estruturas de variância mínima.

Segundo Tsay (2005), a volatilidade não pode ser observada diretamente dos dados de retorno uma vez que existe somente uma observação em um dia de pregão. Se forem considerados dados intradiários (retorno de 15 minutos ou frequências maiores) é possível o cálculo da volatilidade diária, mas, será

necessária uma análise cuidadosa dos dados uma vez que dados de alta frequência possuem pouca informação sobre a volatilidade diária.

Embora não seja diretamente observável, a volatilidade apresenta os seguintes comportamentos comuns em séries de retornos de ativos financeiros: (i) agrupamento de volatilidade pode ser alta para alguns períodos e baixa para outros; (ii) evolui de maneira contínua ao longo do tempo, é raro acontecer saltos; (iii) não diverge para o infinito, as variações ficam restritas a uma faixa de valores (significa que a volatilidade é estatisticamente estacionária); e (iv) reage de forma diferenciada quando comparado seu comportamento frente a um retorno significativamente positivo e um retorno significativamente negativo. Este comportamento é conhecido como efeito de alavancagem.

É necessário considerar as seguintes desvantagens dos modelos ARCH: (i) o modelo assume que choques positivos e negativos têm o mesmo efeito sobre a volatilidade uma vez que ele depende do quadrado dos choques anterior, mas, na prática é sabido que as séries financeiras respondem de forma diferente diante de choques negativos e positivos; (ii) o modelo ARCH é restritivo uma vez que α_1^2 deve assumir valores no intervalo $[0, 1/3]$, se a série tem curtose finita. Na prática, essa restrição dificulta a modelagem com muitos parâmetros e limita a habilidade dos modelos ARCH em capturar excesso de curtose; (iii) o modelo ARCH não fornece informações sobre o que gerou as variações nas séries financeiras. É simplesmente uma forma de descrever o comportamento da variância condicional; (iv) é possível que o modelo superestime a volatilidade porque responde lentamente a grandes choques da série de retornos.

Outra desvantagem dos modelos ARCH é a necessidade frequente de se ajustar uma quantidade elevada de parâmetros para que seja possível a modelagem de uma série de retornos. Com a intenção de resolver o problema de excesso de parâmetros do modelo ARCH, Bollerslev (1986) propôs uma extensão conhecida como modelo ARCH generalizado (GARCH). O modelo adiciona uma variável à equação da variância para capturar o comportamento da variância condicional. Para uma série de retornos r_t , onde a variância depende do último resíduo da equação da média e da última variância condicional, segue o modelo AR(1)+GARCH(1, 1):

$$r_t = \phi_0 + \phi_1 r_{t-1} + v_t \quad (6)$$

$$v_t = \sqrt{\sigma_t^2} \varepsilon_t \quad (7)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 v_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \quad (8)$$

onde $\{v_t\}$ é uma sequência de variáveis aleatórias independentes e identicamente distribuídas com média zero e variância 1, $\alpha_0 > 0$, $\alpha_1 \geq 0$, $\beta_1 \geq 0$ e $(\alpha_1 + \beta_1) < 1$.

Os modelos GARCH podem ser comparados aos modelos ARMA no sentido de que são mais parcimoniosos e geram resultados melhores do que seus componentes, modelo ARCH puro para a modelagem da variância ou modelos AR ou MA no caso da modelagem da média condicional, quando utilizados de forma separada.

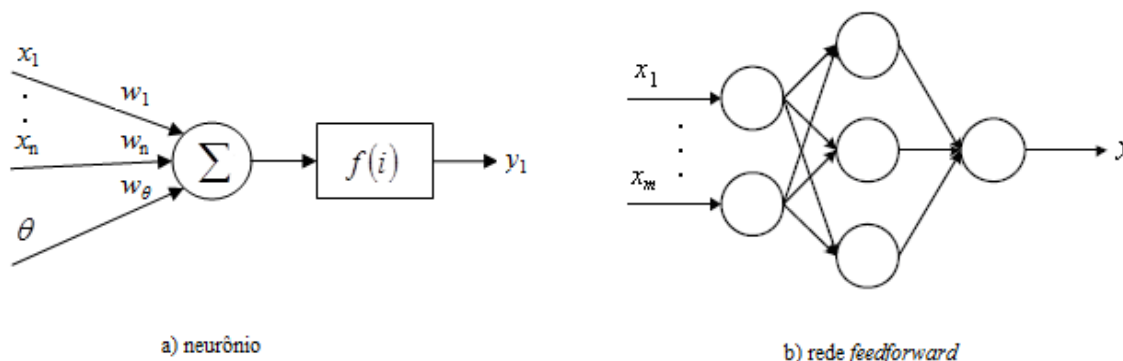
2.3 Redes Neurais Artificiais

Uma RNA é uma técnica para aprendizado, fundamentada em um modelo matemático que representa uma simplificação da representação do funcionamento do cérebro humano. A RNA pode ser descrita como um grafo no qual cada nó i executa uma função de transferência f_i na forma de

$$y_i = f_i \left(\sum_{j=1}^n w_{ij} x_j - \theta_i \right), \quad (9)$$

onde y_i é a saída do nó i , x_j é a j -ésima entrada para o nó, e w_{ij} é o peso da conexão entre os nós i e j . θ_i é o *bias* do nó. Usualmente, f_i é uma função não-linear, também denominada função de ativação, tais como função degrau, sigmóide ou gaussiana. As funções ativação mais comumente utilizadas são: função logística e função tangente hiperbólica. A Figura 1a apresenta a estrutura geral de um neurônio (HAYKIN, 2001).

Figura 1 - a) neurônio b) rede *feedforward*



Fonte: Adaptado de Haykin, 2001

Em termos de topologia, para definir-se uma RNA deve-se determinar as seguintes variáveis: (a) o número de nós na camada de entrada, (b) o número de camadas escondidas bem como a quantidade de neurônios a serem colocados nessas camadas, (c) o número de neurônios na camada de saída. O número de neurônios na camada de entrada corresponde ao número de variáveis mais importantes para o problema em estudo. Não existe um critério geral para definir o número de neurônios na camada escondida. Em geral, RNAs com poucos neurônios tendem a generalizar melhor a informação, o que conseqüentemente reduz a possibilidade de *overfitting*. Em relação ao fluxo de dados, quando as saídas dos neurônios de uma determinada camada forem propagadas somente para uma camada posterior, essa RNA é dita *feedforward* (RECH, 2002).

O aprendizado da RNA acontece por meio de ajustes nos pesos das conexões de cada entrada disponível em cada neurônio da RNA. Em termos de tipo de aprendizado tem-se o aprendizado supervisionado, o não supervisionado e o aprendizado por reforço como um caso específico de aprendizado supervisionado. O processo de aprendizado supervisionado é aquele onde a RNA recebe dados que representam o conteúdo a ser aprendido e a saída desejada. Geralmente os dados são separados em dois conjuntos: conjunto de treinamento e conjunto de testes. A partir da apresentação iterativa dos dados do conjunto de treinamento, o aprendizado é alcançado pelo ajuste dos pesos das conexões da RNA. O aprendizado supervisionado tem base na comparação direta entre a saída atual da rede e a saída desejada, também conhecida como saída alvo. O conjunto de testes

é utilizado para verificar a capacidade de generalização da RNA em condições reais de utilização (BÓDIS, 2004).

O treinamento dos pesos de uma RNA é formulado, usualmente, como a minimização de uma função de erro. Uma métrica frequentemente utilizada é a raiz do erro quadrático médio (REQM) que pode ser calculada com base no conjunto de valores obtidos como saída atual da rede e as saídas desejadas fornecidas no processo de treinamento. O processo de treinamento ajusta iterativamente os pesos das conexões até que seja encontrado um mínimo para a função. Muitos algoritmos de treinamento, tal como o *backpropagation* e o gradiente conjugado, utilizam o gradiente descendente da função de erro. Esses algoritmos frequentemente ficam presos em mínimos locais e são incapazes de encontrar um mínimo global se a função de erro não for diferenciável e/ou unimodal (BERTSEKAS, 1995).

Uma forma de contornar os problemas dos algoritmos baseados em gradiente é a adoção de algoritmos evolutivos, onde o processo de atualização dos pesos das conexões seja realizado por meio de cruzamentos e mutações resultantes da execução do algoritmo. Os algoritmos evolutivos (AEs) podem ser utilizados efetivamente na evolução global, para encontrar um conjunto de pesos das conexões próximos ao ótimo, e sem cálculo do gradiente (YAO, 1999). O valor do erro, objetivo de uma RNA, deve ser definido com base nas necessidades específicas da tarefa a ser executada. Dois fatores frequentemente considerados na formulação da função de erro são: o erro entre a saída deseje e a saída atual, e a complexidade da RNA.

Um ciclo típico da evolução das conexões dos pesos é mostrado na Figura 2, onde a evolução termina quando o erro atinge um valor menor ou igual a um valor predefinido ou a população tenha convergido.

Figura 2 - Ciclo típico da evolução das conexões dos pesos

1. Decodifica cada indivíduo da geração atual em um conjunto de pesos de conexões e construa uma RNA com os pesos.
2. Avalia a RNA computando o REQM total entre as saídas desejadas e obtidas. O *fitness* de um indivíduo pode ser determinado pelo erro. Quando mais alto o erro, menor o *fitness*. O mapeamento ótimo do erro é dependente do problema. Um termo de regularização pode ser incluído na função de *fitness* para penalizar pesos grandes.
3. Seleciona os pais para reprodução baseados em seus valores de *fitness*.
4. Aplica os operadores de busca (cruzamento e mutação) aos pais, para gerar os descendentes, para compor a próxima geração.

Fonte: Adaptado de Yao (1999)

2.4 Algoritmos Evolutivos

Os algoritmos evolutivos (AEs) referem-se à classe de algoritmos de busca, estocásticos, baseados em população, que são desenvolvidos a partir de idéias e princípios da evolução natural. Seguindo essas idéias estão incluídas as estratégias evolutivas, programação evolutiva, algoritmos genéticos e evolução diferencial. Uma característica de todos esses algoritmos é a busca baseada em população, onde os indivíduos de uma população competem e trocam informação entre si a fim de executar alguma tarefa. Uma estrutura geral de AEs pode ser descrita pela Figura 3.

Figura 3 - Estrutura geral de um algoritmo evolutivo

1. Gera uma população inicial $G(0)$ aleatoriamente e atribui $i=0$;
2. REPITA
 - (a) Avalia cada indivíduo na população;
 - (b) Seleciona os pais a partir da população $G(i)$ com base em suas funções de *fitness* em $G(i)$;
 - (c) Aplica as operações de busca aos pais e produza descendentes para formar $G(i+1)$;
 - (d) $i = i + 1$;
3. ATÉ “critério de parada” ser satisfeito.

Fonte: Adaptado de Yao (1999)

Segundo Yao (1999), os AEs são úteis para a resolução de problemas complexos que possuam vários ótimos locais. Eles são menos susceptíveis a ficar presos em mínimos locais do que os algoritmos de busca baseados em informação

do gradiente (*backpropagation*, Quase Newton BFGS, etc.). Eles podem ser aplicados a problemas onde uma função objetivo não exista ou não esteja explicitamente definida e, como não dependem da informação do gradiente, são adequados a problemas onde tal informação não está disponível ou é obtida ou estimada de forma custosa.

O Algoritmo de Evolução Diferencial (ED) é um algoritmo de otimização, proposto por Price e Storn (1997), tem sua população representada por conjunto de vetores solução, os quais são sucessivamente atualizados por operações de adição e subtração (mutação) e troca de componentes (*crossover*), até a convergência da população. A mutação e o *crossover* são usados para gerar novos conjuntos de vetores solução.

Em uma população de possíveis soluções para um problema de otimização dentro de um espaço de busca n-dimensional, um número fixo de vetores são inicializados aleatoriamente e, em seguida novas populações são evoluídas ao longo do tempo para explorar o espaço de busca e localizar os mínimos da função objetivo.

ED utiliza operações bio-inspiradas de cruzamento, mutação e seleção como mecanismos de evolução. O algoritmo também emprega um cruzamento não-uniforme que privilegia um determinado vetor ancestral em detrimento de outros ancestrais utilizados no cruzamento. A principal diferença entre um algoritmo genético e ED é a operação de mutação. Nos algoritmos genéticos, a mutação é causada por pequenas alterações nos genes e no algoritmo de ED é realizada com base na combinação de indivíduos (DU, 2007). O cruzamento não-uniforme embaralha informações das combinações com melhor *fitness* e faz com que a busca seja executada no espaço de soluções. O algoritmo ED introduz uma operação de mutação baseada em diferenças entre pares de soluções selecionadas de forma aleatória na população.

Como todos os algoritmos genéticos, ED manipula uma população de soluções. A população p de uma geração g tem NP vetores (indivíduos X^g). Cada indivíduo representa uma solução potencial:

$$p = \{X_i^g\}, i = 1, 2, \dots, NP. \quad (10)$$

Por sua vez, a solução X^g contém D variáveis, tal que:

$$X_i^g = x_{ij}^g, j = 1, 2, \dots, D. \quad (11)$$

Para cada geração, os indivíduos são atualizados num esquema de reprodução. Para produzir um novo indivíduo, as operações de diferenciação e recombinação são aplicadas ao conjunto de soluções gerado.

Um conjunto de indivíduos é extraído aleatoriamente para a operação de diferenciação. Um vetor de diferença δ e um vetor base β são projetados com base nestes indivíduos extraídos. Assim, o resultado da operação de diferenciação é

$$\omega = \beta + F\delta, \quad (12)$$

onde $F > 0$ é a constante de diferenciação.

A próxima operação de combinação representa um caso típico de troca de partes de um indivíduo. Um indivíduo tentativa ω herda cromossomos com alguma probabilidade. Assim,

$$\omega = \begin{cases} \omega_j, & \text{se } rand < CR, \\ x_{i,j}^g, & \text{qualquer outro caso,} \end{cases} \quad (13)$$

onde $j = 1, 2, \dots, D$ e $CR \in [0, 1]$ é a constante de recombinação.

Os parâmetros F e CR afetam a velocidade de convergência e a robustez do processo de busca. Os valores ótimos dependem das características da função objetivo e do tamanho da população NP , e assim, a seleção dos valores de parâmetros ótimos depende da aplicação, onde:

$$X_i^{g+1} = \begin{cases} \omega & \text{se } f(\omega) < f(X_i^g), \\ x_i^g & \text{qualquer outro caso.} \end{cases} \quad (14)$$

Para todos os algoritmos evolutivos os parâmetros operacionais controlam o nível entre *exploitation* (usar o material existente na busca de uma solução melhor) e *exploration* (procurar por genes melhores).

Price e Storn (1997) propõem o tamanho da população de 5 a 20 vezes a dimensionalidade do problema, com multiplicadores de 2 e 100 representando os valores, mínimo e máximo. Estes valores têm funcionado na prática, demonstrando que a quantidade de material genético contida nas populações é suficiente.

Entretanto, outras pesquisas com algoritmos evolutivos produziram melhores resultados com multiplicadores entre 1,5 e 2 (MAYER, 2002).

Para seleção de ancestrais, uma grande quantidade de métodos de seleção foi utilizada no passado, incluindo roleta (a escolha tradicional para algoritmo genético), seleção completa, truncamento e torneio (onde o tamanho 2 mostra-se como o padrão atual). O algoritmo ED utiliza a seleção completa (cada ancestral é considerada em cada geração).

Em relação à taxa de recombinação, vários estudos sobre algoritmos evolutivos mostraram que recombinação e mutação têm um efeito sinérgico (MICHALEWICZ; FOGEL, 2000). A ED incorpora ambos em uma só operação usando uma forma de cruzamento uniforme. Em ED, a recombinação é controlada por um *CR* especificado pelo usuário e para cada parâmetro, um alelo modificado ou seu ancestral é utilizado. Storn e Price (1997) sugeriram valores de *CR* entre 0 e 1 para uma otimização detalhada (apesar de lenta), 1 e 0 para convergência mais rápida (apesar de arriscada), e 0,5 como valor recomendado.

ED não tem uma taxa de mutação definida e ao invés disso obtém esse parâmetro como um efeito do *CR*. Estudos anteriores mostraram que tanto taxas baixas (0,01) quanto altas (1,0) são efetivas (MAYER, 2002). Um *CR* de 0,5 pode ser utilizado como valor inicial para taxa de mutação, mas estudos mostraram que a forma de mutação aplicada não é importante contanto que exista uma para guiar o processo de exploração.

De forma única, que permite auto-adaptação da quantidade de mutação ao longo do progresso da otimização, a *ED* adiciona uma diferença entre dois ancestrais para um terceiro ancestral. Esta é uma forma aritmética aplicada para cada valor real que poder ser intermediário ou extrapolativo, dependendo de um fator de escala (*F*). Storn e Price (1997) recomendam um valor de *F* entre 0,4 e 1, tendo 0,5 como escolha inicial.

O funcionamento básico do algoritmo é demonstrado pela equação (15) de forma que para cada vetor solução $X_{i,G}$, $i = 0, 1, 2, \dots, NP-1$, um vetor tentativa \underline{V} é gerado conforme

$$\underline{V} = X_{r_1,G} + F(X_{r_2,G} - X_{r_3,G}), \quad (15)$$

com $r_1, r_2, r_3 \in [0, NP-1]$, inteiro e diferentes, e $F > 0$.

Os inteiros r_1 , r_2 e r_3 são escolhidos aleatoriamente de um intervalo $[0, NP-1]$ e são diferentes do índice corrente i . F é um fator real e constante que controla a amplificação da variação diferencial $(\underline{X}_{r_2,G} - \underline{X}_{r_3,G})$. O vetor \underline{X}_{r_1} é selecionado aleatoriamente. Os vetores \underline{X}_{r_2} e \underline{X}_{r_3} são selecionados aleatoriamente, executada a operação de diferença e seu resultado multiplicado pelo fator de amplificação F . Em seguida, é executada a operação de adição para a geração do vetor \underline{V} .

Para aumentar a diversidade dos vetores de parâmetro, é gerado um vetor u , definido conforme as equações (16) e (17), e com base em um vetor tentativa, que tem seus valores modificados segundo a probabilidade definida em CR.

$$u = (u_0, u_1, \dots, u_{D-1})^T \quad (16)$$

Com

$$u_j = \begin{cases} v_j, & \text{para } j = \langle n \rangle_D, \langle n+1 \rangle_D, \dots, \langle n+L-1 \rangle_D \\ (X_{i,G})_j, & \text{para todos os outros } j \in [0, D-1], \end{cases} \quad (17)$$

onde $\langle \cdot \rangle_D$ é a função módulo com módulo D .

A equação (16) produz uma seqüência de vetores u com tamanho idêntico a v , onde alguns elementos do vetor recebem os valores originais de v e outros, com base em uma probabilidade determinada pelo parâmetro CR são modificados a título de cruzamento.

3 ASPECTOS METODOLÓGICOS

Para caracterizar o presente trabalho foi utilizada a classificação de pesquisas científicas definida por Gil (2002), que considera a pesquisa em relação aos seus objetivos. Quanto à natureza da pesquisa, pode ser caracterizado como uma pesquisa descritiva, pois tem como objetivo de avaliar a existência de associações entre variáveis. Trata-se de uma pesquisa predominantemente quantitativa de corte temporal seccional com avaliação longitudinal.

3.1 População e amostra

Os dados da série temporal do índice BOVESPA à vista, foram conseguidos no sítio *www.finance.yahoo.com*. O período de dados considerado tem início em

01/06/2004 e término em 30/05/2008 considerando cotações diárias com base nos valores de fechamento do índice.

Do conjunto de valores inicialmente selecionado para a amostra, 01/06/2004 e término em 30/05/2008, um subconjunto com início em 01/06/2004 e término em 30/07/2007 foi utilizado para a fase de estimação (treinamento) e primeira previsão um passo à frente. Em seguida, ciclos de previsão um passo à frente foram realizados, sempre considerando um retorno mais novo e desprezando-se o mais antigo. As previsões obtidas nesses ciclos são consideradas para fins de validação (teste). Um total de 197 ciclos é considerado, entre as datas 01/08/2007 e 30/05/2008.

3.2 Especificações dos modelos

Para a modelagem da série de retornos do IBOVESPA utilizando a técnica GARCH, o requisito de que os dados sejam estacionários deve ser considerado. Com a intenção de obter um conjunto de dados estacionários e dessa forma eliminar componentes de tendência e sazonalidade, é realizada uma transformação logarítmica na série de preços de fechamento e tomada a sua primeira diferença de forma a conseguir a série de retornos estacionária.

O teste de raízes unitárias de Dickey-Fuller aumentado rejeita a hipótese nula de existência de raízes unitárias. Nesse caso confirma-se a hipótese alternativa de estacionariedade da série de retornos conforme apresentado na tabela (1).

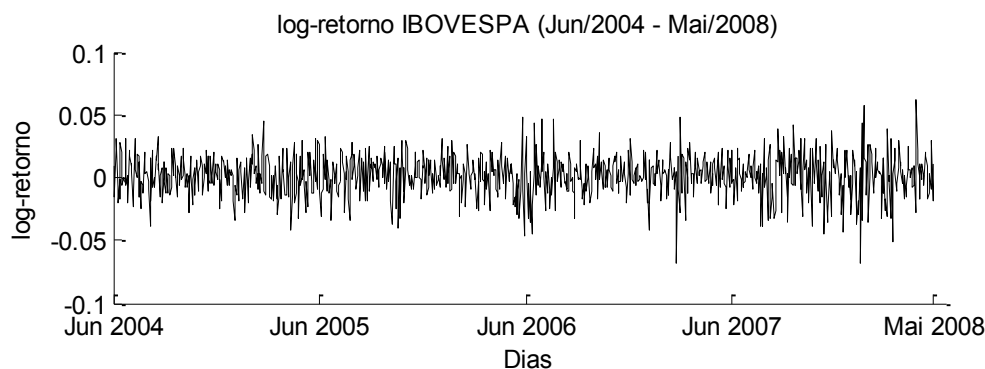
Tabela 1 – Dickey-Fuller aumentado aplicado à série de retornos

1 – Rejeita Hipótese de Raiz Unitária	p-value	Estatística	Valor Crítico
0 – Aceita Hipótese de Raiz Unitária			
1.0000	0.0010	-27.8772	-1.9414

Fonte: Pesquisa Direta

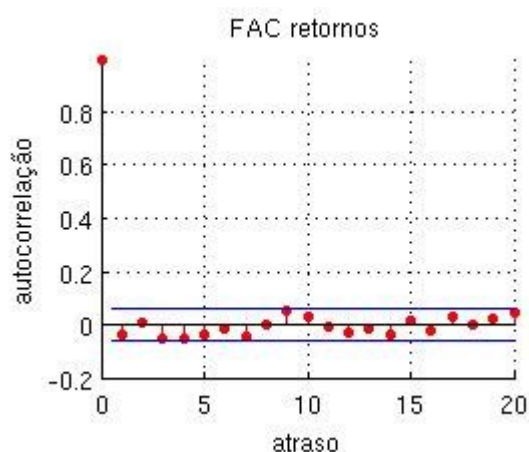
A figura (4) mostra o gráfico da série de retornos obtida após as transformações e diferenciação. As figuras (5) e (6) apresentam as funções de autocorrelação e autocorrelação parcial da série de retornos.

Figura 4 - Série de log-retornos do IBOVESPA



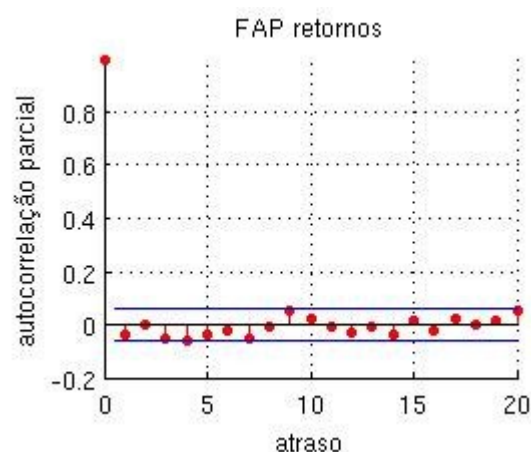
Fonte: Pesquisa Direta

Figura 5 - Função de autocorrelação da série parcial de retornos



Fonte: Pesquisa Direta

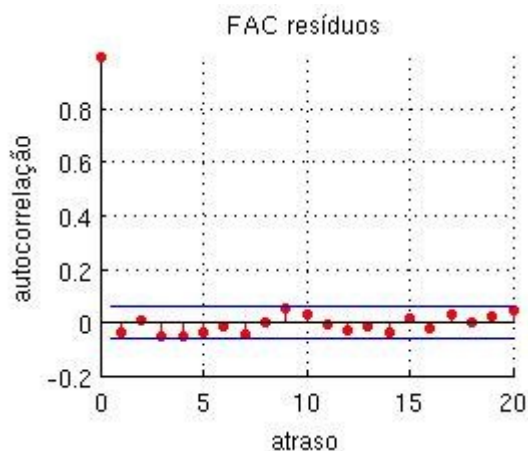
Figura 6 - Função de autocorrelação da série de retornos.



Fonte: Pesquisa Direta

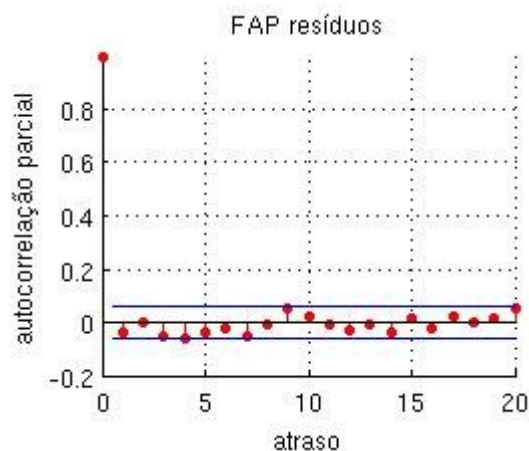
Ajustou-se um modelo AR(1) para a série de retornos buscando modelar os retornos em relação ao último retorno e uma constante que capturasse o comportamento médio de longo prazo da série. As figuras (7) e (8) mostram as funções de autocorrelação e autocorrelação parcial, com base nos resíduos do modelo AR(1) ajustado.

Figura 7 - Função de autocorrelação da série de resíduos do AR(1).



Fonte: Pesquisa Direta

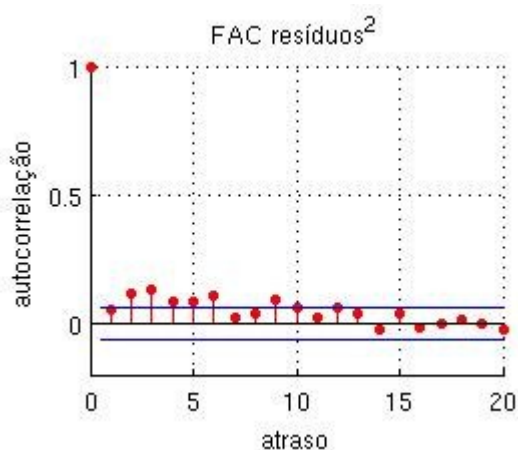
Figura 8 - Função de autocorrelação parcial da série de resíduos do AR(1).



Fonte: Pesquisa Direta

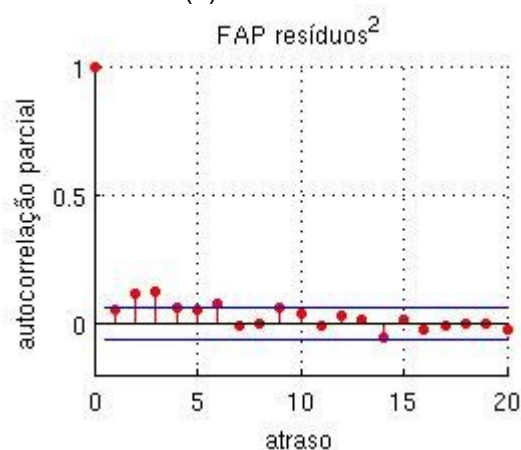
Para verificar a presença de heterocedasticidade pode-se analisar o gráfico da função de autocorrelação dos resíduos quadrados. No caso do IBOVESP, foi verificada a presença de autocorrelação na série de resíduos quadrados e por consequência a possibilidade de aplicação de um modelo GARCH. As figuras (9) e (10) mostram as funções de autocorrelação e autocorrelação parcial, com base nos quadrados dos resíduos do modelo AR(1) ajustado.

Figura 9 - Função de autocorrelação da série de resíduos quadrados do AR(1)



Fonte: Pesquisa Direta

Figura 10 - Função de autocorrelação parcial da série de resíduos quadrados do AR(1)



Fonte: Pesquisa Direta

O teste de Ljung-Box e foi aplicado para confirmar a existência de efeito ARCH na série de retornos, conforme mostra a tabela (2).

Tabela 2 – Ljung-Box aplicado à série de resíduos quadrados

1 – Aceita Hipótese de Efeito ARCH	p-value	Estatística	Valor Crítico
0 – Rejeita Hipótese de Efeito ARCH			
1.0000	0.0000	80.5859	31.4104

Fonte: Pesquisa Direta

Considerando as características da série de retornos, um modelo AR(1)+GARCH(1,1) foi ajustado para capturar o movimento anterior dos retornos e resíduos, e fazer previsões um passo a frente para a séries de retornos do IBOVESA. A tabela (3) mostra o modelo especificado e seus parâmetros estimados

Tabela 3 – Modelo AR(1)+GARCH(1,1) estimado

Parâmetro	p-value	Erro Padrão	Estatística T
C	0.0015675	0.00054341	2.8847
AR(1)	-0.0076707	0.040972	-0.1872
K	1.7141e-05	9.4375e-06	1.8162
GARCH(1)	0.8767	0.055496	15.7975
ARCH(1)	0.048005	0.01979	2.4257

Fonte: Pesquisa Direta

Para confirmar a capacidade do modelo em eliminar o efeito ARCH da série de resíduos, aplicou-se novamente o teste de Ljung-Box sobre a série de resíduos quadrados após o ajuste do modelo AR(1)+GARCH(1,1). A tabela (4) apresenta o resultado do teste.

Tabela 4 – Ljung-Box aplicado à série de resíduos padronizados quadrados do AR(1)+GARCH(1,1)

1 – Aceita Hipótese de Efeito ARCH	p-value	Estatística	Valor Crítico
0 – Rejeita Hipótese de Efeito ARCH			
0	0.5100	19.1824	31.4104

Fonte: Pesquisa Direta

A tabela (5) mostra os resultados de estatística descritiva para os desvios padronizados após o ajuste do modelo AR(1)+GARCH(1,1). As figuras (11), (12), (13) e (14) mostram a distribuição dos resíduos padronizados, os histograma que

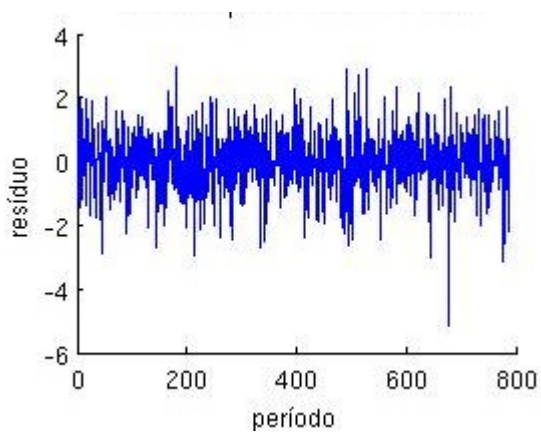
apresenta uma distribuição aproximadamente normal e as funções de autocorrelação e autocorrelação parcial respectivamente.

Tabela 5 – Estatística descritiva dos resíduos padronizados AR(1)+GARCH(1,1)

Média	Desvio Padrão	Assimetria	Curtose
-0.0185	1.0017	-0.3365	3.8979

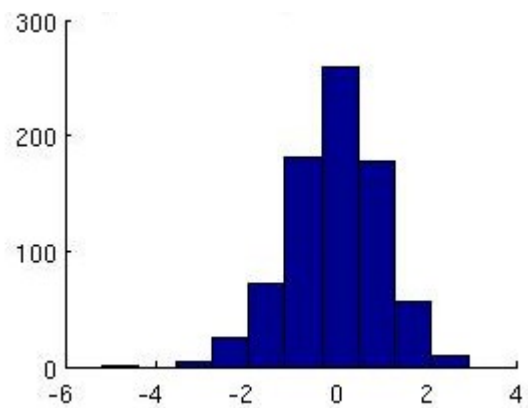
Fonte: Pesquisa Direta

Figura 11 - Série de resíduos padronizados AR(1)+GARCH(1)



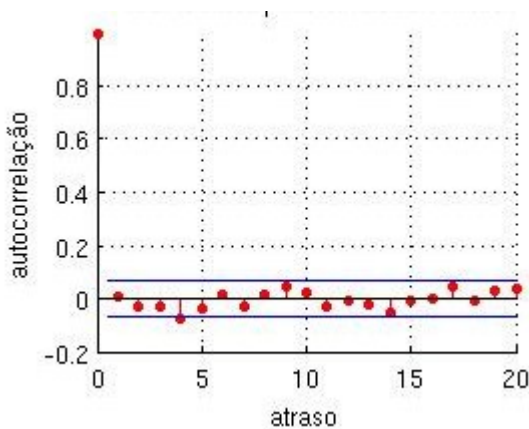
Fonte: Pesquisa Direta

Figura 12 - Histograma de resíduos padronizados AR(1)+GARCH(1,1)



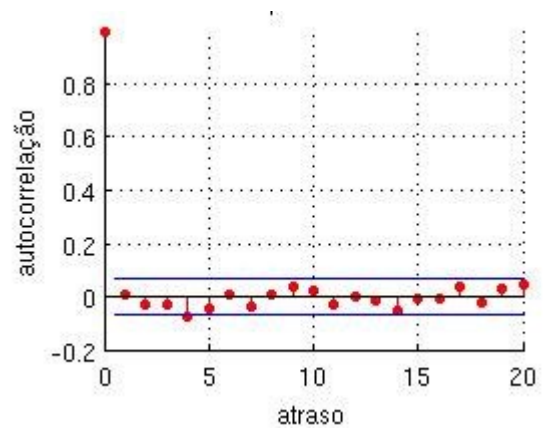
Fonte: Pesquisa Direta

Figura 13 - FAC de resíduos padronizados AR(1)+GARCH(1).



Fonte: Pesquisa Direta

Figura 14 - .FAP de resíduos padronizados AR(1)+GARCH(1,1)



Fonte: Pesquisa Direta

A arquitetura das RNA *feedforward* é composta por três neurônios na camada de entrada, seis neurônios na camada escondida e um neurônio na camada de saída que representa a previsão um passo à frente em relação aos valores fornecidos na camada de entrada. Como o objetivo da RNA é a previsão um passo à frente dos retornos da série do IBOVSPA à vista, a escolha de um único neurônio de saída foi obtida de forma direta.

A quantidade de entradas, inicialmente, foi ajustada para 1 retorno defasado e variações da quantidade de entradas no intervalo de 1 até 15 foram testadas a fim de obter o melhor ajustamento, em relação ao coeficiente U-Theil, com o menor número de entradas defasadas.

A decisão da quantidade de neurônios na camada escondida foi baseada no ajustamento do coeficiente U-Theil tendo também o intervalo de 1 até 10 como faixa de testes. A arquitetura contendo apenas 3 entradas defasadas e 6 neurônios na camada escondida foi a que obteve o melhor ajustamento U-Theil e possui tempo de execução similar ao método GARCH.

Para fins de critério de parada, são executadas até 150 chamadas à função objetivo. Neste caso, a função objetivo é o resultado do REQM com base nas previsões obtidas pela RNA utilizando-se o conjunto de pesos atuais. Se for atingido o critério de parada no qual o erro é menor ou igual a 1×10^{-6} , o treinamento também é encerrado.

Neste artigo é usado o algoritmo ED auto-adaptável proposto por Salman (2007). A idéia proposta por Salman é adaptar o parâmetro F e atribuir, de forma aleatória, uma probabilidade de mutação, parâmetro CR, para cada elemento de uma solução. Dessa forma, os dois únicos parâmetros do algoritmo, não precisam ser acertados manualmente.

A estratégia *best/2/bin* é utilizada como variante do algoritmo de ED. O tamanho da população é de 2 vezes a quantidade de indivíduos necessários para representar todos os pesos da RNA. *best/2/bin* é uma variante do esquema de algoritmo de ED que leva o melhor indivíduo de uma geração considerada como atual, para a geração posterior.

A tabela (6) mostra os resultados de estatística descritiva para os desvios padronizados após o ajuste do modelo RNAE ED. As figuras (15), (16), (17) e (18) mostram a distribuição dos resíduos padronizados, os histograma que apresenta

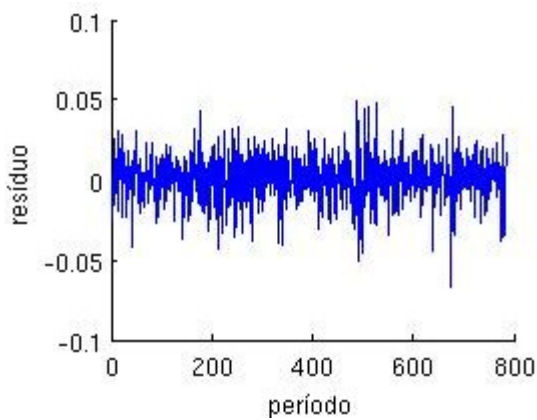
uma distribuição aproximadamente normal e as funções de autocorrelação e autocorrelação parcial respectivamente.

Tabela 6 – Estatística descritiva dos resíduos RNAE ED

Média	Desvio Padrão	Assimetria	Curtose
0.0001	0.0151	-0.2290	3.7754

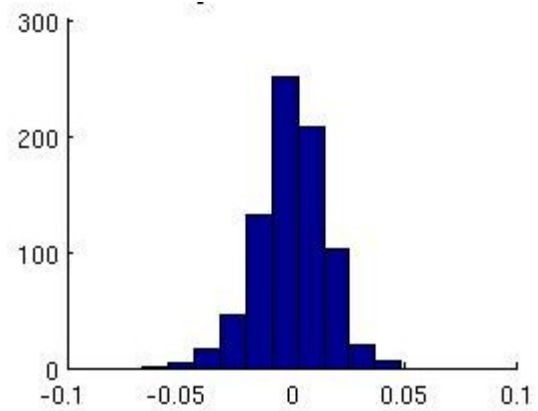
Fonte: Pesquisa Direta

Figura 15 - Série de resíduos RNAE



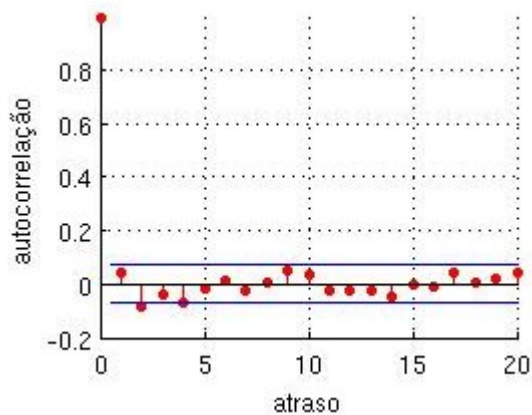
Fonte: Pesquisa Direta

Figura 16 - Histograma de resíduos RNAE



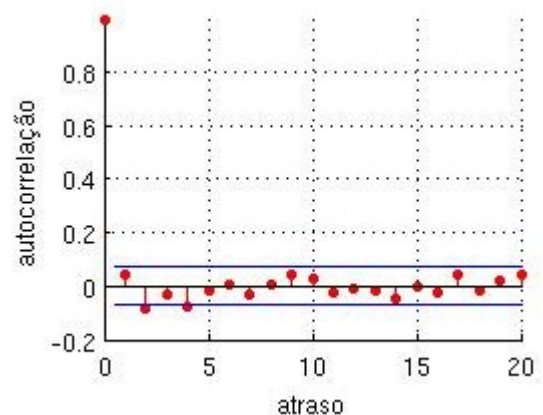
Fonte: Pesquisa Direta

Figura 17 - FAC de resíduos RNAE



Fonte: Pesquisa Direta

Figura 18 - FAP de resíduos RNAE



Fonte: Pesquisa Direta

3.3 Métricas de análise

Para a avaliação dos dados gerados pelos métodos de previsão, foram utilizadas métricas predominantemente quantitativas, a fim de que se tenham resultados matemáticos que corroborem com a teoria utilizada na previsão de séries temporais. Neste artigo, foram adotadas as medidas de desempenho detalhadas a seguir.

- Raiz do Erro Quadrático Médio (REQM): Avalia o tamanho do erro de previsão e é calculada através da média de erros de cada previsão em relação ao valor real, elevados ao quadrado:

$$REQM = \sqrt{\frac{1}{N} \sum_{t=1}^N (x_t - \hat{x}_t)^2} \quad (18)$$

onde x_t e \hat{x}_t representam o valor previsto e o real e N é o tamanho do conjunto de dados. Quando mais próximo de zero é o REQM, melhor é o modelo de previsão.

- Índice de Desigualdade de Theil (U-Theil): Avalia o ajustamento da série prevista à série original. Quanto mais próximo a zero, maior é o ajustamento da série prevista em relação a série original. Em contrapartida, valores mais próximos da unidade indicam que o modelo não conseguiu fazer boas previsões. O índice U-Theil é definido por

$$U - Theil = \frac{\sqrt{\frac{1}{N} \sum_{t=1}^N (\hat{y}_t - y_t)^2}}{\sqrt{\frac{1}{N} \sum_{t=1}^N (\hat{y}_t)^2 + \frac{1}{N} \sum_{t=1}^N (y_t)^2}}, \quad (19)$$

onde \hat{y}_t é o valor previsto para o período t , y_t é o valor observado e N é o número de observações.

4 ANÁLISE DE DADOS

Esta seção apresenta os resultados da aplicação dos modelos especificados na seção 3.2, considerando a realimentação e ajustamento do modelo para a previsão de somente um passo a frente, após a apresentação de cada observação de validação.

Na tabela (7) apresenta-se, resumidamente, os resultados do REQM das previsões obtidas nos ciclos de avaliação, o valor índice de desigualdade U-Theil e o percentual previsões, na direção correta, de cada método avaliado. A última coluna mostra o último retorno real e os retornos previstos pelos métodos.

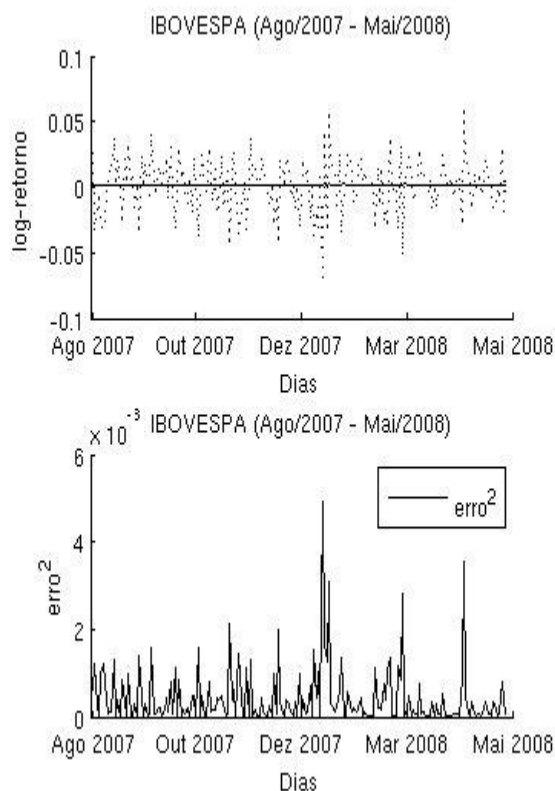
Tabela 7 – Tabela dos resultados da previsão do IBOVESPA

Método	REQM	U-Theil	Percentual de previsões na direção correta	Último previsto x retorno real (0.0110)
AR(1) + GARCH(1,1)	0.0202	0.9123	75.63%	0.0020
RNA Feedforward best/2/bin	0.0199	0.7157	76.14%	0.0113

Fonte: Pesquisa Direta

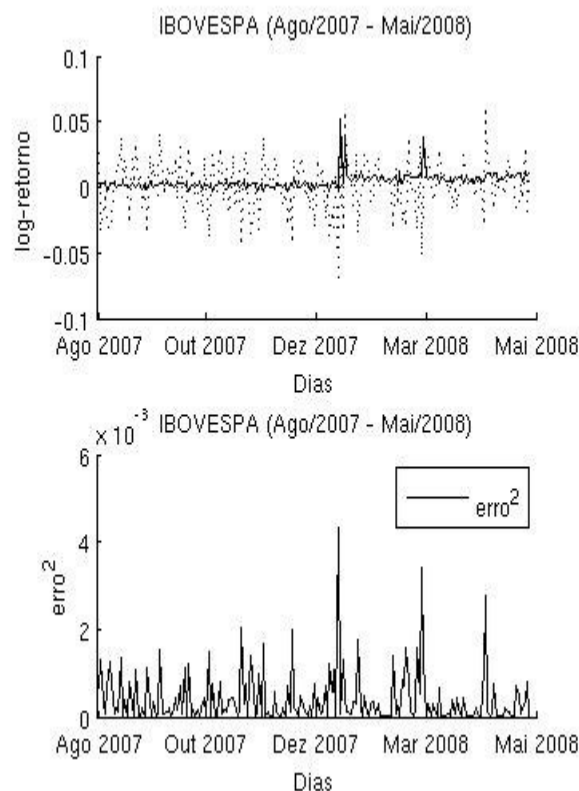
A RNA *Feedforward* ED *best/2/bin* mostrou-se superior no que se refere a capturar o comportamento não linear dos retornos e ainda procurou modelar a extensão dos retornos extraordinários, como é possível observar na Figura 20, em relação ao comportamento de janeiro de 2008. O modelo AR(1)+GARCH(1,1), apesar de tentar acompanhar o comportamento diferenciado, não consegue capturar a extensão do movimento. O modelo RNA *Feedforward* ED *best/2/bin*, por tentar modelar um retorno extraordinário no fim de janeiro de 2008 acaba tendo seu REQM um pouco prejudicado.

Figura 19 - Série prevista x real e erro quadrático obtido pelo método AR(1)+GARCH(1,1)



Fonte: Pesquisa Direta

Figura 20 - Série prevista x real e erro quadrático obtido pelo método RNA Feedforward ED best/2/Bin



Fonte: Pesquisa Direta

Em relação a coeficiente de desigualdade U-Theil, medida de referência para este artigo, o modelo RNA *Feedforward* ED *best/2/bin*, mostrou-se superior ao modelo AR(1)+GARCH(1,1), visto que quanto menor o coeficiente de desigualdade, melhor a previsões de um método se ajustaram a série real.

O percentual de previsões na direção correta e a comparação entre o último retorno real e o último retorno previsto confirmam a maior acurácia obtida pelo modelo RNA *Feedforward* ED *best/2/bin*.

5 CONSIDERAÇÕES FINAIS, LIMITAÇÕES E RECOMENDAÇÕES

Este artigo verificou o desempenho da previsão um passo à frente do IBOVESPA considerando uma RNA *Feedforward* e um modelo AR(1)+GARCH(1,1) como referência de mercado. Os resultados obtidos pela RNA *Feedforward*, treinada pelo algoritmo de Evolução Diferencial Auto-Adaptável foram comparados com os

Revista Produção Online. Florianópolis, SC, v.11, n. 4, p. 1114-1140, out./dez. 2011.

resultados obtidos por um modelo clássico para a previsão de séries temporais financeiras AR(1)+GARCH(1,1).

A RNA *Feedforward* obteve melhores resultados em relação ao coeficiente de desigualdade U-Theil, portanto suas previsões ajustaram-se melhor a série real de retornos. A RNA *Feedforward* ED *best/2/bin* mostrou-se superior no que se refere a capturar o comportamento não linear dos retornos e ainda procurou modelar a extensão dos retornos extraordinários, como é possível observar na Figura 20, em relação ao comportamento de janeiro de 2008. O modelo AR(1)+GARCH(1,1), apesar de tentar acompanhar o comportamento diferenciado, não consegue capturar a extensão do movimento. Embora o modelo estudado ainda ofereça limitações de uso e interpretação, é inquestionável que a utilização de modelos econométricos adequados oferece suporte na tomada de decisão e minimiza significativamente os riscos da organização.

A limitação desta pesquisa está em utilizar somente a própria série de retornos passados para prever o retorno um passo à frente. Sugere-se para pesquisa futura considerar a utilização de outra informação como o erro da previsão anterior ou alguma informação externa de variáveis econômicas, além da própria série de retornos, uma vez que a RNA pode ser projetada com múltiplas entradas.

REFERÊNCIAS

ANDERSEN, T. G.; BOLLERSLEV, T.; LANGE, S. Forecasting financial market volatility: sample frequency vis-à-vis forecast horizon. **Journal of Empirical Finance**. v. 6, p. 457-477, 1999.

BERTSEKAS, D. P. **Nonlinear programming**. Athena Scientific, Belmont, MA, 1995.

BÓDIS, L. **Financial times series forecasting using artificial neural networks**. Master Thesis – “Babes-Bolyai” University, Faculty of Mathematics and Computer Science – Department of Computer Science, Cluj-Napoca, Romania, 2004.

BOLSA DE VALORES, MERCADORIAS E FUTURO (BMFBOVESPA). Disponível em:<www.bmfbovespa.com>. Acesso em: 02 set. 2010

BOLLERSLEV, T. Generalized autoregressive conditional heteroskedasticity. **Journal of Econometrics**, v. 31, n. 3, p. 307-327, 1986.

CALÔBA, G. M.; CALÔBA, L. P.; SALIBY, E. Cooperação entre redes neurais artificiais e técnicas “clássicas” para previsão de demanda de uma série de vendas de cerveja na Austrália. **Pesquisa Operacional**, v. 22, n. 3, p. 345-358, 2002.

CHEN,C,W.S. GERLACH, R.H. TAI, A. P.J. Testing for nonlinearity in mean and volatility for heteroskedastic models. **Mathematics and Computers in Simulation**, v. 79, p. 489–499, 2008.

DELURGIO, S. A. **Forecasting: principles and applications**. Boston, USA: Irwin McGraw-Hill, 1998. 802 p.

DU, J.; HUANG, D.; WANG, X.; GU, X. Shape recognition based on neural networks trained by differential evolution algorithm. **Neurocomputing**, v. 70, n. 4-6, p. 896-903, 2007.

GIL, A. C. **Como elaborar projetos de pesquisa**. 4. ed. São Paulo: Atlas, 2002. 175p.

HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed., Porto Alegre/RS:Bookman, 2001.

MAYER, D. G.; KINGHORN, B. P.; ARCHER, A. A. Differential evolution an easy and efficient evolutionary algorithm for model optimization. **Agricultural Systems**, v. 83, n. 3, p. 315-328, 2005.

MICHALEWICZ, Z.; FOGEL, D. B. **How to solve it: modern heuristics**. Berlin: Springer Verlag, 2000. 467p.

MORETTIN, P. A.; TOLOI, C. M. C. **Previsão de séries temporais**. 2. ed. São Paulo, SP: Atual Editora, 1987. 436p.

PINDYCK, R. S.; RUBINFELD, D. L. **Econometria modelos & previsões**. 4. ed. Rio de Janeiro, RJ: Elsevier, 2004. 726p.

RECH G. **Forecasting with artificial neural network models**. SSE/EFI Working Paper Series in Economics and Finance, n. 491, 2002.

SALMAN, A. ENGELBRECHT, A. P.; OMRAN, M. G. H.; empirical analysis of self-adaptive differential evolution. **European Journal of Operational Research**, v. 183, n. 2, p. 785-804, 2007.

STORN, R.; PRICE K. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341-359, 1997.

SUBUDHI, B.; JENA, D. A differential evolution based neural network approach to nonlinear system identification. **Applied Soft Computing**, 2010, doi:10.1016/j.asoc.2010.01.006.

TSAY, R. S. **Analysis of financial time series**. 2 ed., New Jersey: Wiley Series In Probability and Statistics, New Jersey, USA, 2005.

YAO, X. Evolving artificial neural networks. **Proceedings of IEEE**, v. 87, n. 9, p. 1423-1439, 1999.

ZHANG, G. P. Time series forecasting using a hybrid ARIMA and neural network model. **Neurocomputing**, v. 50, n. 1, p. 159-175, 2003.



Artigo recebido em 02/09/2010 e aceito para publicação em 28/09/2011.