

SISTEMA HÍBRIDO BASEADO EM EXTREME LEARNING MACHINE E HARMONY SEARCH

Edgar Wellington Marques de Almeida
Universidade de Pernambuco
POLI, Mestrado em Engenharia da Computação
Recife, Brasil
ewma@ecomp.poli.br

Mêuser Jorge da Silva Valença
Universidade de Pernambuco
POLI, Mestrado em Engenharia da Computação
Recife, Brasil
meuser@ecomp.poli.br

Abstract—In this work was used a technique of seek associated to a training algorithm of artificial neural network. The aim is to develop a hybrid system which can be used for people from others knowledge area. Besides generate the weight matrix, it selects the input variables and builds the neural network architecture, choosing the number of neurons in the hidden layer. The neural network evaluating metrics guide the search, choosing the configurations which show the minor: forecast error, input variables and neurons in the hidden layer. In general, the hybrid system reached a satisfactory performance, obtaining a better or equivalent response to the methods used by National Operator of Electrical System (ONS) [1].

Keywords—Artificial Neural Network; Variables Selection; Extreme Learning Machine; Time Series Forecasting; Harmony Search;

I. INTRODUÇÃO

Em geral, a maneira como é feita a escolha dos parâmetros de uma rede neural (RN) acontece de forma empírica, necessitando assim, da intervenção do usuário, que na maioria dos casos, não tem domínio sobre a problemática. Dessa forma, técnicas facilmente configuráveis e capazes de gerar bons modelos tem sido estudadas e desenvolvidas.

Com esta finalidade, propomos desenvolver um algoritmo híbrido de treinamento de uma rede neural MLP. O algoritmo deve ser capaz de selecionar as melhores combinações das variáveis de entrada - podendo representar uma porção menor da quantidade de variáveis de entrada originais - e os parâmetros de configuração da rede neural, sem diferenças significativas na qualidade da aprendizagem.

Para a construção do algoritmo proposto, foi utilizado o algoritmo de busca *Harmony Search* [2], em conjunto com a técnica de treinamento *Extreme Learning Machine* [3] – O primeiro, é baseado em populações e tem como peculiaridade a produção de apenas um novo indivíduo a cada iteração do processo de busca. Sendo responsável pela seleção das variáveis de entrada e número de neurônios da camada escondida. O segundo, treina a rede neural com a arquitetura gerada pelo primeiro.

Em [3], o desempenho do ELM foi comparado a outros algoritmos de treinamento. Em suma, mostrou-se dezenas - em alguns casos centenas - de vezes mais rápido que os outros algoritmos citados, e obteve valores semelhantes em relação às

métricas de avaliação de desempenho de redes neurais, como Erro Médio Quadrático (EMQ) e seu Desvio-padrão.

Este trabalho propõe um sistema híbrido para a seleção de variáveis de treinamento de uma rede neural MLP. Sendo o artigo dividido da seguinte forma: Sessão II apresenta alguns trabalhos relacionados. A sessão III aborda nossa proposta. Na sessão IV, os experimentos e resultados obtidos. Finalmente, na sessão V nossas conclusões.

II. TRABALHOS RELACIONADOS

O sistema proposto visa simplificar a utilização de redes neurais por usuários de outras áreas de conhecimento, dada que a ausência de sensibilidade por parte desses usuários durante a escolha dos parâmetros de configuração da rede neural, torna o uso da técnica extremamente difícil, senão inviável. Dessa forma, sistemas híbridos são cada vez mais testados e desenvolvidos.

Uma abordagem híbrida de ELM e *Adaboost* foi proposta por [4] para aproximação de funções e classificação. Em [5], algoritmos genéticos foram utilizados em conjunto com redes neurais para a previsão de carga de curto prazo do sistema elétrico.

III. SISTEMA PROPOSTO

O sistema proposto utiliza os algoritmos de busca HS e ELM em conjunto. O primeiro faz a seleção das variáveis de entrada e escolhe o número de neurônios na camada escondida. O segundo, treina a rede neural com os parâmetros gerados pelo HS. Dessa forma, as métricas de avaliação do treinamento da rede guiam a avaliação da busca, onde os melhores indivíduos são aqueles que possuem um menor número de variáveis de entrada e de neurônios na camada escondida, sem apresentar diferenças significativas na qualidade da aprendizagem.

A. *Harmony Search*

O HS é um algoritmo metaheurístico de otimização proposto por [2], foi inspirado no ato de improvisação dos músicos, onde cada músico toca uma nota e encontram juntos a melhor harmonia.

No algoritmo HS é feita uma analogia entre improvisação e otimização, onde cada músico corresponde a cada variável de decisão; faixas de tons dos instrumentos musicais

correspondem as faixas de valor das variáveis de decisão; harmonias musicais de certo instante correspondem ao vetor de solução em certa iteração; e o padrão estético do público corresponde a função objetivo. Assim como a harmonia musical é aperfeiçoada a cada instante, o vetor solução é aperfeiçoado a cada iteração.

Os parâmetros de configuração do algoritmo são:

hms (*harmony memory size*) – Tamanho da memória harmônica, representa o valor máximo de harmonias que a memória harmônica poderá armazenar.

hmcr (*harmony memory choosing rate*) – Taxa de escolha dos valores da memória harmônica. Representa o percentual de escolha dos valores armazenados na memória harmônica.

O HS tenta encontrar um vetor X que otimize (minimize ou maximize) uma função objetivo. Dessa forma, o algoritmo faz uso de uma memória (memória harmônica MH) onde são armazenados os melhores vetores. Seu funcionamento está dividido da seguinte forma:

1. Gerar vetores com valores aleatórios até atingir o tamanho hms (X^1, \dots, X^{hms}); avaliar os vetores e depois armazená-los na memória harmônica MH (Fig. 1);
2. Gerar e avaliar um novo vetor X' . Para cada componente x'_i :
 - Com probabilidade $hmcr$, selecionar valor armazenado na MH;
 - Com probabilidade $1 - hmcr$, selecionar um valor aleatório da faixa de valor permitida;
3. Se X' é melhor que o pior vetor na MH, ocorrerá a substituição por X' .

Os passos 2 e 3 são repetidos até que o critério de parada seja alcançado, como por exemplo, o máximo de iterações ou o valor médio do fitness dos vetores da MH estiver muito próximo do fitness do melhor vetor. Seu pseudocódigo pode ser visto no algoritmo 1.

$$MH = \begin{bmatrix} x_1^1 & \dots & x_n^1 \\ \vdots & \ddots & \vdots \\ x_1^{hms} & \dots & x_n^{hms} \end{bmatrix} \begin{bmatrix} f(X^1) \\ \vdots \\ f(X^{hms}) \end{bmatrix}$$

Fig. 1. Representação da memória harmônica MH.

Algoritmo 1: Pseudocódigo do *Harmony Search*

- 1 Inicializar a memória harmônica (MH);
- 2 **enquanto** critério de parada não for alcançado **faça**
- 3 Improvisar uma nova harmonia (vetor) a partir da MH;
- 4 Atualizar a MH com a nova harmonia;

B. Redes neurais e Extreme Learning Machine

A RN é um aproximador universal de funções, que mapeia conjuntos de dados de entrada para uma saída apropriada através de um aprendizado por experiência, onde seus pesos são ajustados à medida que são fornecidos novos conjuntos de entrada. Uma das redes mais utilizadas atualmente é a *Multilayer Perceptron* (MLP), com uma arquitetura que

possui várias camadas de nós, em um grafo direcionado, com cada elemento de uma camada ligado a próxima (Fig. 2).

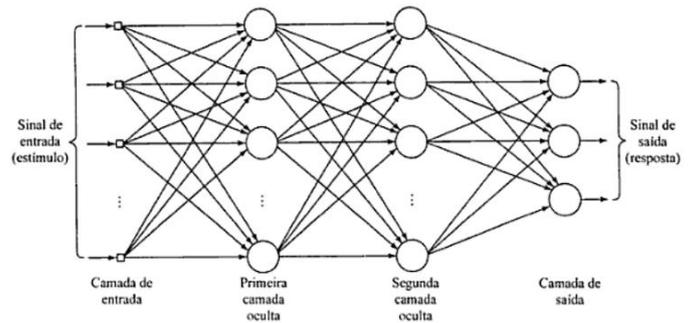


Fig. 2. Exemplo de uma rede MLP.

O ELM, é um algoritmo de treinamento para redes *single hidden layer feedforward networks* (SLFNs) - redes MLP com uma camada escondida - proposto por [3]. Considerada a rede como um sistema linear, onde os pesos da camada de saída (ligam a camada escondida a camada de saída) são analiticamente determinados através do cálculo da inversa generalizada da matriz de saída da camada escondida (i.e., não há ajuste iterativo dos parâmetros da rede). Uma descrição mais detalhada de seu funcionamento pode ser vista no Algoritmo 2.

Algoritmo 2: Pseudocódigo do *Extreme Learning Machine*

- 1 Dado um conjunto de treinamento $N = \{(x_i, t_i) | x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m, i = 1, \dots, N\}$, função de ativação $g(x)$, e número de nós na camada escondida \tilde{N} ;
- 2 Atribua aleatoriamente valores para os pesos de entrada w_i e bias b_i ;
- 3 Calcule a matriz de saída da camada escondida H ;
- 4 Calcule a matriz de pesos de saída B ;
- 5 $B = H^\dagger T$, onde $T = [t_1, \dots, t_N]^T$, H^\dagger é a inversa generalizada de Moore-Penrose da matriz H .

C. Formatação dos operadores

Os operadores utilizados pelos algoritmos HS e ELM foram ajustados e sua descrição pode ser vista a seguir:

VMNCE – Valor máximo de neurônios da camada escondida, representa o número máximo de neurônios que a camada escondida da RN poderá assumir.

VME – Valor máximo de entradas, representa o número máximo de entradas da RN. Sendo igual ao número de variáveis de entrada do conjunto de dados a serem selecionadas.

VB – Vetor de busca, representa o indivíduo. Foi dividido em duas partes, onde a primeira indica as variáveis de entrada selecionadas (indicadas pelo índice dos elementos que tiverem valor um), e a segunda, o número de neurônios na camada escondida que serão utilizados no treinamento da rede neural (sendo igual a quantidade de elementos de valor um).

Um exemplo de configuração de um VB pode ser visto na Fig. 3, onde o VME = 4; VMNCE = 6; as variáveis de entrada selecionadas para o treinamento foram as de índice um, três e quatro; e o número de neurônios da camada escondida igual a três;

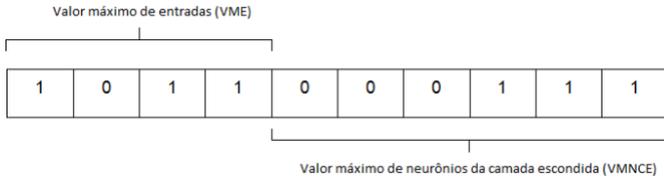


Fig. 3. Exemplo de configuração de um vetor de busca.

Neste trabalho, foram utilizadas duas funções de avaliação, a primeira é uma variação do critério PSE (Predicted Squared Error) [6] e está descrita em (1).

$$f_1(x) = EMQ_{val} + 2\sigma_\varepsilon^2 \frac{p}{N} + \frac{n}{VMNCE} \quad (1)$$

onde, EMQ_{val} é o erro médio quadrático do conjunto de validação; $2\sigma_\varepsilon^2$ é a variância do EMQ_{val} ; p é o número de variáveis de entrada selecionadas para o treinamento; N o número de exemplos do conjunto de validação; e n é a quantidade de neurônios da camada escondida.

A segunda, foi criada segundo uma análise empírica das variáveis do sistema, levando-se em conta as variáveis foco da otimização. Sua estrutura pode ser vista em (2).

$$f_2(x) = EMQ_{val} + \frac{p}{VME} + \frac{n}{VMNCE} \quad (2)$$

D. Funcionamento do sistema

O sistema proposto treina uma rede neural utilizando o vetor de busca para configurar os parâmetros da RN, onde o vetor ζ representa o melhor conjunto de configurações da rede neural. As métricas de avaliação aplicadas a rede, guiam a construção dos novos indivíduos. Seu funcionamento é dividido nos seguintes passos:

1. Inicializar os parâmetros de configuração do sistema:
 - Carregar as bases com os dados de treinamento, validação e teste;
 - Definir o valor das variáveis VME, VMNCE, hms e $hmcr$;
 - Definir o valor máximo de iterações do sistema;
 - Definir a função de avaliação do sistema;
 - Definir a função de ativação da rede neural;
2. Gerar vetores para o preenchimento da MH. Na geração da segunda parte do vetor, distribuir os valores de forma a abranger todas as faixas de valores da variável VMNCE. Por exemplo, supondo que o VMNCE = 100 e o $hms = 5$, as faixas de valores são as seguintes 1-20, 20-40, 40-60, 60-80, 80-100.
3. Gerar um novo vetor X' . Para cada componente X'_i :

- Com probabilidade $hmcr$, selecionar o valor armazenado na MH;
- Com probabilidade $1-hmcr$, selecionar o valor 0 ou 1 com probabilidade de 50%;

4. A partir da configuração que X' representa, treinar uma RN utilizando o algoritmo ELM e a base de dados de treinamento. Com os pesos do treinamento gerar o EMQ_{val} (utilizando a base de dados de validação);
5. Avaliar X' . Se X' é melhor que o pior vetor na MH, substituir o pior vetor da MH por X' .

Os passos 3, 4 e 5 são repetidos até que o critério de parada seja alcançado - máximo de iterações ou valor médio do fitness dos vetores da MH for igual ou superior a 90% do valor do fitness do melhor vetor. O pseudocódigo do sistema proposto está representado no Algoritmo 3.

Algoritmo 3: Pseudocódigo do Sistema proposto

- 1 Inicializar os parâmetros do sistema;
 - 2 Inicializar a memória harmônica (MH);
 - 3 **enquanto** critério de parada não for alcançado **faça**
 - 4 Gerar um novo vetor X' a partir da MH;
 - 5 Treinar e avaliar a RN a partir da configuração representada por X' ;
 - 6 Atualizar a MH com X' ;
 - 7 **retorna** o melhor vetor ζ ;
-

IV. EXPERIMENTOS E RESULTADOS

A fim de medir o desempenho do sistema proposto em cenários com dados reais, foram selecionadas bases de dados contendo a vazão média diária em m^3/s das barragens de Salto Caxias e Furnas. O objetivo de se utilizar estas séries é que se dispõe dos erros atuais cometidos pelos modelos estatísticos, em uso, disponibilizados pelo Operador Nacional do Sistema Elétrico (ONS) [1] [7].

Salto Caxias - Construída no trecho final do Rio Iguaçu, entre os municípios de Capitão Leônidas Marques e Nova Prata do Iguaçu, distantes cerca de 400 quilômetros da capital Curitiba, no estado do Paraná. Esta série representa a vazão diária no período de 31 de dezembro de 1967 a 31 de dezembro de 2007. Na Tabela 1 pode-se ver suas características estatísticas.

Tabela 1. Características Salto Caxias.

Características	
#Mínimo	123
#Máximo	24711
#Média	1501.295
#Variância	2073945
#Desvio-padrão	1440.169
#Observações	14611

Furnas - Está localizada no curso médio do Rio Grande, entre os municípios de São José da Barra e São João Batista do Glória, no estado de Minas Gerais Esta série representa a vazão diária no período de 31 de dezembro de 1967 a 31 de dezembro de 2007. Pode-se ver suas características estatísticas na Tabela 2.

Tabela 2. Características Furnas.

Características	
#Mínimo	111
#Máximo	7497
#Média	914.2143
#Variância	480893.100
#Desvio-padrão	693.4645
#Observações	14611

A. Experimentos

Este experimento consiste na previsão da vazão (m^3/s) de sete dias à frente, utilizando como variáveis de entrada para seleção, trinta dias passados. Neste caso, as duas bases tiveram suas variáveis de entrada e saída (alvo) normalizadas entre [0.15, 0.85]. O sistema treinará a rede e fará a seleção de variáveis de entrada e o número de neurônios da camada escondida.

O algoritmo proposto será chamado de ELM_{HS1} quando estiver utilizando a função de avaliação (1) e será chamado de ELM_{HS2} quando utilizar a função de avaliação (2).

Com o intuito de gerar amostras para testes estatísticos, cada base foi executada trinta vezes nos algoritmos ELM_{HS1} e ELM_{HS2} com os parâmetros da Tabela 3. Para ambos, a função de ativação da rede neural utilizada foi a Sigmoide Logística.

Tabela 3. Configuração dos parâmetros do sistema.

#Observações		#Máximo neurônios	#Máximo iterações	#hms
Treinamento	Validação			
75%	25%	200	200	10

B. Resultados

A métrica de avaliação utilizada foi o EPMA (Erro Percentual Médio Absoluto). Sua fórmula está descrita em (3).

Dados $O = \begin{bmatrix} o_1 \\ \vdots \\ o_n \end{bmatrix}$ e $P = \begin{bmatrix} p_1 \\ \vdots \\ p_n \end{bmatrix}$, onde o conjunto O , representa os valores observados e P , os valores previstos, tem-se que:

$$EPMA = \frac{1}{n} \sum_{i=1}^n \frac{(o_i - p_i)}{o_i} 100 \quad (3)$$

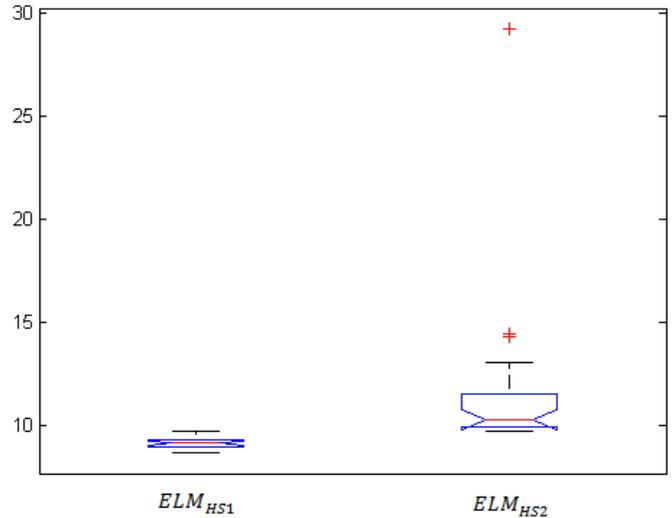
onde n é o número total de ocorrências.

Para cada base foi calculada a média do EPMA e seu desvio-padrão para ELM_{HS1} e ELM_{HS2} . De posse destes resultados, foi possível calcular o teste t de Student com um nível de confiança de 95%. Além disto, também foi calculada a média da quantidade de neurônios na camada escondida e a média de variáveis de entrada selecionadas pelo sistema.

Salto Caxias - Para esta base, o sistema proposto obteve menor EPMA para as duas funções de avaliação, tendo um melhor resultado para ELM_{HS1} . O menor desvio-padrão foi alcançado por ELM_{HS1} . O t de Student mostrou com 95% de chance que há diferença estatística entre os modelos ELM_{HS1} e ELM_{HS2} . Os resultados podem ser vistos na Tabela 4. Um gráfico *boxplot* do EPMA dos algoritmos pode ser visto na Fig. 4 e um gráfico dos resultados da previsão para ELM_{HS1} no período de 01/01/2007 a 21/11/2007 pode ser visto na Fig. 5.

Tabela 4. Comparação dos resultados Salto Caxias.

	ONS	ELM_{HS1}	ELM_{HS2}
EPMA	33.600%	9.181%	11.396%
Desvio-padrão	-	0.243%	3.594%
t	-	-	3.426
Neurônios escondida	-	138	13
Variáveis selecionadas	-	21	21

Fig. 4. Gráfico *boxplot* do EPMA para Salto Caxias.Fig. 5. Gráfico dos resultados de previsão do ELM_{HS1} para Salto Caxias.

Furnas - Para esta base, o sistema proposto obteve menor EPMA para as duas funções de avaliação, tendo um melhor resultado para ELM_{HS1} . O menor desvio-padrão foi alcançado por ELM_{HS1} . O teste t de Student mostrou com 95% de chance que há diferença estatística entre os modelos ELM_{HS1} e ELM_{HS2} . Os resultados podem ser vistos na Tabela 5. Um gráfico *boxplot* do EPMA dos algoritmos pode ser visto na Fig. 6 e um gráfico dos resultados da previsão para ELM_{HS1} no período de 01/01/2007 a 25/11/2007 pode ser visto na Fig. 7.

Tabela 2. Comparação dos resultados para Furnas.

	ONS	ELM_{HS1}	ELM_{HS2}
EPMA	27.600%	10.562%	14.906%
Desvio-padrão	-	0.241%	4.231%
t	-	-	5.712
Neurônios escondida	-	150	12
Variáveis selecionadas	-	21	23

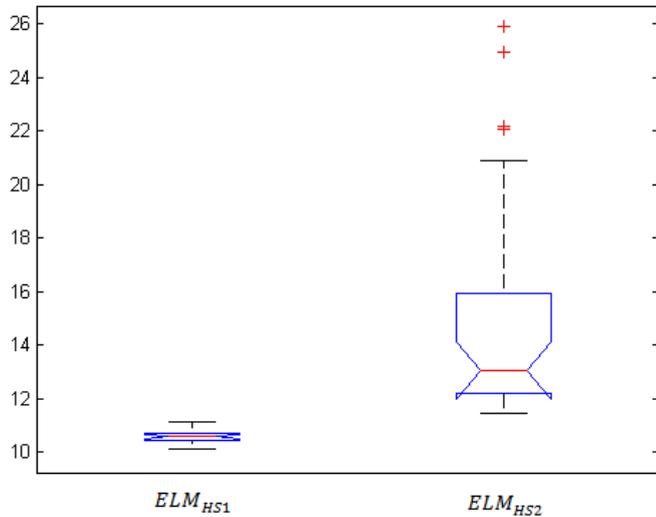


Fig. 6. Gráfico boxplot do EPMA para Furnas.

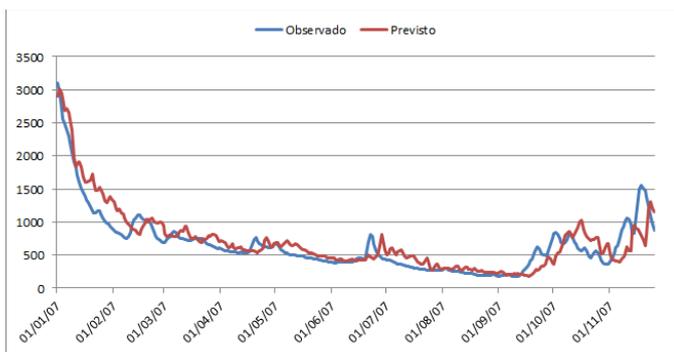


Fig. 7. Gráfico dos resultados de previsão do ELM_{HS1} para Furnas.

V. CONCLUSÃO E TRABALOS FUTUROS

Neste trabalho foi proposto um sistema híbrido para o treinamento e seleção das variáveis de entrada de uma rede neural artificial MLP com uma camada escondida. Experimentos mostraram que há diferenças estatísticas entre o sistema proposto e os dados fornecidos pelo ONS. A partir deste trabalho será possível desenvolver uma ferramenta capaz de auxiliar usuários de outras áreas de conhecimento,

facilitando o uso de redes neurais artificiais, uma vez que poderão utilizar menos parâmetros de configuração.

No geral, a função de avaliação ELM_{HS1} mostrou-se mais eficiente em relação a ELM_{HS2} , por gerar modelos com menor erro e desvio. No entanto, gerou arquiteturas de redes neurais maiores, por escolher um maior número de neurônios na camada escondida.

Em trabalhos futuros, pretende-se criar novas funções de avaliação para o sistema, que melhorem o foco na busca da quantidade de neurônios na camada escondida, a fim de tornar o tempo de treinamento e a arquitetura da rede mais uniforme.

Além disso, pretende-se adaptar a busca atual para levar em conta o coeficiente de correlação entre as variáveis de entrada na geração de novos indivíduos, aperfeiçoando sua escolha.

Ainda, pretende-se comparar os resultados obtidos com outras técnicas de treinamento de redes neurais artificiais.

Outras variações da técnica HS também poderão ser testadas, como por exemplo a [8], [9] e [10]. Desta forma, o sistema terá uma maior robustez, podendo resolver de forma mais eficiente um determinado tipo problema. Por fim, variações do ELM podem ser testadas, como a [11] e [12].

REFERÊNCIAS

- [1] ONS (2013, Março 10). Site do Operador Nacional do Sistema elétrico, Disponível: http://www.ons.org.br/operacao/previsao_vazoes.aspx
- [2] Z.W. Geem, J.H. Kim, G Logonathan, A new heuristic optimization algorithm: Harmony search. SIMULATION, v. 76, n. 2, 2001, pp. 60-68.
- [3] Q.Y. HUAN, C.K. SIEW, Extreme learning machine: Theory and applications. Science Direct, Neurocomputing, v. 70, 2006, pp. 489-501.
- [4] W. Gaitang, Dynamic Adaboost ensemble extreme learning machine. Advanced Computer Theory and Engineering, v. 3, 2010, pp. 54-58.
- [5] Du Xin-hui; Tian Feng; Tan Shao-qiong, "Study of Power System Short-term Load Forecast Based on Artificial Neural Network and Genetic Algorithm," *Computational Aspects of Social Networks (CASoN), 2010 International Conference on*, pp.725-728.
- [6] M. John, Prediction risk and architecture selection for neural networks, Springer-Verlag, 1994, pp.147-165.
- [7] ONS (2013, Março 10), I Workshop Previsão de Vazões, 2017, Disponível: http://www.ons.org.br/home/previsao_vazoes.htm
- [8] M.P. Lucas, P.A. Carolina, A.G. Richard, Harmony Search for Multi-objective Optimization . Brazilian Symposium on Neural Networks, 2012, pp. 220-225.
- [9] S. Li, G. Chen, Y. Lei, Q. Zhang, I. Wu, Improved Harmony Search Algorithm with Better Local Convergence Speed. 31st Chinese Control Conference, 2012, pp 2368-2373.
- [10] K. Zhi, W. Lifu, W. Zhaoxia, Q. Shiqing, Z. Dexuan, An Improved Adaptive Harmony Search Algorithm . 10th World Congress on Intelligent Control and Automation, 2012 pp. 743-774.
- [11] M. Yoan, OP-ELM: Optimally Pruned Extreme Learning Machine. Neural Networks, v. 21, 2010, pp. 158-162.
- [12] R. Zhang, Y. Lan, G.B. Huang, Y.C. Soh, Dynamic Extreme Learning Machine and Its Approximation Capability, Cybernetics, 2013, pp. 1-12.