

# Replicação do trabalho "Extreme learning machine ensemble model for time series forecasting boosted by PSO: Application to an electric consumption problem"

Filipe Coelho de Lima Duarte  
fclld@cin.ufpe.br

José Flávio Vieira Melo  
jfvvm@cin.ufpe.br

**Abstract**—O objetivo deste trabalho foi replicar a pesquisa de Larrea et al. [1], os quais propuseram um Ensemble de ELM otimizado pelo algoritmo Particle Swarm Optimization (PSO) para previsão de séries temporais. Nesse sentido, o modelo proposto tem início com a geração de um pool de ELM ajustados ao conjunto de treinamento; em seguida, combina-se os modelos pela média ponderada; e, por fim, emprega-se o algoritmo PSO para ajustar os pesos com a finalidade de reduzir o erro de previsão. Seguimos o protocolo experimental do referido estudo, o qual utilizou-se de duas bases de dados, Santa Fé laser strength e Consumo de energia espanhola. Em adição ao trabalho supracitado, decidimos utilizar um conjunto de validação para ajustar os pesos do Ensemble. Para ambos os experimentos, os resultados evidenciaram que a utilização do conjunto de validação reduziu os erros dos Ensembles otimizados pelo PSO. Entretanto, o Ensemble otimizado pelo PSO não superou, em termos de erro de previsão, a combinação pela média dos modelos ELM. Ainda, o aumento do tamanho do pool do Ensemble reduziu o erro de previsão, contudo a partir de 20 modelos o erro não foi reduzido. Como limitação deste trabalho, denotamos que, por conta do tempo de processamento, não pudemos realizar mais de uma execução independente de ambos os experimentos.

**Index Terms**—Ensemble, ELM, PSO, Séries temporais

## I. INTRODUÇÃO

Este trabalho se trata de uma reprodução referente ao artigo de Larrea et al. [1] publicado Neurocomputing em 2020. A tese deste trabalho é que a hibridização de técnicas Extreme Learning Machines (ELM), Ensemble Model (EM) e Particle Swarm Optimization (PSO) para realizar a tarefa de previsão de séries temporais produziria melhores resultados que a utilização de apenas um modelo forte de redes neurais. Cada técnica tem sua própria missão: o ELM é utilizado para criar modelos individuais das Séries Temporais. O EM combina as diferentes ELMs por meio de uma regra de agregação. Por fim, o utiliza-se o PSO para ajustar os pesos de cada modelo da combinação do EM.

Em suma, a proposta busca investigar se combinar modelos de redes neurais ELM por meio da média ponderada otimizada pelo algoritmo PSO é superior do que utilizar apenas um modelo de ELM.

Tendo isso em perspectiva, a seguir detalharemos cada uma dessas técnicas: I-A *Extreme Learning Machine*, I-B *Ensemble Model* e I-C *Particle Swarm Optimization*.

### A. Extreme learning machines (ELM)

O modelo Extreme Learning Machine é um algoritmo de aprendizado classificado como Redes Neurais Feed-forward de Camada Única Oculta (SLFN) Huang [2]. O ELM estabelece aleatoriamente os pesos que conectam as entradas e os nós da camada oculta. Depois que esses pesos são definidos, os pesos da camada de saída são calculados analiticamente. Uma topologia ELM genérica é representada em Figura 1

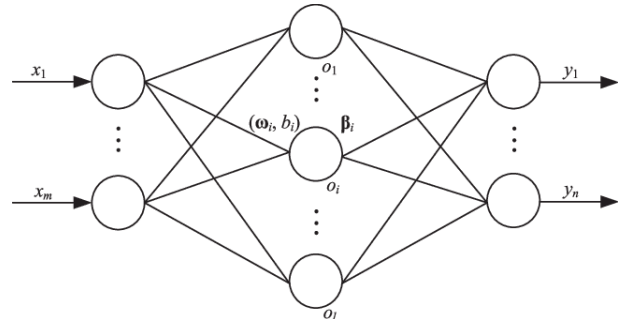


Fig. 1. ELM Topologia genérica

O seu entendimento é simples:

- 1) multiplique as entradas por pesos
- 2) adicionar preconceito
- 3) aplique a função de ativação
- 4) repita as etapas 1-3, número de camadas vezes
- 5) calcular saída
- 6) retropropagar
- 7) repita tudo

O ELM remove a etapa 4 (porque é sempre SLFN), substitui a etapa 6 pela matriz inversa e faz isso apenas uma vez, então a etapa 7 também vai embora.

A função de ativação da camada de entrada utilizada nesta pesquisa é a sigmoideal, enquanto que a função de saída é a identidade, uma vez que se trata de modelos de regressão.

### B. Ensemble models (EM)

Os modelos EM são entendidos como combinação de diferentes modelos com o objetivo de melhorar a previsão final [1]. Os autores indicam que a combinação utiliza-se de modelos de aprendizagem de máquina, pois eles apresentam bons desempenhos individuais. Entretanto, deixam claro que essa não é a única maneira de combinar modelos.

Evidências [3] mostram que a combinação de modelos resulta em desempenho superior aos de modelos individuais. No caso deste estudo, a combinação de modelos de regressão tem o objetivo de combinar diversos modelos de regressão no sentido de melhorar o desempenho final de previsão de valores numéricos distribuídos em dados sequenciais. Segundo [4], a combinação de modelos apresenta maior robustez e melhor acurácia que modelos individuais.

A maneira mais simples de combinar modelos de regressão se dá pela média aritmética dos resultados dos modelos individuais. Dessa forma, avalia-se o desempenho do EM de regressão por meio da equação 1:

$$E_{ens} = \frac{1}{k} E_{média} \quad (1)$$

em que  $k$  é a quantidade de modelos do Ensemble e  $E_{ens}$  é o erro da combinação. Neste trabalho, seguindo os ditames de [1], a combinação de modelos é realizada por meio de uma média ponderada, cuja formulação está disposta na seção II.

Ainda, é importante destacar o fato de que o aumento da quantidade de modelos na combinação nem sempre produzirá melhoria, pois segundo [1], incluir mais modelos ajuda a reduzir o erro final do Ensemble notadamente quando os erros dos modelos individuais não são autocorrelacionados. Isso ocorre, geralmente, quando o processo de geração do pool de regressores produz diversidade dentre os modelos.

### C. Particle swarm optimization (PSO)

Otimização do Enxame de Partículas (PSO) Kennedy [5] é um dos algoritmos de otimização de enxame mais Utilizado. Ele foi desenvolvido para imitar os movimentos de bandos de pássaros ou cardumes de peixes. O PSO foi aplicado com sucesso a muitos problemas: treinamento de rede neural, otimização de função, classificação de padrões e assim por diante. O processo de otimização se inicia com a seleção das variáveis e as restrições existentes e, por meio do espaço de busca, verifica-se qual é o ponto ótimo da função objetivo do problema em questão. As possíveis soluções, chamadas de partículas, “voam” no espaço da situação, seguindo as partículas ótimas, como destacado na Figura 2.

A formula de atualização é dada pela Eq.2

$$P_{i+1} = P_i + V_{i+1} \quad (2)$$

Onde  $P_i$  é a posição da partícula e  $V_i$  é a velocidade da partícula na interação  $i$ . A velocidade é calculada pela Eq.3

$$v_{i+1} = w_{i+1}v_i + c_1rand_1(pbest_i - P_i) + c_2rand_2(gbest_i - P_i) \quad (3)$$

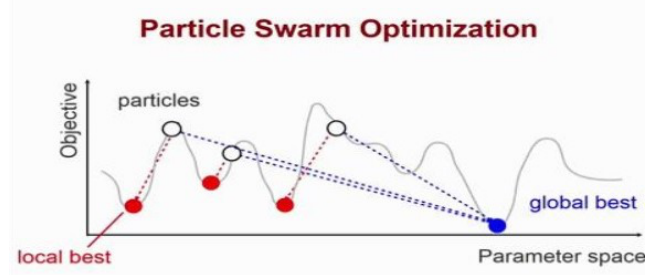


Fig. 2. Processo de otimização PSO

A cada iteração  $i + 1$ , cada partícula  $P$  atualiza a sua velocidade e posição onde  $rand_1$  e  $rand_2$  são números aleatórios independentes e uniformemente distribuídos no intervalo de 0 a 1;  $c_1$  e  $c_2$  são os fatores cognitivo e social, que controlam o fluxo de informação entre a partícula e seu enxame. Quando  $c_2 > c_1$ , a partícula tem mais confiança no enxame, ou seja, na melhor posição alcançada pelo enxame no tempo  $i$ , denotada como  $gbest$ . No caso de  $c_1 > c_2$ , a melhor posição individual da partícula até o tempo  $i$  ( $pbest_i$ ) tem maior influência na atualização da sua velocidade;  $w$  é chamado de fator de inércia e controla a influência da velocidade anterior na atualização da velocidade da partícula Zhang [6]. A Eq. (4) define o valor de  $w$ , onde  $k_{max}$  é o número máximo de iterações do PSO,  $W_{max}$  e  $W_{min}$  são os valores máximo e mínimo para  $w$ , respectivamente. Na abordagem utilizada nesse trabalho  $w$  é inicializado com o valor de  $W_{max}$ . A Tabela I elenca os valores dos parâmetros do PSO.

$$w_{i+1} = \frac{(w_i - i)}{i_{max}} (W_{max} - W_{min}) \quad (4)$$

TABLE I  
PARÂMETROS ADOTADOS NO PSO.

Parâmetro	Valor
Parâmetro Cognitivo ( $c_1$ )	1.49618
Parâmetro Social ( $c_2$ )	1.49618
Fator de Inércia Inicial( $w$ )	0.7298
Fator de Inércia Máximo ( $W_{max}$ )	0.8
Fator de Inércia Mínimo ( $W_{min}$ )	0.2

## II. ELM ENSEMBLE MODEL BOOSTED BY PSO

Nesta seção descrevemos o método proposto por [1] o qual combina três técnicas: *Extreme Learning Machines*, *Ensemble Model*, *Particle Swarm Optimization*. Esse método é utilizado para a tarefa de previsão de séries temporais. Conforme [1] a idealização da combinação dessas técnicas surgiu para modelar sistemas sequencial online [7], assim como para resolver problemas de séries temporais [8].

O método idealizado inicia o processo pela modelagem dos dados por meio das redes ELM. No primeiro momento, é gerado um *pool* de modelos ELM treinados na porção de

dados reservada para tal tarefa. Após a geração dos modelos, constrói-se o *Ensemble* com o método da média ponderada, o qual se propõe a adicionar mais complexidade e melhorar os resultados quando se compara como método da média [1]. Tal fato ocorre, segundo os autores, porque pressupõe-se que exista diversidade no *pool* de modelos. Essa diversidade garante que alguns modelos podem ser melhores do que outros em determinadas partes dos conjuntos de dados.

Sob a ótica de adicionar complexidade ao *Ensemble*, o trabalho supracitado afirma ser possível utilizar algumas heurísticas para otimizar os pesos (representam as ponderações de cada modelo individual) da combinação de modelos. Dentre as heurísticas tem-se Algoritmos Genéticos e Particle Swarm Optimization. Ainda, adicionalmente implementam a sistemática de otimização da combinação com PSO por meio do ranqueamento do desempenho dos modelos ELM individuais.

A previsão final gerada pelo *Ensemble* otimizado pelo PSO está representada na equação 5

$$\hat{Y}_{ens}(X_{M \times 1}) = \sum_{j=1}^k \alpha_j \hat{Y}_j(X_{M \times 1}) \quad (5)$$

Em que,  $\alpha_j$  representa o peso associado ao  $j$ -ésimo modelo ELM (*expert*) do *Ensemble*,  $\hat{Y}_j()$  é a previsão (saída) do  $j$ -ésimo modelo ELM para os dados de entrada  $X_{M \times 1}$ . Vale salientar que a soma de todos os pesos é igual ao valor um (1) como demonstra-se abaixo na Equação 6,

$$\sum_{j=1}^k \alpha_j = 1. \quad (6)$$

Para que isso ocorra é necessário aplicar uma normalização nos dados dos pesos  $\alpha$  por meio da aplicação da função softmax, uma vez que os valores dos pesos gerados pelo algoritmo PSO variam no intervalo dos números reais, podendo apresentar valores negativos. Isto é, aplica-se a equação a seguir:

$$A_{normalizado} = \frac{e^A}{\sum_{j=1}^k e^{\alpha_j}} \quad (7)$$

Em que  $A$  representa o vetor de pesos  $A = [\alpha_1, \dots, \alpha_k]$ , sendo  $\alpha_i$  o peso individual do modelo  $i$  e  $\alpha_k$  o peso do  $k$ -ésimo modelo do *Ensemble*. Destacamos que no numerador, a função exponencial é aplicada a cada elemento do vetor de pesos  $A$ . Neste ponto, nós discordamos da função de normalização utilizada por Larrea et al [1], tendo em vista que eles utilizaram a normalização  $A_{normalizado} = \frac{A}{\sum_{j=1}^k \alpha_j}$ , a qual utiliza-se apenas para valores positivos, não gerando valores percentuais ranqueados em função da magnitude.

Essa normalização é realizada antes de se avaliar a função de erro, traduzida na função de aptidão dentro do algoritmo PSO.

$$NMSE(F) = \frac{\sum_{h \in \mathcal{T}} (observação_h - previsão_h)^2}{\sum_{h \in \mathcal{T}} (observação_h - média_{\mathcal{T}})^2} \quad (8)$$

$$\approx \frac{1}{\hat{\sigma}_{\mathcal{T}}^2} \frac{1}{F} \sum_{h \in \mathcal{T}} (x_h - \hat{x}_h)^2$$

Em que  $h = 1, \dots, F$  representa a quantidade de observações (tamanho amostral) para calcular o erro;  $média_{\mathcal{T}}$  e  $\hat{\sigma}_{\mathcal{T}}$  são a média e a variância das observações. Ainda, ressalta-se que foi realizado um pré-processamento chamado filtro gaussiano (*Standard Scale*), o qual consiste em um escalonamento dos dados, tal qual é feito da seguinte forma: para cada observação, remove-se a média e divide-se a subtração pelo desvio padrão da amostra do conjunto de dados.

O último passo do método proposto é o ajuste fino dos parâmetros do algoritmo PSO, quais sejam, a inércia  $w$  e os componentes cognitivo ( $c_1$ ) e social ( $c_2$ ). Segundo Larrea et al. [1], é possível utilizar valores previamente ajustado de outros trabalhos com problema semelhante.

### III. EXPERIMENTOS E RESULTADOS

No que diz respeito aos Experimentos deste trabalho, seguimos [1] e utilizamos dois experimentos, dos quais o primeiro, Santa Fé laser strength, teria a finalidade de validar o método proposto, ao passo que o segundo seria o experimento responsável por investigar o método em um caso de aplicação real, isto é a Previsão de consumo de energia espanhola.

Como métrica de avaliação dos resultados, para ambos os experimentos, utilizamos o *Root Mean Squared Error* (RMSE), cuja fórmula está descrita na equação 9,

$$RMSE = \sqrt{\frac{1}{F} \sum_{h=1}^F (\hat{Y}_h - Y_h)^2} \quad (9)$$

em que  $F$  é a quantidade de observações previstas,  $\hat{Y}_h$  é a previsão do modelo e  $Y_h$  é o valor real da série temporal.

Sendo assim, os experimentos e resultados são divididos em duas subseções: Experimento I: Santa Fé laser strength e Experimento II: Consumo de energia espanhola.

#### A. Experimento I: Santa Fé laser strength

O conjunto de dados Santa Fé laser strength possui 1100 observações, das quais foram selecionadas as primeiras 1000 instâncias para treinamento e as 100 restantes para teste. No que diz respeito à topologia das redes ELM, selecionamos alguns valores para as quantidades de dados de entrada e de neurônios para a camada escondida como seguem: [10, 20, 30, 40, 50, 100, 150] e [20, 50, 100, 110, 150, 160, 170], respectivamente. Esses valores seguiram os ditames do trabalho base para este relatório. No que diz respeito aos parâmetros do algoritmo PSO, optamos por manter os mesmos valores utilizados por Larrea et al. [1], os quais estão detalhados na Tabela I.

Diante da figura 4, é possível notar que a série apresenta autocorrelação estatisticamente significativa até o valor 30,

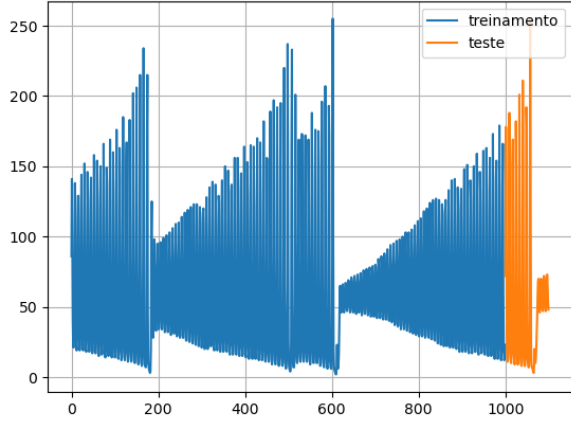


Fig. 3. Série Santa Fé

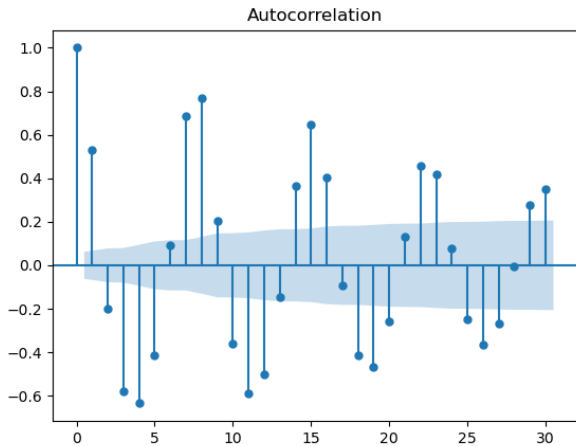


Fig. 4. ACF da série Santa Fé

contudo Larrea et al. [1] afirmam que a maior importância está associada ao valor 10, que representa o instante passado de ordem 10. Isso implica dizer que a melhor configuração para a quantidade de neurônios na camada escondida seria de dez observações passadas para prever uma observação futura.

Nesse sentido, construímos os modelos com a estratégia detalhada na metodologia deste trabalho, ou seja, geramos 100 modelos ELM iniciais treinados para os dados de treinamento. Em seguida, partimos para o processo de geração do Ensemble com o PSO, iniciando a partir de um mecanismo incremental do melhor modelo individual até o pior, salvando os resultados de todos os ensembles para cada configuração detalhada anteriormente, cujos pesos foram ajustados a partir do algoritmo de swarm PSO padrão.

Observamos que a média da variação do erro de treinamento (RMSE) em relação ao erro de teste foi de 94,23%. Entendemos que esse valor é elevado tendo em vista ser uma

taxa de variação percentual. Isso pode ter ocorrido em virtude de não definição de um conjunto de validação para otimizar os pesos com o algoritmo PSO. Conforme artigo base, os autores não informaram um conjunto de validação, constando apenas conjuntos de treinamento e teste. Tendo o primeiro a participação pela geração do pool e otimização dos pesos pelo PSO. Sendo assim, entendemos que o aumento no valor da média do erro de previsão dos Ensembles do conjunto de treinamento para o de teste pode ter sido entendido como sobreajustamento, ou melhor, *overfitting*.

Diante disso, decidimos que seria importante definir um conjunto de validação para treinar o algoritmo PSO. Então, definimos que o conjunto de treinamento inicial seria separado em dois conjuntos, dos quais o primeiro contendo as 80% primeiras observações com a finalidade de geração do pool de ELM, enquanto o segundo sendo formado pelas 20% observações restantes (consideramos como validação), para treinar o algoritmo PSO.

Nesse sentido, seguimos a estratégia de utilizar o pool para prever as observações do conjunto de validação na mesma perspectiva empregada para gerar as previsões para os dados de teste, levando em consideração a utilização das previsões dos dados futuros como *inputs* para os modelos ELM, em outras palavras, evitando usar as observações do conjunto de validação como dados de entrada passados. Portanto, as previsões são feitas conforme janelamento móvel de um passo por vez, diante do qual se prevê uma observação em um determinado instante futuro, digamos  $t + 1$  e a utiliza no próximo instante como dados de entrada para prever o instante  $t + 2$ . Essa estratégia tem a finalidade de evitar o *overfitting* ao se utilizar o PSO nos mesmos dados de geração do pool de ELM.

Em seguida, computamos a variação média do erro de treinamento em relação ao erro de teste considerando a otimização do pso com o conjunto de validação e observamos um valor de 57,69%. Esse valor foi consideravelmente inferior aos 94,23% observados pela primeira estratégia empregada. Dessa feita, optamos por manter essa estratégia de levar em consideração o conjunto de validação para otimizar os pesos com o algoritmo PSO.

A tabela II apresenta cinco ensembles treinados pelo PSO com os menores erros calculados a partir do RMSE. Em comparação com o estudo realizado com [1], notou-se uma diferença quando comparamos a quantidade de *inputs* (observações passadas para prever uma observação futura). No nosso estudo, verificamos que as topologias que resultaram nos melhores resultados continham 30, 40 e 20 neurônios na camada de entrada, enquanto que o estudo supracitado faz referência apenas a 10 observações passadas. Além disso, os autores afirmaram que quanto maior a quantidade de neurônios na camada escondida, menores eram os erros produzidos pelos modelos. Nossos resultados demonstraram que isso nem sempre é o observado, pois o segundo e o quarto melhores modelos possuíram apenas 20 neurônios.

Sendo assim, diante dos resultados do nosso Experimento I, observamos que o objetivo da pesquisa não foi atendido, uma

TABLE II  
MELHORES MODELOS ELM EM RELAÇÃO AO RMSE

n_input	n_hidden	RMSE
30	150	1.275
40	20	1.300
30	160	1.309
20	20	1.356
20	150	1.356

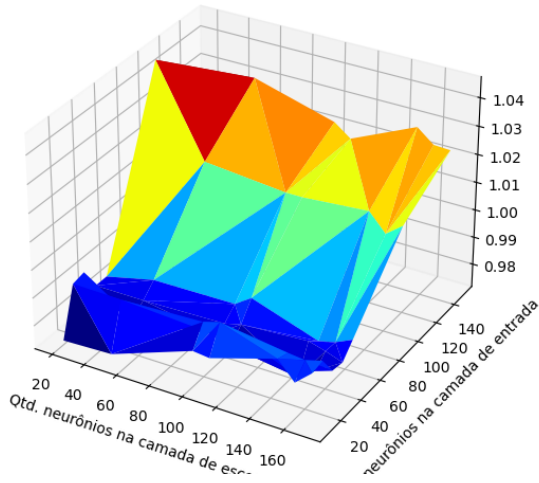


Fig. 5. RMSE em função das quantidades de neurônios na camada de entrada e na camada escondida para o conjunto de treinamento

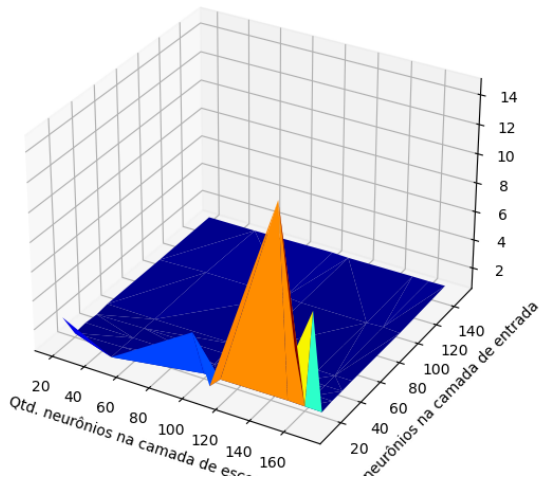


Fig. 6. RMSE em função das quantidades de neurônios na camada de entrada e na camada escondida para o conjunto de validação

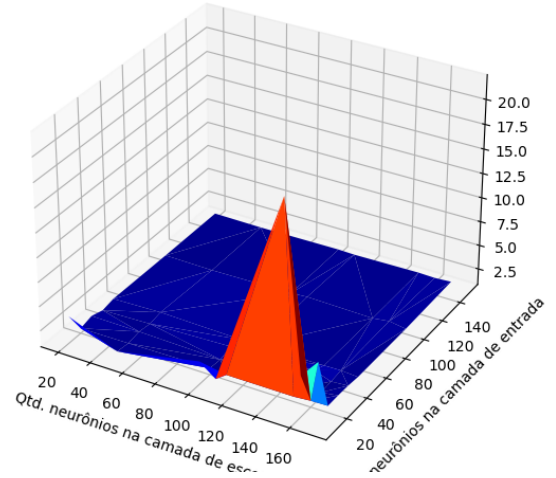


Fig. 7. RMSE em função das quantidades de neurônios na camada de entrada e na camada escondida para o conjunto de teste

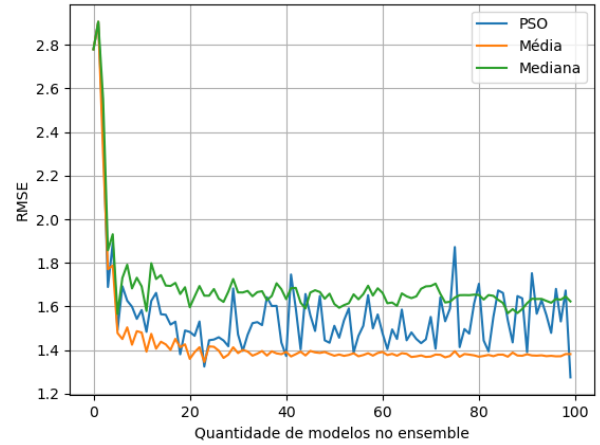


Fig. 8. RMSE em função da quantidade de ELM do pool com melhor desempenho

vez que o algoritmo PSO não conseguiu produzir Ensembles melhores do que a média. Ainda, notamos que os melhores modelos ELM individuais podem ser superiores à combinação de mais de um regressor conforme Figura 9.

#### B. Experimento II: Consumo de energia espanhola

A seleção dos dados de entrada e camada de neurônios escondida no experimento II seguem como [30,40,50,60,90] e [100,140,150,160]. Esses valores seguem o consumo por hora (MW) fornecidos pela rede espanhola em seu site, o conjunto de dados consiste de dados do dia 1 de janeiro de 2014 as 00:00 horas à 31 de dezembro de 2016 as 11:00. A princípio tentamos além do gráfico de autocorrelação Figura 11, uma janela de dados incremental indo de 10,20,...,300 tanto para inputs(observações passadas para prever uma observação futura) quanto neurônios escondidos afim de encontra



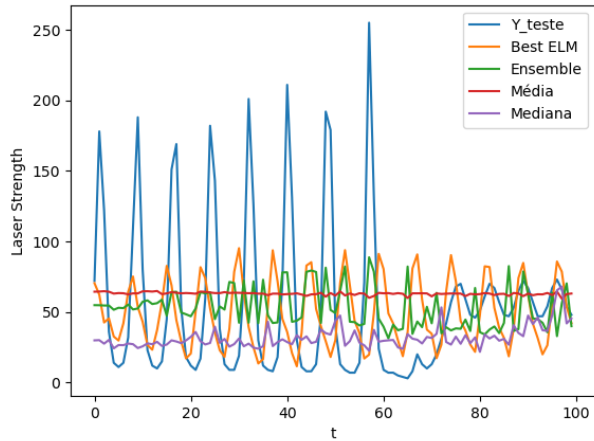


Fig. 9. Valores observados para o conjunto de teste e previsões pelo melhor modelo ELM, PSO, Média e Mediana

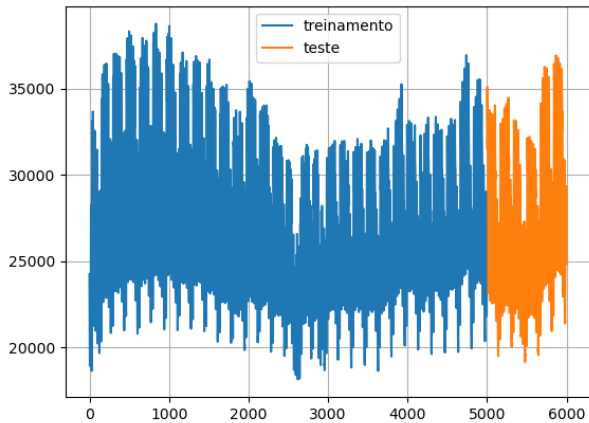


Fig. 10. Série Consumo de energia espanhola

as melhores topologias para o ELM como proposto por [1], entretanto essa abordagem demanda muito tempo para execução e teste então optamos por escolher os melhores modelos apontados no artigo base [1]. Esta base conta com 26304 instancias então esse experimento foi conduzido em duas etapas:

- 1) A primeira segue as diretrizes do trabalho citado onde foram utilizadas 5.000 instancias para o treinamento e o restante para teste. Depois da partição da série, o processo de treinamento é iniciado processando e gerado as 100 ELM para cada topologia, criando o ensemble que será uma partícula do PSO e assim criando uma população de maneira incremental até 100 partículas que serão otimizadas. Esse procedimento mostrou-se bastante custoso, cada topologia leva cerca de 100 minutos para ser otimizada; a execução levou cerca de 3 dias para o treinamento e 7 dias para o teste, totalizando 10

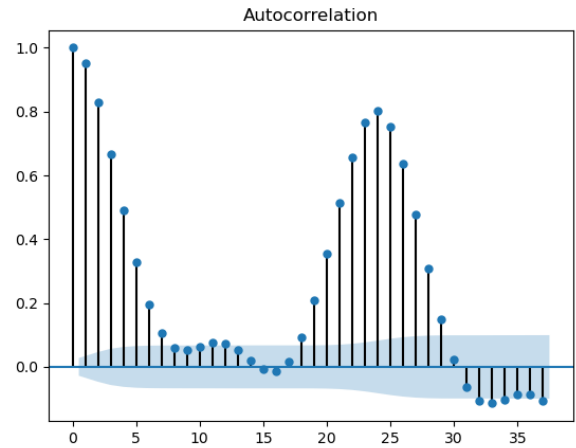


Fig. 11. ACF da série Consumo de energia espanhola

dias para uma única execução, por conta disso não foi possível fazer as 10 execuções do experimento como descrito no trabalho. [1].

- 2) A segunda segue uma segunda estratégia onde é criado um conjunto de validação nas mesmas proporções do experimento I mantendo o mesmo tamanho do conjunto de treino com 5.000 instancias e do conjunto de teste é tirada uma amostra de 1.000 instancias para testes, o conjunto de treinamento é subdividido em treino 80% das instancias e validação 20%, o processo de otimização via PSO é feito com o conjunto de validação. A execução dessa estratégia foi mais rápida em relação a etapa anterior levando aproximadamente um dia para treinamento e 6 horas no teste para fins de comparação entre as estratégias foi feita uma etapa de teste com um conjunto de 1.000 instancias na estratégia sem validação.

Ao analisarmos a estratégia que não usa validação observamos que a média da variação do erro de treinamento (RMSE) em relação ao erro de teste foi de 452%. Entendemos que esse valor é muito elevado tendo em vista ser uma taxa de variação percentual. Em contrapartida temos a estratégia com a validação que obteve -9% na sua média da variação do erro de treinamento em relação ao teste. Como não foi informado qual quer menção a utilização de validação no artigo citado optamos por manter essa estratégia com validação para otimizar o PSO.

A tabela III apresenta os cinco melhores ensembles com base nos menores RMSE. Comparando os o resultado com o estudo feito no trabalho [1] nota-se uma diferença ao compararmos a quantidade de inputs. No nosso experimento para a base espanhola as topologias que apresentaram os melhores resultados contêm 90 e 60 neurônios na camada de entrada entre os cinco citados a topologia com 90 neurônio está em quatro, enquanto isso no estudo supracitado faz referências a 30,40,60,90 inputs. Nesse experimento também não foi notado uma melhoria com o aumento de neurônios da camada escondida.

TABLE III  
MELHORES MODELOS ELM EM RELAÇÃO AO RMSE - CONSUMO DE  
ENERGIA ESPANHOLA

n_input	n_hidden	RMSE
90	140	1.897
90	160	2.159
90	100	2.232
90	150	2.545
60	160	2.882

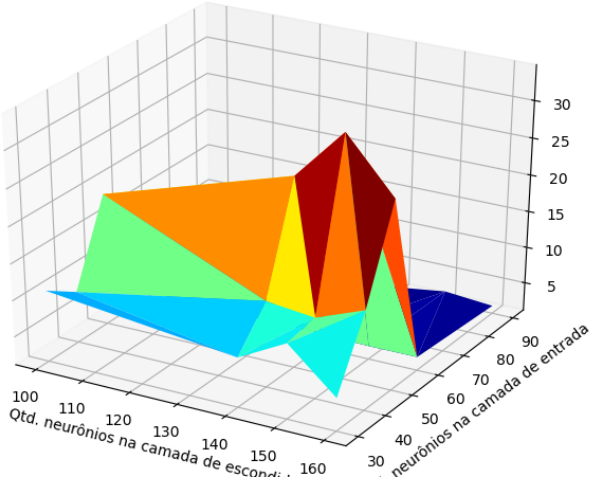


Fig. 12. RMSE em função das quantidades de neurônios na camada de entrada e na camada escondida para o conjunto de treinamento - Base Consumo de energia espanhola

Diante os resultados do experimento II observando a figura 15 notamos que o objetivo de pesquisa também não foi atendido já que a média foi melhor que o ensemble gerado pelo PSO e ainda é possível notar modelos ELM individuais com bons resultados.

### C. Conclusões

Diante do objetivo deste trabalho, isto é, replicar o estudo [1], observamos algumas divergências nos resultados que podem ter decorrido da metodologia de implementação do Ensemble otimizado com o PSO. Nesse sentido, um dos pontos cruciais que representa a divergência entre os trabalhos foi a utilização, no nosso caso, de um conjunto de validação para otimizar os pesos pelo algoritmo PSO. Essa decisão foi importante e nos mostrou que houve uma redução na taxa média da diferença entre o erro de treinamento em relação ao erro de teste.

Ademais, notamos que o Ensemble construído pela média produziu melhores resultados (*i.e.*, menor valor para a métrica de erro RMSE), em média, que o método proposto, para ambos os experimentos. No entanto, a metodologia replicada demonstrou menores valores de erros que a combinação pela mediana.

Como crítica ao trabalho replicado, nós decidimos por comentar o que se segue: Pensamos que o Ensemble proposto por Larrea et al. [1] deveria ter sido otimizado por meio de

RMSE vs Neurônios camada escondida vs Neurônios camada de entrada

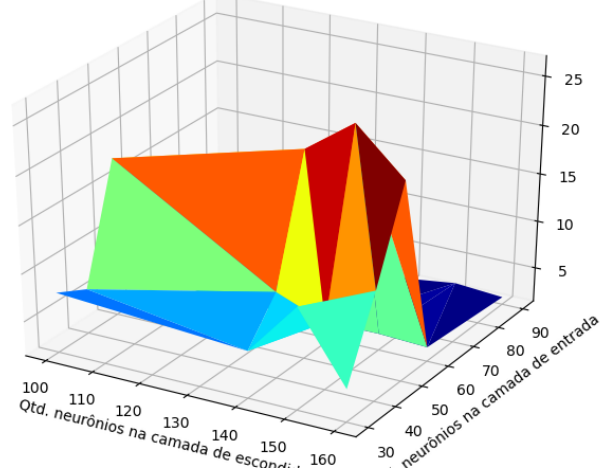


Fig. 13. RMSE em função das quantidades de neurônios na camada de entrada e na camada escondida para o conjunto de teste - Base Consumo de energia espanhola

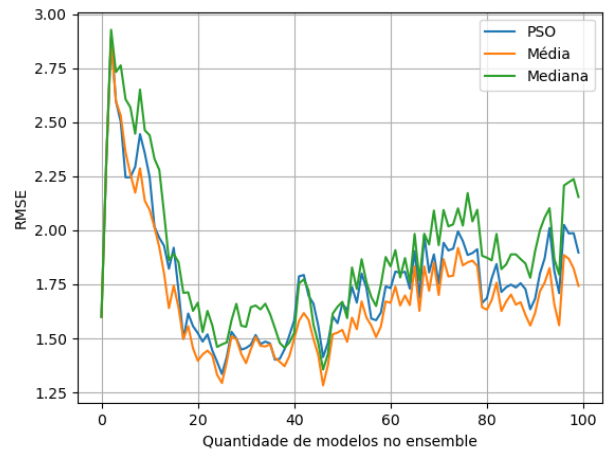


Fig. 14. RMSE em função da quantidade de ELM do pool com melhor desempenho - Base Consumo de energia espanhola

alguma técnica de seleção dinâmica ou algo que se adaptasse à dinâmica da série temporal como uma validação cruzada ao longo do tempo, metodologia esta que falarei adiante. No caso em questão o modelo proposto gerou pesos estáticos no processo de validação e ajuste do algoritmo PSO.

Além disso, os autores Larrea et al. [7] poderiam ter utilizado outro modelo individual (para modelar os dados) que fosse menos performático como a Regressão linear. A utilização de redes ELM como regressor base monolítico do ensemble pode gerar pouca diversidade no processo de geração. Ademais, não foi utilizado um processo de janela móvel que adaptasse os pesos do PSO ao longo da dinâmica temporal.

Dito isso, entendemos que seria salutar implementar alguma técnica de detecção de mudança de distribuição dos dados,

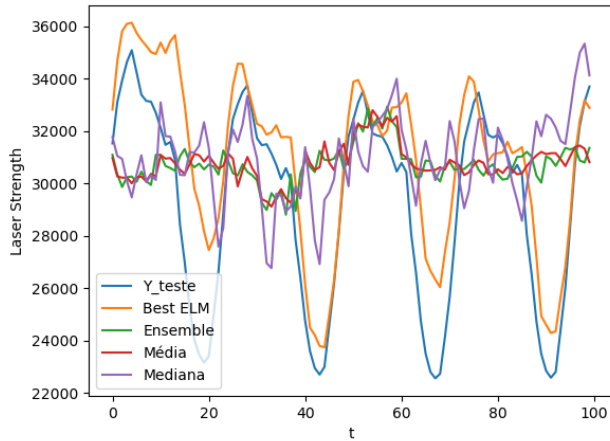


Fig. 15. Valores observados para o conjunto de teste e previsões pelo melhor modelo ELM, PSO, Média e Mediana - Base Consumo de energia espanhola

caso houvesse esse padrão na série temporal. Para tanto, existe metodologia [9] que combina as técnicas de PSO e redes neurais artificiais para prever séries temporais em ambientes dinâmicos, ou seja, séries temporais com mudança de distribuição ao longo do tempo. Um dos algoritmos de PSO seria o CQSO (cooperativo quântico).

Ademais, o processo de treinamento e teste como foi determinado, isto é, utilizar apenas as primeiras observações como dados de treinamento, ao passo que empregar as amostras subsequentes como dados de teste seria uma forma simplória (reduzindo a complexidade) para o tratamento de séries temporais, uma vez que a própria natureza desse dado é a dinâmica, mudança temporal dos dados. Portanto, sugere-se utilizar um processo de validação cruzada com um janelamento móvel, isto é, definidas as quantidades de janelas, fixa-se as proporções de treinamento, validação e teste e realiza-se a validação cruzada de forma dinâmica, janela por janela, computando-se os erros e avaliando o desempenho ao longo do tempo.

## REFERENCES

- [1] M. Larrea, A. Porto, E. Irigoyen, A. J. Barragán, and J. M. Andújar, "Extreme learning machine ensemble model for time series forecasting boosted by PSO: Application to an electric consumption problem," *Neurocomputing*, nov 2020.
- [2] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*. IEEE.
- [3] J. Mendes-Moreira, C. Soares, A. M. Jorge, and J. F. D. Sousa, "Ensemble approaches for regression: A survey," *Acm computing surveys (csur)*, vol. 45, no. 1, pp. 1–40, 2012.
- [4] N. García-Pedrajas, C. Hervás-Martínez, and D. Ortiz-Boyer, "Cooperative coevolution of artificial neural network ensembles for pattern classification," *IEEE transactions on evolutionary computation*, vol. 9, no. 3, pp. 271–302, 2005.
- [5] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*. IEEE.
- [6] Y. Zhang, S. Wang, and J. G., "A comprehensive survey on particle swarm optimization algorithm and its application," *Mathematical Problems in Engineering*, vol. 2015, p. 38, 2015.

- [7] Y. Liu, B. He, D. Dong, Y. Shen, T. Yan, R. Nian, and A. Lendasse, "Particle swarm optimization based selective ensemble of online sequential extreme learning machine," *Mathematical Problems in Engineering*, vol. 2015, 2015.
- [8] A. Porto, E. Irigoyen, and M. Larrea, "A pso boosted ensemble of extreme learning machines for time series forecasting," in *The 13th International Conference on Soft Computing Models in Industrial and Environmental Applications*. Springer, 2018, pp. 324–333.
- [9] S. A. Abdulkarim and A. P. Engelbrecht, "Time series forecasting with feedforward neural networks trained using particle swarm optimizers for dynamic environments," *Neural Computing and Applications*, pp. 1–17, 2020.