



DOI: 10.5335/rbca.v13i2.12067

Vol. 13, № 2, pp. 58-72

Homepage: seer.upf.br/index.php/rbca/index

#### ARTIGO ORIGINAL

# Um estudo comparativo de ensembles híbridos para aplicações de previsão de séries temporais

# A comparative study of hybrid ensembles for forecasting time series applications

Flávio Mineiro do Amaral Filho<sup>1</sup>, Saymon Galvão Bandeira <sup>6,2</sup>, Symone Gomes Soares Alcalá <sup>6,2</sup>, Talles Marcelo G. de A. Barbosa <sup>6,1</sup>

<sup>1</sup>Pontifícia Universidade Católica de Goiás, Goiânia, GO, Brasil, <sup>2</sup>Universidade Federal de Goiás, Faculdade de Ciências e Tecnologia, Aparecida de Goiânia, GO, Brasil

\*flaviofilho2908@gmail.com, saymongb@gmail.com, saymongb@gmail.com, talles@pucgoias.edu.br

Recebido: 30/11/2020. Revisado: 31/05/2021. Aceito: 05/07/2021.

#### Resumo

Previsões de séries temporais auxiliam a tomada de decisão em diversas áreas como marketing, economia e indústria, sendo que a principal finalidade é estimar o comportamento futuro de uma sequência de observações. Nesse sentido, conjuntos de modelos (ensembles) híbridos, que combinam modelos de aprendizado de máquina e estatísticos, têm se mostrado eficientes para prever séries temporais. Entretanto, a correta seleção dos modelos e da combinação em um ensemble é importante para assegurar o desempenho do sistema. Assim, este trabalho propõe e compara diferentes abordagens de ensembles híbridos para melhorar a previsão de séries temporais. São propostos um conjunto de modelos de previsão e diferentes estratégias de combinação para lidar com séries temporais de diferentes padrões. A primeira abordagem de ensembles combina um conjunto de modelos com acurácia utilizando quatro estratégias de combinação. Já a segunda abordagem ensemble seleciona automaticamente modelos e a combinação utilizando meta-heurísticas. As abordagens são comparadas utilizando um novo conjunto de dados de uma empresa de distribuição de cosméticos e um conjunto de dados público. Os resultados demonstram que os ensembles propostos são eficientes para diminuir o erro de previsão.

Palavras-Chave: Aprendizado de Máquina; Ensemble; Meta-heurística; Previsão de Séries Temporais;

#### Abstract

Time series forecasting assists decision making in several areas such as marketing, economics and industry, being that the main purpose is to estimate the future behavior of a sequence of observations. Thus, sets of hybrid models (ensembles), which combine machine learning and statistical models, have been shown efficient in predicting time series. However, the correct selection of models and combination in an ensemble is important to ensure the system performance. Thus, this work proposes and compares different approaches of hybrid ensembles to improve time series forecasting. A set of forecasting models and different combination strategies are proposed to deal with time series of different patterns. The first ensemble approach combines a set of accurate models using four combination strategies. The second ensemble approach selects automatically models and combination using meta-heuristics. The approaches are compared using a new data set from a cosmetic distribution company and a public data set. The results demonstrate that the proposed ensembles are efficient to reduce the prediction errors.

Keywords: Machine Learning; Ensemble; Metaheuristic; Time Series Forecasting;

# 1 Introdução

Previsão é uma tarefa que utiliza conhecimentos disponíveis para prever comportamentos futuros com a maior exatidão possível (Hyndman and Athanasopoulos, 2018). De acordo com Montgomery et al. (2008), áreas como gerenciamento de operações, marketing, finanças, economia, controle de processos industriais e demografia, enxergam as previsões como algo crítico, visto que elas são importantes para realizar planejamentos e auxiliar a tomada de decisão. Na área financeira, previsões são utilizadas em investimentos na bolsa de valores, em que elas servem para nortear os investidores a tomarem suas decisões de compra e venda (Thiele and Adami, 2016, Machado et al., 2020).

Em geral, previsões podem ser dividas em duas categorias: qualitativa e quantitativa. Abordagens qualitativas não utilizam dados históricos para realizar previsões e, por outro lado, abordagens quantitativas utilizam informações históricas disponíveis. Séries temporais são aplicações comuns em abordagens quantitativas. Elas são constituídas por dados que possuem uma observação sequencial sobre um período de tempo, sendo que elas possuem padrões como tendência, sazonalidade e ciclos. A principal finalidade de prever uma série temporal é estimar o comportamento futuro da sequência das observações (Hyndman and Athanasopoulos, 2018).

Um desafio em previsões de séries temporais é definir o horizonte de previsão, isto é, a quantidade de observações futuras a serem previstas (Montgomery et al., 2008). Dependendo do problema, pode ser necessário um horizonte de previsão curto ou longo. Além disso, quanto mais distante está o evento previsto, mais incerta é a previsão (Hyndman and Athanasopoulos, 2018). Outro desafio é uma quantidade de dados suficientes da série temporal para modelagem, uma vez que modelos de previsão complexos necessitam de grandes quantidades de dados.

Exemplos de modelos de previsão para séries temporais incluem modelos simples, como o método ingênuo (Makridakis et al., 2020), que utiliza a última observação da série como previsão da próxima observação; ou modelos mais sofisticados, como Redes Neurais Artificiais (RNAs) (Zhang et al., 1998), que extraem padrões e relações ocultas nos dados para realizar previsões.

Novas abordagens que combinam múltiplos modelos de previsão têm sido propostas na literatura (Montero-Manso et al., 2020, Moon et al., 2020, Chen and Liu, 2020, Li et al., 2020). Estas abordagens são conhecidas como *ensemble learning* ou aprendizado em conjunto. A principal motivação para combinar modelos é que, na maioria dos casos, um conjunto de modelos pode obter melhor desempenho na previsão do que um modelo único do conjunto (Bolón-Canedo and Alonso-Betanzos, 2019). Em *ensembles*, as previsões dos modelos podem ser combinadas utilizando estratégias como média ponderada (Wichard, 2016) e média simples (Bandeira et al., 2020).

Trabalhos utilizando séries temporais têm demostrado que *ensembles* alcançam melhor resultado quando combinam modelos de aprendizado de máquina (em inglês, *Machine Learning* – ML) e modelos estatísticos (Makridakis et al., 2018). De acordo com Makridakis et al. (2020), estes resultados incentivam pesquisadores a explorar métodos

de combinação híbridos, que beneficiam-se das vantagens dos modelos estatísticos e ML, procurando novas maneiras de selecionar modelos e de combinar os mesmos.

Além disso, observa-se na literatura trabalhos que propõem abordagens de seleção precisa dos modelos para constituir ensembles de modo a minimizar o erro das previsões (Soares et al., 2013, Wang and Alhamdoosh, 2013, Pulido et al., 2014, Adhikari et al., 2015, Chen and Liu, 2020, Li et al., 2020). A maioria destes trabalhos utiliza metaheurísticas para otimizar esta seleção. Segundo Adhikari et al. (2015), a correta seleção dos modelos do ensemble pode melhorar potencialmente a exatidão geral da previsão. Isso porque nem todos os modelos produzem boas previsões para séries temporais específicas. Em geral, o principal desafio na seleção de modelos do ensemble é a escolha de uma heurística ou meta-heurística eficiente e rápida. Além disso, baixa quantidade de dados ou observações com alta variabilidade nas séries pode impactar negativamente no desempenho da seleção dos modelos para o ensemble.

Assim, este artigo propõe e compara abordagens de *ensembles* híbridos para melhorar a previsão de aplicações de séries temporais. Para isso, são propostos modelos de previsão para lidar com séries temporais de diferentes padrões, sendo eles: método ingênuo, Suavização Exponencial Simples (SES), método Holt, Modelo Autorregressivo (AR), *AutoRegressive Integrated Moving Average* (ARIMA), *Support Vector Regression* (SVR), RNA, *Extreme Learning Machine* (ELM) e Rede Neural Recorrente (RNR). Além disso, são propostas diferentes estratégias de combinação: média, mediana, média aparada e média ponderada.

A primeira abordagem de *ensembles* híbridos combina um conjunto de modelos com acurácia utilizando as estratégias de combinação descritas. Já a segunda abordagem seleciona automaticamente e de forma otimizada os modelos e a combinação utilizando meta-heurísticas, sendo elas Algoritmo Genético (AG), *Simulated Annealing* (SA) e Otimização por Enxame de Partículas (em inglês, *Particle Swarm Optimization* – PSO). As abordagens são comparadas utilizando um novo conjunto de dados de uma empresa de distribuição de cosméticos; e um conjunto de dados público, chamado CIF2016, um *benchmark* para a previsão de séries temporais (Štěpnička and Burda, 2017). Os resultados demonstram que os *ensembles* híbridos propostos são eficientes para diminuir o erro geral das previsões.

Este artigo está organizado da seguinte forma. A Seção 2 apresenta uma revisão da literatura sobre as estratégias de combinação e a seleção de modelos em *ensembles*. Na Seção 3 são apresentados e propostos os modelos de previsão utilizados, as abordagens de *ensembles* híbridos e as meta-heurísticas utilizadas neste trabalho. A Seção 4 apresenta os dados considerados, a configuração dos modelos de previsão, os resultados e as discussões. Na Seção 5 são apontadas as considerações finais.

# 2 Revisão da Literatura

Nesta seção são apresentados trabalhos que demonstram a eficiência de abordagens *ensemble*, sendo que a Seção 2.1 descreve *ensembles* que utilizam diferentes estratégias de combinação e a Seção 2.2 apresenta trabalhos que realizam

seleção de modelos para ensemble.

# 2.1 Estratégias de Combinação para Ensembles

Estratégias comuns para combinar modelos em um ensemble são: média simples, mediana, média aparada, média ponderada e empilhamento. A média simples soma as saídas (previsões) dos modelos e divide pela quantidade dos modelos para obter a saída final do ensemble. Na mediana, os valores das saídas são ordenados de forma crescente, e o valor central é definido como saída final. Na média aparada, os valores das saídas também são ordenados de forma crescente e uma porcentagem dos valores das extremidades é excluída, em seguida, é obtida a média dos valores restantes (Peck et al., 2007). Para média ponderada, é realizada uma soma ponderada das saídas, em que cada modelo tem um peso atribuído (por exemplo, baseado na exatidão do modelo) (Soares et al., 2013). No empilhamento, as saídas dos modelos são tratadas como entradas para o treinamento de um novo modelo, que prevê o resultado final (Kumar and Jain, 2020).

Soares et al. (2013) utilizam média, mediana, média aparada e média ponderada como estratégias de combinação, sendo que a média ponderada teve os pesos ajustados pela exatidão de cada modelo baseado em um erro. Já Wichard (2016) propõe um *ensemble* híbrido para previsão de séries temporais que combina cinco modelos utilizando média ponderada, sendo que o peso de cada modelo é ajustado baseado na métrica de erro *Symmetric Mean Absolute Percentage Error* (SMAPE).

O trabalho de Montero-Manso et al. (2020) propõe um ensemble híbrido composto por nove modelos e média ponderada como combinação. A diferença desta proposta está no cálculo dos pesos para cada modelo, que são escolhidos por um modelo de ML previamente treinado, denominado metamodelo. No treinamento deste modelo, são utilizados os erros das previsões dos modelos do ensemble e as características da séries como entrada.

Moon et al. (2020) desenvolveram um *ensemble* que utiliza empilhamento. A proposta usa quatro modelos de RNA profundas como entradas para uma regressão de componente principal que realiza a previsão final. O método proposto foi utilizado para fazer a previsão do consumo de energia elétrica em edifícios. De acordo com o trabalho, a proposta apresentou resultados melhores que outros métodos para previsão de consumo de energia.

Bandeira et al. (2020) comparam a média e a média ponderada como estratégias de combinação em *ensembles* para uma aplicação de previsão de demanda. As abordagens foram avaliadas utilizando uma base de dados da competição M3 e uma base de dados de uma indústria de elevadores. Os resultados do artigo demonstram que a média ponderada supera a média simples, e que *ensembles* podem ter bom desempenho na previsão de demanda de séries intermitentes.

# 2.2 Seleção de Modelo para Ensembles

Soares et al. (2013) propõem uma abordagem automática para o desenvolvimento de *ensembles* com RNAs para regressão. A abordagem foi feita utilizando AG e SA que realizam a seleção de um subconjunto de RNAs e a melhor estratégia de combinação para formar um *ensemble*. O artigo apresenta uma comparação do desempenho de cada meta-heurística. Os resultados mostram que AG e SA obtiveram resultados semelhantes, porém o algoritmo AG teve desempenho ligeiramente superior.

O artigo de Wang and Alhamdoosh (2013) apresentou um *ensemble* evolutivo baseado em modelos ELM. No trabalho, a seleção de modelos ELM é feita por meio de um AG, sendo que o *ensemble* é iniciado com dois modelos e a quantidade de modelos é incrementada a cada etapa iterativa do algoritmo até encontrar a melhor quantidade. A proposta foi comparada com outros *ensembles* tradicionais e com um modelo individual de ELM em doze conjuntos de dados de regressão. Os resultados mostraram que o *ensemble* proposto tem melhor performance em todos os conjuntos de dados.

Pulido et al. (2014) desenvolveram um ensemble de RNAs otimizado com algoritmo PSO e agregação fuzzy para prever ações na bolsa de valores do México. O PSO otimiza o ensemble escolhendo o número de RNAs para o ensemble, a quantidade de camadas ocultas e o número de neurônio das RNAs. Além disso, o artigo utiliza fuzzy dos tipos 1 e 2, que são algoritmos que trabalham com incerteza por meio de regras e funções de pertinência para realizar a agregação de RNAs no ensemble. Os resultados demostram que a abordagem proposta tem melhor desempenho na previsão que outros métodos do estado-da-arte.

Adhikari et al. (2015) propõem uma metodologia de *ensemble* seletivo baseada no desempenho individual dos modelos, sendo que cada modelo é avaliado e classificado em um conjunto de validação e, em seguida, são selecionados os *N* melhores modelos para compor um *ensemble* com combinação linear ponderada. Para avaliar a metodologia, os autores testaram o *ensemble* com nove modelos individuais em quatro séries temporais de dados reais. O artigo demonstrou que o *ensemble* proposto alcançou resultados melhores que todos os métodos individuais e da combinação de todos os nove métodos com várias estratégias de combinação.

O estudo de Chen and Liu (2020) apresenta uma abordagem *ensemble*, na qual quatro modelos de ML são treinados para uma série temporal com quatro resoluções de tempos diferentes, formando assim um total de dezesseis submodelos. Estes submodelos são selecionados utilizando Otimização da Borboleta (OB) para compor um *ensemble* de empilhamento, em que os submodelos escolhidos são as entradas para um modelo SVR que realiza a previsão final. OB é uma otimização que busca uma solução ótima inspirada nos movimentos das borboletas para encontrar fontes de alimentos (Arora and Singh, 2019). Os resultados do artigo mostram que a solução proposta supera outros modelos do estado-da-arte.

Li et al. (2020) propõem um *ensemble* para prever a vida útil remanescente de equipamentos. Na proposta do trabalho, os modelos bases são selecionados utilizando otimização *Non-dominated Sorting Genetic Algorithm-II* (NSGA-II), uma variação de AG, que escolhe um conjunto de modelos com acurácia e diversos. As saídas dos modelos escolhidos são as entradas para um modelo RNA que faz uma previsão final por meio de empilhamento. O *ensemble* proposto obteve desempenho superior aos métodos individuais e a

um ensemble com otimização de objetivo único.

Os trabalhos apresentados nesta subseção abordaram seleção de modelos para *ensembles*. Os resultados dos artigos demostraram que a seleção correta de modelos para um *ensemble* podem melhorar a acurácia na previsão de diversas aplicações.

# 3 Propostas de Ensembles Híbridos

Nesta subseção são apresentados os modelos de previsão individuais utilizados e as abordagens com *ensembles* híbridos.

#### 3.1 Modelos de Previsão

Neste trabalho, foram utilizados nove tipos de modelos de previsão de séries temporais com diferentes características para compor os *ensembles* híbridos propostos. São eles:

**Método Ingênuo.** É um método simples, e bastante utilizado como *benchmark*, que define sua previsão com base na última observação da série temporal (Hyndman and Athanasopoulos, 2018).

Suavização Exponencial Simples (SES). Neste modelo, a previsão é uma média ponderada das observações da série temporal, sendo que pesos maiores são atribuídos para as observações mais recentes e pesos menores para observações mais distantes. O parâmetro de suavização  $\alpha$  do método deve ser definido no intervalo de 0 a 1. Se  $\alpha$  estiver próximo de 0, valores antigos terão mais relevância na média ponderada; e, se  $\alpha$  estiver próximo de 1, o método se tornará semelhante a uma previsão do método ingênuo (Hyndman and Athanasopoulos, 2018).

**Método Holt.** Este método é um tipo de suavização exponencial para séries temporais com tendência. Ele é composto por duas suavizações exponenciais, sendo que uma faz a média ponderada do nível da observação e a outra para a tendência. O método Holt precisa de dois parâmetros: um para a suavização do nível  $(\alpha)$  e outro para tendência  $(\beta)$ . Além disso, é possível incluir um parâmetro de amortecimento  $(\phi)$  para que o método não caía em uma previsão com tendência infinita (Hyndman and Athanasopoulos, 2018).

**Modelo Autorregressivo (AR).** Este modelo realiza previsões a partir da combinação linear dos preditores com uma defasagem de ordem *p*. Portanto, são modelos que efetuam uma auto-regressão na variável preditora. Além disso, são modelos que conseguem lidar com vários tipos de padrões de séries temporais (Hyndman and Athanasopoulos, 2018).

Autoregressive Integrated Moving Average (ARIMA). É um modelo que combina a previsão do modelo AR e média móvel. São modelos que possuem como parâmetros a ordem *p* da autorregressão, ordem *q* da média móvel e o grau *d* da diferenciação. Além disso, é possível incluir sazonalidade por meio da adição de parâmetros sazonais parecidos com os componentes não sazonais do modelo (Hyndman and Athanasopoulos, 2018).

Support Vector Regression (SVR). É um modelo de regressão inspirado no modelo Support Vector Machine (SVM). SVM lida com problemas de classificação, sendo que ele otimiza a separação de classes para encontrar um hiperplano

com uma margem máxima de distância de cada classe diferente, por meio da minimização do erro e manutenção da generalização (Awad and Khanna, 2015). Uma diferença entre SVR e SVM é que o primeiro possui como saída valores numéricos reais. Além disso, no modelo SVR foi adicionada uma margem de tolerância de erro chamada épsilon (Sayad, 2020).

Rede Neural Artificial (RNA). É um método baseado nos neurônios humanos para capturar padrões que relacionem entradas e saídas (Hyndman and Athanasopoulos, 2018). Ele é formado por neurônios artificiais que são interligados por pesos, sendo que cada neurônio possui uma função de ativação que permite o mesmo transmitir os dados para outro neurônio. Com esta estrutura, uma RNA consegue armazenar os padrões aprendidos. Os neurônios artificiais são organizados em várias camadas. A primeira camada é a camada de entrada na qual as informações chegam à rede; as camadas ocultas, que podem ser uma ou várias, auxiliam no aprendizado dos padrões; e, a camada de saída transmite o resultado da rede (de Melo Junior et al., 2020, Zhang et al., 1998).

Extreme Learning Machine (ELM). É um modelo RNA com apenas uma camada oculta. Nesta proposta, os pesos entre a camada de entrada e a camada oculta são escolhidos aleatoriamente, e os pesos entre estas camadas não precisam ser ajustados. Já os pesos entre a camada oculta e a camada de saída são ajustados analiticamente. Assim, o modelo ELM não precisa de várias iterações para determinar os pesos, e isso o torna mais veloz que outras RNAs (Huang et al., 2006).

Rede Neural Recorrente (RNR). É uma RNA utilizada para processar dados com valores em sequência. Uma característica deste modelo é o compartilhamento de parâmetros ao longo do tempo de treinamento, possibilitando que o modelo trabalhe com dados sequenciais de modo mais eficiente. Um compartilhamento é feito na unidade de processamento desta rede que é alimentada pela saída da unidade anterior, formando um gráfico recorrente profundo (Goodfellow et al., 2016).

# 3.2 Estratégias de Combinação de Ensemble

Este trabalho propõe a utilização das principais estratégias de combinação em *ensembles*: média, mediana, média aparada e média ponderada. Sendo assim, são propostos quatro *ensembles* com combinação: *ensemble* com média ( $EC_{média}$ ) como estratégia de combinação, *ensemble* com mediana ( $EC_{médiaNparada}$ ), *ensemble* com média aparada ( $EC_{médiaNparada}$ ) e *ensemble* com média ponderada ( $EC_{médiaPonderada}$ ). Cumpre ressaltar que para o desenvolvimento de cada *ensemble*, é gerado um conjunto de modelos de previsão, e todos eles são combinados sem estratégia de seleção.

Assumir que  $f_i(x)$  é a saída do modelo i e F(x) é a saída final de um *ensemble* com n modelos (Adhikari et al., 2015, Soares et al., 2013). Então, o método  $EC_{média}$  utiliza a Eq. (1) para obter sua previsão. O método  $EC_{mediana}$  realiza a ordenação das previsões dos modelos para cada observação futura e, em seguida, é escolhida a previsão central (mediana) das ordenações. Para a abordagem  $EC_{médiaAparada}$ ,

primeiramente são retiradas 5% das previsões mais baixas e 5% das mais altas e, em seguida, é calculada a média das previsões para obter a previsão final. Já o método  $EC_{m\'ediaPonderada}$  usa a Eq. (2), em que o peso de um modelo  $w_i$  é determinado pela Eq. (3), onde o erro SMAPE é calculado em um conjunto de validação.

$$F(x) = \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$
 (1)

$$F(x) = \sum_{i=1}^{n} w_i \cdot f_i(x)$$
 (2)

$$w_{i} = \frac{SMAPE_{i}^{-1}}{\sum_{k=1}^{n} SMAPE_{k}^{-1}}$$
(3)

# Ensemble com Seleção Otimizada

Este artigo propõe uma abordagem Ensemble com Seleção Otimizada (ESO), em que os modelos e a estratégia de combinação são escolhidos por uma meta-heurística. As meta-heurísticas consideradas são AG, PSO e SA. A abordagem ESO foi desenvolvida com base no trabalho de Soares et al. (2013). Cumpre ressaltar que o trabalho mencionado tem como foco o desenvolvimento automático de ensembles de RNAs para problemas de regressão. No entanto, este trabalho tem o objetivo de desenvolver automaticamente ensembles com um conjunto de modelos híbridos (estatísticos e ML) para problemas de previsão de séries temporais.

Embora a tarefa de previsão de séries temporais seja uma tarefa de regressão, séries temporais possuem outras características que devem ser consideradas. Por exemplo, na divisão dos dados para treino e teste dos modelos, a série deve ser dividida de maneira que não perca a sua sequência de observação, uma vez que esta sequência é uma informação importante. Além disso, a tarefa de previsão lida com o número de observações futuras que serão previstas.

A Fig. 1 mostra o pseudocódigo da abordagem ESO para séries temporais. Já a Figura Fig. 2 apresenta a arquitetura detalhada da abordagem ESO. O pseudocódigo e a arquitetura mostram as etapas utilizadas na abordagem ESO para realizar a seleção dos modelos e a estratégia de combinação para uma série temporal.

Entradas: Série Temporal (X) e Modelos de previsão ( $M_1,...,M_n$ ).

- Divida a série temporal X em dados de treino X<sub>tr</sub>, validação X<sub>v</sub> e teste X<sub>te</sub>.
   Treine todos os modelos (M<sub>1</sub>,..,M<sub>n</sub>) com os dados X<sub>tr</sub>.
   Realize a previsão com os modelos (M<sub>1</sub>,..,M<sub>n</sub>) nos dados X<sub>v</sub>.
   Envie as previsões para uma meta—heurística (otimizador) que retorna o melhor subconjunto de modelos  $(M_i,...,M_k)$  e a melhor estratégia de combinação

- 5. Retreine os modelos  $(M_i,...,M_k)$  com os dados  $X_{tr}$ e  $X_{\nu}$ .
  6. Realize a previsão dos modelos  $(M_i,...,M_k)$  nos dados  $X_{te}$ .
  7. Realize a agregação das previsões com a estratégia de combinação E para
- gerar previsão final F. 8. Avalie a previsão do ensemble F com as observações dos dados  $X_{te}$ .

Figura 1: Pseudocódigo da abordagem ensemble com seleção otimizada de modelos e combinação.

# 3.4 Meta-heurísticas para Ensembles com Seleção Otimizada

Nesta subseção são apresentadas as implementações das meta-heurísticas AG, PSO e SA que são utilizadas na abordagem ESO. Elas foram inspiradas pelo trabalho de Soares et al. (2013), que utiliza AG e SA para problemas de regressão. Antes de apresentá-las, é necessário definir:

Codificação da solução. Esta é uma cadeia binária, onde os *n* primeiros bits representam os *n* modelos de previsão que podem ser escolhidos para o ensemble e os dois últimos bits representam a estratégia de combinação. Os modelos escolhidos são representados na cadeia binária com o número "1", enquanto aqueles que não foram selecionados são representados por "0". Os dois *bits* da combinação podem possuir os seguintes valores: "00" para média, "01" para mediana, "10" para média aparada e "11" para média ponderada.

Função de avaliação. Ela avalia o desempenho do ensemble nos dados de validação utilizando a métrica SMAPE. A fórmula da função de avaliação, também chamada de função de aptidão, é definida por 1/SMAPE. Portanto, quanto menor for o erro de uma solução com um ensemble, melhor esta será avaliada.

## 3.4.1 Algoritmo Genético

AG é um otimizador inspirado nos princípios da genética e na seleção natural (Haupt and Haupt, 2004). A proposta do AG é simular a evolução de uma população, em que os indivíduos são as possíveis soluções de um problema para buscar convergir os indivíduos para a melhor solução. As principais operações do AG são: seleção, crossover e mutação (Coley, 1999).

A seleção é a operação que escolhe os indivíduos da população para fazer o cruzamento (crossover) e produzir novos descendentes (Haupt and Haupt, 2004, Coley, 1999). O método da roleta é amplamente utilizado na seleção. Ele dá maior probabilidade de seleção aos indivíduos com maior aptidão, apesar de também possibilitar a seleção de indivíduos com menor aptidão.

O crossover é uma forma de troca de informações entre indivíduos da população, podendo ser visto como um cruzamento, no qual os pais selecionados trocam informações genéticas para gerar descendentes (Haupt and Haupt, 2004, Coley, 1999). Este trabalho utiliza o crossover uni-

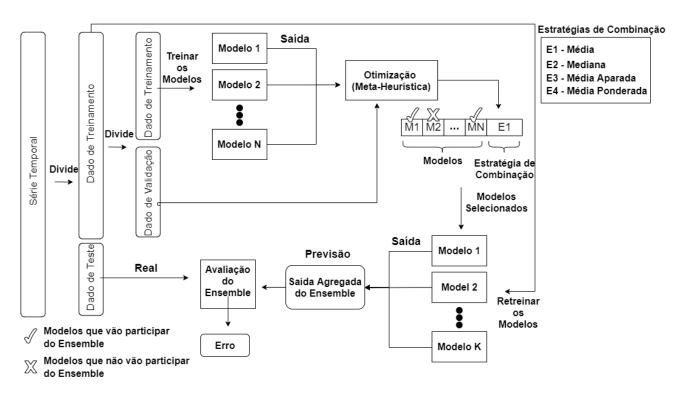


Figura 2: Arquitetura da abordagem ensemble com seleção otimizada de modelos e combinação.

forme que emprega uma máscara binária para trocar os genes (*bits*) de dois pais para gerar dois descendentes.

A mutação é uma operação que introduz novas informações na população original. Isto permite aumentar a busca da solução nos espaços de soluções. Em AG com soluções binárias, a mutação pode ser feita pela mudança aleatória de uma porcentagem de *bits* presentes na população (Haupt and Haupt, 2004).

A Fig. 3 apresenta o pseudocódigo do método AG (Coley, 1999, Soares et al., 2013) para a seleção de modelos em *ensembles*. Ele é empregado na etapa 4 do pseudocódigo da Fig. 1. Inicialmente o método AG gera uma população que é avaliada pela função de avaliação. Em seguida, a população passa pelas etapas de seleção, *crossover*, mutação, avaliação e elitismo (etapa que remove indivíduos mais fracos) por *G* gerações. O método retorna o melhor indivíduo da população que contém os modelos e a combinação selecionada.

# 3.4.2 Simulated Annealing

SA é uma meta-heurística baseada em um processo metalúrgico para endurecer metais e vidros, conhecido como têmpera. Neste caso, o material é aquecido para ficar em altas temperaturas e depois resfriado lentamente para que este possa atingir um estado cristalino de baixa energia (Norvig and Russell, 2013).

A ideia do método SA é primeiramente encontrar uma solução aleatória no espaço de busca e, em seguida, percorrer soluções vizinhas para alcançar o máximo global. Enquanto o algoritmo está percorrendo soluções vizinhas, a temperatura inicial é reduzida. Um dos pontos-chaves deste algoritmo é a temperatura, visto que quando está

alta, o mesmo tem um comportamento parecido com o algoritmo de caminhada aleatória, aceitando com mais facilidade soluções piores, mas sempre aceitando soluções melhores. Quando a temperatura diminui, o método SA tem comportamento semelhante ao método da subida de encostas, aceitando com menos facilidade soluções piores (Michalewicz and Fogel, 2004, Norvig and Russell, 2013).

A Eq. (4) determina como o algoritmo SA aceita uma solução com resultado pior que a atual, em que Rand() é uma função que devolve um número aleatório entre 0 e 1, Exp() é uma função de transformação exponencial,  $s_V$  é o resultado da função de avaliação da solução vizinha

Entradas: Previsão dos Modelos  $(M_1,...,M_n)$ , Número Máximo de Geração (G), Tamanho da População (P) e Taxa de Mutação (M).

1. Defina geração atual  $(G_a)$  igual a 1.

- Gere uma população inicial de cromossomos de tamanho P com cadeias binárias aleatórias e com o tamanho proposto pela codificação da solução.
- 3. Decodifique cada cromossomo, encontrando os modelos e a estratégia de combinação escolhida; em seguida, avalie o *ensemble* formado utilizando a função de avaliação proposta.
- Selecione pares de cromossomos utilizando o método da roleta para serem os pais. O número de pares devem ser a metade do tamanho da população.
- Realize o crossover com os pares selecionados com o método crossover uniforme para gerar dois novos descendentes para cada par.
- 6. Realize mutação na cadeia binária dos novos descendentes com a taxa M.
   7. Avalie os novos descendentes na função de aptidão proposta, como no passo 3.
- 7. Avancos invos descendentes com a antiga população para formar a nova população.
- 9. Elimine metade dos cromossomos que tiveram os valores da função de aptidão mais baixos.
- 10. Incremente  $G_a$  em 1; Retorne ao passo 4 se  $G_a$  é menor ou igual a G.

**Figura 3:** Pseudocódigo da meta-heurística AG para a seleção de modelos em *ensembles*.

que está sendo avaliada,  $s_a$  é o resultado da avaliação da solução atual, e  $T_a$  é a temperatura atual.

$$Aceita = \begin{cases} 1 & \text{Se Rand()} < \text{Exp(} s_{\nu} - s_{a}/T_{a})) \\ 0 & \text{Senão} \end{cases}$$
 (4)

A Fig. 4 apresenta o pseudocódigo do método SA utilizado para a seleção de modelos em ensembles. O método gera inicialmente uma cadeia binária aleatória (solução atual) e a avalia. Depois, ele define a temperatura atual como a temperatura máxima. Em seguida, o método passa pelas etapas de geração e avaliação da solução vizinha, troca da solução atual pela solução vizinha (se necessário, utilizando a Eq. (4)) e decremento do valor da temperatura atual. Estas etapas são repetidas até que a temperatura mínima seja alcançada. No final, o método retorna a última solução atual contendo os modelos e a combinação selecionada.

Entradas: Previsão dos Modelos  $(M_1,...,M_n)$ , Temperatura Máxima  $(T_{mi\alpha})$ , Temperatura Mínima  $(T_{min})$ , Taxa de Decaimento (D), Número de Tentativas (NT) e Distância de Hamming (H).

- Gere uma cadeia binária aleatória do tamanho proposto pela codificação da solução.

  2. Decodifique a cadeia binária, encontrando os modelos e a estratégia de
- combinação escolhida; e avalie o ensemble formado utilizando a função de
- 3. Armazene a cadeia binária e o valor da função de avaliação como solução
- atual. Defina a temperatura atual  $T_a$  como  $T_{max}$ . 4. Selecione uma nova solução vizinha que é uma cadeia binária aleatória utilizando o método Hamming, no qual apenas H bits são alterados da Solução
- 5. Decodifique e avalie a solução vizinha utilizando a função de avaliação. 6. Compare o valor da função de avaliação da solução atual com a solução vizinha. Se a solução vizinha for melhor, então esta se torna nova solução atual, senão a troca pode ocorrer utilizando a Eq. (4).
- 7. Retorne ao passo 4 até atingir a quantidade máxima do NT para  $T_a$ .
- 8. Decremente  $T_a$  utilizando a taxa de decaimento D e retorne ao passo 4 até atingir  $T_{min}$

Figura 4: Pseudocódigo da meta-heurística SA para a seleção de modelos em ensembles.

#### 3.4.3 Otimização por Enxame de Partículas (PSO)

O algoritmo de otimização PSO foi proposto por Kennedy and Eberhart (1995). Ele é baseado no comportamento de bandos de pássaros, cardumes de peixes, enxames e na computação evolutiva. A ideia principal do algoritmo é que um indivíduo do enxame possa aprender com suas próprias experiências e com as experiências de todo enxame. O algoritmo coloca cada solução como uma partícula que tem sua próxima posição calculada com uma velocidade que guia o movimento da partícula. Esta velocidade é definida por uma equação que contém uma parte cognitiva da partícula (que utiliza a melhor posição da partícula) e uma parte social (que utiliza a melhor posição do enxame).

A proposta original do PSO tinha como foco resolver problemas não lineares contínuos. Porém, os autores publicaram em 1997 uma versão do PSO para problemas discretos (Kennedy and Eberhart, 1997). Esta proposta altera o conceito da velocidade, que passa ser a probabilidade de mudança da partícula. Outras modificações no PSO foram propostas, como a adição de um parâmetro de inércia para

velocidade (Shi and Eberhart, 1998). Este artigo utiliza os seguintes princípios matemáticos do PSO discreto:

$$v_{id} = v_{id}w + c_1r_1(p_{id} - x_{id}) + c_2r_2(g_{id} - x_{id})$$
 (5)

$$x_{id} = \begin{cases} 1 & \text{Se Rand}() < S(v_{id}) \\ 0 & \text{Senão} \end{cases}$$
 (6)

onde  $v_{id}$  é a velocidade da partícula na dimensão d, w é um parâmetro de inércia, c1 é uma constante cognitiva, c2 é uma constante social,  $r_1$  e  $r_2$  são variáveis aleatórias no intervalo de 0 e 1,  $x_{id}$  é o estado da partícula na dimensão d que pode assumir o valor de o ou 1, S() é uma função de transformação sigmoide usada para manter os valores da velocidade no intervalo de 0 e 1, Rand() é uma função que gera um número aleatório entre 0 e 1,  $p_{id}$  é a melhor posição da partícula na dimensão d, e  $g_{id}$  é a melhor posição entre todas as partículas na dimensão d.

A Fig. 5 exibe o pseudocódigo da otimização PSO utilizado neste trabalho para a escolha de modelos em ensembles. O algoritmo PSO gera inicialmente cadeias binárias aleatórias (partículas) para serem as soluções candidatas. Em seguida, estas partículas passam pelas etapas de avaliação, armazenamento (etapa na qual as melhores posições de cada partícula e do enxame são guardadas), cálculo das novas velocidades e das novas posições por NP iterações. No final, o algoritmo retorna a melhor partícula encontrada que contém os modelos e a combinação selecionada.

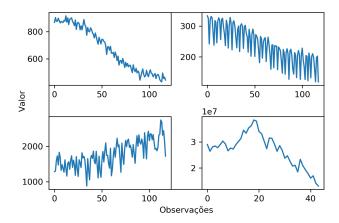
Entradas: Previsão dos Modelos  $(M_1,...,M_n)$ , Número de Partículas (NP), Número de Iterações (NI), Constante Cognitiva  $c_1$ , Constante Social  $c_2$ , Inércia da Velocidade w.

- 1. Defina iteração atual  $(I_a)$  igual a 1.
- 2. Gere as NP partículas com a dimensão do tamanho proposto para representar a codificação da solução. Cada partícula é uma cadeia binária gerada
- 3. Define uma variável p para cada partícula e uma variável g para o enxame.
- 4. Decodifique as soluções das partículas para avaliar cada previsão formada pelo Ensemble na função de avaliação.
- 5. Verifique se o resultado da função de avaliação foi a melhor posição de cada partícula até atual iteração. Caso seja o melhor resultado para alguma partícula, esta armazena este como sua melhor posição na variável p.
- 6. Verifique em todas partículas se existe um resultado melhor da função de avaliação do enxame até iteração atual. Se sim, é substituído a melhor posição
- 7. Calcule as novas velocidades das partículas com a Eq. (5)
- 8. Calcule as novas posições das partículas com a Eq. ( $\acute{6}$ ). 9. Incremente  $I_a$  em 1; e retorne ao passo 4 se  $I_a$  for menor ou igual a NI.

**Figura 5:** Pseudocódigo da meta-heurística PSO para a seleção de modelos em ensembles.

# 4 Resultados experimentais

Nesta seção, os ensembles híbridos propostos são avaliados em dois conjuntos de dados, sendo eles um conjunto de séries temporais da competição Computational Intelligence in Forecasting (CIF2016) e um conjunto de dados de demanda de produtos de uma distribuidora de cosméticos. As abordagens foram desenvolvidas utilizando linguagem



**Figura 6:** Exemplos de séries temporais da base CIF2016.

de programação *Python* e testadas em um computador equipado com um Intel® CoreTM Processador i3-8100 3.60GHz de 4 núcleos e 8 GB de RAM.

# 4.1 Descrição dos Dados

A Tabela 1 apresenta a descrição das bases de dados utilizadas neste trabalho. CIF2016 é um conjunto de dados público com 72 séries temporais mensais utilizadas em uma competição de modelos da área da inteligência computacional. De acordo com Štěpnička and Burda (2017), a base contém 24 séries reais e 48 séries geradas artificialmente, sendo que as séries reais têm de 23 a 69 observações e as séries artificiais têm 108 observações. Além disso, as séries artificiais são formadas por diferentes combinações de tendências, sazonalidades e erros aleatórios.

Neste trabalho, foi obtido um Coeficiente de Variação (CV), que é uma métrica relativa da variabilidade que divide o desvio padrão pela média para medir a variação das observações nestas séries (Peck et al., 2007). O valor CV médio encontrado para as séries deste conjunto foi 25%. A Fig. 6 apresenta quatro exemplos de séries desta base. Observa-se que este conjunto de séries apresenta dados com diferentes escalas.

O segundo conjunto de dados é de uma empresa do ramo de distribuição de cosméticos e, por motivos de confidencialidade não será revelado o nome da empresa e não serão exibidos gráficos com os dados. Os dados contêm séries temporais mensais de demanda de produtos do período de maio de 2016 a fevereiro de 2020.

Para este conjunto, foram utilizadas séries temporais de produtos que possuem observações em todos os períodos e com no mínimo 30 observações. Isto porque este trabalho não foca em séries temporais com observações zeradas (intermitentes), e além disso, os modelos de revisão utilizados não possuem bom desempenho em séries com poucas observações. Assim, foram selecionadas e analisadas 113 séries temporais, sendo que o CV médio encontrado foi de 42%, mostrando que as séries têm alta variabilidade de valores nas suas observações.

**Tabela 1:** Descrição dos conjunto de dados com séries temporais.

		-		
Conjunto	N° de séries	Média de observações nas séries	Dados públi- cos?	Média do CV
CIF2016	72	97,3	Sim	25%
Distribuidora	113	45,74	Não	42%

# 4.2 Metodologia de Avaliação

A métrica utilizada para avaliar os modelos e *ensembles* é o SMAPE. Ela é menos susceptível a problemas de escala e foi utilizada em uma competição com os dados CIF2016. O erro SMAPE é calculado da seguinte forma:

SMAPE = 
$$\frac{1}{h} \sum_{t=1}^{h} \frac{|f_t - a_t|}{(|a_t| + |f_t|)/2},$$
 (7)

onde  $f_t$  é o valor previsto no tempo t,  $a_t$  é a observação real no tempo t e h é o horizonte de tempo de previsão que define o número de observações a serem previstas (Štěpnička and Burda, 2017).

# 4.3 Implementação dos Modelos de Previsão

Este trabalho adota 9 tipos diferentes de modelos de previsão para compor as abordagens *ensembles*, conforme descrito na Seção 3.1. Alguns modelos receberam parâmetros diferentes para produzir novos modelos. Assim, um total de 20 modelos foram utilizados para compor os *ensembles*. Este número é frequentemente utilizado em trabalhos de *ensembles* do estado-da-arte (Soares et al., 2013). Bibliotecas e parâmetros dos modelos de previsão são descritos a seguir:

**Método Ingênuo.** Este modelo foi implementado utilizando o método SES com o parâmetro  $\alpha$  de suavização igual a 1, visto que o modelo SES com este parâmetro tem comportamento de um método ingênuo (Hyndman and Athanasopoulos, 2018). Foi utilizada a biblioteca *Statsmodels*, que é uma biblioteca com foco em análise de séries temporais (Seabold and Perktold, 2010).

SES. Este modelo tem seu parâmetro  $\alpha$  escolhido pelo otimizador da biblioteca *Statsmodels*, que maximiza a probabilidade de log e também utiliza a busca em grade para encontrar valores iniciais ótimos para  $\alpha$  (Seabold and Perktold, 2010).

**HOLT.** Este método também foi implementado utilizando a biblioteca *Statsmodels*, em que os parâmetros  $\alpha$ ,  $\beta$  e  $\phi$  são escolhidos por uma otimização que maximiza a probabilidade de log (Seabold and Perktold, 2010).

AR. Para implementação do método e escolha de p, foi utilizada a configuração padrão da biblioteca *Statsmodels*, já que ela seleciona o atraso ideal por meio de uma análise da série temporal (Seabold and Perktold, 2010).

**ARIMA.** Ele foi implementado utilizando a biblioteca Pmdarima, que seleciona automaticamente a ordem ideal dos parâmetros p, q e d (Smith et al., 2017). Ela também escolhe os componentes sazonais P, Q e D. É imple-

mentada uma função que busca os parâmetros p, P, q e Q em grade para minimizar um critério de informação de Akaike, sendo que para d é utilizado um teste de estacionariedade e para D um teste de sazonalidade.

**SVR.** Este modelo foi desenvolvido utilizando a biblioteca *Sklearn* (Pedregosa et al., 2011). Rbf foi utilizado como *kernel*, e os parâmetros C, Epsilon e Gama são selecionados por uma busca em grade que minimiza o erro de raiz quadrático médio do modelo. Foram feitas 6 implementações diferentes do modelo SVR (isto é,  $SVR_{a1}$ ,  $SVR_{a2}$ ,  $SVR_{a3}$ ,  $SVR_{a4}$ ,  $SVR_{a5}$  e  $SVR_{a6}$ ), onde cada modelo tem como entrada um atraso de tempo diferente (ou seja, 1, 2, 3, 4, 5 e 6, respectivamente).

RNA. Para este modelo, foram implementadas quatro versões diferentes. A primeira implementação, chamada nnetar<sup>1</sup>, utiliza o modelo RNA da linguagem R. As outras três versões foram implementadas utilizando o modelo RNA da competição M4 (Makridakis et al., 2020), sendo que o código deste modelo está disponível em linguagem Python<sup>2</sup>. Cada versão utiliza um atraso de tempo diferente (o atraso define quantas observações passadas a RNA utiliza para prever), isto é RNA $_{a1}$ , RNA $_{a2}$  e RNA $_{a3}$ , onde a numeração define o atraso de tempo utilizado (ou seja, 1, 2 e 3, respectivamente). Cumpre ressaltar que as quatro versões foram construídas em Python por meio da biblioteca Sklearn (Pedregosa et al., 2011). Além disso, os protocolos de teste dos modelos RNA seguem os mesmos protocolos dos modelos originais do modelo nnetar e da RNA do M4. Isto é, o resultado do teste do modelo nnetar é fornecido pela média em 20 execuções do modelo, e os resultados dos modelos baseados na competição M4 são fornecidos pela média em 30 execuções.

RNR. Este modelo também tem 4 versões diferentes. A primeira versão, chamada nnetar RNR, é também baseada na função nnetar. As outras três versões são baseadas na implementação do modelo RNR da competição M4 (Makridakis et al., 2020), sendo elas denominadas RNR<sub>a1</sub>, RNR<sub>a2</sub> e RNR<sub>a3</sub>, possuindo atrasos de tempo diferentes (ou seja, 1, 2 e 3, respectivamente). As 4 versões foram implementadas utilizando a biblioteca Keras (Chollet, 2015). Os resultados de teste são exibidos em 10 execuções de cada modelo para garantir estabilidade nas previsões.

ELM. Este modelo foi implementado utilizando a biblioteca *Python* ELM (Almeida, 2015). Para a seleção do parâmetro *C* de regularização, da função de ativação e do atraso ideal, foi realizada uma busca em grade. Para minimizar a estocasticidade do modelo, a previsão do modelo é a mediana das previsões em 20 execuções do algoritmo. Foi utilizado um pré-processamento de dados da competição M4 para tratar a tendência e a sazonalidades das séries.

# 4.4 Preparação dos Dados

Para avaliar os modelos e *ensembles* propostos, os dados das séries temporais foram divididos em treino, validação e teste. Na base CIF2016, para os dados de teste, foram utilizados horizontes de tempo de 6 e 12 observações (Štěp-

nička and Burda, 2017); já os dados de treino e validação ficaram com 80% e 20% do restante dos dados, respectivamente. Para as séries da distribuidora de cosméticos, adotou-se a seguinte divisão: 60% para treino, 20% para validação e 20% para teste.

#### 4.5 Resultados dos Modelos de Previsão

Na Tabela 2 e na Tabela 3 são apresentados os resultados dos erros médios e Desvio Padrão (DP) dos 20 modelos de previsão nos dados de teste das séries, além do tempo de execução dos mesmos (em segundos). O modelo com menor erro (médio e DP) e o modelo com menor tempo de execução (médio e DP) são destacados em verde nas duas tabelas.

Os resultados dos modelos na base CIF2016 demostram que modelos simples obtiveram menores erros, sendo que o modelo Holt apresentou o menor erro. Isso pode ser explicado porque na base CIF2016, as séries apresentam padrões de tendência e, por isso, o método Holt consegue captar bem estes padrões.

Entretanto, na base da distribuidora, o método Holt obteve o pior desempenho entre os modelos. Nesta base, o método SES apresentou o menor error. Isto porque as séries temporais desta base possuem maior variação de valores nas suas observações. Ademais, pode-se notar que os erros dos modelos são maiores que os erros apresentados na base CIF2016. Isso mostra portanto, que a base da distribuidora possui séries mais difíceis de prever.

Por fim, pode ser observado que os métodos da área de ML (por exemplo, nnetar, RNAs, RNRs, etc.) apresentam maior tempo de execução na Tabela 2 e na Tabela 3 do que os métodos estatísticos clássicos (por exemplo, SES, método ingênuo, Holt e AR).

# 4.6 Resultados dos *Ensembles* com Estratégias de Combinação

Os resultados dos *ensembles* com diferentes estratégias de combinação são exibidos na Tabela 4 e na Tabela 5. A estratégia com menor erro (médio e DP) e o modelo com menor tempo (médio e DP) de execução são destacados em verde nas duas tabelas.

Os resultados mostram que a abordagem  $EC_{m\'ediaPonderada}$  obteve o menor erro médio nas duas bases de dados. Isto confirma os resultados do trabalho de Bandeira et al. (2020), onde esta abordagem exibe melhor desempenho do que a abordagem ensemble com média simples. Porém, cumpre destacar que a abordagem  $EC_{m\'ediaPonderada}$  apresentou maior tempo de execução, uma vez que ela retreina os modelos nos dados de validação após o cálculo dos pesos.

# 4.7 Resultados do Ensemble com Seleção Otimizada por Meta-heurística AG

Foram feitos 12 experimentos do *ensemble* com seleção otimizada por AG ( $ESO_{AG}$ ) para cada base de dados, sendo que para cada experimento foram utilizados 20 modelos candidatos e 4 estratégias de combinação, já descritos an-

<sup>&</sup>lt;sup>1</sup>https://github.com/robjhyndman/forecast/blob/master/R/nnetar.
R

<sup>2</sup>https://github.com/Mcompetitions/M4-methods

**Tabela 2:** Resultados dos modelos de previsão individuais na base CIF2016.

Modelo	SMAPE (Média)	SMAPE (DP)	Tempo* (Média)	Tempo* (DP)
SES	0,150	0,134	0,009	0,001
Ingênuo	0,159	0,133	0,008	0,001
Holt	0,148	0,147	0,016	0,004
AR	0,164	0,185	0,006	0,001
ARIMA	0,185	0,238	0,768	0,517
SVR <sub>a1</sub>	0,217	0,159	0,654	0,051
SVR <sub>a2</sub>	0,192	0,147	0,640	0,040
SVR <sub>a3</sub>	0,184	0,149	0,642	0,041
SVR <sub>a4</sub>	0,176	0,141	0,642	0,041
SVR <sub>a5</sub>	0,168	0,139	0,650	0,054
SVR <sub>a6</sub>	0,160	0,130	0,644	0,043
nnetar	0,171	0,164	1,870	0,493
nnetar RNR	0,171	0,197	85,271	38,087
RNA <sub>a1</sub>	0,287	0,388	0,782	0,019
RNA <sub>a2</sub>	0,280	0,390	0,772	0,111
RNA <sub>a3</sub>	0,280	0,390	0,791	0,109
RNR <sub>a1</sub>	0,241	0,341	59,798	18,901
RNR <sub>a2</sub>	0,230	0,322	64,553	20,652
RNR <sub>a3</sub>	0,215	0,290	68,739	22,372
ELM	0,240	0,327	5,085	1,368

<sup>\*</sup>O tempo está representado em segundos.

teriormente. Cada experimento foi construído alterando os seguintes parâmetros do AG: tamanho da população (P), taxa de mutação (M) e número de gerações (G); sendo que eles podem assumir os valores:

- Tamanho da população (P): 5, 10 e 15;
- Taxa de mutação (*M*): 0,10 e 0,15;
- Número de gerações (*G*): 75 e 100.

A Tabela 6 e a Tabela 7 mostram os resultados em 20 execuções da abordagem  $ESO_{AG}$  nas duas bases de dados, sendo que o experimento com menor erro médio está em verde em ambas tabelas. Os tempos de execução apresentam o tempo gasto nas otimizações e, portanto, não incluem o tempo de treino dos modelos. Além disso, é apresentado o número de vezes que a Função de Avaliação (NFA) foi usada durante a otimização AG.

Na Tabela 6 é possível observar que experimentos com *M* igual a 15% obtiveram melhores resultados; por outro lado, na Tabela 7, valores menores de *P* e *M* conseguiram menores erros. Além disso, nota-se que o aumento do tempo de execução possui como principal fator o aumento das operações próprias do algoritmo AG, uma vez que nas tabelas, o aumento do número de avaliações de função não reflete em um aumento expressivo do tempo.

# 4.8 Resultados do Ensemble com Seleção Otimizada por Meta-heurística SA

Foram realizados 12 experimentos do *ensemble* com seleção otimizada por SA (*ESO*<sub>SA</sub>) para cada base de dados,

**Tabela 3:** Resultados dos modelos de previsão individuais na base da distribuidora de cosméticos.

Modelo	SMAPE (Média)	SMAPE (DP)	Tempo* (Média)	Tempo* (DP)
SES	0,436	0,160	0,007	0,001
Ingênuo	0,485	0,186	0,006	0,001
Holt	0,531	0,273	0,013	0,003
AR	0,480	0,195	0,004	0,001
ARIMA	0,485	0,324	0,387	0,224
SVR <sub>a1</sub>	0,485	0,210	0,589	0,036
SVR <sub>a2</sub>	0,476	0,201	0,581	0,021
SVR <sub>a3</sub>	0,470	0,209	0,578	0,012
SVR <sub>a4</sub>	0,454	0,174	0,579	0,020
SVR <sub>a5</sub>	0,446	0,157	0,580	0,019
SVR <sub>a6</sub>	0,450	0,168	0,581	0,020
nnetar	0,493	0,178	1,999	0,517
nnetar RNR	0,467	0,182	30,305	7,636
RNA <sub>a1</sub>	0,488	0,380	0,793	0,012
RNA <sub>a2</sub>	0,481	0,359	0,756	0,147
RNA <sub>a3</sub>	0,484	0,366	0,813	0,013
RNR <sub>a1</sub>	0,484	0,367	27,480	0,894
RNR <sub>a2</sub>	0,481	0,364	28,906	0,992
RNR <sub>a3</sub>	0,481	0,361	30,215	1,065
ELM	0,491	0,360	2,924	0,163

<sup>\*</sup>O tempo está representado em segundos.

**Tabela 4:** Resultados dos *ensembles* com estratégias de combinação na base CIF2016.

Abordagem ensemble	SMAPE (Média)	SMAPE (DP)	Tempo* (Média)	Tempo* (DP)
EC <sub>média</sub>	0,160	0,128	292,341	97,225
EC <sub>mediana</sub>	0,157	0,132	292,340	97,225
EC <sub>médiaAparada</sub>	0,160	0,128	292,330	97,225
EC <sub>médiaPonderada</sub>	0,153	0,136	525,400	173,041

<sup>\*</sup>O tempo está representado em segundos.

sendo que 20 modelos e 4 combinações foram utilizados. Os parâmetros de temperatura máxima e de temperatura miníma foram fixados respectivamente em 1000 e 10<sup>-20</sup> (Soares et al., 2013). Os outros parâmetros podem assumir os seguintes valores:

- Taxa de decaimento (*D*): 0,85; 0,90 e 0,95;
- Número de tentativas (NT): 1 e 2;
- Distância de Hamming (*H*): 2 e 3.

Os resultados da abordagem  $ESO_{SA}$  em 20 execuções são apresentados na Tabela 8 e na Tabela 9, sendo que os experimentos com menor erro médio são destacados em verde. Pode notar-se nas duas tabelas que, quanto maior é a taxa de decaimento (D), maiores são os erros SMAPE. Além disso, na Tabela 9 é possível observar que os experimentos que utilizaram a distância de Hamming (H) com valor 3 tiveram menores erros.

Outra observação é que os tempos de execução da abordagem  $ESO_{SA}$  são menores que os tempos apresentados na

**Tabela 5:** Resultados dos *ensembles* com estratégias de combinação na base da distribuidora de cosméticos.

Abordagem ensemble	SMAPE (Média)	SMAPE (DP)	Tempo* (Média)	Tempo* (DP)
EC <sub>média</sub>	0,422	0,153	128,098	8,968
EC <sub>mediana</sub>	0,426	0,167	128,098	8,968
EC <sub>médiaAparada</sub>	0,420	0,154	128,098	8,968
EC <sub>médiaPonderada</sub>	0,419	0,152	227,609	14,595

<sup>\*</sup>O tempo está representado em segundos.

abordagem  $ESO_{AG}$ . Isto é motivado pelo fato da abordagem  $ESO_{SA}$  possuir menos operações e ser mais simples do que a abordagem  $ESO_{AG}$ . Porém, é observado que para as duas bases, o melhor experimento da abordagem  $ESO_{AG}$  possui erro SMAPE médio ligeiramente menor que o melhor experimento da abordagem  $ESO_{SA}$ .

# 4.9 Resultados do Ensemble com Seleção Otimizada por Meta-heurística PSO

Da mesma forma, foram realizados 12 experimentos do *ensemble* com seleção otimizada por PSO ( $ESO_{PSO}$ ) para cada base de dados, sendo que 20 modelos e 4 combinações foram utilizados. O parâmetro w foi fixado no valor de 0,9, já os outros parâmetros podem assumir os seguintes valores:

- Número de partículas (NP): 15, 20 e 25;
- Número máximo de iterações (*MI*): 30 e 50;
- Constantes cognitivas c<sub>1</sub> e c<sub>2</sub>: 1 e 2.

O resultado dos experimentos nas duas bases podem ser vistos na Tabela 10 e na Tabela 11, onde o melhor experimento de cada base de dados está destacado. Os resultados da Tabela 10 mostram que experimentos com constantes  $c_1$  e  $c_2$  igual a 1 têm seu desempenho melhorado quando o número de partículas (NP) aumenta. Por outro lado, na Tabela 11 pode-se notar que experimentos com  $c_1$  e  $c_2$  igual a 1 apresentam resultados melhores que  $c_1$  e  $c_2$  igual a 2.

# 4.10 Comparação dos Resultados dos Ensembles com Seleção Otimizada

A Tabela 12 e a Tabela 13 exibem os resultados dos melhores experimentos de cada otimização nas duas bases de dados, onde o experimento com menor erro (médio e DP) e o experimento com menor tempo de execução (médio e DP) são destacados em verde. Os tempos de execução apresentados correspondem ao tempo total das abordagens, que incluem os tempos de otimização, além dos tempos de treino, teste e validação dos modelos.

Nos dados CIF2016 (Tabela 12), pode ser observado que a abordagem  $ESO_{PSO}$  alcançou o menor erro médio e menor desvio padrão. Já nos dados da empresa de distribuição (Tabela 13), a abordagem  $ESO_{AG}$  obteve o melhor resultado na métrica de erro SMAPE. Assim, a meta-heurística SA, utilizada por  $ESO_{SA}$ , apresentou erros médios SMAPE ligeiramente superiores que AG e PSO nas duas bases. Entretanto, nota-se que a meta-heurística SA possui menor tempo médio de execução.

Além disso, pode ser observado que o tempo total das abordagens *ensemble* quando comparadas aos modelos de previsão individuais é bastante elevado. Isso porque, estas abordagens precisam treinar 20 modelos e retreinar os modelos selecionados.

# 4.11 Discussão Final dos Resultados

Na base CIF2016, as abordagens ESO apresentaram os menores erros médios quando comparadas aos *ensembles* com estratégias de combinação e modelos individuais. Neste caso,  $ESO_{PSO}$  apresentou o menor erro médio. Nos *ensembles* com estratégia de combinação, o *ensemble* com a média ponderada ( $EC_{médiaPonderada}$ ) obteve o melhor resultado. Porém, esse *ensemble* teve pior resultado que dois modelos individuais (isto é, Holt e SES), sendo que o modelo Holt apresentou o menor erro médio entre os modelos individuais.

Os resultados apontaram que as abordagens ESO reduzem o erro médio e o desvio padrão médio na base CIF2016 e, portanto, o desempenho é comparável ao desempenho dos melhores modelos individuais. Além disso, estes resultados são promissores quando comparados aos resultados dos modelos da competição CIF2016, em que os erros médios estão entre 0,105 e 0,288 (Štěpnička and Burda, 2017).

Nos dados da distribuidora, os menores erros médios SMAPE pertencem às abordagens *ensembles* com estratégia de combinação, onde  $EC_{médiaPonderada}$  teve o melhor desempenho. Por outro lado, as abordagens ESO apresentaram resultados inferiores ao método individual SES. Isso foi devido à alta variação de valores das observações e da pouca quantidade de amostras das séries deste conjunto.

As abordagens ESO e os ensembles com estratégia de combinação possuem maior tempo de execução que os modelos individuais nas duas bases de dados. Isso deve-se ao uso dos modelos RNR nos ensembles que possuem alto custo computacional. Nota-se também que as abordagens ESO possuem tempo de execução menor que a abordagem  $EC_{médiaPonderada}$ . Isso porque  $EC_{médiaPonderada}$  treina e retreina todos os modelos, enquanto as abordagens ESO retreinam apenas os modelos selecionados para o ensemble. Portanto, observa-se que as abordagens ESO além de diminuírem o erro, também auxiliam na redução do custo computacional.

# 5 Conclusão

Este trabalho compara a exatidão de várias abordagens de ensembles híbridos e modelos individuais para a previsão de séries temporais. Primeiramente, foram propostos quatro ensembles com diferentes estratégias de combinação (isto é, média, mediana, média aparada e média ponderada). Além disso, foi proposta uma abordagem ensemble que realiza a seleção otimizada dos modelos (ESO) e da estratégia de combinação, sendo que para esta seleção foram propostas as meta-heurísticas AG, PSO e SA.

Os ensembles foram avaliados em duas bases de dados utilizando a métrica SMAPE e o tempo computacional. Os resultados apresentados demostraram que as abordagens ESO possuem melhor desempenho na base CIF2016. Nos dados da distribuidora de cosméticos, os ensembles com es-

 $\textbf{Tabela 6:} \ \text{Resultados do} \ \textit{ensemble} \ \text{com seleção} \ \text{otimizada} \ \text{por meta-heurística} \ \text{AG} \ (\textit{ESO}_{\textit{AG}}) \ \text{na base} \ \text{CIF2016}.$ 

Experimento	1	2	3	4	5	6	7	8	9	10	11	12
P	5	5	5	5	10	10	10	10	15	15	15	15
M	0,10	0,15	0,10	0,15	0,10	0,15	0,10	0,15	0,10	0,15	0,10	0,15
G	75	75	100	100	75	75	100	100	75	75	100	100
SMAPE (Média)	0,1424	0,1420	0,1427	0,1421	0,1419	0,1420	0,1427	0,1421	0,1424	0,1419	0,1432	0,1415
SMAPE (DP)	0,1221	0,1221	0,1231	0,1228	0,1208	0,1215	0,1224	0,1205	0,1206	0,1213	0,1209	0,1208
Tempo* (Méd.)	1,425	1,482	1,779	1,852	11,296	11,814	17,837	19,244	12,298	13,086	19,033	20,698
Tempo* (DP)	0,307	0,312	0,354	0,395	2,564	2,873	4,133	4,707	2,765	3,132	4,306	4,917
NAF <sup>†</sup>	375	375	500	500	750	750	1000	1000	1125	1125	1500	1500

<sup>\*</sup>O tempo está representado em segundos; †Número de vezes que a Função de Avaliação foi utilizada.

**Tabela 7:** Resultados do ensemble com seleção otimizada por meta-heurística AG ( $ESO_{AG}$ ) na base da distribuidora.

Experimento	1	2	3	4	5	6	7	8	9	10	11	12
P	5	5	5	5	10	10	10	10	15	15	15	15
M	0,10	0,15	0,10	0,15	0,10	0,15	0,10	0,15	0,10	0,15	0,10	0,15
G	75	75	100	100	75	75	100	100	75	75	100	100
SMAPE (Média)	0,4429	0,4435	0,4422	0,4434	0,4440	0,4440	0,4450	0,4424	0,4451	0,4436	0,4446	0,4437
SMAPE (DP)	0,2315	0,2346	0,2303	0,2346	0,2376	0,2365	0,2375	0,2359	0,2381	0,2367	0,2364	0,2381
Tempo* (Méd.)	0,956	0,977	1,217	1,241	7,234	7,372	11,563	11,885	8,036	8,213	12,461	12,842
Tempo* (DP)	0,169	0,169	0,224	0,235	1,865	1,745	3,193	2,991	2,083	2,028	3,472	3,331
NAF <sup>†</sup>	375	375	500	500	750	750	1000	1000	1125	1125	1500	1500

<sup>\*</sup>O tempo está representado em segundos; †Número de vezes que a Função de Avaliação foi utilizada.

 $\textbf{Tabela 8:} \ \text{Resultados do} \ \textit{ensemble} \ \text{com seleç\~ao} \ \text{otimizada} \ \text{por meta-heur\'istica} \ \text{SA} \ (\textit{ESO}_{SA}) \ \text{na} \ \text{base} \ \text{CIF2016}.$ 

Experimento	1	2	3	4	5	6	7	8	9	10	11	12
D	0,85	0,85	0,85	0,85	0,90	0,90	0,90	0,90	0,95	0,95	0,95	0,95
NT	1	2	1	2	1	2	1	2	1	2	1	2
Н	2	2	3	3	2	2	3	3	2	2	3	3
SMAPE (Média)	0,1424	0,1440	0,1426	0,1425	0,1451	0,1444	0,1426	0,1431	0,1460	0,1455	0,1435	0,1435
SMAPE (DP)	0,1213	0,1256	0,1203	0,1209	0,1298	0,1267	0,1226	0,1213	0,1316	0,1296	0,1228	0,1226
Tempo* (Méd.)	0,777	1,411	0,791	1,454	1,097	2,045	1,155	2,173	2,127	4,042	2,230	4,313
Tempo* (DP)	0,151	0,276	0,155	0,283	0,204	0,405	0,229	0,449	0,438	0,830	0,444	0,893
NAF <sup>†</sup>	326	652	326	652	503	1006	503	1006	1033	2066	1033	2066

<sup>\*</sup>O tempo está representado em segundos; †Número de vezes que a Função de Avaliação foi utilizada.

**Tabela 9:** Resultados do *ensemble* com seleção otimizada por meta-heurística SA (*ESO<sub>SA</sub>*) na base da distribuidora.

Experimento	1	2	3	4	5	6	7	8	9	10	11	12
D	0,85	0,85	0,85	0,85	0,90	0,90	0,90	0,90	0,95	0,95	0,95	0,95
NT	1	2	1	2	1	2	1	2	1	2	1	2
Н	2	2	3	3	2	2	3	3	2	2	3	3
SMAPE (Média)	0,4441	0,4470	0,4432	0,4430	0,4447	0,4467	0,4432	0,4446	0,4453	0,4468	0,4444	0,4447
SMAPE (DP)	0,2307	0,2364	0,2296	0,2302	0,2323	0,2357	0,2287	0,2328	0,2327	0,2357	0,2335	0,2350
Tempo* (Méd.)	0,582	1,071	0,610	1,158	0,855	1,604	0,907	1,738	1,637	3,166	1,768	3,470
Tempo* (DP)	0,070	0,140	0,065	0,132	0,113	0,223	0,097	0,189	0,226	0,439	0,186	0,401
NAF <sup>†</sup>	326	652	326	652	503	1006	503	1006	1033	2066	1033	2066

<sup>\*</sup>O tempo está representado em segundos; †Número de vezes que a Função de Avaliação foi utilizada.

Experimento 3 NP 2.0 20 20 20 15 15 15 15 25 25 25 25 MI 30 50 30 50 30 50 30 50 30 50 30 50  $c_1$  e  $c_2$ 1 1 2 2 1 1 2 2 1 1 2 2 SMAPE (Média) 0,1421 0,1425 0,1419 0,1421 0,1413 0,1410 0,1419 0,1425 0,1411 0,1409 0,1416 0,1427 SMAPE (DP) 0,1221 0,1181 0,1206 0,1168 0,1202 0,1167 0,1185 0,1180 0,1175 0,1183 0,1199 0,1188 Tempo\* (Méd.) 2,835 1,404 2,166 1,326 2,026 1,832 1,711 2,637 2,252 2,097 3,238 3,515 Tempo\* (DP) 0,264 0,405 0,246 0,363 0,346 0,528 0,314 0,467 0,419 0,652 0,388 0,578 NAF<sup>†</sup> 450 750 450 750 600 1000 600 1000 750 1250 750 1250

**Tabela 10:** Resultados do *ensemble* com seleção otimizada por meta-heurística PSO (ESO<sub>PSO</sub>) na base CIF2016.

**Tabela 11:** Resultados do *ensemble* com seleção otimizada por meta-heurística PSO (*ESO*<sub>PSO</sub>) na base da distribuidora.

Experimento	1	2	3	4	5	6	7	8	9	10	11	12
NP	15	15	15	15	20	20	20	20	25	25	25	25
MI	30	50	30	50	30	50	30	50	30	50	30	50
c <sub>1</sub> e c <sub>2</sub>	1	1	2	2	1	1	2	2	1	1	2	2
SMAPE (Média)	0,4428	0,4434	0,4451	0,4455	0,4451	0,4447	0,4458	0,4449	0,4441	0,4449	0,4446	0,4453
SMAPE (DP)	0,2355	0,2367	0,2367	0,2364	0,2371	0,2374	0,2365	0,2371	0,2361	0,2363	0,2359	0,2361
Tempo* (Méd.)	1,047	1,602	0,980	1,527	1,371	2,065	1,285	2,001	1,713	2,672	1,577	2,486
Tempo* (DP)	0,067	0,111	0,086	0,140	0,092	0,140	0,116	0,191	0,110	0,182	0,133	0,224
NAF <sup>†</sup>	450	750	450	750	600	1000	600	1000	750	1250	750	1250

<sup>\*</sup>O tempo está representado em segundos; †Número de vezes que a Função de Avaliação foi utilizada.

**Tabela 12:** Resultados dos melhores experimentos dos *ensembles* otimizados na base CIF2016.

Abordagem ensemble	SMAPE (Média)	SMAPE (DP)	Tempo* (Média)	Tempo* (DP)
ESO <sub>AG</sub>	0,1415	0,1208	393,769	152,135
ESO <sub>SA</sub>	0,1424	0,1213	355,168	140,961
ESO <sub>PSO</sub>	0,1409	0,1175	362,855	143,096

<sup>\*</sup>O tempo está representado em segundos.

**Tabela 13:** Resultados dos melhores experimentos dos *ensembles* otimizados na base da distribuidora.

Abordagem ensemble	SMAPE (Média)	SMAPE (DP)	Tempo* (Média)	Tempo* (DP)
$ESO_{AG}$	0,4422	0,2303	171,320	30,643
$ESO_{SA}$	0,4430	0,2302	169,445	32,804
ESO <sub>PSO</sub>	0,4428	0,2355	176,287	29,132

<sup>\*</sup>O tempo está representado em segundos.

tratégias de combinação obtiveram melhor resultado. Em geral, as abordagens ensembles tiveram melhores resultados que a maioria dos modelos individuais, demostrando o potencial dessas abordagens. Entretanto, conforme apresentado neste artigo, essas abordagens possuem maior tempo computacional. Desse modo, modelos individuais com menor tempo computacional devem ser escolhidos para reduzir o tempo computacional total do ensemble.

Para trabalhos futuros, propõe-se o uso de outras otimizações para a abordagem ESO, como otimização da borboleta e NSGA-II. Além disso, deve-se considerar a escolha de outras estratégias de combinação, como o empilhamento que agrega as saídas dos modelos individuais para obter a

saída final do ensemble.

# Referências

Adhikari, R., Verma, G. and Khandelwal, I. (2015). A model ranking based selective ensemble approach for time series forecasting, *Procedia Computer Science* 48: 14 – 21. https://doi.org/10.1016/j.procs.2015.04.104.

Almeida, A. (2015). Python Extreme Learning Machine (ELM). Disponível em https://elm.readthedocs.io/en/latest/.

Arora, S. and Singh, S. (2019). Butterfly optimization algorithm: a novel approach for global optimization, *Soft Computing* **23**: 715–734. https://doi.org/10.1007/s00500-018-3102-4.

Awad, M. and Khanna, R. (2015). Support vector regression, *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*, Apress, Berkeley, CA, pp. 67–80. https://doi.org/10.1007/978-1-4302-5990-9\_4.

Bandeira, S. G., Alcalá, S. G. S., Vita, R. O. and Barbosa, T. M. G. A. (2020). Comparison of selection and combination strategies for demand forecasting methods, *Production* **30**: e20200009. https://doi.org/10.1590/0103-6513.20200009.

Bolón-Canedo, V. and Alonso-Betanzos, A. (2019). Ensembles for feature selection: A review and future trends, Information Fusion 52: 1 – 12. https://doi.org/10.1016/j.inffus.2018.11.008.

<sup>\*</sup>O tempo está representado em segundos; †Número de vezes que a Função de Avaliação foi utilizada.

- Chen, C. and Liu, H. (2020). Medium-term wind power forecasting based on multi-resolution multi-learner ensemble and adaptive model selection, *Energy Conversion and Management* **206**: 112492. https://doi.org/10.1016/j.enconman.2020.112492.
- Chollet, F. (2015). Keras. Disponível em https://keras.io.
- Coley, D. A. (1999). An Introduction to Genetic Algorithms for Scientists and Engineers, World Scientific, River Edge, NJ, USA.
- de Melo Junior, G., Alcalá, S., Furriel, G. and Vieira, S. (2020). Missing data analysis using machine learning methods to predict the performance of technical students, *Revista Brasileira de Computação Aplicada* 12(2): 134–143. https://doi.org/10.5335/rbca.v12i2.10565.
- Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep Learning*, MIT Press, Cambridge, MA, USA.
- Haupt, R. L. and Haupt, S. (2004). Practical Genetic Algorithms, Wiley InterScience electronic collection, John Wiley & Sons Inc., USA.
- Huang, G. B., Zhu, Q. Y. and Siew, C. K. (2006). Extreme learning machine: Theory and applications, *Neurocomputing* **70**(1): 489 501. https://doi.org/10.1016/j.neucom.2005.12.126.
- Hyndman, R. and Athanasopoulos, G. (2018). Forecasting: principles and practice, 2 edn, OTexts, Melbourne, Australia.
- Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization, *Proceedings of ICNN'95 International Conference on Neural Networks*, Vol. 4, IEEE, Perth, WA, Australia, pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968.
- Kennedy, J. and Eberhart, R. C. (1997). A discrete binary version of the particle swarm algorithm, 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Vol. 5, IEEE, Orlando, FL, USA, pp. 4104–4108. https://doi.org/10.1109/ICSMC.1997.637339.
- Kumar, A. and Jain, M. (2020). Ensemble Learning for AI Developers: Learn Bagging, Stacking, and Boosting Methods with Use Cases, 1 edn, Apress, Berkeley, CA, USA.
- Li, F., Zhang, L., Chen, B., Gao, D., Cheng, Y., Zhang, X., Yang, Y., Gao, K. and Huang, Z. (2020). An optimal stacking ensemble for remaining useful life estimation of systems under multi-operating conditions, *IEEE Access* 8: 31854–31868. https://doi.org/10.1109/ACCESS.2020.2973500.
- Machado, E., de Assis, C. and Pereira, A. (2020). Modelagem, implementação e avaliação de estratégias de negociação baseadas em algoritmos de aprendizado de máquina para o mercado financeiro, *Revista Brasileira de Computação Aplicada* 12(1): 16–31. https://doi.org/10.5335/rbca.v12i1.9106.

- Makridakis, S., Spiliotis, E. and Assimakopoulos, V. (2018). The M4 competition: Results, findings, conclusion and way forward, *International Journal of Forecasting* **34**(4): 802 808. https://doi.org/10.1016/j.ijforecast.2018.06.001.
- Makridakis, S., Spiliotis, E. and Assimakopoulos, V. (2020). The M4 competition: 100,000 time series and 61 forecasting methods, *International Journal of Forecasting* **36**(1): 54 74. https://doi.org/10.1016/j.ijforecast. 2019.04.014.
- Michalewicz, Z. and Fogel, D. (2004). How to Solve It: Modern Heuristics, Springer Berlin Heidelberg, Berlin, Heidelberg.
- Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J. and Talagala, T. S. (2020). Fforma: Feature-based forecast model averaging, *International Journal of Forecasting* **36**(1): 86 92. https://doi.org/10.1016/j.ijforecast. 2019.02.011.
- Montgomery, D. C., Jennings, C. L. and Kulahci, M. (2008). *Introduction to Time Series Analysis and Fore-casting*, Wiley-Interscience, Hoboken, N.J., USA.
- Moon, J., Jung, S., Rew, J., Rho, S. and Hwang, E. (2020). Combination of short-term load forecasting models based on a stacking ensemble approach, *Energy and Buildings* **216**: 109921. https://doi.org/10.1016/j.enbuild.
- Norvig, P. and Russell, S. (2013). *Inteligência Artificial*, Elsevier Editora, Rio de Janeiro, RJ.
- Peck, R., Olsen, C. and Devore, J. (2007). *Introduction to Statistics and Data Analysis*, Cengage Learning, Boston, MA, USA.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* 12: 2825–2830. https://dl.acm.org/doi/10.5555/1953048.2078195.
- Pulido, M., Melin, P. and Castillo, O. (2014). Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the mexican stock exchange, *Information Sciences* **280**: 188 204. https://doi.org/10.1016/j.ins.2014.05.006.
- Sayad, S. (2020). Support Vector Machine: Regression (SVR). Disponível em: https://www.saedsayad.com/support\_vector\_machine\_reg.htm.
- Seabold, S. and Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with python, *in* Stéfan van der Walt and Jarrod Millman (eds), *9th Python in Science Conference*, SciPy 2010, Austin, Texas. http://dx.doi.org/10.25080/Majora-92bf1922-011.
- Shi, Y. and Eberhart, R. (1998). A modified particle swarm optimizer, 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360), IEEE,

- Anchorage, AK, USA, pp. 69-73. https://doi.org/10.1109/ICEC.1998.699146.
- Smith, T. G. et al. (2017). Pmdarima: ARIMA estimators for Python. Disponível em http://www.alkaline-ml.com/ pmdarima.
- Soares, S., Antunes, C. H. and Araújo, R. (2013). Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development, *Neurocomputing* **121**: 498 511. https://doi.org/10.1016/j.neucom.2013.05.024.
- Thiele, C. and Adami, A. (2016). Previsão de séries temporais financeiras: modelo baseado em redes neurais artificiais, *Revista Brasileira de Computação Aplicada* 8(2): 113—128. https://doi.org/10.5335/rbca.v8i2.5638.
- Štěpnička, M. and Burda, M. (2017). On the results and observations of the time series forecasting competition cif 2016, 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, Naples, IT, pp. 1–6. https://doi.org/10.1109/FUZZ-IEEE.2017.8015455.
- Wang, D. and Alhamdoosh, M. (2013). Evolutionary extreme learning machine ensembles with size control, *Neurocomputing* **102**: 98 110. https://doi.org/10.1016/j.neucom.2011.12.046.
- Wichard, J. D. (2016). An adaptive forecasting strategy with hybrid ensemble models, 2016 International Joint Conference on Neural Networks (IJCNN), IEEE, Vancouver, BC, CA, pp. 1495–1498. https://doi.org/10.1109/IJCNN.2016.7727375.
- Zhang, G., Eddy Patuwo, B. and Y. Hu, M. (1998). Forecasting with artificial neural networks: The state of the art, *International Journal of Forecasting* **14**(1): 35 62. https://doi.org/10.1016/S0169-2070(97)00044-7.