
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO CIÊNCIA DA COMPUTAÇÃO

GABRIELA MOTA DE LACERDA PADILHA SCHETTINI

# Um método baseado em correção de erros para previsão de séries temporais em ambiente online e na presença de concept drift.

Recife

2020

GABRIELA MOTA DE LACERDA PADILHA SCHETTINI

# Um método baseado em correção de erros para previsão de séries temporais em ambiente online e na presença de concept drift.

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

**Área de Concentração**: Inteligência computacional

**Orientador (a)**: Paulo Salgado Gomes de Mattos Neto

**Coorientador (a)**: Leandro Lei Minku

Recife

2020

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S327s Schettini, Gabriela Mota de Lacerda Padilha
Um método baseado em correção de erros para previsão de séries temporais em ambiente online e na presença de *concept drift* / Gabriela Mota de Lacerda Padilha Schettini. – 2020.
57 f.: il., fig., tab.

Orientador: Paulo Salgado Gomes de Mattos Neto.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2020.
Inclui referências e apêndices.

1. Inteligência computacional. 2. Séries temporais. I. Mattos Neto, Paulo Salgado Gomes de (orientador). II. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2021 – 34

**Gabriela Mota de Lacerda Padilha Schettini**

**“Um método baseado em correção de erros para previsão de séries temporais em ambientes online e na presença de concept drift”**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 30/11/2020.

**BANCA EXAMINADORA**

_____

Prof. Dr. Adriano Lorena Inacio de Oliveira
Centro de Informática/ UFPE

_____

Prof. Dr. João Fausto Lorenzato de Oliveira
Escola Politécnica de Pernambuco / UPE

_____

Prof. Dr. Paulo Salgado Gomes de Mattos Neto
Centro de Informática / UFPE
(**Orientador**)

*Dedico este trabalho a toda minha família que perante as dificuldades foi meu porto seguro.*

## RESUMO

Há um tempo que o mundo digitalizado não é novidade, mas cada vez mais adaptar-se aos novos dados recebidos é uma questão de sobrevivência de certos serviços. Diante da alta frequência de recebimento de novas informações e da rápida mudança que estamos sujeitos são poucas as estratégias existentes para adaptar-se. De olho neste cenário de rápida tomada de decisão e da precisão nos resultados que este trabalho é proposto. Fazendo um comparativo com a área de inteligência computacional, soluções que exigem essa adaptabilidade constante são plausíveis de ficarem desatualizados e resultarem em baixo desempenho ao estarem subtreinados para novos comportamentos. Sendo necessário, com o passar do tempo, manter os modelos em uso atualizados para resolver uma determinada tarefa em relação aos dados mais recentes, evitando - desta forma - o *underfitting* que é caracterizado pelo subtreinamento. Na tentativa de mitigar esse problema é comum utilizar abordagens baseadas em um sistema de combinação de preditores que realizam retreinamento dos modelos, ou a utilização de modelos com aprendizado online. Este trabalho visa comparar a estratégia de ajuste de parâmetros com correção dos resíduos de modelos monolíticos em 3 séries temporais financeiras com mudança de conceito. Propondo-se a investigar as previsões e resíduos de forma a estender o leque de estratégias que resolvem de maneira eficaz o tipo de problema sugerido. A contribuição principal deste trabalho é introduzir o conceito de correção da previsão através da série residual, que até o presente momento nunca foi visto para séries que apresentam de mudança de conceito e no ambiente de aprendizado online.



**Palavras-chaves**: Previsão de séries temporais. Aprendizado online. Série Residual. Fluxo de dados. Ambiente não-estacionário. Mudança de conceito.

## ABSTRACT

The digital world is not new to the society, but yet some services struggle to survive while not adapting to the new ways of doing things. Given the high information frequency and fast the changes that required there are only a few strategies that try to cope with such changes. With this scenario in mind we review some of the problems on the existing strategies. When comparing the real world with artificial intelligence, solutions that require these kinds of changes are plausible to be under-fitted to new behaviors (concept drift). In this panorama, with time, it is required to have the models updated to the latest data so the new behaviors are predictable in real-time and avoiding under-fitting. As an attempt to avoid this last problem it is common to use multiple predictors systems approaches that get retrained or that use online learners, to keep training at execution time. Anticipating the future data is key to use the time left to take action. This dissertation aims to compare the performance of models enhanced by grid search and monolithic models with error correction within 3 financial time series containing concept drift. Besides that, the forecast of the residuals are used to study a potential increase in the prediction's performance. The main contribution of the present work is to introduce the concept of prediction correction using the past error's prediction, that has not been seen until the present moment for time series in an online and non-stationary environment. In this way, this work proposes to investigate the prediction and residuals to formalize one more strategy to approach the problems lately disclosed.



**Keywords**: Time Series Forecasting. Online learning. Residual Series. Data stream. Non-stationary environments. Concept Drift.

## LISTA DE FIGURAS

## LISTA DE TABELAS

## SUMÁRIO

# 1 INTRODUÇÃO

A informatização das atividades cotidianas está fazendo o mundo digital passar por uma crise pela quantidade de dados gerados e que não são automaticamente processados. Embora os dados gerados possam ser valiosos, eles podem não estar sendo aproveitados da melhor maneira. Tendo em vista a complexidade das tarefas que exigem o monitoramento de dados em tempo real e dos desafios do recebimento e da análise de grandes volumes de dados, muitos trabalhos têm sido desenvolvidos com o objetivo de explorar essas informações recebidas e tomar decisões precisas em tempo hábil. Através da recepção de informações obtidas em tempo real, de forma contínua e sem um fim previsto (fluxo de dados) é arriscado assumir a estacionariedade [1] dos dados, principalmente quando o ambiente não é monitorado. E tratá-los como tal pode desdobrar-se em previsões com baixa precisão devido à possíveis mudanças na distribuição dos dados (GAMA et al., 2014).

Tarefas de aprendizado em ambientes não estacionários são também referenciados por aprendizado em ambientes dinâmicos, evolutivos, incertos, ou simplesmente, e mais conhecido: aprendizado com mudança de conceito, em inglês *concept drift* (SCHLIMMER; GRANGER, 1986a; SCHLIMMER; GRANGER, 1986b; WIDMER; KUBAT, 1996). Tais problemas requerem abordagens efetivas na identificação e/ou adaptação aos dados que apresentam mudança de conceito. A adaptação dos algoritmos em relação ao aprendizado sob a presença de *concept drift* pode ser feita de duas formas. A primeira é a abordagem ativa que propõe detectar mudança de conceito e a partir disso se adaptar aos dados (KRAWCZYK et al., 2017; OLIVEIRA et al., 2017). E a segunda é a abordagem passiva que visa atualizar continuamente o modelo, independente da identificação de uma mudança na distribuição dos dados (BROCKWELL; DAVIS, 2002; BOX; JENKINS, 2008).

## 1.1 MOTIVAÇÃO

Ao assumir - de forma implícita ou não - que o processo gerador dos dados é estacionário subentende-se que os dados são coletados de uma fonte de distribuição de probabilidade fixa e desconhecida. Sendo assim, existe o risco do modelo treinado não se adaptar apropriadamente

[1] A estacionariedade é considerada uma sequênciade informações relacionadas com tempo, mas que se desenvolve de forma aleatória ao redor de uma média constante, refletindo uma ideia de um equilíbrio estável. Mais detalhes na seção 2.3.1.

aos dados de treinamento pelo fato dos dados serem não-estacionários. Como consequência, o modelo treinado pode apresentar perda de precisão e tornar-se obsoletos, podendo tomar decisões completamente equivocadas por falta de adaptabilidade para com os dados (*underfitting*) (Lu et al., 2019; GAMA et al., 2014; DITZLER et al., 2015; KRAWCZYK et al., 2017; BROCKWELL; DAVIS, 2002).

A preocupação com os parâmetros do modelo em relação aos dados utilizados sempre foi uma questão muito estudada, incluindo o conflito entre o viés e a variância, *overfitting* e *underfitting*. Além do problema acima, a complexidade do aprendizado na presença de mudanças de conceito pode ser considerada maior, justificada pela mudança de comportamento dos dados e diante das diversas formas com que essas mudanças podem ocorrer.

Outro fator motivador relevante são estudos acerca do resíduo das previsões. Através da previsão da série de erros foi possível criar modelos híbridos que se beneficiam dos erros de previsão para gerar uma previsão final mais acurada (ZHANG, 2003; FIRMINO; FERREIRA, 2015; FIRMINO; NETO; FERREIRA, 2014; NETO et al., 2017; SILVA et al., 2018). Em contrapartida, tais estudos foram apenas testados em ambientes offline, mas são argumento forte de que essa mesma estratégia possa vir a ser relevante também em ambientes online.

## 1.2 OBJETIVOS E PERGUNTAS DE PESQUISA

O estudo desenvolvido ao logo deste trabalho buscou investigar e modelar o comportamento da série residual [2] em ambientes online baseando-se nos seguintes questionamentos:

1. É possível modelar a série de resíduos proveniente da previsão de um método online?
2. O uso da previsão dos resíduos por modelos online podem ser usados para melhorar as estimativas das séries temporais em ambientes online?
3. A correção do erro em ambientes online pode ser considerada uma opção para reduzir os impactos de uma previsão sem ajuste de parâmetros?

[2] Série de erros ou residual é obtida pela diferença entra a série original e a previsão feita pelo modelo. Mais detalhes na Seção 2.1.2

## 1.3 CONTRIBUIÇÕES

Este trabalho mostra como a modelagem da série residual pode ser usada para aumentar a acurácia da previsão de séries temporais em ambientes online.

A metodologia tradicional de ajuste de parâmetros pode se mostrar ineficiente diante dos desafios que a mudança de conceito pode oferecer.

## 1.4 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação foi dividida em 5 capítulos:

1. **Introdução:** apresenta o contexto de séries temporais e o ambiente em que esta dissertação está inserida, bem como a motivação e os objetivos da realização deste trabalho.
2. **Fundamentação Teórica:** Conceitos e Técnicas: descreve do ponto de vista da autora os conceitos e técnicas fundamentais para o embasamento teórico desta dissertação.
3. **Método Proposto:** explica desde as ideias e inspirações até a ilustração a arquitetura do método proposto. Bem como a escolha dos modelos e dos parâmetros.
4. **Protocolo Experimental:** abrange as séries temporais utilizadas destacando suas estatísticas e comportamentos, métricas e o processo de execução dos experimentos.
5. **Resultados Experimentais:** apresenta os experimentos realizados, e os respectivos resultados são apresentados.
6. **Conclusão e Trabalhos Futuros:** resume o trabalho, as conclusões e limitações com base nos experimentos. Por fim, cita propostas de trabalhos futuros.

# 2 FUNDAMENTAÇÃO TEÓRICA: CONCEITOS E TÉCNICAS

## 2.1 SÉRIES TEMPORAIS

Uma série temporal pode ser entendida como uma sequência de valores $x_t$, observados e registrados, geralmente, de forma equidistante no tempo $t$ (BROCKWELL; DAVIS, 2002), dada por:

$$X_t = \{x_t \in \mathbb{R} \quad | \quad t = T_0 + n.\Delta t, \quad n = 1, 2, ..., N\}, \tag{2.1}$$

em que $t$ é o índice temporal, $T_0$ é o instante inicial das observações, $\Delta t$ é o intervalo de tempo de coleta entre as observações, e $N$ é a quantidade de observações obtidas.

Uma série temporal discreta, muitas vezes provêm de uma série temporal contínua amostrada em intervalos de tempo iguais. A depender da frequência com a qual se deseja fazer as análises, é possível agrupar as amostras por minuto, hora, dia, mês, semestre, ano, e assim sucessivamente (MORETTIN, 2004).

### 2.1.1 Objetivos da Análise de Séries Temporais

Ao possuir uma série temporal é possível que haja interesse em:

a) investigar qual o mecanismo gerador do fenômeno temporal;
b) descrever o comportamento da dados a fim de identificar padrões;
c) realizar a previsão de valores futuros da série.

Neste trabalho o enfoque foi em torno da análise e previsão de séries temporais, embora a descrição do comportamento temporal tenha sido utilizada para embasar algumas decisões e compreender certos comportamentos.

### 2.1.2 Previsão de Séries Temporais

A previsão de valores futuros $(x_{t+1})$ de uma série temporal é feita a partir de uma ou mais observações passadas. O objetivo por trás da previsão é identificar padrões temporais nos

dados, de forma a construir um modelo capaz de estimar, com precisão, os valores futuros da série.

Diante de um instante de tempo qualquer $t$, a diferença entre o valor observado $x_t$ e o valor predito $\hat{x}_t$ é chamada de erro residual e pode ser descrita da seguinte forma:

$$\epsilon_t = x_t - \hat{x}_t. \tag{2.2}$$

Portanto, o erro coletado dentro de um período de tempo é chamado de série residual: $\epsilon_1, \epsilon_2, \epsilon_3, ...\epsilon_n$ (COWPERTWAIT, 2009a). Esta nova série gerada pode ser construída iterativamente à medida que o valor observado é conhecido.

### 2.1.3 Ruído Branco

Por definição, uma série temporal é considerada um ruído branco quando os valores desta série são independentes e identicamente distribuídos e com média zero. Isto implica que a variância desses elementos são, todos, iguais a zero e quando $(i \not\equiv j)$ a $Correlação(\epsilon_i, \epsilon_j) = 0$. Na prática, é comum considerar um nível de significância de $5\%$ para assumir que uma série é um ruído branco. O ruído branco também pode ser chamado de série puramente randômica (COWPERTWAIT, 2009b).

## 2.2 REDES NEURAIS PARA PREVISÃO DE SÉRIES TEMPORAIS

### 2.2.1 Máquina de vetor de suporte para regressão

Máquinas de vetores de suporte foram inicialmente desenvolvidas para problemas de classificação (VAPNIK; LERNER, 1963), onde exemplos de treinamento nas fronteiras de decisão são utilizados para criar um hiperplano ideal de separação equidistante das classes distintas. A máquina de vetor para regressão (SVR) (VAPNIK, 1995), por sua vez, visa em encontrar uma função $f(x)$ com liberdade de erro menor ou igual a $\varepsilon$. A SVR é descrita pela função 2.3,

$$f(x) = b + (w \cdot \phi(x)), \tag{2.3}$$

onde $b$ é o bias, $w$ é o vetor de pesos e $\phi(x)$ é uma função de kernel. Diante de $j$ exemplos de treinamento, o algoritmo de treinamento tem como objetivo encontrar o vetor de pesos $z$

e um limiar $b$ que minimize a função 2.4:

$$\frac{1}{2} \parallel z \parallel^2 + C \sum_{i=1}^{j} (\xi_i + \xi_I^*), \tag{2.4}$$

respeitando as restrições apresentadas nas equações 2.5, 2.6 e 2.7.

$$y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \tag{2.5}$$

$$\langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \tag{2.6}$$

$$\xi_i, \xi_I^* \geq 0. \tag{2.7}$$

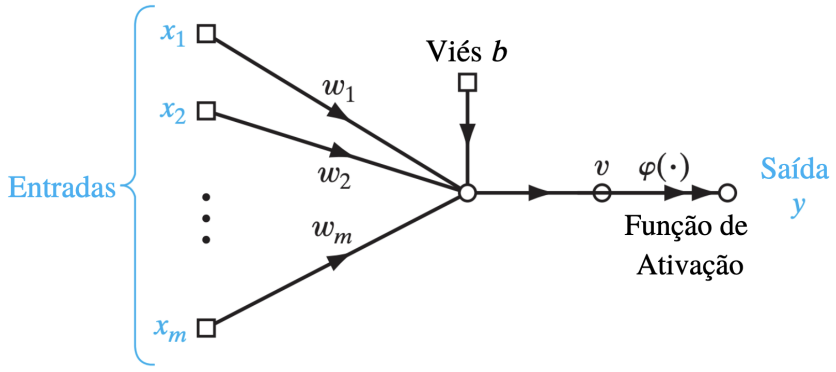
De forma que $y_i$ é o valor desejado, $C > 0$ é o fator de regularização, sendo assim responsável por regular a complexidade da função e a quantidade de vezes que serão tolerados desvios maiores do que $\varepsilon$. $xi_i$ e $\xi_I^*$ representam as variáveis de folga, criadas para o caso em que, no processo de treinamento, não seja encontrada uma função que limite os erros em $\varepsilon$, essas duas variáveis meçam os custos dos erros de previsão (SMOLA A. J.; SCHÖLKOPF, 2004).

### 2.2.2 Perceptron

Inspirado pelo funcionamento dos neurônios e pelo desenvolvimento da lógica simbólica, o perceptron foi proposto por Frank Rosenblatt como forma de ilustrar algumas das propriedades fundamentais de sistema inteligente, sem a necessidade de imergir afundo nas condições biológicas do organismo (ROSENBLATT, 1958).

O perceptron é considerado a forma mais simples de uma rede neural artificial (RNA), utilizado inicialmente para classificação de padrões linearmente separáveis e de forma supervisionada. A estrutura do perceptron é de apenas um neurônio adaptável através de um viés e de pesos sinápticos (HAYKIN, 2009). Devido ao uso de um único neurônio, o perceptron é classificado com uma rede neural uma única camada. A Figura 1 é uma representação do funcionamento desta RNA

Figura 1 – Diagrama de fluxo do funcionamento do perceptron.




**Fonte:** Imagem adaptada para o português a partir de Haykin (2009)

A função 2.8 descreve o neurônio do perceptron utilizado por Rosemblatt,

$$v = \sum_{i=1}^{m}(w_I \cdot x_i) + b, \tag{2.8}$$

onde os exemplos de entrada são representados por $x_1, x_2 \ldots x_m$, os pesos sinápticos são representados por $w_1, w_2 \ldots w_m$ e b é o viés. O perceptron teve sua notoriedade pelo uso da funcão de ativação não-linear denominada função degrau após o cálculo da função acima (BISHOP, 2006). A função degrau pode ser descrita é descrita equação 2.9.

$$f(x) = \begin{cases} +1, a \geq 0 \\ -1, a < 0. \end{cases} \tag{2.9}$$

A função de ativação é o mecanismo de transformação dos estímulos em uma decisão. No caso do perceptron para regressão é possível utilizar como função de ativação a função identidade, em que a linearidade da RNA é mantida, mas há a possibilidade de incluir uma função de ativação não-linear como a tangente, sigmóide logística, pseudo-linear, etc.

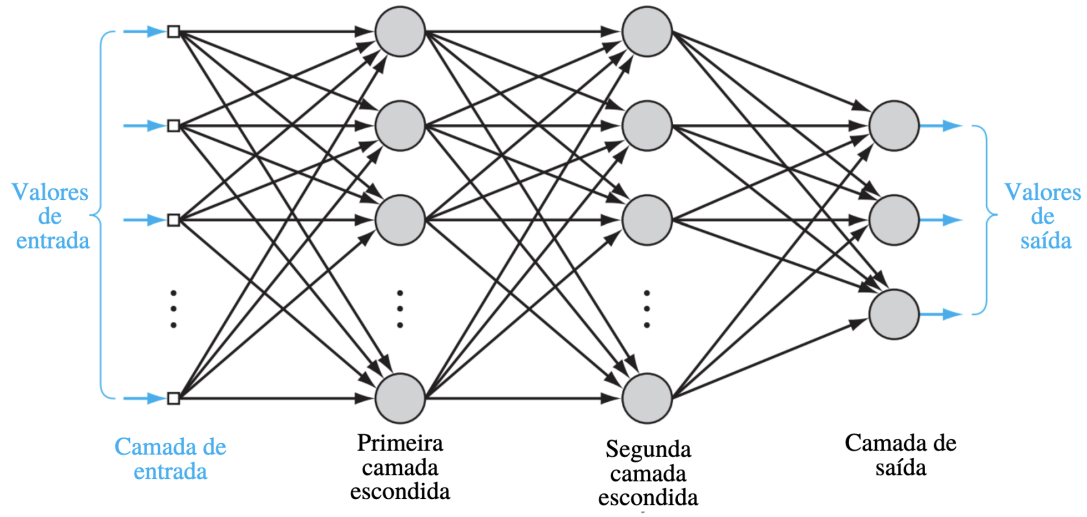
Para esse trabalho, a função pseudo-linear para a série original utilizada foi a representada pela equação 2.10.

$$f(x) = \begin{cases} 0, v < 0 \\ v, 0 \geq v \geq 1 \\ 1, v > 1. \end{cases} \tag{2.10}$$

### 2.2.3 Perceptron de Múltiplas Camadas

A limitação do perceptron em operar bem apenas em classes linearmente separáveis expandiu os horizontes para estudo de redes neurais mais complexas, como o perceptron de múltiplas camadas. A figura 2 representa um exemplo de uma estrutura desse tipo:

Figura 2 – Arquitetura em forma de grafo de um perceptron de duas camadas escondidas.




**Fonte:** Imagem adaptada para o português a partir de Haykin (2009)

O perceptron de múltiplas camadas é considerado uma rede neural artificial completamente conectada, pois toda camada é conectada com todos neurônios da camada anterior (HAYKIN, 2009). A função de ativação mais comum para a MLP para todas as camadas é a função logística, representada na equação 2.11.

$$g(x) = \frac{1}{1 + exp(-x)}, \tag{2.11}$$

em que, $x$ é a soma do valor de entrada ponderado pelo peso sináptico. Essa função tem como objetivo restringir a amplitude da saída do neurônio, comumente no intervalo $[0, 1]$ ou $[-1, 1]$.

O treinamento desse tipo de série ocorre em duas fases: a fase de propagação, onde se tem uma configuração fixa de de pesos e os valores de entrada são propagados pela rede, camada por camada, até atingir a camada de saída. E a fase de retropropagação: em que a saída esperada e a saída real tem seu erro calculado e esse erro é retropropagado - novamente camada por camada - realizando ajustes nos pesos sinápticos (HAYKIN, 2009). O cálculo do

erro pode ser realizado a partir de métricas como o erro médio absoluto (MAE) descrito na seção 4.7.1.

## 2.3 CARACTERÍSTICAS DOS AMBIENTES DE APRENDIZADO

O ambiente no qual o aprendizado é realizado indica primariamente o comportamento da série temporal, e tem influencia direta na modelagem e estratégias que devem ser utilizadas para realizar uma boa previsão.

### 2.3.1 Ambiente Estacionário

**Definição:**
Seja $X_t$ uma série temporal com $E(X_t^2) < \infty$. A função média de $|X_t|$ é:

$$\mu_{X_t} = E(X_t), \tag{2.12}$$

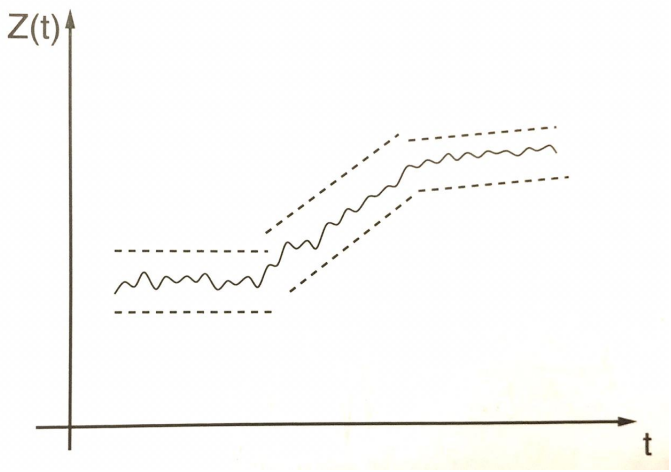
onde E é o valor esperado. Em termos básicos, a estacionariedade é considerada uma sequência de informações relacionadas com tempo, mas que se desenvolve de forma aleatória ao redor de uma média constante, refletindo uma ideia de um equilíbrio estável. Uma observação é que uma série temporal estritamente estacionária $X_t, t = 0, \pm 1, ...$ também é definida pela condição que $(X_1, ...X_n)$ e $(X_{1+h}, ...X_{n+h})$ possuam a mesma distribuição de probabilidade conjunta para todos os inteiros $h$ e $n > 0$ (BROCKWELL; DAVIS, 2002). É possível que uma série tenha um comportamento disciplinado - estacionário - por um período de tempo longo, mas o oposto também pode ocorrer.

### 2.3.2 Ambientes Não-Estacionários

A mudança na distribuição dos exemplos com o passar do tempo é característico de ambientes não-estacionários, também referidos por ambientes que possuem mudança de conceito (Lu et al., 2019). Formalmente, considere que a cada instante de tempo $t$, os exemplos gerados a partir de uma fonte que possua a mesma distribuição de probabilidade conjunta $P^t(x, y)$. Desta forma, é possível afirmar que a mudança de conceito ocorre quando, existem dois instantes de tempo distintos $t_1$ e $t_2$ em que $P^{t_1}(x, y) \neq P^{t_2}(x, y)$ (GAMA et al., 2014; COWPERTWAIT, 2009b). Estes conceitos podem mudar ou variar de diversas formas: de forma

abrupta ou gradual, temporal ou indeterminadamente, ciclicamente ou aciclicamente, etc. Um exemplo comum da não-estacionariedade pode ser vista na Figura 3, a tendência. Um tipo de comportamento simples de ser identificado é a tendência linear, onde a série flutua sobre uma reta de inclinação positiva ou negativa.

Figura 3 – Série não-estacionária quanto ao nível e inclinação.




**Fonte:** Cowpertwait (2009b)

Problemas de aprendizado decorrentes de ambientes não-estacionários são também referenciados por aprendizado em ambientes dinâmicos, evolutivos, incertos, ou simplesmente, e mais conhecido: aprendizado com mudança de conceito, em inglês *concept drift*. Tais problemas requerem abordagens efetivas na identificação e/ou adaptação aos dados que apresentam mudança de conceito. A adaptação dos algoritmos em relação ao aprendizado sob a presença de concept drift pode ser feita de duas formas. A primeira é a abordagem ativa que propõe detectar mudança de conceito e a partir disso se adaptar aos dados. E a segunda é a abordagem passiva que visa atualizar continuamente o modelo, independente da identificação de uma mudança na distribuição dos dados (BROCKWELL; DAVIS, 2002; NETO et al., 2017; BOX; JENKINS, 2008).

### 2.3.3 Disponibilidade dos dados

Ao desenhar a resolução de um problema, a forma com que os dados serão disponibilizados influenciam na determinação de qual estratégia e modelo serão utilizados. Em (KASABOV, 2003) se fala de duas situações: o aprendizado offline e o online. Essas formas de aprendizado

fazem referência a maneira com que os dados são recebidos para então, serem utilizados pelos modelos.

#### *2.3.3.1 Aprendizado Offline*

De forma abrangente, o aprendizado offline se dá pelo treinamento de um modelo sobre uma base de dados fixa e imutável. Neste caso, a partição de dados utilizada para treinamento aprendida pelo modelo de aprendizagem é limitada a uma parte dos dados disponíveis. Após a fase de aprendizado o modelo não é mais atualizado, sobrando apenas a função de consulta daquele conhecimento que foi aprendido. Devido a essa limitação dos dados de aprendizado, deve-se ter cuidado com forma com que o aprendizado é realizado diante da possibilidade de ocorrer o sobretreinamento, que acontece quando o modelo fica super ajustado ao conjunto de treino.

A exemplo de um modelo que foi treinando apropriadamente em cima da detecção de uma tendência linear numa série temporal. Tendo sido entregue ao modelo exemplos diversos e representativos deste comportamento, e sabendo que os dados são estacionários, não se têm evidências diretas que o modelo precisa ser retreinado. Até o momento em que seja detectado que os dados possuem mudança de conceito.

#### *2.3.3.2 Aprendizado Online*

Como o nome sugere, esse tipo de aprendizagem se baseia na forma com que o sistema opera - comumente em tempo real em forma de fluxo de dados. O sistema de previsão realiza adaptações no processo de otimização da função objetivo a cada nova informação disponível. Um cenário comum é a análise em tempo real de séries financeiras (KASABOV, 2003).

### 2.3.4 Abordagens de Aprendizagem

Em (KRAWCZYK et al., 2017) os métodos existentes para evitar os problemas de mudança são separados em 2 grupos: aprendizado ativo e aprendizado passivo. Os métodos de aprendizado ativo buscam identificar o exato ponto da mudança de conceito. Como consequência da detecção, há uma atualização do modelo ou um novo modelo é criado de forma condizente com o novo comportamento dos dados. Os métodos de aprendizado passivo não se preocupam

ativamente em identificar mudanças de conceito. No entanto, a abordagem possui um processo de atualização que acontece de forma contínua e diante do recebimento de novas instâncias, mas sem tomar conhecimento explícito sobre possíveis mudanças no comportamento dos dados. Os métodos de aprendizado passivo ainda podem ser subcategorizados em modelos únicos e sistema de múltiplos preditores. Os modelos únicos de abordagem passiva são atrativos por seu baixo custo computacional, quando comparados com os métodos de múltiplos preditores. Os sistemas de múltiplos preditores são um conjunto de modelos que realizam a previsão e que contam com uma regra para reunir os resultados, como a média ponderada dos resultados. Em DITZLER et al., 2015 também são citadas 3 vantanges de utilizar o sistema de múltiplos preditores: (i) método se torna mais atrativo do que utilizar um único modelo por ser preciso, ao ter a variância dos resultados reduzidos; (ii) A possibilidade de incorporar novos modelos no conjunto de preditores traz diversidade para a previsão;(iii) A flexibilidade de remover modelos de baixo rendimento, ou modelos mais antigos agregam ao conjunto um mecanismo natural de esquecimento de dados possivelmente obsoletos.

## 2.4 TRABALHOS RELACIONADOS

Nesta seção são apresentadas 3 subseções que tem como objetivo fundamentar o uso das estratégias escolhidas para compor o trabalho, bem como uma explicação de como esses métodos funcionam.

### 2.4.1 Abordagens híbridas

O modelo híbrido proposto por Zhang (2003) teve fortes motivações, e estas também são firmemente relacionadas com as motivações por trás deste presente trabalho. A primeira, é a frequente dificuldade em determinar quando uma série temporal é gerada a partir de um processo linear ou não-linear ou qual seria o método mais efetivo para lidar com uma previsão de um conjunto de teste desconhecido. As duas questões levantadas acima já demonstram a dificuldade em escolher a técnica correta para cada situação específica. O que é comumente é feito para resolver este tipo de problema é experimentar com uma determinada quantidade de modelos e selecionar o modelos cujo resultado é o mais acurado. Por outro lado, o melhor modelo dentre um conjunto não é necessariamente a melhor técnica futura, o autor cita a variação amostral, as incertezas do modelo, e a mudança estrutural como fatores para isso.

Além do mais, o autor afirma que séries temporais do mundo real raramente são puramente lineares ou não lineares, possuindo muitas vezes ambas as características. E como solução para isso ele sugere uma abordagem híbrida que una essas duas características. Principalmente pelo fato de que a literatura da área de previsão concorda que não existe um único método que que seja melhor em todas as situações (JENKINS, 1982; CHATFIELD, 1988). Este fato é diretamente ligado com a alta complexidade dos problemas do mundo real e nenhum modelo ser capaz de capturar todos os diferentes padrões com a mesma alta qualidade sempre.

Há vários anos existem pesquisas sobre métodos híbridos e as melhorias obtidas pela previsão da série de resíduos como em (ZHANG, 2003; FIRMINO; FERREIRA, 2015; FIRMINO; NETO; FERREIRA, 2014; NETO et al., 2017). Essas pesquisas visam aproveitar o melhor de cada modelo de previsão para compor uma previsão mais acurada através de informações remanecentes na série de resíduo. O objetivo dessa composição é aproximar-se da série temporal desejada (ZHANG, 2003; SILVA et al., 2018).

A série residual pode ser obtida pela diferença entre a série temporal e a sua previsão (BROCKWELL; DAVIS, 2002). Também pode ser chamada de série de erros, ou resíduos. Resíduos são úteis para avaliar a qualidade da previsão realizada, ao checar se o modelo foi capaz de compreender e estimar valores futuros a partir das informações passadas.

A partir de uma série residual, por mais que a série pareça ter um comportamento randômico, é possível que a série original não tenha sido modelada adequadamente. Isso significa dizer que é possível que ainda existam valores correlacionados. E estes - quando identificados - podem ser utilizados para melhorar a previsão, principalmente se as correlações forem significativas (COWPERTWAIT, 2009b; HYNDMAN; ATHANASOPOULOS, 2018). Essa análise pode ser realizada utilizando o correlograma, e caso não hajam correlações significativas, a série residual é considerada como ruído branco, como ilustrado na Figura 11.

### 2.4.2 Additive Experts para dados contínuos

O sistema de múltiplos preditores (KOLTER; MALOOF, 2005) tem como objetivo realizar previsão em dados que podem mudar de comportamento. Para isso, o sistema conta com a ajuda de um modelo online qualquer e uma uma estratégia de ajuste da previsão. O sistema tem como objetivo construir um conjunto de preditores que se adaptam aos dados de forma a mitigar o problema de não-estacionariedade dos dados. O algoritmo funciona mantendo um conjunto (pool) ou sub-conjunto (ensemble) de preditores, onde a previsão do conjunto é a

média ponderada desses preditores, ou como chama o artigo: experts. Esse pool é autorregulado criando - e removendo, no caso do ensemble - experts, de forma dinâmica, e em resposta à mudanças de performance, como o pseudo-algoritmo em apresentado no Apêndice A.

### 2.4.3 Previsão de séries temporais utilizando uma abordagem de otimização por enxame de partículas

A metodologia abordada em OLIVEIRA et al., 2007 obteve destaque dentre os preditores online de abordagem ativa em comparação com o estado da arte da mesma categoria. Relembrando que a abordagem ativa está relacionada com a detecção da mudança de conceito a priori para, então, atualizar o modelo de previsão. No mesmo trabalho são propostas duas abordagens de detecção baseada no comportamento do enxame: (1) detecção baseada no comportamento do enxame inteiro (IDPSO-ELM-B); e (2) detecção baseada na avaliação de algumas partículas utilizadas como sensores (IDPSO-ELM-S). Em ambos os casos cada partícula do enxame é representado por uma máquina de aprendizado extremo (do inglês, Extreme Learning Machine ou ELM) (HUANG; ZHU; SIEW, 2006) e é treinado conforme a estratégia de otimização de enxame de partículas auto-adaptativas (ZHANG; XIONG; ZHANG, 2013), conhecida pelo título em inglês por *Improved Self-Adaptive Particle Swarm Optimization* (IDPSO).

O IDPSO é uma estretégia evolutiva que visa encontrar o conjunto de pesos que melhor represente os dados de treinamento. No início do treinamento cada ELM tem seu conjunto de pesos atribuído de maneira aleatória e através de modificações esses pesos são otimizados visando o menor valor de erro médio absoluto. O treinamento cessa com a estabilização do melhor resultado após uma certa quantidade de iterações, e o melhor conjunto de pesos passa a ser utilizado para realizar as previsões.

Após o treinamento, a ELM com o melhor conjunto de pesos da fase de treinamento passa a ser responsável por realizar a previsão de cada nova instância recebida. Esse modelo é responsável pelas previsões até ser substituído por outro modelo, o que indica que uma mudança de conceito ocorreu.

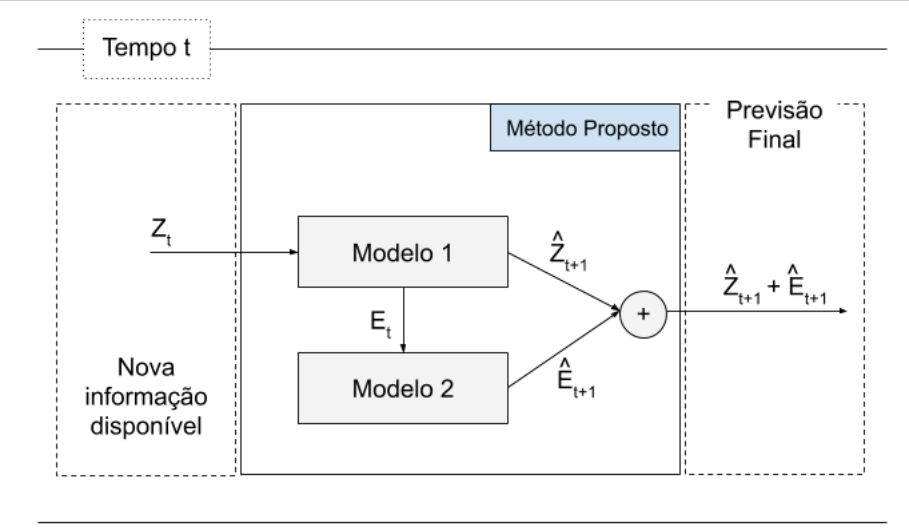
A detecção da mudança de conceito varia para as duas abordagens mencionadas anteriormente. A mudança de conceito baseada no comportamento do enxame (IDPSO-ELM-B) calcula primeiramente a média aritmética simples e o desvio padrão do erro de todos os modelos ELM presentes no enxame no momento do treinamento. Após isso, uma condição de mudança é avaliada a cada nova instância recebida. Esta condição considera uma média e um

desvio padrão, que por sua vez são calculados através da média móvel ponderada exponencialmente (EWMA), estimando a média de erro de uma certa quantidade de variáveis e as ponderando de forma que as mais recentes são consideradas mais importantes. Já a ideia da detecção de mudança de conceito baseada no comportamento dos sensores (IDPSO-ELM-S) consiste em selecionar um conjunto com os melhores modelos, calcular a média aritmética simples e o desvio padrão e os monitorar. O monitoramento, por sua vez, será apenas com o conjunto previamente selecionado e a detecção será levada em conta apenas quando o conjunto - unanimamente - concordar que uma mudança ocorreu.

# 3 MÉTODO PROPOSTO

O método proposto faz uso de dois modelos de previsão, como pode ser visto na Figura 4 que ilustra a arquitetura do método proposto. A Figura 4 mostra o papel de cada modelo na geração da previsão final do método proposto. O primeiro modelo é responsável pela previsão da série temporal sob estudo e o segundo busca aperfeiçoar a previsão do método, através da previsão do erro do primeiro modelo. A composição da previsão dos modelos, para este trabalho introdutório da previsão do resíduos na presença de concept drift, será feita através da soma, seguindo trabalhos como (ZHANG, 2003; FIRMINO; FERREIRA, 2015; NETO et al., 2017).

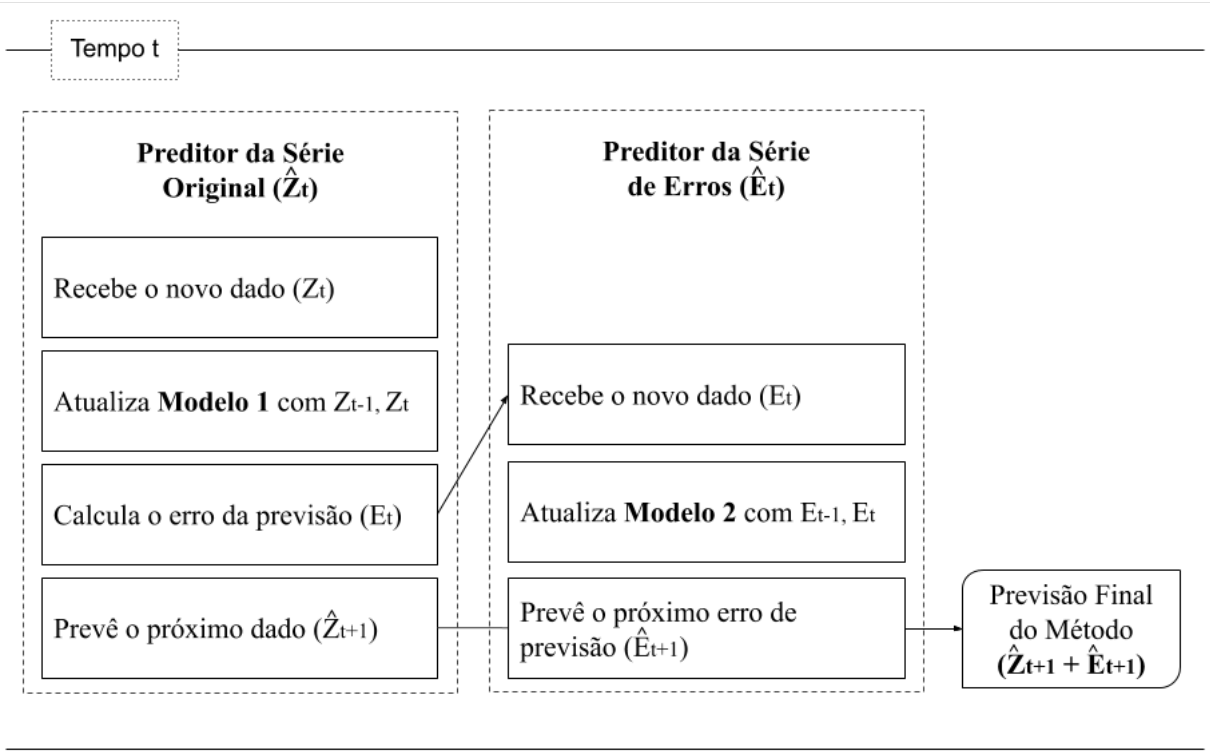
Figura 4 – Arquitetura do método proposto.




**Fonte:** Elaborada pela autora (2019)

A sequência geral de passos realizada para o ciclo de previsão online pode ser visto na Figura 5. A ideia é que a cada novo instante, quando um novo dado é disponibilizado, o Modelo 1 realiza a previsão da série original e em seguida este pode ser atualizado com $Z_{t-1}$ e $Z_t$; onde $Z_{t-1}$ sendo a informação recebida no tempo anterior; e $Z_t$ - a nova informação recebida - é utilizada como alvo da previsão. Após isso, é possível realizar o cálculo do erro da previsão do valor $Z_t$. O cálculo do erro é medido pela distância entre o alvo e previsão dele, dada por: $E_t = Z_t - \hat{Z}_t$. Com a nova informação do valor do erro, é possível atualizar o Modelo 2 e compor a previsão final do método somando a previsão do Modelo 1 e do Modelo 2 para o tempo $t + 1$.

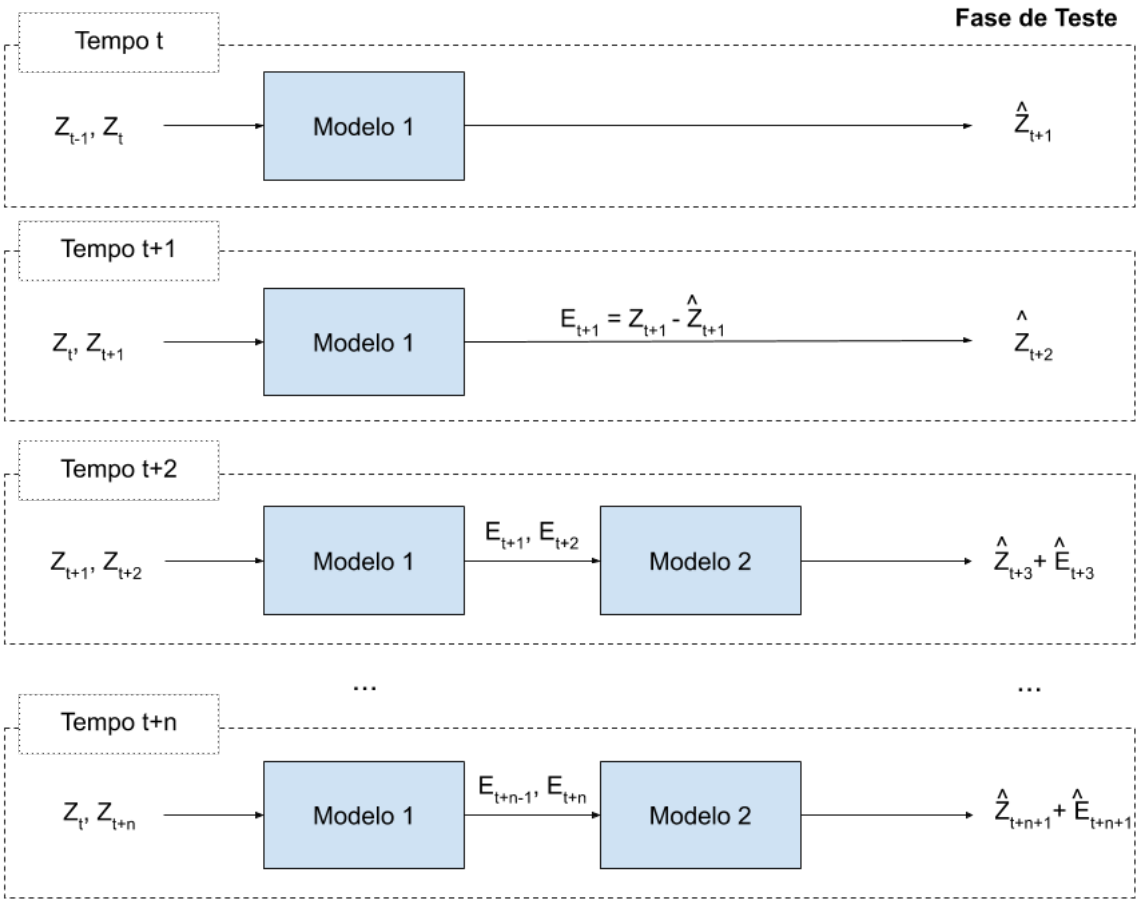
Figura 5 – Etapas do funcionamento do método proposto.




**Fonte:** Elaborada pela autora (2019)

A partir de um fluxo contínuo de dados, existem diversas abordagens para tratar e fornecer os dados de entrada para um modelo de previsão. No caso deste trabalho, o fluxo de dados é recebido e processado incrementalmente pelos modelos, ou seja, dado a dado. O objetivo é simplificar o método, extinguindo a necessidade de estimar tal parâmetro, já que a seleção da janela de tempo ideal é, por si só, considerada um desafio (GAMA et al., 2014). Na Figura 6 é possível observar os primeiros exemplos recebidos pelo método proposto. No tempo t é possível notar que são necessárias ao menos duas instâncias da série original para iniciar o treinamento do Modelo 1. No tempo t, o Modelo 1 recebe como par de treinamento $Z_t, Z_t$, sendo a instância $Z_{t-1}$ dada como entrada e $Z_t$ representando o alvo. Ao treinar com as duas primeiras instâncias, o Modelo 1 já está apto a fazer a primeira previsão. No tempo $t+1$, ao receber o alvo da previsão realizada no tempo $t$, é possível calcular o primeiro valor do erro $E_{t+1}$. É importante notar que cálculo do erro é realizado apenas para o Modelo 1. Assim como no Modelo 1, são necessárias ao menos duas instâncias do erro para que o Modelo 2 consiga obter seu primeiro exemplo de treinamento. A partir do tempo t+2, ambos os modelos cooperam na previsão do método proposto compondo o resultado através da soma.

Figura 6 – Funcionamento do método proposto a partir das primeiras informações recebidas.




**Fonte:** Elaborado pela autora (2019)

# 4 PROTOCOLO EXPERIMENTAL

## 4.1 ABORDAGEM UTILIZADA

A partir do objetivo de estudar a possibilidade do resíduo das previsões de modelos online conterem informações remanescentes úteis, quer-se determinar se tais informações são relevantes para contribuir para o aprimoramento da previsão. Diante disso, o início do estudo deu-se através da escolha de séries em que tenham sido detectadas mudanças de conceito, a subseção 4.2 realiza o detalhamento dessas séries. Em seguida o trabalho foi dividido em três partes:

1. Seleção dos modelos;
2. Busca de parâmetros ideais para esses modelos;
3. Correção do erro dos modelos monolíticos.

Cada uma das abordagens acima será desdobrada em uma etapa do desenvolvimento da pesquisa. De forma geral, na subseção 4.3 serão apresentados os 3 modelos clássicos escolhidos para conduzir os experimentos. Na subseção 4.4 serão apresentados os parâmetros utilizados no *gridsearch*. E na última subseção de desenvolvimento, 4.5, serão apresentadas as combinações do modelos monolíticos com os modelos de correção de erro. Já as métricas utilizadas para avaliar o desempenho das previsões serão detalhadas na seção 4.7.

## 4.2 SÉRIES UTILIZADAS

Séries temporais financeiras são conhecidas por não possuírem uma previsibilidade estável. Em finanças, essa instabilidade é caracterizada por contextos escondidos dos modelos de aprendizagem (WIDMER; KUBAT, 1996), como a taxa de juros ou o humor do mercado (HARRIES; HORN; SAMMUT, 1998), e tendo por consequência um componente estocástico (WANG; WANG, 2015). Devido à todas as variações e da previsibilidade não ser modelada por modelos estatísticos simples, é possível afirmar que o modelo utilizado para esta tarefa precisa de adaptações e/ou de um modelo de aprendizado contínuo. As séries escolhidas para serem parte do experimento deste trabalho são reais e fazem parte do mundo das finanças.

A escolha das séries temporais foi motivada por trabalhos como OLIVEIRA et al., 2017, onde a detecção dos pontos de mudança de conceito são realizadas e utilizadas para aprimorar a previsão do IDPSO. Nesse trabalho, a abordagem destacou-se obtendo resultados relevantes em comparação com outros métodos de previsão de séries temporais na presença de mudanças de conceito. As 3 bases de dados reais e seus respectivos períodos estão disponíveis no Yahoo Finance:

- Dow Jones de 29/Jan/1985 à 12/Mai/2017;
- Nasdaq de 02/Mai/1985 à 12/Mai/2017;
- S&P-500 de 15/Mai/1950 à 12/Mai/2017.

As séries captadas foram normalizadas entre [0,1] devido à exigência do algoritmo Additive Experts para dados contínuos (KOLTER; MALOOF, 2005). Com isso, além de descrever os dados estatísticos da série original, também serão descritas as informações para a série normalizada.

### 4.2.1 Dow Jones

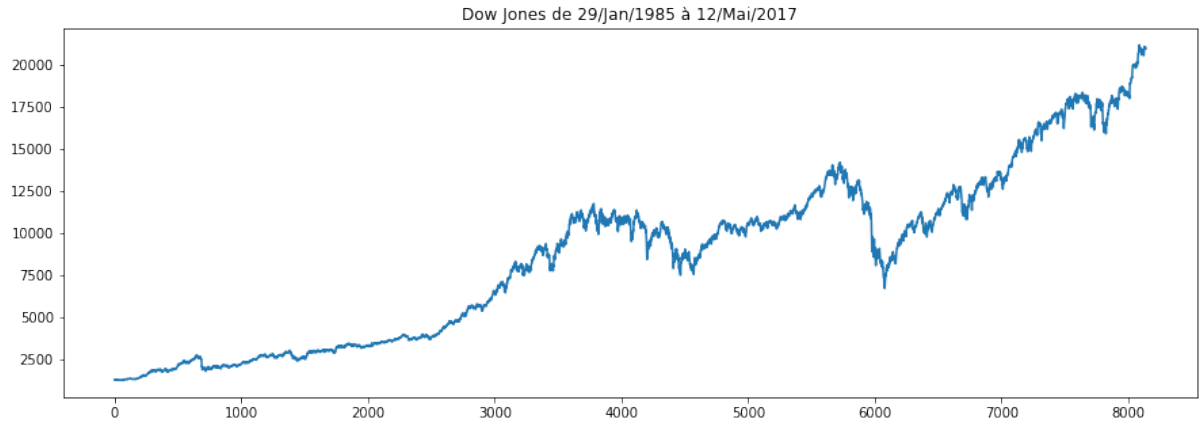
O nome completo do índice da bolsa de valores americana, cuja série temporal diária foi extraída para análise, é Dow Jones Industrial Average. Esse índice mensura a performance média das 30 maiores indústrias norte americanas listadas na bolsa de valores nova iorquina (NYSE). E embora o nome industrial conste no seu título, atualmente muitas empresas listadas não são relacionadas com as indústrias pesadas comuns da época em que foi fundada. A lista das indústrias que compõem o valor de índice, bem como a forma com que a média é calculada, tem evoluído e se modificado desde a sua criação (GANTI, 2020). A Tabela 1 descreve as estatísticas da série, enquanto a Figura 7 ilustra o comportamento da série Dow Jones para o intervalo de tempo previamente informado.

Tabela 1 – Estatísticas da série financeira Dow Jones antes e após a normalização

| **Estatística** | **Valores** | |
|---|---|---|
| **Mínimo** | 1251,21 | 0,00 |
| **Máximo** | 21169,10 | 1,00 |
| **Média** | 8649,64 | 0,371447 |
| **Desvio Padrão** | 5032.76 | 0,252676 |
| **Quantidade de Pontos** | 8.139 | |

**Fonte:** Elaborada pela autora (2019)

Figura 7 – Visualização do comportamento da séries Dow Jones




**Fonte:** Elaborada pela autora (2019)

### 4.2.2 NASDAQ

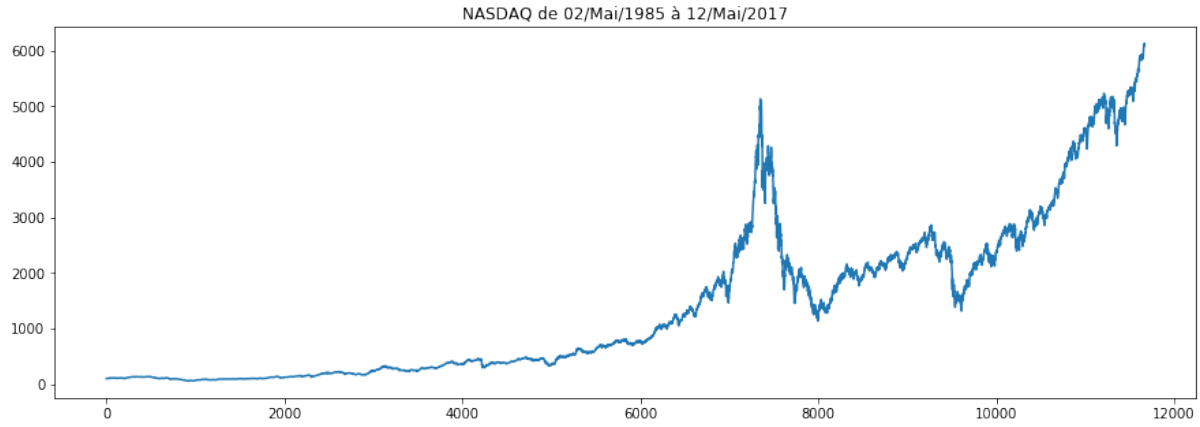
A sigla NASDAQ vem do inglês: *National Association of Securities Dealers Automated Quotations System*, e se traduz em associação nacional de corretores de títulos de cotações automáticas. A associação NASDAQ foi fundada em 1971 com o objetivo de ser um mercado de ações eletrônico (NASDAQ, 2020), e anos depois foi pioneiro em negociações online - atraindo empresas de tecnologia. Atualmente, é o segundo maior mercado de ações, precedido apenas pela bolsa de valores de Nova York (NYSE) (CFI, 2020). O índice *NASDAQ Composite* é constituído por ações listadas no mercado de ações NASDAQ e conta, em sua maioria, com empresas do setor de tecnologia (KENNON, 2019). A Tabela 2 descreve algumas estatísticas da série, enquanto a Figura 8 ilustra o comportamento da série NASDAQ para o intervalo de tempo previamente informado.

Tabela 2 – Estatísticas da série financeira NASDAQ antes e após a normalização

| **Estatística** | **Valores** | |
|---|---|---|
| **Mínimo** | 54,87 | 0,00 |
| **Máximo** | 6.133,00 | 1,00 |
| **Média** | 1.408,12 | 0,222644 |
| **Desvio Padrão** | 1.440,30 | 0,236965 |
| **Quantidade de Pontos** | 11.667 | |

**Fonte:** Elaborada pela autora (2019)

Figura 8 – Visualização do comportamento da séries NASDAQ




**Fonte:** Elaborada pela autora (2019)

### 4.2.3 S&P 500

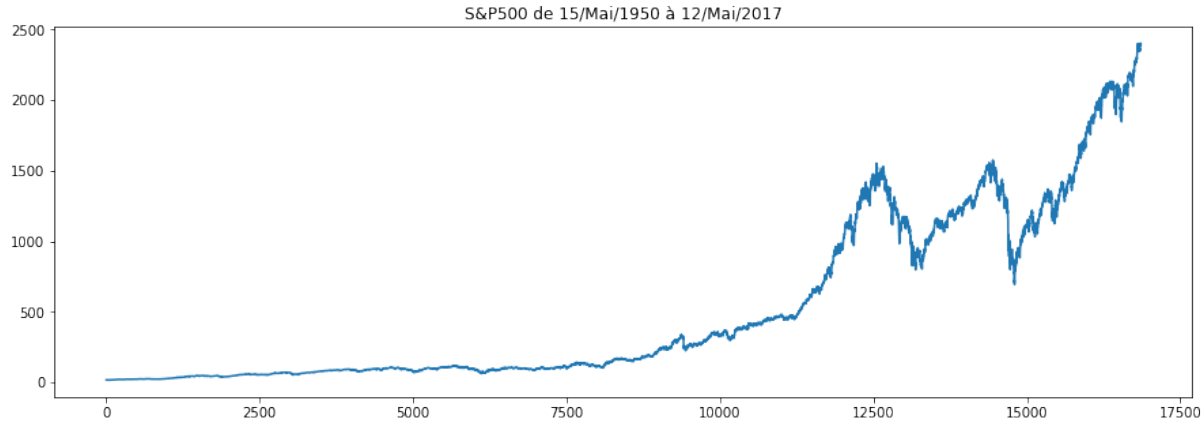
A empresa conhecida por *Standard & Poor's 500* foi fundada como um serviço de informações financeiras. Essas informações eram embasadas em pesquisas e dados estatísticos, utilizadas como fonte de informação para investimentos e que, anos depois, criou seu prório índice: S&P 500 (S&P-GLOBAL, 2020). Este índice mantém-se até o momento e representa a performance das 500 maiores companhias de capital aberto listadas na bolsa de valores estado-unidense (S&P-DOW-JONES, 2020). A Tabela 3 descreve algumas estatísticas da série, enquanto a Figura 9 ilustra o comportamento da série S&P500 para o intervalo de tempo previamente informado.

Tabela 3 – Estatísticas da série financeira S&P500 antes e após a normalização

| **Estatística** | **Valores** | |
|---|---|---|
| **Mínimo** | 16,68 | 0,00 |
| **Máximo** | 2.403,87 | 1,00 |
| **Média** | 523,75 | 0,212416 |
| **Desvio Padrão** | 603,39 | 0.252764 |
| **Quantidade de Pontos** | 16.857 | |

**Fonte:** Elaborada pela autora (2019)

Figura 9 – Visualização do comportamento da séries S&P500




**Fonte:** Elaborada pela autora (2019)

## 4.3 SELEÇÃO DOS MODELOS

Em estatística, um modelo é considerado uma descrição probabilística de uma série temporal. Esta descrição é um meio de fornecer informações para a tomada de decisão conforme o objetivo que se quer alcançar (MORETTIN, 2004). Em redes neurais, pela mimetização da função biológica do cérebro, um modelo é algo mais complexo do que uma descrição probabilística, porém da mesma forma como na estatística, os dados são utilizados para moldar o comportamento dos modelos conforme o objetivo.

Seguindo a necessidade de estudar o comportamento das séries residuais, o raciocínio foi iniciar o estudo por modelos clássicos da literatura, mas que também estavam disponíveis em sua forma online, pelo bom desempenho e uso corrente. Seguindo esse racioncínio, três modelos estão presentes nos experimentos: Perceptron, MLP e SVR.

As versões dos modelos clássicos utilizados são adaptações para o aprendizado online e de forma incremental, para que o modelo consiga ter a flexibilidade de adaptar-se com novos exemplos. A forma mais básica de um modelo incremental é um modelo único que suporta esse tipo de treinamento. Para fins de referência chamaremos de monolítico os modelos que não possuem nenhuma forma de ajuste de parâmetros.

Mais detalhadamente, foram utilizados o Perceptron, e os modelos MLP e SVR disponíveis pela biblioteca Scikit-Learn (PEDREGOSA et al., 2011). Os parâmetros originais utilizados estão disponíveis no Apêndice B. A biblioteca Scikit-Learn (PEDREGOSA et al., 2011) e tais modelos foram utilizados devido à padronização de implementação e ao curto tempo de realização deste estudo, portanto foram utilizados os modelos disponíveis pela biblioteca da linguagem

Python.

### 4.3.1 Uso dos Modelos

Os modelos selecionados foram utilizados em dois experimentos diferentes: o primeiro experimento foi utilizar a abordagem tradicional de um modelo ajustado, através da busca de parâmetros. Neste caso não houve a correção dos erros de previsão e os detalhes dessa busca de parâmetros é discutida na seção 4.4. E o segundo experimento, que é o método proposto, nenhuma modificação é realizada nos parâmetros originais dos modelos fornecidos pela biblioteca (PEDREGOSA et al., 2011), cujos detalhes estão disponíveis no Apêndice B.

## 4.4 AJUSTE DE PARÂMETROS

O ajuste de parâmetros foi realizado em 25% da série original, através de um *gridsearch* dentre os parâmetros principais dos modelos selecionados. Essa porção de dados utilizada foi considerada suficiente por representar uma quantidade absoluta de mais de dois mil pontos.

Os parâmetros de melhor desempenho deste *grid search* foram então selecionados para realizar a previsão da série completa e o desepenho dessas previsões foram utilizados como comparativo com os resultados do método proposto.

As tabelas 4, 5 e 6 apresentam os parâmetros e valores utilizados no *grid search* dos modelos Perceptron, MLP e SVR respectivamente.

Tabela 4 – Lista de parâmetros de teste utilizados no *grid search* do modelo Perceptron.

| **Descrição do Parâmetro** | **Valores** |
|---|---|
| **Taxa de Aprendizado:** | 0,01 / 0,1 / 0,5 / 1 |
| **Função de Ativação:** | Linear / Pseudo-Linear / Sigmóide Logística / Tangente Hiperbólica |

**Fonte:** Elaborada pela autora (2019)

Tabela 5 – Lista de parâmetros de teste utilizados no *grid search* do modelo MLP.

| Descrição do Parâmetro | Valores |
|---|---|
| **Otimizador** | SGD / ADAM |
| **Taxa de Aprendizado:** | 0,001 / 0,01 / 0,1 |
| **Regularização** | 0,0001 / 0,001 / 0,01 |
| **Neurônios** | 5 / 10 / 30 |
| **Camadas** | 1 |

**Fonte:** Elaborada pela autora (2019)

Tabela 6 – Lista de parâmetros de teste utilizados no *grid search* do modelo SVR.

| Descrição do Parâmetro | Valores |
|---|---|
| **Taxa de Aprendizado:** | 0,001 / 0,01 / 0,1 |
| **Regularização** | 0,0001 / 0,001 / 0,01 |
| **Epsilon** | 0,001 / 0,01 / 0,1 |

**Fonte:** Elaborada pela autora (2019)

## 4.5 CORREÇÃO DOS ERROS

A hipótese deste trabalho é que realizar a previsão da série residual possa resultar em uma previsão geral mais precisa - mesmo dentro de um ambiente online e com mudança de conceito - do que realizar apenas uma previsão da série original. A concretização desta correção neste trabalho foi realizada da forma mais básica, isto é, utilizando apenas um modelo clássico com aprendizado incremental. Consequentemente, caso uma unidade simples consiga aperfeiçoar os resultados, é possível que um modelo ou estratégia mais complexa consiga - também - prover melhorias relevantes ou que modelos complexos sejam substituídos por modelos simples acrescidos da correção do erro destes.

A correção dos erros de previsão foram realizadas utilizando apenas modelos monolíticos, isto é, modelos sem nenhuma otimização de parâmetro. Mais detalhes sobre os parâmetros utilizados nos modelos monolíticos diante da versão da biblioteca utilizada podem ser encontrados no apêndice B.

A tabela 7 apresenta as combinações possíveis realizadas dentre os modelos de correção da série original e os modelos de correção do erro.

Tabela 7 – Combinações entre os modelos usados na série original e dos modelos monolíticos usados na série de erros.

| **Modelo 1** | Perceptron | | | MLP | | | SVR | | |
|---|---|---|---|---|---|---|---|---|---|
| **Modelo 2** | Perceptron | MLP | SVR | Perceptron | MLP | SVR | Perceptron | MLP | SVR |

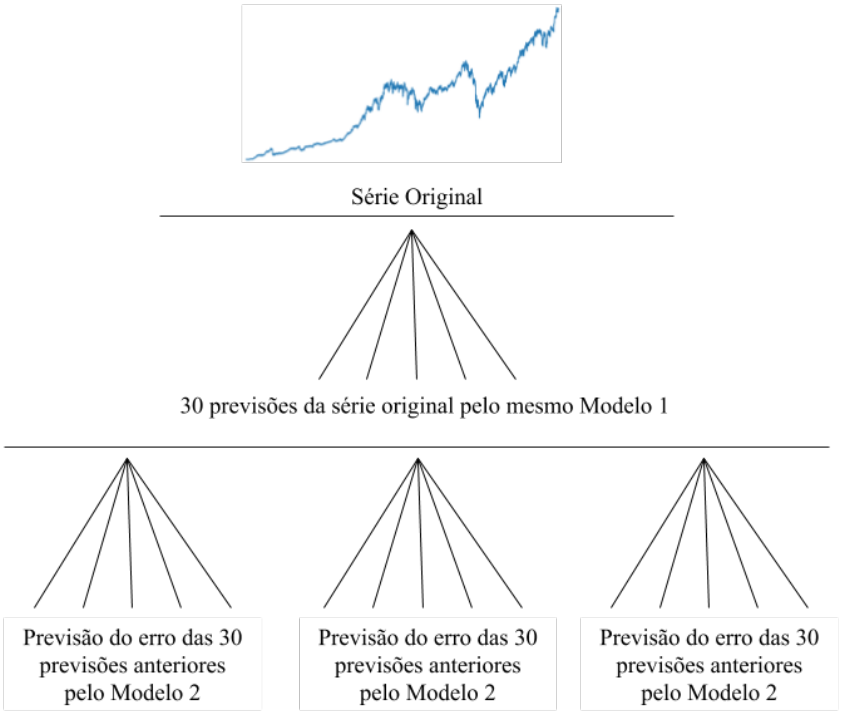
**Fonte:** Elaborada pela autora (2019)

## 4.6 EXPERIMENTOS REALIZADOS

O mesmo procedimento de previsão foi aplicado em todos os experimentos e a figura 10 o ilustra os modelos monolíticos e os modelos com parâmetros ajustados aplicam-se apenas para onde tem Modelo 1, ou seja, o primeiro nível de previsão.

Para as experimentações do método proposto, um modelo de previsão da série original realiza trinta previsões e essas mesmas trinta previsões são corrigidas por todos os modelos de previsão do erro. Ao realizar trinta previsões distintas usando um mesmo Modelo 1, os modelos de previsão do erro tem oportunidade igual de prever as mesmas trinta séries erros gerada pelo Modelo 1. Realizar o procedimento anterior, ao invés de executar cada conjunto (preditor da série original e preditor dos erros) na mesma execução trinta vezes, permite uma comparação mais justa entre os modelos de previsão do erro. Sendo trinta uma quantidade relevante de repetições para calcular a significância estatística.

Figura 10 – Metodologia dos experimentos visando uma comparação mais justa.




**Fonte:** Elaborada pela autora (2019)

## 4.7 MÉTRICAS DE AVALIAÇÃO E TESTES ESTATÍSTICOS

### 4.7.1 Erro Médio Absoluto (MAE)

Sua sigla vem do inglês: *Mean Absolute Error*. O MAE é calculado a partir da diferença absoluta média entre a série original $x_t$ e a previsão do modelo $\hat{x}_t$, como pode ser visto na Equação 4.1:

$$MAE = \frac{1}{N}\sum_{t=1}^{N}(|x_t - \hat{x}_t)|). \tag{4.1}$$

### 4.7.2 Ganho Percentual

A métrica de ganho percentual (GP) foi utilizada para quantificar o desempenho do método proposto. Dessa forma, esse indicador irá mensurar o percentual de refinamento realizado a partir da previsão inicial e diante da previsão do erro. Sendo G o MAE do método proposto e

S o MAE da previsão sem a correção do erro, o GP é calculada como:

$$GP = \frac{S - G}{S} \cdot 100. \tag{4.2}$$

### 4.7.3 Função de Autocorrelação (ACF)

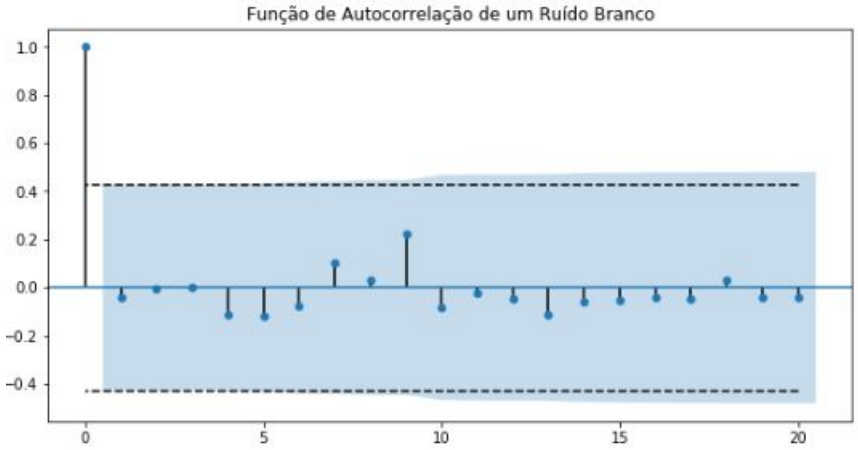
A medida de correlação tem por objetivo avaliar o grau de dependência e mensurar a relação entre duas variáveis. Já a autocorrelação mede o relacionamento linear entre valores da mesma série (auto), porém com atraso (também conhecido como lag) em relação aos valores da série. Sendo a medida de $r_k$ o cálculo da autocorrelação entre os dados da série com os valores dos dados da série com atraso de $k$ na medida temporal:

$$r_k = \frac{\sum_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T} (y_t - \bar{y})^2}, \tag{4.3}$$

Portanto, uma comparação entre a relação entre os dados no tempo original $y_t$ e dados os mesmos dados atrasados em 1 (uma) unidade de tempo $y_{t-1}$, seria indicado pelo cálculo de $r1$, ou $r2$ se o desejo for da comparação entre a autocorrelação entre os dados atuais e os mesmos dados com lag de 2 (duas) unidade de tempo $y_t$ e $y_{t-2}$.

Além dos valores obtidos com a equação (4.3), a autocorrelação também pode ser observada com um gráfico específico para a observação de correlações, chamado de *correlograma*. Além de ser um excelente aliado como um recurso visual da observação do comportamento do resíduo. Um exemplo deste gráfico pode ser visto na Figura 11; que mostra um ruído branco simulado. Os valores contidos entre as linhas tracejadas indicam que estes valores de lag são significativamente iguais a zero. E a região além das linhas tracejadas indica que as correlações são significativamente diferentes de zero. Para uma série de ruído branco é esperado ao menos 95% dos pontos correspondentes ao valor de correção estarem dentro da região azul. Se um ou mais valores altos de correlação estiverem fora deste limiar ou se muito mais do que 5% dos valores de autocorrelação estiverem fora deste limiar, então é provável que esta série não seja um ruído branco (HYNDMAN; ATHANASOPOULOS, 2018; BROCKWELL; DAVIS, 2002).

Figura 11 – Correlograma de um ruído branco simulado, onde todas as autocorrelações são zero (exceto o lag 0). É possível que exista algum lag estatísticamente relevante devido à variação da amostra.




**Fonte:** Elaborada pela autora (2019)

### 4.7.4 Teste de Autocorrelação de Portmanteau - Teste de Ljung-Box

De maneira a formalizar e complementar a leitura do correlograma da função de autocorrelação (ACF) é possível considerar um conjunto (ou grupo) de autocorrelações $r_iuk$. Um teste realizado com um grupo de autocorrelações é chamado de **Portmanteau test**, devido ao significado da palavra francesa *Portmanteau* que significa mala, fazendo um paralelo com o conjunto de itens/valores que estão nela. Fazer uma análise conjunta, ao invés de fazer uma análise separada dos possíveis valores de $r_k$ minimiza os riscos da ocorrência de um falso positivo. Desta forma é possível testar se as primeiras $h$ autocorrelações diferem significativamente do que é esperado de um ruído branco.

O teste utilizado neste trabalho foi o teste de **Ljung-Box** baseado na estatística abaixo, onde:

$T$: Tamanho da série;

$h$: o valor máximo do lag;

$k$: número de parâmetros do modelo utilizado.

$$Q^* = T(T+2)\sum_{k=1}^{h}(T-k)^{-1}r_k^2 \tag{4.4}$$

E sendo a avaliação da teste:

$H_0$: Não há autocorrelação significativa entre os lags.

$H_a$: Existe autocorrelação significativa entre os lags.

Caso os dados utilizados sejam dados brutos (ao invés de ser resíduos de um modelo) ou caso o modelo utilizado não utilize parâmetros utilize K=0. Ao olhar para o p-valor, caso ele seja menor do que o nível de significância é possível rejeitar a hipótese nula.

### 4.7.5 Teste de Friedman

O teste de Friedman é um teste estatístico não paramétrico que ranqueia os algoritmos separadamente para cada conjunto de dados. O algoritmo de melhor performance sendo ranqueado como o primeiro, o segundo melhor como o segundo e assim sucessivamente. Em caso de empate, o ranque dos algoritmos é a média dos ranques dentre as posição dos algoritmos empatados.

Seja $r_i^j$ o ranque do j-ésimo de k algoritmos na i-ésima de N séries temporais. O teste de Friedman compara a média do ranque dos algoritmos, $R_j = \frac{1}{N}\sum r_i^j$. Sob a hipótese nula de que os algoritmos possuem desempenho semelhante e portanto seus tanques $R_j$ devem ser iguais, a estatística de Friedman

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{4.5}$$

segue a distribuição $\chi_F^2$ com $K-1$ graus de liberdade, quando $N > 10$ e $k > 5$ (DEMŠAR, 2006). Para uma menor quantidade de séries e algoritmos, há uma tabela mais adequada com os valores críticos exatos (SHESKIN, 2000).

### 4.7.6 Teste post-hoc Nemenyi

O teste de Nemenyi (NEMENYI, 1963) é utilizado como um teste discriminante para quando modelos são comparados entre si. Dois a dois, os testes tem sua estatística definida por:

$$z = (R_i - R_j)/\sqrt{\frac{k(k+1)}{6N}}, \tag{4.6}$$

onde k a quantidade de algoritmos e N a quantidade de séries temporais. O valor de $z$ é utilizado para encontrar a probabilidade correspondente a partir da tabela de distribuição normal, então esse valor é comparado com o nível de significância ($\alpha$) apropriado (DEMŠAR, 2006).

# 5 RESULTADOS EXPERIMENTAIS

## 5.1 IDENTIFICAÇÃO DE RUÍDO BRANCO

Visando identificar a existência de informação remanescente nas séries residuais foram utilizados dois testes de identificação de ruído branco a função de autocorrelação e o teste de Portmanteau. Ambos os testes foram realizados utilizando 20 lags nos resíduos das 30 execuções das três séries. Um resumo descritivo dos modelos testados, e os respectivos valores máximos e mínimos de lags significativos são apresentados na tabela 8. A contagem dos lags significativos reproduzida nesta última tabela não corresponde apenas aos lags significativos consecutivos.

Tabela 8 – Quantidade mínima e máxima de lags significativos - não necessariamente consecutivos - diante dos autocorrelogramas gerados. O lag 0 (zero) da função de autocorrelação não foi contabilizado.

| **Modelo 1** | **Dow Jones** | | **Nasdaq** | | **S&P 500** | |
|---|---|---|---|---|---|---|
| | **Lags** | | **Lags** | | **Lags** | |
| **Monolítico** | **Min** | **Max** | **Min** | **Max** | **Min** | **Max** |
| Perceptron | 0 | 0 | 0 | 1 | 0 | 1 |
| MLP | 2 | 4 | 2 | 3 | 2 | 4 |
| SVR | 4 | 4 | 4 | 4 | 2 | 6 |

**Fonte:** Elaborada pela autora (2020)

O resíduo destes modelos, para as três séries selecionadas e para cada uma das 30 execuções produziu no teste de Portmanteau o p-valor unânime e igual a 0 para os modelos SVR e MLP. No caso do Perceptron, como era de se esperar, pela baixa quantidade lags significativos de autocorrelação para a série Dow Jones, existiram resultados do teste estatístico de Portmanteau que não indicaram índícios estatísticos de rejeição da hipótese nula. Muito embora, 90% dos p-valores dentre as 30 execuções indicaram estatísticamente a existência de informações nos resíduos com nível de significância de 5%, correspondendo também à 86,7% com nível de significância de 1%. Para as séries Nasdaq e S&P 500 o p-valor do teste de Portmanteau foi o harmonicamente abaixo do nível de significância de 1%.

Através dos resultados detalhados acima é possível responder a primeira pergunta de pesquisa, concluindo que o resultado de ambas abordagens não possuem indicativos suficientes que identifiquem os resíduos como ruído branco. Portanto, com os testes realizados, tais re-

sultados corroboram para a existência de informações remanescentes nos resíduos, e indicam a possibilidade de modelagem dos resíduos proveniente da previsão de um método online. Este trabalho, também, sugere que tais informações podem auxiliar em uma previsão ainda mais precisa.

## 5.2 COMPARAÇÃO DENTRE OS RESULTADOS

### 5.2.1 Resultados do grid search

Diante dos experimentos de *grid search* referenciados nas tabelas da seção de ajuste de parâmetros 4.4, apenas os melhores resultados de MAE foram selecionados e estão descritos na Tabela 9. Como explicitado anteriormente na seção 4.6, o método proposto será comparado com os mesmos modelos, só que ao invés de utilizar modelos monolíticos, serão utilizados modelos com parâmetros ajustados.

Tabela 9 – Parâmetros e respectivos valores decorrente dos melhores resultados de acurácia do *grid search*.

| **Série** | **Perceptron** | | **MLP** | | **SVR** | |
|---|---|---|---|---|---|---|
| | **Parâmetros** | **Valores** | **Parâmetros** | **Valores** | **Parâmetros** | **Valores** |
| Dow Jones | Taxa de Aprendizado | 1 | Otimizador | SGD | Taxa de Aprendizado | 0,1 |
| | Função de Ativação | Pseudo-Linear | Taxa de Aprendizado | 0,01 | Regularização | 0,0001 |
| | | | Regularização | 0,1 | Epsilon | 0,1 |
| | | | Neurônios | 30 | | |
| Nasdaq | Taxa de Aprendizado | 1 | Otimizador | SGD | Taxa de Aprendizado | 0,1 |
| | Função de Ativação | Linear | Taxa de Aprendizado | 0,1 | Regularização | 0,0001 |
| | | | Regularização | 0,1 | Epsilon | 0,1 |
| | | | Neurônios | 30 | | |
| S&P 500 | Taxa de Aprendizado | 1 | Otimizador | SGD | Taxa de Aprendizado | 0,1 |
| | Função de Ativação | Tangente | Taxa de Aprendizado | 0,001 | Regularização | 0,0001 |
| | | | Regularização | 0,1 | Epsilon | 0,1 |
| | | | Neurônios | 30 | | |

**Fonte:** Elaborada pela autora (2020)

Para comparar os modelos de forma justa, nas tabelas de resultados 12a, 13a, 14a constam, além do tempo médio da execução da série inteira com o melhor conjunto parâmetro de um modelo para uma série, o tempo de execução médio de cada combinação de parâmetros do *grid search*.

### 5.2.2 Desempenho dos Modelos

Visando entender a relação entre modelos ajustados e modelos não ajustados (monolíticos) são apresentados nas Tabelas 12a, 13a, 14a uma visualização que favorece a comparação entre a acurácia e o tempo médio de execução dessas abordagens. É possível visualizar nestas tabelas que as previsões dos modelos monolíticos do perceptron e da MLP são, em sua maioria, mais acurados do que os modelos ajustados. Uma excessão dentre as três séries é a SVR que sempre apresenta um melhor resultado utilizando o conjunto de parâmetros resultante do gridsearch. O contraponto do *gridsearch* para o SVR e também para os outros modelos é que essa busca por melhores parâmetros reflete um alto tempo de execução quando comparado com os modelos sem ajuste de parâmetros e até sem correção de erro.

Em contrapartida, nas Tabelas 12b, 13b, 14b são apresentados os resultados do método proposto na ordem de um ranking ascendente pelo MAE. O ranking permite uma comparação breve entre os melhores resultados do método proposto com os modelos de parâmetros ajustados. Como primeira observação, é possível apontar que o método proposto possui a melhor acurácia para as três séries propostas, além do tempo de execução também ter sido menor. Outro ponto positivo é fato de não haver grande diversidade dentre as três melhores combinações de modelos. Dentre os três melhores resultados, sempre estão presentes as combinações de SVR + Perceptron, Perceptron + SVR e MLP + Perceptron. Embora os modelos de correção de erro devem ser estudados com mais profundidade, podendo - inclusive - trazer maiores ganhos para o método.

O teste de Friedman foi, então, conduzido visando atestar se existe ou não diferença significativa entre os resultados de MAE. Com o nível de significância de 5% e p-valor igual a 0,0, o teste de hipótese rejeita a hipótese nula de que não existe diferença entre os modelos avaliados. Após o teste de Friedman, foi realizado o teste de hipótese Nemenyi que realizada uma comparação múltipla para indicar quais resultados são significativamente diferentes, com resultados apresentados na Tabela 10. Com nível de significância de 5%, não houve diferença significativa entre os modelos com ajuste de parâmetro e o método proposto. Em contrapartida, entre os modelos monolíticos e o método proposto o teste Nemenyi destaca a melhoria entre os resultados do modelo monolítico do SVR e de sua composição com o Perceptron.

Tabela 10 – Comparação múltipla entre método proposto, os modelos com ajuste de parâmetroos modelos e os modelos monolíticos, representado pelo p-valor resultante do teste Nemenyi apenas para as 3 melhores abordagens do método proposto.

| **Teste de Nemenyi** | | | | | | |
|---|---|---|---|---|---|---|
| Método | Modelos Ajustados | | | Modelos Monolíticos | | |
| Proposto | PERC. | MLP | SVR | PERC. | MLP | SVR |
| Perc. & SVR | 0,9000 | 0,1673 | 0,9000 | 0,9000 | 0,7648 | 0,0209 |
| SVR & Perc. | 0,9000 | 0,1251 | 0,8576 | 0,9000 | 0,7648 | **0,0209** |
| MLP & Perc. | 0,9000 | 0,5730 | 0,9000 | 0,9000 | 0,9000 | 0,2309 |

**Fonte:** Elaborada pela autora (2020)

Apesar do teste estatístico Nemenyi não indicar diferença significativa entre as diferentes abordagens, é importante citar as condições de comparação entre os modelos. Aproveitando que terceira pergunta de pesquisa visa entender se a correção do erro em ambientes online pode ser considerada uma opção para reduzir os impactos de uma previsão sem ajuste de parâmetros, serão abordadas também neste parágrafo as vantagens do método proposto. Primeiramente, é possível citar a acurácia com baixo tempo de execução e, mais importante, a previsão da série a partir do segundo ponto de treinamento. Como explicado na seção 4.4, o ajuste de parâmetros resultante do *gridsearch* foi realizado utilizando $1/4$ da série original, e portanto - em uma aplicação real - esta parcela de dados é inutilizada após seu uso, já que as informações foram utilizadas para medir o desempenho dos parâmetros e estão obsoletas. Já nos resultados reportados nas Tabelas 12, 13 e 14 em todos os casos - para fins de uniformidade e em todos os casos - a série inteira foi prevista e, portanto, os modelos ajustados tem vantagem nas comparações que serão feitas a seguir. Os melhores parâmetros, por sua vez, não demonstram ser a única e, nem mesmo, a mais relevante forma de obter os melhores resultados de previsão de uma série temporal online na presença de *concept drift*, como pôde demonstrar os experimentos.

Figura 12 – Resultados de MAE, desvio padrão (STD) e tempo para a série Dow Jones. A escala de cores segue o esquema de mapa de calor, onde quanto mais escuro o vermelho, menor o valor de MAE.

| Modalidade | Modelo 1 | MAE | STD | Tempo Médio |
|---|---|---|---|---|
| Ajustado | Perceptron | 0.002971 | 0.000320 | 00:00:50.340870 |
| | MLP | 0.007286 | 0.000348 | 00:59:06.477510 |
| | SVR | 0.005031 | 0.000000 | 00:11:42.635790 |
| Monolítico | Perceptron | 0.002535 | 0.000117 | 00:00:00.291851 |
| | MLP | 0.006158 | 0.000127 | 00:00:11.267013 |
| | SVR | 0.062572 | 0.000000 | 00:00:03.746210 |

(a) Modelos online sem correção de erro.

| Modelo 1 | Modelo 2 | MAE | STD | Tempo Médio |
|---|---|---|---|---|
| Perceptron | SVR | 0.002484 | 0.000115 | 00:00:05.682677 |
| SVR | Perceptron | 0.002694 | 0.000309 | 00:00:04.206926 |
| MLP | Perceptron | 0.003066 | 0.000159 | 00:00:11.796818 |
| Perceptron | MLP | 0.003515 | 0.000228 | 00:00:17.721748 |
| | Perceptron | 0.004886 | 0.000945 | 00:00:00.546398 |
| SVR | MLP | 0.005848 | 0.000108 | 00:00:23.608057 |
| MLP | SVR | 0.006301 | 0.000268 | 00:00:17.040368 |
| | MLP | 0.008716 | 0.000518 | 00:00:26.906029 |
| SVR | SVR | 0.032501 | 0.000000 | 00:00:11.524421 |

(b) Modelos monolíticos com erros corrigidos.

**Fonte:** Elaborada pela autora (2020)

Figura 13 – Resultados de MAE, desvio padrão (STD) e tempo para a série Nasdaq. A escala de cores segue o esquema de mapa de calor, onde quanto mais escuro o vermelho, menor o valor de MAE.

| Modalidade | Modelo 1 | MAE | STD | Tempo Médio |
|---|---|---|---|---|
| Ajustado | Perceptron | 0.002331 | 0.000222 | 00:01:15.167490 |
| | MLP | 0.005563 | 0.000294 | 01:24:09.624450 |
| | SVR | 0.004209 | 0.000000 | 00:16:42.773520 |
| Monolítico | Perceptron | 0.002251 | 0.000167 | 00:00:00.391524 |
| | MLP | 0.005080 | 0.000222 | 00:00:17.649121 |
| | SVR | 0.051799 | 0.000000 | 00:00:05.370103 |

(a) Modelos online sem correção de erro.

| Modelo 1 | Modelo 2 | MAE | STD | Tempo Médio |
|---|---|---|---|---|
| SVR | Perceptron | 0.002056 | 0.000216 | 00:00:06.089923 |
| Perceptron | SVR | 0.002196 | 0.000175 | 00:00:07.868430 |
| MLP | Perceptron | 0.002286 | 0.000098 | 00:00:18.229451 |
| Perceptron | MLP | 0.002705 | 0.000209 | 00:00:30.401312 |
| | Perceptron | 0.004295 | 0.000773 | 00:00:01.061646 |
| SVR | MLP | 0.005089 | 0.000317 | 00:00:32.910493 |
| MLP | SVR | 0.005261 | 0.000545 | 00:00:25.474144 |
| | MLP | 0.006476 | 0.000389 | 00:00:40.419503 |
| SVR | SVR | 0.034188 | 0.000000 | 00:00:15.898840 |

(b) Modelos monolíticos com erros corrigidos.

**Fonte:** Elaborada pela autora (2020)

Figura 14 – Resultados de MAE, desvio padrão (STD) e tempo para a série S&P 500. A escala de cores segue o esquema de mapa de calor, onde quanto mais escuro o vermelho, menor o valor de MAE.

| Modalidade | Modelo 1 | MAE | STD | Tempo Médio |
|---|---|---|---|---|
| Ajustado | Perceptron | 0.001718 | 0.000190 | 00:01:41.099370 |
| | MLP | 0.003858 | 0.000177 | 02:00:45.425130 |
| | SVR | 0.002743 | 0.000000 | 00:24:04.234560 |
| Monolítico | Perceptron | 0.001728 | 0.000186 | 00:00:00.560067 |
| | MLP | 0.003521 | 0.000309 | 00:00:24.712242 |
| | SVR | 0.040642 | 0.000000 | 00:00:07.751726 |

(a) Modelos online sem correção de erro.

| Modelo 1 | Modelo 2 | MAE | STD | Tempo Médio |
|---|---|---|---|---|
| SVR | Perceptron | 0.001468 | 0.000146 | 00:00:08.735173 |
| Perceptron | SVR | 0.001700 | 0.000184 | 00:00:12.121343 |
| MLP | Perceptron | 0.001739 | 0.000133 | 00:00:25.660571 |
| Perceptron | MLP | 0.002118 | 0.000199 | 00:00:31.858869 |
| MLP | SVR | 0.003626 | 0.000482 | 00:00:36.707277 |
| Perceptron | Perceptron | 0.003645 | 0.000842 | 00:00:01.282927 |
| SVR | MLP | 0.003709 | 0.000291 | 00:00:45.237561 |
| MLP | MLP | 0.004722 | 0.000365 | 00:00:56.505656 |
| SVR | SVR | 0.023576 | 0.000000 | 00:00:21.759429 |

(b) Modelos monolíticos com erros corrigidos.

**Fonte:** Elaborada pela autora (2020)

### 5.2.3 Ganho Percentual

Através do ganho percentual é possível avaliar de forma mais direta a segunda pergunta de pesquisa, onde quer-se entender se a previsão dos resíduos por modelos online podem ser utilizados para melhorar as estimativas das séries temporais em ambientes online. Na Tabela 11 o desempenho do método proposto em relação aos modelos monolíticos é representado utilizando-se da métrica de ganho percentual. A tabela também foi ranqueada em ordem decrescente segundo a métrica utilizada e as três linhas grifadas representam os três melhores valores de MAE das Tabelas 12b, 13b, 14b. É possível observar que, por série, em mais de 50% das combinações o uso da correção do resíduo das séries foi benéfico. Inclusive, no mínimo dois casos de melhoria devem ser explorados aqui: o modelo monolítico SVR teve o menor desempenho de MAE para a série Nasdaq - última linha da Tabela 13b, porém este modelo teve seu desempenho elevado à melhor previsão geral dos modelos combinados, quando em combinação com o Perceptron monolítico. O segundo caso interessante dos resultados é o Perceptron, que apesar de não ter apresentado correlação - segundo a função de autocorrelação - para a série Dow Jones ainda obteve ganho percentual dentre as 3 séries, quando combinado com a SVR.

Tabela 11 – Ganho percentual relativo ao MAE do método proposto em relação aos modelos monolíticos.

| Dow Jones | | | Nasdaq | | | S&P 500 | | |
|---|---|---|---|---|---|---|---|---|
| **Modelo 1** | **Modelo 2** | **Ganho Percentual** | **Modelo 1** | **Modelo 2** | **Ganho Percentual** | **Modelo 1** | **Modelo 2** | **Ganho Percentual** |
| SVR | Perceptron | 95,695% | SVR | Perceptron | 96,031% | SVR | Perceptron | 96,389% |
| SVR | MLP | 90,654% | SVR | MLP | 90,176% | SVR | MLP | 90,874% |
| MLP | Perceptron | 50,203% | MLP | Perceptron | 55,001% | MLP | Perceptron | 50,605% |
| SVR | SVR | 48,058% | SVR | SVR | 33,999% | SVR | SVR | 41,99% |
| Perceptron | SVR | 1,989% | Perceptron | SVR | 2,466% | Perceptron | SVR | 1,632% |
| MLP | SVR | -2,332% | MLP | SVR | -3,552% | MLP | SVR | -2,994% |
| Perceptron | MLP | -38,646% | Perceptron | MLP | -20,15% | Perceptron | MLP | -22,572% |
| MLP | MLP | -41,547% | MLP | MLP | -27,478% | MLP | MLP | -34,112% |
| Perceptron | Perceptron | -92,743% | Perceptron | Perceptron | -90,782% | Perceptron | Perceptron | -110,95% |

**Fonte:** Elaborada pela autora (2020)

## 5.3 COMPARAÇÃO COM UMA ESTRATÉGIA DE ABORDAGEM ATIVA

Uma das referências citadas nos trabalhos relacionados é uma abordagem de PSO para previsão de séries temporais (IDPSO) (OLIVEIRA et al., 2017). Sendo este um dos poucos trabalhos recentes que realiza uma comparação com o estado da arte das abordagens ativa de previsão online na presença de mudança de conceito e que utilizada as mesma séries do que este trabalho, será realizada uma comparacão entre os resultados. O conjunto de teste da abordagem ativa possui uma perda de 300 valores da parte inicial da série original, utilizados para treinar o modelo inicial, já o método proposto é reduzido de apenas dois valores iniciais utilizados para o mesmo fim. Dessa forma, o conjunto de teste do método proposto é maior em quase 300 pontos em relação à abordagem IDPSO. Na Tabela 12 estão os dois melhores resultados da abordagem original do IDPSO (OLIVEIRA et al., 2017) e os três melhores resultados do método proposto neste trabalho.

Tabela 12 – Desempenho das abordagens de previsão ativa e dos 3 melhores resultados do método proposto através da métrica MAE e desvio padrão para as 3 séries propostas. Compilação feita através da média aritmética de 30 execuções.

| **Série** | **Estratégia de Abordagem Ativa** | | **Previsão do Método Proposto** $\hat{Z}_t + \hat{E}_t$ | | |
|---|---|---|---|---|---|
| **Temporal** | IDPSO-ELM-S | IDPSO-ELM-B | Perc. & SVR | SVR & Perc | MLP & Perc |
| Dow Jones | 0,0069 (2,76e-03) | 0,0065 (2,87e-03) | 0,0024 (1,15e-04) | 0,0026 (3,09e-04) | 0,0030 (2,59e-04) |
| Nasdaq | 0,0058 (1,51e-03) | 0,0057 (2,73e-03) | 0,0021 (1,75e-04) | 0,0020 (2,16e-04) | 0,0022 (9,80e-05) |
| S&P500 | 0,0046 (2,62e-03) | 0,0053 (2,41e-03) | 0,0017 (1,84e-04) | 0,0014 (1,46e-04) | 0,0017 (1,33e-04) |

**Fonte:** Elaborada pela autora (2020)

Foi aplicado o teste de Friedman, onde a hipótese nula é que não existe evidência significativa de que os modelos possuem desempenhos diferentes em relação ao MAE, e com nível de significância de 5%, a hipótese nula foi rejeitada com p-valor sendo 0,007. Na Tabela

13, são apresentados os resultados do teste estatístico Nemenyi, com apenas p-valor entre a abordagem IDPSO-ELM-S e a MLP & Perceptron apresentando uma diferença significativa com p-valor 0,07, nível de significância um pouco acima de 5%.

Tabela 13 – Comparação múltipla entre método proposto e a abordagem de aprendizado ativo IDPSO-ELM, representado pelo p-valor resultante do teste Nemenyi apenas para as 3 melhores abordagens do método proposto.

| **Teste de Nemenyi** | | | |
|---|---|---|---|
| | Método Proposto | | |
| | Perc. & SVR | SVR &Perc. | MLP & Perc. |
| IDPSO-ELM-S | 0,1814 | 0,5994 | 0,0738 |
| IDPSO-ELM-B | 0,2979 | 0,7459 | 0,1370 |

**Fonte:** Elaborada pela autora (2020)

Através dos resultados acima, mais uma vez, é possível considerar a possibilidade da correção do erro em ambientes online para reduzir os impactos de uma previsão sem ajuste de parâmetros - terceira pergunta de pesquisa. O argumento para essa afirmação se baseia no aproveitamento de quase todos os valores da série para a realização da previsão - com a exceção dos dois primeiros - em contraponto com o uso de uma parcela significativa da série utilizada para o ajuste dos parâmetros do PSO. Nenhum uso de memória pelo método proposto, em comparação com uso de memória para o treinamento inicial e retreinamento quando uma mudança de conceito é identificada. Por fim, é possível ainda destacar a ausência de qualquer de tempo de ajuste até a execução do método proposto, além de uma menor variação de resultados dentre as execuções.

# 6 CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho trouxe o conceito de combinação de previsões e de uso do resíduo para corrigir a previsão original de séries temporais - já conhecido - para o contexto de ambientes online e na presença de mudança de conceito. O resultado obtido foi relevante, mesmo diante do teste de Nemenyi não ter apontado divergências tão significantes diante dos diversos resultados do método proposto. A relevância do método se dá pelas suas vantagens e diante dos métodos que são praticado. O uso de modelos incrementais, por exemplo, faz com que não seja necessário o uso de memória para armazenar os dados, como nos treinamentos em batch ou que utilizam janelas de previsão (KRAWCZYK et al., 2017). O modelos que se utilizam de janela de previsão também precisam estimar qual o tamanho utilizar, enquanto o método proposto treina acrescentando um exemplo de cada vez. Outra vantagem do uso do modelo incremental é que são necessários apenas dois valores da série para iniciar a previsão, enquanto outras abordagens, como o PSO e modelos ajustados, precisa de uma quantidade significativa de dados para poder fazer o treinamento e ajuste dos pesos necessarios aos modelos envolvidos. Ao utilizar modelos monolíticos - sem ajustes de parâmetros - o tempo de preparação termina sendo baixo, tempo utilizado no máximo para determinar o modelo que se quer utilizar. Enquanto modelos ajustados precisam de tempo escolher os parâmetros, determinar uma quantidade de dados que seja significativa para representar as características da série e, por fim, executar o *gridsearch*.

Um outro ponto que vale ressaltar, é que o método proposto esteve em desvantagem em relação aos modelos no qual ele teve seus resultados comparados. Acontece que o IDPSO-ELM utiliza de 300 valores e os modelos ajustados utilizam, no mínimo, 2000 valores apenas para ajustar os parâmetros utilizados na abordagem, enquanto que o método proposto utiliza apenas dois valores inicias da série que se está prevendo.

A simplicidade do método proposto foi bastante considerada quando proposta, devido às aplicações onde normalmente são utilizadas. A rapidez, simplicidade e adaptabilidade do método permite combinações que podem ser aperfeiçoadas caso a caso, diante da existência de poucos ajustes para ser utilizada.

## 6.1 QUESTÕES DE PESQUISA

Ao voltar as questões de pesquisa que iniciaram o trabalho podemos concluir de cada umas delas:

- É possível modelar a série de resíduos proveniente da previsão de um método online? Na seção 5.1 os resultados dos testes foram analisados e seus resultados indicaram evidências significativas da existência de informações remanecentes nos resíduos. Esses testes podem, também, ser apoiados pelos resultados de ganho percentual, onde os resíduos dos três modelos monolíticos foram aprimorados por outros modelos também monolíticos.

- O uso da previsão dos resíduos por modelos online podem ser usados para melhorar as estimativas das séries temporais em ambientes online? Contando com melhorias de até 96,4%, e mais de 50% das combinações de modelos tendo resultados positivos, a Tabela 11 detalha a diversidade de modelos que puderam ter os resíduos aprimorados bem como seus respectivos percentuais. Inclusive, o teste de Nemenyi ressalta exatamente a melhoria do desempenho dentre o modelo monolítico da SVR com a sua combinação com o Perceptron.

- A correção do erro em ambientes online pode ser considerada uma opção para reduzir os impactos de uma previsão sem ajuste de parâmetros? Diante dos dois casos apresentados neste trabalho, um *gridsearch* e um modelo considerado estado da arte, o método proposto apresenta vantagens concretas. Essas vantagens contam com aspectos que vão desde a simplicidade do método, aproveitamento de dados, menor tempo total de preparação e execução da estratégia, uso mínimo de memória, além de uma menor variação nos resultados da abordagem.

## 6.2 TRABALHOS FUTUROS

- Diante dos diferentes protocolos experimentais e dos resultados do método proposto, é válido ampliar e normatizar os estudos comparativos, afim ampliar a compreensão quanto ao real ganho do método proposto em relação aos outros métodos de previsão.

- No presente trabalho a estratégia de combinação das previsões utilizadas foi a soma, com o objetivo estudar o comportamento da correção do erro em um novo contexto de maneira mais simples. Porém é possível que existam outras forma de combinação que sejam mais adequadas.

- Ao que tange as séries de resíduos, o método proposto utiliza de modelos clássicos adaptados para o aprendizado incremental, e com este método consegue melhorar a previsão um modelo monolítico. Diante disso, outras estratégias de previsão podem ser exploradas, inclusive alguma com enfoque na característica ruidosa das séries de resíduo, e que possam vir a ser mais vantajosas.

# REFERÊNCIAS


BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. ISBN 0387310738.

BOX, G. E. P.; JENKINS, G. *Time Series Analysis, Forecasting and Control*. San Francisco, CA, USA: Holden-Day, Inc., 2008. ISBN 0816211043.

BROCKWELL, P. J.; DAVIS, R. A. *Introduction to Time Series and Forecasting*. 2nd. ed. Springer, 2002. Hardcover. ISBN 0387953515. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0387953515>.

CFI. *NASDAQ Composite - Components, Methodology & Criteria for Inclusion*. 2020. Disponível em: <https://corporatefinanceinstitute.com/resources/knowledge/trading-investing/nasdaq-composite/>.

CHATFIELD, C. What is the 'best' method of forecasting? *Journal of Applied Statistics*, v. 15, p. 19–39, 1988.

COWPERTWAIT, A. V. M. P. S. Introductory time series with r. In: _____. [S.l.: s.n.], 2009. ISBN 0387886974, 9780387886978.

COWPERTWAIT, A. V. M. P. S. *Introductory Time Series with R*. 1st. ed. Springer-Verlag New York, 2009. Hardcover. ISBN 978-0-387-88698-5. Disponível em: <https://www.springer.com/gp/book/9780387886978#reviews>.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, JMLR.org, v. 7, p. 1–30, dez. 2006. ISSN 1532-4435.

DITZLER, G.; ROVERI, M.; ALIPPI, C.; POLIKAR, R. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, v. 10, n. 4, p. 12–25, Nov 2015. ISSN 1556-603X.

FIRMINO, P. R. A.; NETO, P. S. de M.; FERREIRA, T. A. Correcting and combining time series forecasters. *Neural Networks*, v. 50, p. 1 – 11, 2014. ISSN 0893-6080. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0893608013002517>.

FIRMINO, P. S. d. M. N. P. R. A.; FERREIRA, T. A. Error modeling approach to improve time series forecasters. *Neurocomputing*, v. 153, p. 242 – 254, 2015. ISSN 0925-2312. Disponível em: <http://www.sciencedirect.com/science/article/pii/S092523121401563X>.

GAMA, J.; ŽLIOBAITĖ, I.; BIFET, A.; PECHENIZKIY, M.; BOUCHACHIA, A. A survey on concept drift adaptation. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 46, n. 4, p. 44:1–44:37, mar. 2014. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/2523813>.

GANTI, A. *Dow Jones Industrial Average - DJIA*. Investopedia, 2020. Disponível em: <https://www.investopedia.com/terms/d/djia.asp>.

HARRIES, M.; HORN, K.; SAMMUT, C. Learning in time ordered domains with hidden changes in context. *AAAI Technical Report WS-98-07*, May 1998.

HAYKIN, S. S. *Neural networks and learning machines*. Third. Upper Saddle River, NJ: Pearson Education, 2009.

HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: theory and applications. *Neurocomputing*, Elsevier, v. 70, n. 1-3, p. 489–501, 2006.

HYNDMAN, R.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. OTexts: Melbourne, Australia., 2018. Disponível em: <https://otexts.com/fpp2/>.

JENKINS, G. Some practical aspects of forecasting in organisations. *J. Forecasting*, v. 1, p. 3–21, 1982.

KASABOV, N. *Evolving Connectionist Systems, Methods and Applications in Bioinformatics, Brain Study and Intelligent Machines*. 1st. ed. Springer-Verlag London, 2003. Hardcover. ISBN 978-1-4471-3740-5. Disponível em: <https://www.springer.com/gp/book/9781447137405>.

KENNON, J. *What to Know About the NASDAQ, the World's Second-Largest Stock Market*. The Balance, 2019. Disponível em: <https://www.thebalance.com/what-is-the-nasdaq-356343>.

KOLTER, J. Z.; MALOOF, M. A. Using additive expert ensembles to cope with concept drift. In: *Proceedings of the 22Nd International Conference on Machine Learning*. New York, NY, USA: ACM, 2005. (ICML '05), p. 449–456. ISBN 1-59593-180-5. Disponível em: <http://doi.acm.org/10.1145/1102351.1102408>.

KRAWCZYK, B.; MINKU, L. L.; GAMA, J.; STEFANOWSKI, J.; WOźNIAK, M. Ensemble learning for data stream analysis: A survey. *Information Fusion*, v. 37, p. 132 – 156, 2017. ISSN 1566-2535. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1566253516302329>.

Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, v. 31, n. 12, p. 2346–2363, 2019.

MORETTIN, C. M. C. T. P. A. *Análise de Séries Temporais.* 1st. ed. Blucher, 2004. Hardcover. ISBN 8521203896. Disponível em: <https://www.blucher.com.br/livro/detalhes/analise-de-series-temporais-147>.

NASDAQ, S. *National Association of Securities Dealers Automatic Quotation System (Nasdaq) Definition*. 2020. Disponível em: <https://www.nasdaq.com/glossary/n/national-association-of-securities-dealers-automatic-quotation-system>.

NEMENYI, P. Distribution-free multiple comparisons. In: *PhD thesis, Princeton University.* [S.l.: s.n.], 1963.

NETO, P. S. de M.; FERREIRA, T. A.; LIMA, A. R.; VASCONCELOS, G. C.; CAVALCANTI, G. D. A perturbative approach for enhancing the performance of time series forecasting. *Neural Networks*, Elsevier, v. 88, p. 114–124, 2017.

OLIVEIRA, G. H. F. M.; CAVALCANTE, R. C.; CABRAL, G. G.; MINKU, L. L.; OLIVEIRA, A. L. I. Time series forecasting in the presence of concept drift: A pso-based approach. In: *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*. [S.l.: s.n.], 2017. p. 239–246. ISSN 2375-0197.

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.; DUCHESNAY, E.

Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, p. 386–408, 1958. ISSN 0033-295X. Disponível em: <http://dx.doi.org/10.1037/h0042519>.

SCHLIMMER, J. C.; GRANGER, R. H. Beyond incremental processing - tracking concept drift. *AAAI*, p. 502–507, 1986. Disponível em: <https://www.aaai.org/Papers/AAAI/1986/AAAI86-084.pdf>.

SCHLIMMER, J. C.; GRANGER, R. H. *Incremental Learning from Noisy Data*. [S.l.]: Kluwer Academic Publishers, 1986. 317-354 p.

SHESKIN, D. *Handbook of parametric and nonparametric statistical procedures*. Florida, United States: CHAPMAN & HALL/CRC, 2000. ISBN N I-58488-133-X.

SILVA, E. G.; JUNIOR, D. S. de O.; CAVALCANTI, G. D. C.; NETO, P. S. G. de M. Improving the accuracy of intelligent forecasting models using the perturbation theory. In: *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2018. p. 1–7. ISSN 2161-4407.

SMOLA A. J.; SCHÖLKOPF, B. A. A tutorial on support vector regression. *Statistics and computing*, v. 14, p. 199–222, 2004.

S&P-DOW-JONES. *Our Company - S&P Dow Jones Indices*. 2020. Disponível em: <https://us.spindices.com/our-company/our-history/>.

S&P-GLOBAL. *Our History, S&P Global*. 2020. Disponível em: <https://www.spglobal.com/en/who-we-are/our-history>.

VAPNIK, V. *The Nature of Statistical Learning Theory.* [S.l.]: Springer, New York, 1995.

VAPNIK, V.; LERNER, A. Pattern recognition using generalized portrait method. *Automation and Remote Control*, v. 24, p. 774–780, 1963.

WANG, J.; WANG, J. Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks. *Neurocomputing*, v. 156, 05 2015.

WIDMER, G.; KUBAT, M. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, v. 23, n. 1, p. 69–101, Apr 1996. ISSN 1573-0565. Disponível em: <https://doi.org/10.1007/BF00116900>.

ZHANG, P. Zhang, g.p.: Time series forecasting using a hybrid arima and neural network model. neurocomputing 50, 159-175. *Neurocomputing*, v. 50, p. 159–175, 01 2003.

ZHANG, Y.; XIONG, X.; ZHANG, Q. An improved self-adaptive pso algorithm with detection function for multimodal function optimization problems. *Mathematical Problems in Engineering*, Hindawi, v. 2013, 2013.

# APÊNDICE A – PSEUDO-CÓDIGO ADDITIVE EXPERTS PARA DADOS CONTÍNUOS

**Parâmetros:**

$\{x, y\}^T \in [0, 1]$ Dados de treinamento

$\gamma \in [0, 1]$ Peso de um novo expert

$\beta \in [0, 1]$ Fator de penalização dos pesos

$\tau \in [0, 1]$ Limiar de inclusão de um novo expert

**Inicializações:**

Inicializa o número de experts para 1, $N_1 = 1$.

Inicializa o peso do 1º expert para 1, $w_{1,1} = 1$.

**For** $t = 1, 2, ..., T$:

**1.** Calcula a previsão dos experts $\xi_{t,1}, ..., \xi_{t,Nt} \in [0, 1]$

**2.** Obtém a previsão através da média ponderada das previsões:

$$\hat{y}_t = \frac{\sum_{i=1}^{N^t} w_{t,i}\xi_{t,i}}{\sum_{i=1}^{N^t} w_{t,i}} \tag{A.1}$$

**3.** Calcula o erro $|\hat{y}_t - y_t|$

**4.** Atualiza os pesos dos experts:

$$w_{t+1,i} = w_{t,i}\beta^{|\xi_{t,i} - y_t|} \tag{A.2}$$

**5.** Se $|\hat{y}_t - y_t| \geq \tau$ adiciona um novo expert:

$$N_{t+1} = N_t + 1 \tag{A.3}$$

$$w_{t+1}, N_{t+1} = \gamma \sum_{i=1}^{N_t} w_{t,i}|\xi_{t,i} - y_t|. \tag{A.4}$$

**6.** Treina cada expert com o exemplo de entrada $x_t, y_t$

# APÊNDICE B – LISTA DOS PARÂMETROS UTILIZADOS

## B.1 PARÂMETROS DO SCI-KIT LEARN

Para fins de reprodutibilidade, este apêndice concede todos os parâmetros do Sci-Kit Learn que foram utilizados. A versão do Sci-Kit Learn utilizada foi a 0.23.1 e as classes utilizadas desta biblioteca foram: *SGDRegressor*, que pertence ao módulo de modelos lineares, para simular a SVR linear; *MLPRegressor*, que pertence ao módulo de redes neurais, para simular a MLP para regressão.

Os modelos foram utilizados sem modificação de parâmetro diante da versão 0.23.1. Mas, caso esta lista de parâmetros fique indisponível ou outra biblioteca seja utilizada, são disponibilizadas aqui os parâmetros incluso no modelos, na versão citada.

– **Modelo:**

  MLP Online

– **Classe:**

  sklearn.neural_network.MLPRegressor

– **Parâmetros:**

  hidden_layer_sizes=(100, ), activation='relu', solver='adam', alpha=0.0001,
  batch_size='auto', learning_rate='constant', learning rate init=0.001,
  power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001,
  verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True,
  early_stopping=False, validation_fraction=0.1, beta_1=0.9,
  beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000

– **Método para treinamento online:**

  partial_fit()

– **Janela:**

  1

– **Modelo:**

  SVR Linear Online

– **Classe:**

  sklearn.linear_model.SGDRegressor

– **Parâmetros:**

loss='squared_loss', penalty='l2', alpha=0.0001, l1_ratio=0.15, fit_intercept=True, max_iter=1000, tol=0.001, shuffle=True, verbose=0, epsilon=0.1, random_state=None, learning_rate='invscaling', eta0=0.01, power_t=0.25, early_stopping=False, validation_fraction=0.1, n_iter_no_change=5, warm_start=False, average=False

– **Método para treinamento online:**

partial_fit()

– **Janela:**

1

## B.2 PARÂMETROS DO PERCEPTRON

O Perceptron para regressão online não é disponibilizado pela biblioteca acima, portanto a implementação foi própria.

– **Modelo:**

Perceptron Online

– **Parâmetros iniciais:**

learning_rate=1, bias=randômico [0,1], weight=randômico [0,1]

– **Método para treinamento online:**

partial_fit()

– **Janela:**

1