

CP-WTG Analytics: Uma ferramenta para análise e detecção de pontos de mudança em sinais de dados de turbinas eólicas

Caio D. Gilo* Marcos V. A. Costa** Luiz A. M. Lima***
Leiva C. Oliveira*,**

* Departamento de Computação
** Programa de Pós-graduação em Engenharia Elétrica
Universidade Federal Rural do Semi-Árido, Mossoró-RN, Brasil
*** Voltalia Energia do Brasil, Rio de Janeiro-RJ, Brasil
(e-mails: [caio.gilo, marcos.costa13388]@alunos.ufersa.edu.br,
l.lima@voltalia.com, leiva.casemiro@ufersa.edu.br)

Abstract: The wind turbine generator components' monitor task is useful to detect non-normal operating behavior, and enhances the operation and maintenance processes. The change points indicates when an unusual patterns in the sensor responses time series happened. This work presents a tool for detection and analysis change points in measurement data generated by wind turbine sensors. With simple and intuitive interface, containing the main functionalities for selection, configuration, adjustments, display and storage of the results for the detected change points, and thus facilitates communication between the different professionals involved in the operation. Experiments were carried out from datas of a wind turbine that generated alarm of possible problems in generator. The detected change points indicated non-normal operation behavior with 60 days in advance, compared to generated alarm. Furthermore, different configurations and adjustments were analyzed to demonstrate the tool feasibility.

Resumo: O monitoramento dos componentes de uma turbina eólica possibilita a detecção de comportamentos não-normal de funcionamento, auxiliando nos processos de operação e manutenção. Os pontos de mudança indicam os instantes em que padrões incomuns nas respostas dos sensores ocorrem. Esse trabalho apresenta uma ferramenta para detecção e análise de pontos de mudança em dados de medição gerados por sensores de turbinas. Possui interface simples e intuitiva contendo as principais funcionalidades para seleção, configuração, ajustes, exibição e armazenamento dos resultados para os pontos de mudanças detectados; facilitando a comunicação entre os diferentes profissionais envolvidos na operação. Experimentos foram realizados com dados de uma turbina que gerou alarme de possíveis problemas em seu gerador. Os pontos de mudança detectados indicaram a existência de comportamento não-normal de operação com mais de 60 dias de antecedência, comparado ao alarme gerado. Ademais, diferentes configurações e ajustes foram analisados para demonstrar a aplicabilidade da ferramenta.

Keywords: Wind Turbine Generator; Change Points Detection; Python; Ruptures; Time Series; Prediction.

Palavras-chaves: Turbina eólica; Ponto de Mudança; Python; Ruptures; Série temporal; Predição.

1. INTRODUÇÃO

E A utilização de energia eólica teve um aumento expressivo nos últimos anos, atingindo uma capacidade de geração na ordem de 837 GW segundo os dados da GWEC (2023). A energia eólica desempenha um papel importante na agenda global de descarbonização e vêm alcançando elevados patamares de investimento, com estimativas de quadruplicar o número de novas instalações anuais de turbinas eólicas (aerogeradores ou WTG, sigla em inglês para *Wind Turbine Generator*) até o final da década (GWEC (2023)).

A busca por crescimento e investimento de forma sustentável do setor perpassa por processos eficientes de operação e manutenção (O&M), os quais estima-se requisitem um quarto das despesas totais para a produção de energia (IRENA (2022)). Estes valores podem ser ainda maiores a depender da localização ou do clima da área onde a WTG está alocada.

O monitoramento dos componentes de uma WTG através da análise dos dados obtidos dos seus inúmeros sensores possibilita verificar situações de mau funcionamento, anomalias e sazonalidades, e têm sido largamente empregado como estratégia de O&M para redução de custos (Dao and et al. (2019); Dao (2022)). A observação das séries

de dados temporais dos sinais referentes a temperatura, pressão, óleo, angulação, velocidade e potência presentes no Sistema de Supervisão e Aquisição de Dados (SCADA, sigla em inglês para *Supervisory Control And Data Acquisition*) de uma turbina eólica, permite identificar mudanças incomuns nos padrões de resposta dos sensores, isto é, que fogem do seu comportamento normal de funcionamento. Tais situações podem indicar uma falha vigente, ou que está prestes a ocorrer naquele determinado setor. O instante da mudança abrupta do sinal é denominado de ponto de mudança (CP, sigla em inglês para *Change Points*).

Muitos algoritmos e métodos podem ser encontrados na literatura para detecção de pontos de mudanças (CPD, sigla em inglês para *Change Points Detection*) em séries temporais (Aminikhanghahi and Cook (2017); Basseville and et al. (1993)). A aplicação de métodos de CPD para monitoramento das condições normais de uma turbina eólica a partir de dados SCADA (Letz Gus (2020); Xu and et al. (2014)), detecção de rupturas em sinais de componentes específicas (Yuan and et (2019)) ou para auxiliar na limpeza de dados utilizados em modelos preditivos (Quanlin and et al. (2020)) podem ser encontrados na literatura.

Há disponibilizados em pacotes de software, algoritmos de CPD implementados em linguagem R¹ e Python² (Killick and et al. (2014)), permitindo a configuração e ajustes dos hiperparâmetros. Entretanto, os pacotes não possuem padronização comum, alguns computam apenas parte do algoritmo de CPD, são otimizados para problemas específicos e/ou não possuem comunidade para suporte.

Apesar da existência destes pacotes, faz-se necessário uma ferramenta com interface amigável, sem necessidade de escrita de código de programação, para facilitar a configuração dos parâmetros necessários em cada parte do algoritmo de CPD, e assim, otimizar seu cálculo/análise e permitir a comunicação entre os diferentes profissionais envolvidos nos processos de O&M.

Nesse sentido, este trabalho tem por objetivo principal o desenvolvimento de uma ferramenta para análise e detecção de pontos de mudanças em sinais de turbinas eólicas. Com uma interface simples e intuitiva, possui funcionalidades para seleção dos dados, configuração e ajustes de parâmetros, exibição gráfica e armazenamento dos resultados para os pontos de mudanças detectados.

2. ASPECTOS TEÓRICOS SOBRE CPD

A teoria sobre detecção de pontos de mudança é bem estabelecida na literatura; trabalhos que apresentam os fundamentos teóricos sobre os algoritmos de CPD, suas características, aplicações e implementações podem ser encontrados nas referências: Truong and et al. (2020); Aminikhanghahi and Cook (2017); Csörgö and et al. (1997) e Basseville and et al. (1993). Brevemente, a detecção de pontos de mudança consiste em determinar os instantes de variação abrupta que surgem na distribuição de uma sequência de dados específica. Assim, em um conjunto de dados é possível detectar os pontos em que as características principais mudam abruptamente a partir de métricas

¹ Pacotes R: **kcpRS**, **bcp**, **changepoint** e **breakfast**.

² Python: Prophet, Ruptures (2023), Luminaire e scikit-multiflow.

como variância, média, desvio padrão, frequência e padrão (Aminikhanghahi and Cook (2017); Truong and et al. (2020)).

Os algoritmos de CPD dividem-se em dois grandes grupos i) *online*: visam detectar mudanças assim que ocorrem em um ambiente de tempo real e ii) *offline*: detectam mudanças retrospectivamente quando todas as amostras são recebidas, ou seja, realizam a segmentação do sinal. Este trabalho aborda a utilização de métodos *offline*.

Dada uma série temporal, o algoritmo de detecção de CPs *offline* consiste em minimizar uma função contraste da forma $\min_t V(t, y) + pen(t)$, sendo que $V(\cdot)$ pode ser expressa como uma soma total dos custos referente a segmentação:

$$V(t, y) = c([y_t]_1^t) + c([y_t]_{t_1+1}^{t_2}) + \dots + c([y_t]_{t_i+1}^{t_{i+1}}) \quad (1)$$

em que y é o sinal de entrada, $t = [t_1, t_2, \dots]$ o conjunto de pontos de mudanças, $c(\cdot)$ a função custo e $pen(t)$ é um regularizador/penalidade no valor da partição t . O algoritmo CPD divide a série temporal em segmentos a partir da detecção dos pontos em que os valores possuem uma divergência considerável com os demais segmentos.

Tipicamente, o cálculo dos pontos de mudança pode ser dividido em três partes principais (Truong and et al. (2020)): i) funções de custo, ii) métodos de busca e iii) restrições. A corretude na computação destas partes é primordial para uma boa predição/segmentação visto que impactam tanto na complexidade quanto na sensibilidade que o algoritmo de CPD terá para considerar rupturas nos dados como pontos de mudança ou não.

2.1 Funções de Custo

A função de custo expressa a homogeneidade dos dados da série temporal, considerando dados heterogêneos aqueles que possuem mais pontos de discrepância evidentes. A escolha da função de custo é arbitrária e irá depender da finalidade e de qual tipo de ponto de mudança é desejado, isto é, mudanças em valores estatísticos (como média, mediana ou ambos) ou em valores não-paramétricos (como mudança de distribuição). Em geral, retornam valores “baixos” para um segmento homogêneo e valores “alto” para um segmento heterogêneo (Truong and et al. (2020)).

Existem algumas funções de custo que podem ser utilizadas, entre as quais destacam-se:

- Função de custo L2: Esta função de custo toma como base o cálculo dos desvios médios de um determinado sinal. Sendo $\{y_t\}_T$ o sinal de entrada em um intervalo $T = [t_1, t_n]$, o custo L2 é dado por:

$$c(y_t) = \sum_{t \in T} |y_t - \bar{y}|^2 \quad (2)$$

com \bar{y} o valor médio do intervalo $\{y_t\}_{t \in T}$;

- Função de custo Linear: utiliza um modelo de regressão linear múltipla para considerar $y_{t,e}$, e as estimativas para a ruptura são computadas minimizando a soma do quadrado dos resíduos (Truong and et al. (2020)).
- Função de custo baseadas em *Kernel*: faz uso de funções *Kernel*, mapeamento os dados do espaço de entrada para um outro espaço de características em que métricas de similaridade ou distância entre dois pontos podem ser computadas aplicando o *Kernel*

(Φ). A projecção do sinal de entrada para $\{\Phi(y_t)\}_T$ em um intervalo $T = [t_1, t_n]$, é então utilizada para o cálculo do custo:

$$c(y_t) = \sum_{t \in T} |\Phi(y_t) - \bar{y}|^2 \quad (3)$$

com \bar{y} o valor médio do intervalo $\{\Phi(y_t)\}_{t \in T}$. As funções Φ baseadas em grafos, polinômios e funções de base radial são comumente empregadas.

2.2 Métodos de busca

Os métodos de busca referem-se ao procedimento de resolução para o problema de detecção de pontos de mudança, tanto quando sabe-se previamente a quantidade de rupturas presentes no sinal, bem como quando esse número é desconhecido e restrições/penalidades são utilizadas para auxiliar na resolução do problema.

Tipicamente, a detecção de CPs para segmentação *off-line* de séries temporais vale-se de métodos de busca com soluções ótimas ou aproximadas. Os algoritmos de CPD com os métodos de Programação dinâmica (*Dynp*) e Segmentação penalizada linearmente (*Pelt*) buscam resolver o problema de forma ótima, enquanto a Segmentação binária (*Binseg*), Segmentação de baixo para cima (*BottomUp*) e o método da Janela deslizante (*Window*) resolvem de forma aproximada. Cada método atinge um equilíbrio entre complexidade computacional e precisão. A seleção do método a ser utilizado vai depender do objetivo/aplicação. A seguir, são apresentados detalhes resumidos sobre cada método.

Programação dinâmica (Dynp): calcula o valor mínimo exato das somas dos custos com base nos conceitos da programação dinâmica. O problema geral é dividido em subproblemas e, com um auxílio de uma memória auxiliar, armazena valores de operações que possam a ser realizadas novamente, garantindo uma maior eficiência e evitando cálculos desnecessários. Tem sua complexidade algorítmica de ordem $O(CKn^2)$, sendo K o número de pontos de mudança, C a complexidade da função de custo utilizada e n o número de amostras (Truong and et al. (2020)).

Segmentação penalizada linearmente (Pelt): também busca calcular o valor mínimo exato das somas dos custos através de regras de poda que o auxiliam a descartar partições menos relevantes, diminuindo assim o custo computacional na busca por uma segmentação otimizada do sinal. Em geral, utiliza-se como entrada uma subamostra do sinal, e configura-se a distância mínima e granularidade entre pontos de mudança (Killick and et al. (2012); Ruptures (2023)). Apresenta complexidade algorítmica média de $O(CKn)$.

Segmentação binária (Binseg): o sinal completo é analisado para detectar o primeiro ponto de ruptura; em seguida, a série é dividida ao meio a partir desse ponto e a detecção é realizada nos subsegmentos replicando o método de detecção recursivamente. O *Binseg* possui baixa complexidade computacional, de $O(Cn \log n)$ (Fryzlewicz (2014)).

Segmentação de baixo para cima (BottomUp): parte do princípio de que o sinal encontra-se, inicialmente, totalmente segmentando e a partir da semelhança entre seções, pontos de mudanças menos significativos são removidos mesclando as seções adjacente. O algoritmo fina-

liza quando só houverem seções com discrepâncias entre si. Possui complexidade logarítmica, dada por $O(n \log n)$ (Fryzlewicz (2007)).

Janela deslizante (Window): utiliza como base dois segmentos do sinal (denominados de janelas) que se deslocam ao longo do fluxo de dados. Dentro de cada janela é calculada uma medida de discrepância que é tomada como base para a detecção de pontos de mudança. Na situação onde as duas janelas estão sobre o mesmo segmento, a discrepância entre elas é irrelevante, no entanto, quando caem em segmentos distintos, a discrepância é relevante indicando a possibilidade de um ponto de mudança entre elas. Esse método apresenta complexidade $O(nw)$, para o número de amostras n , com w sendo o número de índices possíveis onde as janelas poderão estar alocadas.

2.3 Restrições

A última parte envolvida na detecção de pontos de mudança é o cálculo de restrições ou penalidades, as quais referem-se aos métodos agregados ao algoritmo quando o número de pontos de mudança é inicialmente desconhecido. A escolha da restrição está relacionada com a amplitude de mudanças que deverão ser detectadas, isto é, com pequenas restrições inúmeros pontos de mudança serão considerados válidos; por outro lado, com restrições grandes apenas mudanças abruptas/significativas serão levadas em consideração.

Em termos gerais, as funções de penalidade baseiam-se no conceito de penalidade mínima, obtida com base em uma regressão entre a quantidade penalizada e a função de custo associada sem penalização. Como exemplo, listam-se as funções denominadas de pen_{slope} e pen_{cost} :

$$pen_{slope}(t) = \alpha \times \frac{1}{T} \left[-s_1 \log \left(\frac{T-1}{n-1} \right) - s_2 n \right] \quad (4)$$

$$pen_{cost}(t) = \alpha \times c(y_t) \quad (5)$$

com T sendo o número de pontos de dados; $n \in [0.6N_{MAX}, N_{MAX}]$ e N_{MAX} uma estimacão para o número de segmentos; s_1 e s_2 denotam os coeficientes para a regressão de $c(\cdot)$ sobre $\log \left(\frac{T-1}{n-1} \right)$ e sobre $\frac{n}{T}$, respectivamente (Birgé and et al. (2007)). Para mais detalhes sobre penalidades ver Truong and et al. (2020) e Arlot and et al. (2019).

3. A FERRAMENTA CP-WTG ANALYTICS

A Figura 1 apresenta a ferramenta denominada de CP-WTG Analytics. Possui interface simples e fácil, contendo um menu localizado no lado esquerdo para a escolha de todos os parâmetros utilizados na detecção de pontos de mudança. Também contém uma área gráfica para apresentação dos resultados obtidos. Opções de manipulação de cores, *zoom* e exportação dos dados estão disponíveis. A ferramenta foi desenvolvida utilizando a linguagem de programação Python em conjunto com a biblioteca de detecção de pontos de mudança Ruptures (2023). Tal biblioteca está publicamente disponível e implementa os algoritmos de CPDs apresentados.

A ordenação da escolha dos parâmetros segue o fluxo do algoritmo de detecção de CPs. Inicialmente, faz-se necessário a seleção da base de dados, o sinal a ser analisado contido nessa base e a granularidade da série de dados. Em

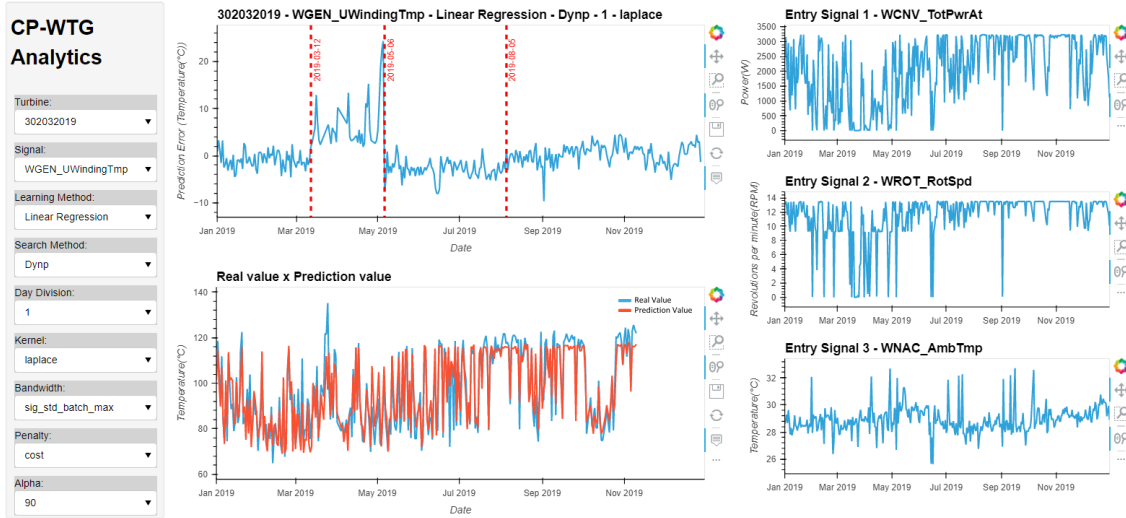


Figura 1. Interface da ferramenta CP-WTG Analytics. Menu à esquerda contendo as opções para seleção de parâmetros para o algoritmo de CPD; Área gráfica central apresentando o sinal analisado e CPs detectados (parte superior); e na parte inferior o gráfico do sinal real e do sinal predito (cor vermelha) pelo algoritmo; sub-gráficos a direita representando os sinais de entrada para predição (X_p , X_{rpm} , X_{Tamb} , ...).

seguida, é possível selecionar o método de busca, a função de custo, o tipo de penalidade e demais configurações para ajustar adequadamente o cálculo dos pontos de mudança. As denominações das opções disponíveis na ferramenta são listadas a seguir. Para maiores detalhes indicamos a documentação da biblioteca Ruptures (2023):

- *Turbine* → Base de dados a ser utilizada (arquivo .CSV contendo todos os sinais da turbina eólica);
- *Signal* → Série temporal com os dados do Sinal a ser analisado (sensor/setor da turbina);
- *Learning Method* → Método de aprendizado a ser utilizado para realizar a predição/normalização (“LR”, “RNA”);
- *Search Method* → Método de busca a ser utilizado (“Dynp”, “Pelt”, “Binseg”, “BottomUp”, “Window”);
- *Day Division* → Granularidade de dados. Expresso pela quantidade de pontos de dados por dia (“1”, “2”, “3”);
- *Kernel* → Função de *kernel* a ser utilizado pelo método de busca (“linear”, “laplace”, “gaussian”);
- *Bandwidth* → Largura de banda utilizada pela função *Kernel*. É baseado na heurística mediana, isto é, utiliza-se o inverso da mediana de todas as distâncias pareadas (“median”, “sig_std”, “sig_std_batch_max”);
- *Penalty* → Função de custo a ser utilizada (“cost”, “slope”);
- *Alpha* → Valor aplicado à penalidade.

O cálculo dos pontos de mudança é realizado na série de dados temporal referente ao erro calculado a partir da diferença do valor real do sinal (X) e do seu valor predito (y) ao longo de um determinado período de tempo: $err(t) = X(t) - y(t)$. Tal procedimento auxilia no processo de visualização e interpretação (Letzguis (2020)). A ferramenta possibilita a predição do sinal através dos métodos aprendizagem de máquina baseados em Rede Neural Artificial (RNA) e Regressão Linear (RL). Os sinais de entrada comumente utilizadas para predição são a potência ativa (X_p), velocidade de rotação (X_{rpm}) e temperatura no

interior da nacela (X_{Tamb}), uma vez que os dois primeiros estarem relacionadas ao estado de operação da turbina e o último por ser um bom excluyente de sazonalidade (Haraldsdóttir (2016)). O valor predito é então expresso pela combinação linear: $y(t) = \omega_1 X_p + \omega_2 X_{rpm} + \omega_3 X_{Tamb} + c$. Os regressores baseados em RNA e RL foram configurados com os valores padrão da biblioteca scikit-learn².

A série temporal normalizada do sinal é então utilizada e os (hiper)parâmetros selecionados e configurados na ferramenta são repassados à Ruptures para realizar a detecção dos CPs. Com os CPs calculados, a ferramenta exibe-os no gráfico como linhas verticais ortogonais ao sinal analisado, indicando os pontos de mudança detectados.

4. RESULTADOS

Para verificar o funcionamento da ferramenta, foram utilizados os dados referentes aos sinais de uma turbina eólica localizada no estado do Rio Grande do Norte, obtidos ao longo do ano de 2019. Esses sinais são referentes aos dados dos sensores presentes na turbina, coletados e armazenados a cada 10 minutos. Esta turbina gerou alarme de possíveis problemas em seu gerador no dia 24 de abril de 2019, tendo seu reparo finalizado e retorno aos estados normais de operação, em 05 de maio de 2019 (após 12 dias).

Foi utilizado para análise um sinal referente a temperatura de um dos sensores instalados na região ou componente que se deseja analisar o comportamento, nesse caso, o gerador. Dentre os sinais diretamente relacionados ao gerador, a turbina apresenta sensores para medição das temperaturas dos rolamentos, anel, refrigerador e enrolamentos. Assim, o sinal de um dos enrolamentos do gerador foi escolhido para detecção e análise dos pontos de mudança.

Processamento de dados

O processamento dos dados perpassa pela remoção de dados incompletos ou com valores inconsistentes; apenas a

² MLPRegressor() e LinearRegression(). Ver scikit-learn.org

inclusão de pontos de dados em que o status de funcionamento do sistema de controle da turbina indica seu pleno funcionamento (sinal *status* igual a 60); e a exclusão de dados em que a turbina opere com um valor de potência abaixo de 10% da potência nominal. Após a seleção e limpeza dos dados, ocorre a predição do sinal com um dos métodos escolhidos pelo usuário. Por fim, é realizado o processo de normalização entre os valores preditos da variável analisada e os valores reais, através do cálculo do erro. Os valores de erro são então aplicados ao algoritmo de CPD.

Análise e Detecção de Pontos de Mudança

A Figura 2 apresenta o resultado da detecção de CPs do enrolamento_U do gerador. Os gráficos obtidos são: a) o sinal normalizado e indicação onde ocorre o CPD (denominado de gráfico principal), b) os valores reais e preditos do sinal analisado. Os gráficos referentes aos sinais de entrada (X_p , X_{rpm} e X_{Tamb}) também são exibidos [ver sub-gráficos da Fig. 1]. Os gráficos são gerados em tempo real de acordo com os resultados obtidos.

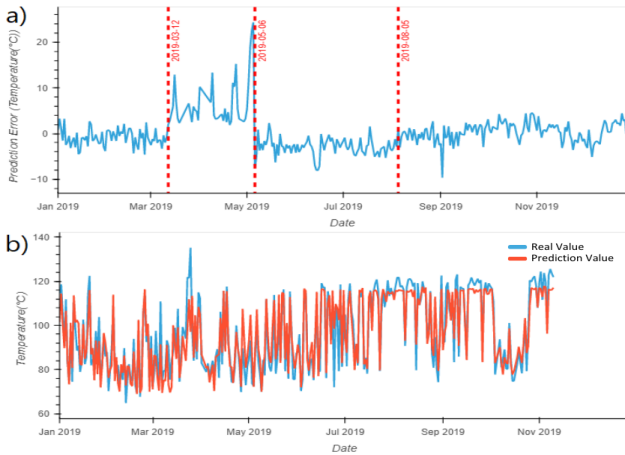


Figura 2. Resposta do algoritmo de CPD para o sinal do enrolamento_U do gerador. a) Sinal normalizado (erro) com indicação dos pontos de mudança em 12/03, 06/05 e 05/08. As linhas verticais representam os instantes dos CPs detectados. b) Sinal predito e sinal real (em °C) utilizados na normalização.

O gráfico principal representa o erro de predição gerado pelo algoritmo ao longo de um determinado período de tempo. O erro apresentado na Fig. 2a) foi obtido utilizando a regressão linear como método de predição. Os pontos de mudança encontrados são denotados por linhas verticais pontilhadas. A Tabela 1 resume as datas em que ocorreram os alarmes gerados pela turbina e os CPs. Como observado, o primeiro CP detectado indica a existência de comportamento não-normal de operação da turbina com mais de 60 dias de antecedência, em relação ao início do alarme gerado. O segundo CP detectado coincidiu com a data de finalização dos reparos da turbina. O terceiro CP indica mudança sazonal das condições ambientais do parque eólico.

Os CPs indicados na Fig. 2a) foram obtidos com a seguinte configuração: [*Search Method* → “Dynp”; *Day Division* → “1”; *Kernel* → “laplace”; *Bandwidth* → “sig_std_batch_max”; *Penalty* → “cost” e *Alpha* → “90”].

Tabela 1. Datas da ocorrência dos alarmes e dos CPs detectados pela ferramenta.

Alarme	CP encontrado
24/05/2019 (início)	12/03/2019
05/05/2019 (fim)	06/05/2019
	05/08/2019

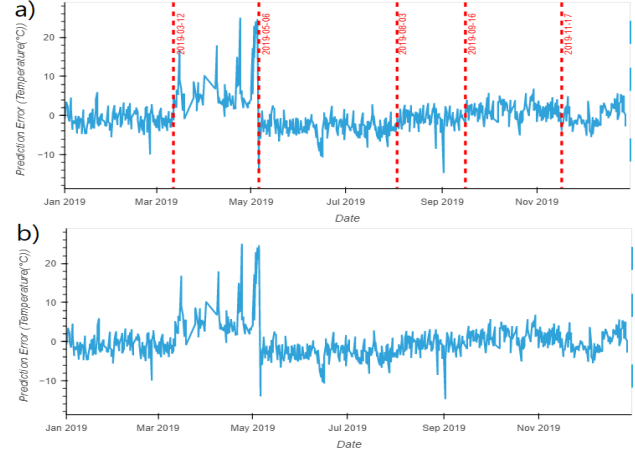


Figura 3. Detecção CPs para o sinal do enrolamento_U do gerador utilizando diferentes conjuntos de parâmetros: a) *Search Method* → “Window”; *Day Division* → “2”; *Kernel* → “gaussian”; *Bandwidth* → “sig_std”; *Penalty* → “slope” e *Alpha* → “4”: neste exemplo 5 CPs foram encontrados. b) *Search Method* → “BottomUp” e os mesmos parâmetros da Fig. 2a): neste exemplo nenhum CP foi encontrado.

Os valores de *Alpha* aplicados à penalidade, variam de 75 a 150 para a penalidade do tipo “cost” e de 4 a 12 para a penalidade do tipo “slope”, dependendo da função Kernel. Comparativo entre diferentes configurações de *Kernel*, *bandwidth* e *Alpha* podem ser encontradas em Letzgs (2020).

A depender dos parâmetros selecionados, o algoritmo de CPD pode ficar mais ou menos sensível, e assim, pode determinar que uma região da série temporal apresenta um *change point* ou não. Como exemplo, a Figura 3a) apresenta o resultado para a utilização do conjunto [*Search Method* → “Window”; *Day Division* → “2”; *Kernel* → “gaussian”; *Bandwidth* → “sig_std”; *Penalty* → “slope” e *Alpha* → “4”], que acarreta na detecção de 5 pontos de mudança para o mesmo sinal de temperatura do enrolamento_U. A alteração do método de busca *Dynp* pelo método *BottomUp*, mantendo-se os demais parâmetros de configuração para obtenção da Fig. 2a), faz com que o algoritmo de CPD deixe de considerar as rupturas na série temporal como CPs [ver Fig. 3b)].

Cumprir destacar que todos os gráficos gerados são atualizados em tempo real de acordo com as alterações nos parâmetros feitas pelo usuário. A depender da complexidade dos métodos de busca, Kernel, penalidade e aprendizado selecionados, o tempo para geração dos gráficos pode aumentar significativamente.

Predição após remoção de CPs

Em uma última análise, verificou-se como a presença de pontos de mudança afeta a predição do comportamento

dos componentes de uma WTG. O comportamento normal da turbina pode ser realizado removendo os CPs, e assim, evitar que a predição reconheça como comportamento normal as medições referentes a estados críticos de funcionamento ou com componentes em eminente falha.

Diante disso, a representação do correto comportamento-normal da WTG foi configurada para ser referente aos dados após o último CP detectado, em 05/08/2019 [ver Fig. 2a)]. A Figura 4 apresenta a série temporal, os valores preditos e o resultado da detecção de pontos de mudança.

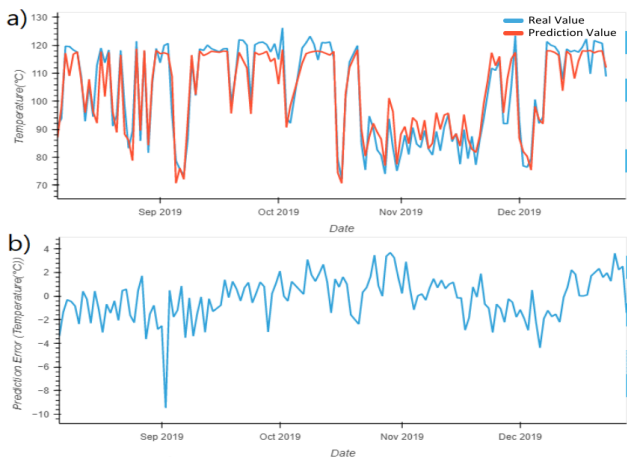


Figura 4. a) Sinal real e predito em °C da série de dados representado o novo comportamento normal da WTG. b) Resposta do algoritmo de CPD para o sinal normalizado de enrolamento_U não indicando CPs.

O algoritmo CPD não identifica rupturas na série temporal, sugerindo um comportamento dentro dos padrões esperados, ou seja, indicando que a WTG está operando em comportamento normal. É importante mencionar que para a predição, escolheu-se separar as medições do último mês de 2019 (Dez/19) como dados de teste, ficando os demais meses como dados de treino (de Ago a Nov/19).

5. CONCLUSÃO

Foi apresentada uma ferramenta para análise e detecção de pontos de mudança em sinais gerados por sensores de turbinas eólicas. Desenvolvida na linguagem Python em conjunto com a biblioteca Ruptures, a ferramenta denominada CP-WTG Analytics possui interface simples e intuitiva, permitindo a configuração de todos os parâmetros necessários para detecção de CPs em séries temporais, bem como análise visual dos resultados gerados.

Experimentos foram realizados com dados de uma turbina eólica localizada no Estado do Rio Grande do Norte, que apresentou alarmes de possíveis falhas em seu gerador no ano de 2019. A configuração das partes do algoritmo de CPD (função de custo, método de busca e restrições) foi demonstrada e os gráficos gerados pela ferramenta identificam o momento em que ocorreu um CP. Em comparação com os sinais de alarme gerados pela turbina, a análise de CP antecipou em mais de 60 dias o início do comportamento não-normal de operação da turbina.

A correta seleção dos métodos e ajuste de valores dos parâmetros determinam a sensibilidade de detecção desejada, e consequentemente, o monitoramento do comportamento

normal de uma turbina eólica. Ademais, diferentes configurações e ajustes foram analisados para demonstrar a aplicabilidade da ferramenta.

AGRADECIMENTOS

Os autores gostariam de agradecer ao CNPq, CAPES, UFERSA e Voltalia Energia do Brasil pelo apoio financeiro concedido a pesquisa.

REFERÊNCIAS

- Aminikhanghahi, S. and Cook, D. (2017). A survey of methods for time series change point detection. *Knowl. Inf. Syst.*, 51, 339–367.
- Arlot, S. and et al. (2019). A kernel multiple change-point algorithm via model selection. *J. Mach. Learn. Res.*, 20, 1–56.
- Basseville, M. and et al. (1993). *Detection of Abrupt Change Theory and Application*. Prentice-Hall, Inc, Rennes, FR.
- Birgé, L. and et al. (2007). Minimal penalties for gaussian model selection. *Prob. Theo. Rel.*, 138, 33–73.
- Csörgö, M. and et al. (1997). *Limit theorems in change-point analysis*. Chichester, New York, USA.
- Dao, C. and et al. (2019). Wind turbine reliability data review and impacts on levelised cost of energy. *Wind Energy*, 22(12), 1848–1871.
- Dao, P. (2022). Condition monitoring and fault diagnosis of wind turbines based on structural break detection in scada data. *Renewable Energy*, 185, 641–654.
- Fryzlewicz, P. (2007). Unbalanced haar technique for nonparametric function estimation. *J. Amer. Stat. Assoc.*, 480, 1318–1327.
- Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. *J. Amer. Stat. Assoc.*, 42, 2243–2281.
- GWEC (2023). Global wind report 2023. URL [gwec.net/wp-content/uploads/2023/04/GWEC-2023_interactive.pdf](https://www.gwec.net/wp-content/uploads/2023/04/GWEC-2023_interactive.pdf). Accessed 2023-02-15.
- Haraldsdóttir, H. (2016). *Lifetime Analysis of a Wind Turbine Component, Mather thesis*. Chalmers University of Technology, Gothenburg.
- IRENA (2022). International Renewable Energy Agency. Renewable power generation costs in 2021. Accessed 2023-04-01.
- Killick, R. and et al. (2012). Optimal detection of change-points with a linear computational cost. *J. Amer. Stat. Assoc.*, 107, 1590–1598.
- Killick, R. and et al. (2014). Changepoint: an R package for changepoint analysis. *J. Stat. Soft.*, 58, 1–19.
- Letzgus, S. (2020). Change-point detection in wind turbine scada data for robust condition monitoring with normal behaviour models. *Wind Energy Sci.*, 5, 1375–1397.
- Quanlin, Z. and et al. (2020). An automatic data cleaning and operating conditions classification method for wind turbines scada system. In *2020 ICCWAMTIP*, 437–443.
- Ruptures (2023). URL [centre-borelli.github.io/ruptures-docs](https://github.com/ruptures-docs). Accessed 2023-01-22.
- Truong, C. and et al. (2020). Selective review of offline change point detection methods. *Sig. Process.*, 167.
- Xu, M. and et al. (2014). Application of change-point analysis to abnormal wind power data detection. In *2014 IEEE PES*, 1–5.

Yuan, T. and et (2019). Gearbox fault prediction of wind turbines based on a stacking model and change-point detection. *Energies*, 12(22).