

# **Relatório Técnico**

## **Um tratamento para outliers e dados ausentes em séries temporais multivariadas em Redes de Sensores Sem Fio**

**Crislânio S. Macêdo<sup>1</sup>**

1

crislanio.ufc@gmail.com

**Abstract.** *Multivariate time series, such as those obtained in dense WSNs, often exhibit spatial and temporal correlations in the data. This property can be used to treat and lessen the effects of two frequent events in data acquisition processes: the presence of outliers and the absence of data. This paper presents and evaluates a joint strategy based on non-parametric regression to treat outliers and missing data. Outliers are detected by statistical techniques and treated as missing data. Fuzzy Inference (FI) is used to estimate the temperatures of the sensors and the decomposition of time series data is performed by the Simple Moving Average Method (SMA).*

**Resumo.** *Séries temporais multivariadas, tais como aquelas obtidas em RSSF densas, frequentemente exibem correlações espacial e temporal nos dados. Esta propriedade pode ser utilizada para tratar e minorar os efeitos de dois eventos frequentes em processos de aquisição de dados: a presença de outliers e a ausência de dados. Este trabalho apresenta e avalia uma estratégia conjunta baseada em regressão não paramétrica para tratar outliers e dados faltantes. Outliers são detectados por técnicas estatísticas e tratados como dados faltantes. A Inferência Fuzzy (FI) é usada para estimar as temperaturas dos sensores e a decomposição dos dados das séries temporais é realizada pelo Método de Média Móvel Simples (SMA).*

### **1. Introdução**

Rede de sensores sem fio (RSSF) é uma rede composta por um conjunto de nós sensores implantados numa região em particular para detectar e monitorar fenômenos físicos como temperatura, umidade, pressão, voltagem, luminosidade e assim por diante. Este conjunto de sensores conduz a coleta dos dados, podendo ou não realizar algum processamento no próprio nó, transferindo a informação para um concentrador que é responsável pelo armazenamento, compilação e transferência destes dados à locais mais distantes.

As redes de sensores são implantadas em vários domínios para obter informações sobre diferentes fenômenos em tempo real. Os dados adquiridos normalmente não são utilizáveis diretamente, pois sofrem de problemas como ruído, falta de dados e incompletude. Existem vários fatores que contribuem para esses problemas. Ruído geralmente ocorre devido à imprecisão na detecção e transmissão do hardware. Dados faltantes geralmente ocorrem devido a perda de pacotes e falha de nó. Como os sensores analisam fenômenos físicos contínuos em intervalos de tempo discretos, os dados adquiridos são incompletos. Portanto, a incompletude é um problema inerente a aquisição de dados em

redes de sensores, enquanto valores ausentes são de natureza accidental. Todos esses problemas impactam na qualidade dos dados obtidos de tais redes [Tan et al. 2005].

Visando nós sensores de baixa capacidade, este trabalho propõe um método para detectar e imputar dados em RSSF, atenuando os problemas descritos acima. O restante deste trabalho está assim organizado. A Seção 2 descreve os trabalhos relacionados. A Seção 3 descreve a implementação técnica do trabalho. A Seção 4 descreve os Dados e Preprocessamento. A Seção 5 descreve o Sistema de Inferência Fuzzy (FI). A Seção 6 descreve o Método de Média Móvel Simples (SMA). A Seção 7 mostra como foi realizado a Detecção de *Outliers*. A Seção 8 descreve a etapa de Imputação de Dados Ausentes. A Seção 9 é de Resultados. A Seção 10 é de Conclusão do estudo. A Seção 11 fala sobre Trabalhos Futuros e finalmente a Seção 12 mostra os Apêndices.

## 2. Trabalhos Relacionados

As técnicas para detecção e imputação de dados são aplicadas nas mais diversas áreas. [Gil et al. 2018] usa técnicas como máquinas de Vetores de Suporte por Mínimos Quadradados (LS-SVM) e Análise de Componentes Principais(PCA) para detectar *outliers* em RSSF online. [Cateni et al. 2007] usa um Sistema de Inferência Fuzzy para combinar várias estratégias de detecção de *outliers* em dados industriais de uma siderúrgica. [Cho et al. 2013] usa Análise Harmônica nos dados e técnicas estatísticas como z-score modificado nos resíduos para detectar *outliers* e fazer imputação dos dados de temperatura da água costeira (CWT). [Tan et al. 2005] apresenta métodos populares de modelagem de dados como filtros de Kalman e regressão para limpeza dos dados do sensor.

Diante desse cenário, fica nítido o arcabouço de ferramentas e métodos para realizar imputação e detecção de outliers, a justificativa da nossa abordagem se dá pelo conhecimento do especialista na criação de regras para o FI e na característica do que é um *outlier* para o nosso estudo, note que no melhor caso é esperado que um dado sensor transmita dados de temperatura com pouca variação entre uma época e outra, o SMA é capaz de capturar as nuances de uma variação maior tratando o dado como um *outlier*.

## 3. Implementação

Nessa seção descrevemos o passo a passo da construção da nossa abordagem. O código deste trabalho pode ser encontrado no github com o link do projeto no rodapé. A Figura 1 mostra as funções para a construção do método SMA juntamente com a remoção dos *outliers* pelo especialista após fazer a leitura do conjunto de dados. Feito a remoção dos *outliers* e removidos os *outliers* pelo SMA é feito uma preparação dos dados para fazer a imputação pelo FI conforme Figura 2. A Figura 3 mostra a preparação dos dados para a aplicação do FI conforme Figura 4.

Depois da aplicação do FI é montado a tabela com as informações estatística dos dados, conforme Figura 5. Note que esse processo é feito para cada sensor. Essa seção mostrou os passos necessários para a construção do modelo Fuzzy bem como o método SMA. Este relatório pode ser encontrado em: <https://github.com/crislanio/OutlierDetectionAndMultivariateImputationFI-SMA>. Nas seções seguintes são mostrados as etapas de preprocessamento, construção e aplicação do sistema FI e SMA e os resultados do estudo, bem como trabalhos futuros.

---

<sup>0</sup>Repositório do trabalho, <https://github.com/crislanio/OutlierDetectionAndMultivariateImputationFI-SMA>

```

outliersIlLista = []
outliersIlListaV = []
for i, row in data.iterrows():
    if row[4] >=26.2 :
        outliersIlLista.append(i)
        outliersIlListaV.append(row[4])
print(data.shape, len(outliersIlLista))
# Removendo outliers no sensor
passo3= data.copy()
print(passo3.head())
print(passo3.shape)
passo3= passo3.drop(outliersIlLista)
print(passo3.shape)
passo3.to_csv('sensor.csv')
data = pd.read_csv("sensor.csv")
data.shape
data.drop(['Unnamed: 0'], axis=1,inplace=True)
# data.head(1)
for key,d in data[:data.shape[0]].groupby('3'):
    print(key,"n",d.head())
    print("-----")

def moving_average(data, window_size):
    window = np.ones(int(window_size))/float(window_size)
    return np.convolve(data, window, 'same')

def explain_anomalies(y, window_size, sigma=1.0):
    avg = moving_average(y, window_size).tolist()
    residual = y - avg
    # Calculando a variação da distribuição dos resíduos
    std = np.std(residual)
    return {'standard_deviation': round(std, 2),
            'anomalies_dict': collections.OrderedDict([(index, y_i) for index, y_i, avg_i in zip(count(), y, avg)
                if (y_i > avg_i + (sigma*std)) | (y_i < avg_i - (sigma*std))])}

# Esta função é responsável por exibir o desempenho da função no conjunto de dados fornecido.
def plot_results(x, y, window_size, sigma_value=1,
                 text_xlabel="X Axis", text_ylabel="Y Axis"):
    plt.figure(figsize=(15, 8))
    plt.plot(x, y, "k.")
    y_av = moving_average(y, window_size)
    plt.plot(x, y_av, color='gray')
    plt.xlim(0, 30000)
    plt.xlabel(text_xlabel)
    plt.ylabel(text_ylabel)

    # Query para as anomalias e o plot da anomalias
    events = {}
    events = explain_anomalies(y, window_size=window_size, sigma=sigma_value)
    x_anomaly = np.fromiter(events['anomalies_dict'].keys(), dtype=int, count=len(events['anomalies_dict']))
    y_anomaly = np.fromiter(events['anomalies_dict'].values(), dtype=float,
                           count=len(events['anomalies_dict']))
    plt.plot(x_anomaly, y_anomaly, "r*")
    # add grid e linhas
    plt.grid(True)
    plt.show()

```

**Figure 1. Média Móvel Simples**

#### 4. Dados e Preprocessamento

As RSSF possui um fator crítico que é o poder de processamento dos nós sensores, todavia o mal funcionamento nas transmissões podem influenciar na qualidade dos dados monitorados, afetando nas tomadas de decisões. Embora na literatura existam diversos trabalhos com o intuito de detectar *outliers* e realizar imputação nos dados em RSSF, dada a natureza do problema, seja no processamento de dados massivos ou na detecção de anomalias em RSSF online, ocorre uma dificuldade na execução das técnicas analisadas.

A análise exploratória dos dados é uma etapa crucial e de suma importância no entendimento de um determinado conjunto de dados, sendo a imputação e detecção de *outliers* um passo essencial para uma análise estatística. Nesse estudo, sugerimos um método de detectar *outliers* e realizar a imputação de dados. A abordagem proposta foi avaliada utilizando o Intel Lab data set<sup>1</sup> de domínio público contendo medições de 54 sensores, como mostra na Figura 6 realizadas no Intel Berkeley Research Lab. O dataset contém um total de 2.3 milhões de medidas e um total de 65535 épocas. Os nós sensores coletaram medições de umidade, temperatura, luminosidade e energia a cada 31 segun-

```

indiceInput = []
for i in range(65536):
    indiceInput.append(i)
np.max(indiceInput)
passo0Ep = passo0['2'].tolist()
len(passo0Ep)

idInputEq = set(indiceInput)
idInputNEq = set(passo0Ep)
resultIntersecEp = list( idInputEq - idInputNEq )

print(len(resultIntersecEp)) # {4, 6, 10}

print(len(idInputEq), len(idInputNEq))
# série temporal de X elementos, um a 31 segundos a partir de 2004-02-28
time = pd.date_range('2004-02-28', periods=65536, freq='31s')

series = pd.Series(np.random.randint(100, size=65536), index=time)
dfHora =pd.DataFrame(series, columns=['1'])
data.shape, type( len(resultIntersecEp) ), dfHora.head(1)
# tamanho resultIntersecEp # CSV das épocas faltantes
listofzeros = [0] * len(resultIntersecEp)

myDicNewdf2 = {
    '0':dplot[:len(resultIntersecEp)][['Date']],
    '1':dplot[:len(resultIntersecEp)][['Time']],
    '2':resultIntersecEp[:len(resultIntersecEp)],
    '3':dplot[:len(resultIntersecEp)][['Motaid']],
    '4':listofzeros,
    '5':dplot[:len(resultIntersecEp)][['Humidity']],
    '6':dplot[:len(resultIntersecEp)][['Light']],
    '7':dplot[:len(resultIntersecEp)][['Voltage']]
}

dfOut = pd.DataFrame(myDicNewdf2)
print(dfOut.shape[0])
dfOut.to_csv('csvAlloutliers.csv', index=False)
# Gerando o csv para fazer reconstrução dos dados
dffinalStep = pd.concat((dfOut,passo0), sort='2', axis=0)
dffinalStep.sort_values(by = '2', inplace = True)

# dfFinalStep.set_index('2', inplace=True)
print(dffinalStep.shape)

dffinalStep.tail(10)
# 65536-65623
dffinalStep.to_csv('dfImputation.csv', index=False)

```

**Figure 2. Preparação do csv para o passo 3 da reconstrução dos dados**

dos. Apenas dados de temperatura foram utilizados nestes experimentos. De acordo com o conhecimento do especialista foi feito uma filtragem na medida da temperatura, em seguida o método de Média Móvel Simples foi usado para detectar *outliers*. Depois de remover os *outliers*, os dados ausentes ou removidos são preenchidos com o conjunto de regras conduzidos com o Sistema de Inferência Fuzzy.

## 5. Sistema de Inferência Fuzzy (FI)

Para inferência fuzzy foi utilizado um modelo de Mandani [Jang et al. 1997]. Um sistema fuzzy baseado no modelo de Mandani pode ser dividido em 4 partes: variáveis linguísticas. Fuzzificação das variáveis de entrada. Regras de avaliação. Máquina de inferência. Defuzzificação. O modelo construído foi baseado em conhecimento do especialista do domínio. Para explicar a construção do sistema de Mandani, a Figura 6 mostra um mapa da distribuição dos sensores.

### 5.1. Detalhamento do FI

Para construir o FI foi usado como variável linguística a temperatura. Uma variável linguística é uma variável cujos valores são nomes de conjuntos fuzzy. Por exemplo, a temperatura de um determinado processo pode ser uma variável linguística assumindo valores baixa, média, e alta. Estes valores são descritos por intermédio de conjuntos fuzzy, representados por funções de pertinência trapezoidal e triangular definidos na perspectiva

---

<sup>1</sup>Intel Berkeley Laboratory Data, <http://berkeley.intel-research.net/labdata/>

```

data_column_names = ["Date", "Time", "Epoch", "Moteid", "Temperature", "Humidity", "Light", "Voltage"]
dplot = pd.read_csv("../data.txt", sep = ' ', names = data_column_names)
s = 50.0
S49df = dplot[dplot['Moteid']==49]
S51df = dplot[dplot['Moteid']==51]
S50df = pd.read_csv('dfImputation.csv')
S50df['3'] = s

# Removendo os outliers pelo especialista para os vizinhos do sensor
outliersIL = []
outliersILValues = []
s1 = S49df
for i, row in s1.iterrows():
    if row['Temperature'] >=26.2:
        outliersIL.append(i)
        outliersILValues.append(row['Temperature'])
# Removendo outliers no Sensor
passo3= s1.copy()
passo3= passo3.drop(outliersIL)
passo3.to_csv('S49FINAL.csv')
# Removendo os outliers pelo especialista para os vizinhos do sensor
outliersIL2 = []
outliersILV2 = []
s2 = S51df
for i, row in s2.iterrows():
    if row['Temperature'] >=26.2:
        outliersIL2.append(i)
        outliersILV2.append(row['Temperature'])
print(len(outliersIL2))
# Removendo outliers no S2
passo3_3= s2.copy()
passo3_3= passo3_3.drop(outliersIL2)
passo3_3.to_csv('S51FINAL.csv')
# Removendo os outliers pelo especialista para os vizinhos do sensor
outliersIL_3 = []
outliersILV_3 = []
s3 = S50df
for i, row in s3.iterrows():
    if row['4'] >=26.2:
        outliersIL_3.append(i)
        outliersILV_3.append(row['4'])
# Removendo outliers no sensor
passo3_2= s3.copy()
passo3_2= passo3_2.drop(outliersIL_3)
passo3_2.to_csv('S50FINAL.csv')
S49df_= pd.read_csv('S49FINAL.csv')
S50df_= pd.read_csv('S50FINAL.csv')
S51df_= pd.read_csv('S51FINAL.csv')
S50df_[‘3’] = s

```

**Figure 3. Sistema Fuzzy - Leitura dos dados e remoção dos *outliers* pelo especialista**

do especialista, conforme mostrado na Figura 8. Para se obter os parâmetros aplicáveis às funções de pertinência analisou-se o gráfico de todas as medições de temperatura contidas no conjunto de dados baseados na Figura 9.

Um FI é criado para estimar a temperatura em cada sensor. As regras de inferência fuzzy foram construídas *ad hoc* examinando a Figura 6. Para estimar a temperatura em cada sensor foi usado apenas os dois sensores mais próximos na vizinhança. Assim, cada conjunto de regras possui no mínimo 3 e no máximo 9 regras. A Figura 7 mostra um exemplo de dependência.

Como observamos na Figura 7, a imputação da temperatura para o sensor 1 em uma determinada época depende da temperatura dos sensores 2 e 33 na mesma época, caso uma época não exista nos sensores 2 ou 33 é obtido o resultado da última medição do sensor. O mesmo procedimento é válido para os outros sensores. Cabe ressaltar que para os experimentos feitos, não foi considerado os sensores 5, 15, 28 por questões de inconsistências.

O operador de agregação utilizado foi a composição max-min e para defuzzificação foi utilizado o método do maior do máximo. Outros operadores de agregação e defuzzificação foram testados mas não geraram melhores resultados.

## 6. Método de Média Móvel Simples (SMA)

O método da média móvel é utilizado para controlar a tendência em determinados dados da série temporal. Na SMA, cada ponto de dados das séries temporais é ponderado independentemente de onde ele ocorre na sequência, pois não há nenhum fator de ponderação aplicado a qualquer um dos pontos de dados conforme equação 6.

$$SMA = \frac{X_n + X_{n-1}}{2}$$

Em que:

$X_n$  significa o valor do ponto de dados no momento  $n$ ;  $n$  significa o instante do valor dos dados. Para uma determinada janela contínua de período de tempo, a média e o desvio padrão das entradas são calculados. Se a próxima entrada no conjunto de dados estiver entre a média  $+ - 2 * \text{desvio padrão}$ , é considerado normal caso contrário é considerado uma anomalia.

## 7. Detecção de *Outliers*

Basicamente, *outliers* (ou anomalias) são definidos como as anormalidades que não seguem os comportamentos normais esperados bem definidos, a maioria dos métodos para detecção de *outliers* é aplicável apenas a dados que seguem uma distribuição normal [Cho et al. 2013].

As anomalias podem ser induzidas por vários fatores, como atividade maliciosa, fraude, interferências do usuário, etc. Deve-se notar que a detecção de anomalias é distinta, mas relacionada à "remoção de ruído", dados indesejados que não são de interesse do especialista do domínio [Teng et al. 1990].

De forma geral, a identificação de *outliers* é utilizada em aprendizagem de máquina, mineração de dados, estatística, teoria da informação e no preprocessamento de dados para a detecção de intrusão em fraude [Chandola et al. 2009]. Porém, a aplicação dessas técnicas na área de rede de sensores tem ocorrido recentemente [Zhang et al. 2010]. O processo de detecção de *outliers* é desafiador para vários contextos de atuação. Podemos observar alguns impasses no processo de detecção de anomalias como: identificação e definição da região de normalidade dos dados; o processo de detecção de *outliers* torna-se notavelmente complicado se os dados contiverem anomalias juntamente com ruído; a definição de anomalia em um domínio de aplicação pode ser dramaticamente diferente em outros domínios [Chandola et al. 2009].

Portanto o método sugerido neste estudo analisa o comportamento temporal dos sensores, outros métodos foram usados para detectar outliers, mas esses resultados foram menos sensíveis. Certos critérios especificados que definem os outliers são mais sensíveis [Cho and Oh 2012]. Assim, os seguintes passos foram usados para detectar outliers na série temporal de cada sensor:

Passo 1: Dado o conhecimento do especialista e depois de analisar graficamente a escala da temperatura dos sensores foi removida dados com medição maiores do que 26.2 graus.

Passo 2: O método SMA foi usado para detectar *outliers* para uma determinada janela de tempo: em nosso estudo usamos uma janela do tamanho de 10 épocas. Caso

o valor da próxima entrada no conjunto de dados estiver entre a média  $+ - 2 * \text{desvio padrão}$ , é considerado normal, caso contrário é considerado uma anomalia.

## 8. Imputação dos Dados Ausentes

Muitos dos métodos conhecidos para realizar imputação dos dados não são aplicados em estudos longitudinais, que são estudos de pesquisa correlacional que envolvem observações repetidas das mesmas variáveis ao longo dos períodos de tempo. Para superar esse impasse foi desenvolvido um método multivariado para realizar a imputação dos dados baseado em FI. Os seguintes passos foram usados para realizar a imputação dos dados:

Passo 1: O FI foi usado para imputar os valores através de regras **ad hoc** em que para estimar a temperatura em um nó sensor em uma determinada época é observado no máximo os 2 vizinhos mais próximos naquela época, caso o nó sensor vizinho não tenha valores de temperatura para a época estimada é usado o valor da época seguinte existente.

## 9. Resultados

No apêndice podemos observar as tabelas e séries temporais para cada sensor. Foi observado uma variação nas medições para as séries temporais de cada sensor capturadas pelo método SMA, que teve resultados satisfatórios na detecção de *outliers*, visto que para o nosso contexto um *outlier* é uma medida atípica em uma janela de tamanho 10, como as medidas de temperatura não oscilam tanto quanto uma medida de luminosidade, o método foi capaz de se comportar bem. Olhando para o caso geral na Figura 12 observamos que a média da temperatura do passo 2 para o passo 3 varia pouco, o desvio padrão e o MAD diminuíram de forma acentuada no passo 3 em relação ao passo 2.

De forma geral podemos constatar que o FI possui um comportamento aceitável para realizar imputação em sensores com uma grande sequencia de épocas faltantes, ou seja, em muitos casos o valor imputado para o sensor foi determinado baseado na última época válida dos sensores vizinhos e esse comportamento é frequente. A Figura 10 e Figura 11 mostra a interseção das épocas entre os seus sensores vizinhos, por exemplo, o sensor 46 possui em torno de 78% de épocas em comum para os seus sensores vizinhos, em consequência desse fenômeno, em torno de 22% dos dados será imputado um valor obtido pelo FI com uma época anterior a época que se deseja imputar.

O fato de uma época ser imputada por valores de épocas anteriores fizeram com que o FI não chegasse a resultados mais satisfatórios. O alto índice de valores repetidos se deu em consequência desse comportamento, porém, como observamos no apêndice 2 nas tabelas dos sensores, os resultados após a detecção de *outliers* e reconstrução dos dados tiveram a informação estatística mantida.

As Tabelas do apêndice 2 contém informações estatísticas básica dos dados antes e depois da remoção dos *outliers* (ARO / DRO), e após a imputação dos dados faltantes. Como podemos observar nas tabelas há uma diferença de épocas faltantes em relação ao número de épocas existentes. Para efeito de explicação considere por exemplo a Figura 16 que representa a série temporal do sensor 1, nela os valores para o passo 1 (ARO), a média e mediana do passo 1 ARO aumentou ligeiramente em relação ao passo 2 (DRO) assim

como o desvio padrão e o MAD. A média e mediana do passo 2 (DRO) diminuiu em relação ao passo 3, já o desvio padrão e o MAD aumentaram. Em comparação com os dados originais, a média e a mediana dos novos dados (passo 3) diminuíram substancialmente, enquanto o desvio padrão diminuiu por causa dos grandes valores positivos de *outliers*.

## 10. Conclusão

A estratégia abordada pelo FI para imputar os dados de um sensor em uma determinada época baseados nos valores da mesma época dos 2 sensores vizinhos tiverem diferenças quando se comparou a informação estatística entre o passo 2 (DRO) e o passo 3 da reconstrução dos dados. Esse comportamento se deu devido ao grande número de intervalos de épocas faltantes para um dado sensor.

O método sugerido usado para detectar e remover outliers obteve resultados satisfatórios. Em uma comparação de cada MAD, mediana, as mudanças foram relativamente pequenas. No entanto, outliers e dados ausentes causam vieses ou distorções nos resultados estatísticos. Assim sendo, outliers e dados em falta devem ser revistos e tratados antes da realização de análises estatística. O FI ou qualquer outro método de imputação devem ser robustos e mostrar a tendência dos dados intrínsecos, quanto maior o intervalo de dados em falta, maiores os erros dos dados estimados. No caso de um intervalo de dados muito longo, o preenchimento dos dados perdidos deve ser realizado com cuidado para não distorcer os resultados.

Em geral para todos os sensores foram observadas diferenças quando se comparou a informação estatística entre os dados brutos e tratados (passo 3). Contudo, a detecção “mais provável” da perspectiva da significância estatística, os *outliers* foram removidos com sucesso e os dados perdidos foram preenchidos com sucesso para este estudo, então, o método proposto é esperado para produzir alto desempenho.

## 11. Trabalhos Futuros

Como trabalho futuro pode-se criar o conjunto de regras para o FI de forma automática baseado em uma Árvore de Regressão. Pode-se testar a imputação de dados levando em consideração um conjunto maior de vizinhos para cada sensor e comparar os resultados, bem como estimar os parâmetros do FI baseado em um algoritmo genético. Em relação ao SMA foi observado que para o propósito do estudo o método teve resultados satisfatórios, visto que para o nosso contexto um *outlier* é uma medida atípica em uma janela de tamanho 10, como uma nova perspectiva apesar das medidas de temperatura não oscilarem tanto quanto uma medida de luminosidade, pode-se testar novos métodos como Análise Harmônica e comparar os resultados.

## 12. Apêndice

Imagens e tabela com as informações estatísticas das séries temporais dos sensores.

## References

- Cateni, S., Colla, V., and Vannucci, M. (2007). A fuzzy logic-based method for outliers detection. In *Artificial Intelligence and Applications*, pages 605–610.

- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15.
- Cho, H.-Y. and Oh, J.-H. (2012). Outlier detection of the coastal water temperature monitoring data using the approximate and detail components. *Journal of the Korean Society for Marine Environment & Energy*, 15(2):156–162.
- Cho, H. Y., Oh, J. H., Kim, K. O., and Shim, J. S. (2013). Outlier detection and missing data filling methods for coastal water temperature data. *Journal of Coastal Research*, 65(sp2):1898–1903.
- Gil, P., Martins, H., and Januário, F. (2018). Outliers detection methods in wireless sensor networks. *Artificial Intelligence Review*, pages 1–26.
- Jang, J.-S. R., Sun, C.-T., and Mizutani, E. (1997). Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence.
- Tan, Y. L., Sehgal, V., and Shahri, H. H. (2005). Sensoclean: Handling noisy and incomplete data in sensor networks using modeling. *Main*, pages 1–18.
- Teng, H. S., Chen, K., and Lu, S. C. (1990). Adaptive real-time anomaly detection using inductively generated sequential patterns. In *Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on*, pages 278–284. IEEE.
- Zhang, Y., Meratnia, N., and Havinga, P. J. (2010). Outlier detection techniques for wireless sensor networks: A survey. *IEEE Communications Surveys and Tutorials*, 12(2):159–170.

## Sistema Fuzzy

```

# Modelo Fuzzy
temp = np.arange(-26.2, 26.2, 0.1)
tempF = np.arange(-26.2, 26.2, 0.1)
# Gerando as funções de pertinência
temp_lo = fuzz.trimf(temp, [-26.2, 0.0, 5.0])
temp_md = fuzz.trimf(temp, [0.0, 5.0, 20.0])
temp_hi = fuzz.trimf(temp, [17.20, 22.8, 26.2])
mfx = fuzz.trapmf(tempF, [-26.2, 5.0, 17.20, 26.2])
tempFinal_lo = fuzz.trapmf(tempF, [-26.2, 5.0, 17.20, 26.2])
tempFinal_md = fuzz.trimf(tempF, [0, 5.0, 20.0])
tempFinal_hi = fuzz.trimf(tempF, [17.20, 22.8, 26.2])
# Fuzzificação
def fuzzification(i, data, data1, data2):
    if data['3'].ix[i]== s:
        temp_level_lo0 = fuzz.interp_membership(temp, temp_lo, data1['Temperature'][i]) # ex: [52, [51, 52]]
        temp_level_md0 = fuzz.interp_membership(temp, temp_md, data1['Temperature'][i])
        temp_level_hi0 = fuzz.interp_membership(temp, temp_hi, data1['Temperature'][i])
        temp_level_lo1 = fuzz.interp_membership(temp, temp_lo, data2['Temperature'][i])
        temp_level_md1 = fuzz.interp_membership(temp, temp_md, data2['Temperature'][i])
        temp_level_hi1 = fuzz.interp_membership(temp, temp_hi, data2['Temperature'][i])
        active_rule1_0 = np.fmin(temp_level_lo0, temp_level_lo1)
        active_rule1_1 = np.fmin(temp_level_lo0, temp_level_md1)
        active_rule1_2 = np.fmin(temp_level_lo0, temp_level_hi1)
        active_rule1_3 = np.fmin(temp_level_md0, temp_level_lo1)
        active_rule1_4 = np.fmin(temp_level_md0, temp_level_md1)
        active_rule1_5 = np.fmin(temp_level_md0, temp_level_hi1)
        active_rule1_6 = np.fmin(temp_level_hi0, temp_level_lo1)
        active_rule1_7 = np.fmin(temp_level_hi0, temp_level_md1)
        active_rule1_8 = np.fmin(temp_level_hi0, temp_level_hi1)
        active_rule1_Final_0 = np.fmax(active_rule1_0, tempFinal_lo)
        active_rule1_Final_1 = np.fmax(active_rule1_1, tempFinal_lo)
        active_rule1_Final_2 = np.fmax(active_rule1_2, tempFinal_md)
        active_rule1_Final_3 = np.fmax(active_rule1_3, tempFinal_lo)
        active_rule1_Final_4 = np.fmax(active_rule1_4, tempFinal_hi)
        active_rule1_Final_5 = np.fmax(active_rule1_5, tempFinal_hi)
        active_rule1_Final_6 = np.fmax(active_rule1_6, tempFinal_md)
        active_rule1_Final_7 = np.fmax(active_rule1_7, tempFinal_hi)
        active_rule1_Final_8 = np.fmax(active_rule1_8, tempFinal_hi)
        r1_0 = np.fmax(active_rule1_Final_8, np.fmax(active_rule1_Final_7, np.fmax(active_rule1_Final_6, active_rule1_Final_5) ))
        r1_1 = np.fmax(active_rule1_Final_5, np.fmax(active_rule1_Final_3, np.fmax(active_rule1_Final_2, np.fmax(active_rule1_Final_1, np.fmax(active_rule1_Final_0, aggregated))))
        aggregated = np.fmax(r1_0,r1_1)
    return aggregated
def defuzz(x, mfx, mode):
    mode = mode.lower()
    x = x.ravel()
    mfx = mfx.ravel()
    n = len(x)
    assert n == len(mfx), 'Length of x and fuzzy membership function must be \ identical.'
    if 'Centroid' in mode or 'bisector' in mode:
        tot_area = mfx.sum()
        assert tot_area != 0, 'Total area is zero in defuzzification!'
        if 'Centroid' in mode:
            return centroid(x, mfx)
        elif 'bisector' in mode:
            tmp = 0
            for k in range(n):
                tmp += mfx[k]
                if tmp >= tot_area / 2.:
                    return x[k]
    elif 'mom' in mode:
        return np.mean(x[mfx == mfx.max()])
    elif 'som' in mode:
        tmp = x[mfx == mfx.max()]
        return tmp[tmp == np.abs(tmp).min()][0]
    elif 'lom' in mode:
        tmp = x[mfx == mfx.max()]
        return tmp[tmp == np.abs(tmp).max()][0]
    else:
        raise ValueError('The input for `mode` , %s, was incorrect.' % (mode))
aux = 0.0
for i, row in S50df_.iterrows():
    if S50df_[4][i] == 0.0:
        try:
            S50df_.at[i, '4'] = defuzz(tempF,fuzzification(i, S50df_, S49df_,S51df_), 'lom')
            aux = defuz(tempF,fuzzification(i, S50df_, S49df_,S51df_), 'lom')
        except:
            S50df_.at[i, '4'] = aux
S50df_.to_csv('finalsensor.csv', index=False)

```

Figure 4. Construção e aplicação do Sistema Fuzzy

## Estatística do Sensor

```
# Passo 0
sensor = s
myDicInit = {
    'Data type':['Raw Data'],
    'size':[dplot[dplot['Moteid']==sensor].shape[0]], # Pegamos o shape dos dados brutos de dplot para cada sensor
    'mean':dplot[dplot['Moteid']==sensor][:dplot.shape[0]]['Temperature'].mean(),
    'SD':dplot[dplot['Moteid']==sensor][:dplot.shape[0]]['Temperature'].std(),
    'median':dplot[dplot['Moteid']==sensor][:dplot.shape[0]]['Temperature'].median(),
    'MAD':dplot[dplot['Moteid']==sensor][:dplot.shape[0]]['Temperature'].mad()
}
df0 = pd.DataFrame(myDicInit)
# Passo 1
myDic = {
    'Data type':['step 1 (BOR)'],
    'size':[d.shape[0]], # d.shape[0] sensor
    'mean':d[:d.shape[0]]['4'].mean(),
    'SD':d[:d.shape[0]]['4'].std(),
    'median':d[:d.shape[0]]['4'].median(),
    'MAD':d[:d.shape[0]]['4'].mad()
}
df = pd.DataFrame(myDic)
# Passo 2
myDicResidualRemoved = {
    'Data type':['step 2 (AOR)'],
    'size':[passo0.shape[0]],
    'mean':passo0['4'].mean(),
    'SD':passo0['4'].std(),
    'median':passo0['4'].median(),
    'MAD':passo0['4'].mad()
}
dfResAOR = pd.DataFrame(myDicResidualRemoved)
dfResAOR.head()
# Passo 3 - Passo feito depois de ser gerado pelo scrip para os vizinhos do sensor.
myDicFinalImputation = {
    'Data type':['step 3'],
    'size':[teste.shape[0]],
    'mean':teste['4'].mean(),
    'SD':teste['4'].std(),
    'median':teste['4'].median(),
    'MAD':teste['4'].mad()
}
dffinalI = pd.DataFrame(myDicFinalImputation)
# Estatística do sensor - Média. Mediana. Desvio Padrão. MAD
print( df0.head() )
print( df.head() )
print( dfResAOR.head() )
print( dffinalI.head() )
```

Figure 5. Informação estatística de sensor

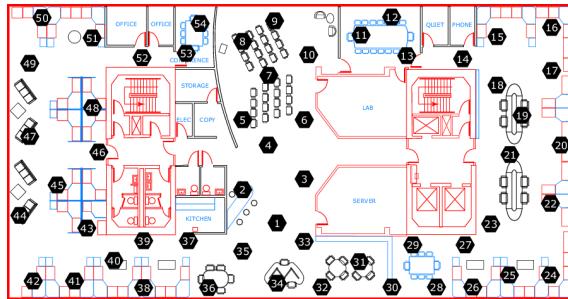


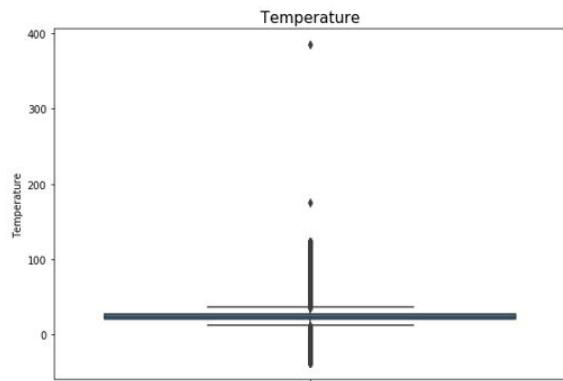
Figure 6. Distribuição dos sensores no laboratório

```

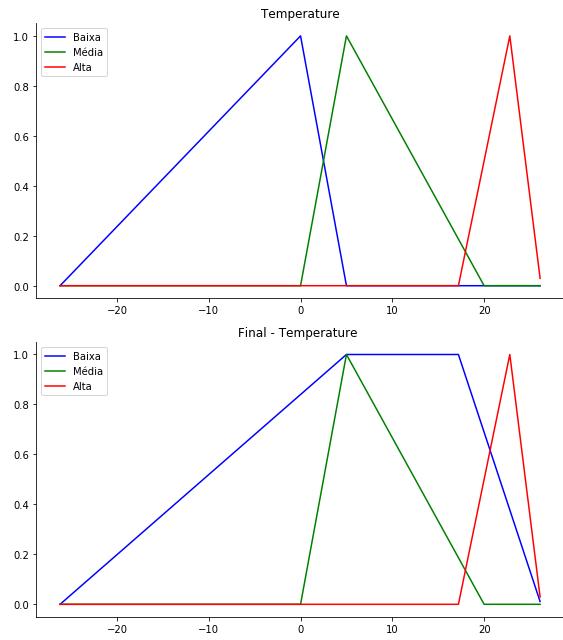
Dependência dos sensores
[1, [2, 33]]
[2, [1, 3]]
[3, [1, 4]]
[4, [2, 6]]
[6, [4, 2]]
[7, [8, 10]]
[8, [7, 9]]
[9, [8, 10]]
[10, [7, 9]]
[11, [12, 13]]
[12, [11, 13]]
[13, [11, 12]]
[14, [13, 18]]

```

**Figure 7. Exemplo de dependência entre os nós sensores**



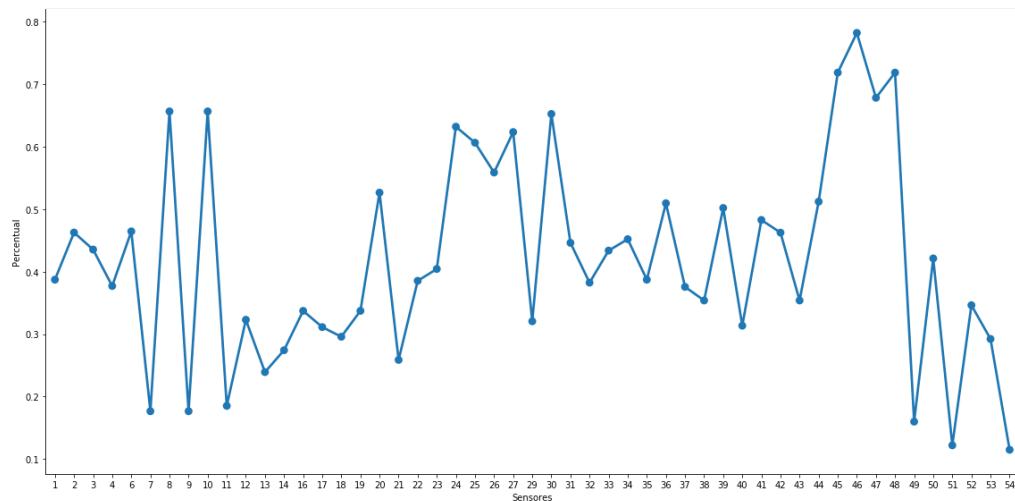
**Figure 8. Boxplot para a variável temperatura**



**Figure 9. Gráfico de pertinência para a variável temperatura**

| Sensores | Tamanho | Percentual |
|----------|---------|------------|
| 46       | 51272   | 0.782349   |
| 48       | 47105   | 0.718765   |
| 45       | 47105   | 0.718765   |
| 47       | 44465   | 0.678482   |
| 8        | 43035   | 0.656662   |
| 10       | 43035   | 0.656662   |
| 30       | 42766   | 0.652557   |
| 24       | 41429   | 0.632156   |
| 27       | 40868   | 0.623596   |
| 25       | 39750   | 0.606537   |
| 26       | 36621   | 0.558792   |
| 20       | 34511   | 0.526596   |
| 44       | 33566   | 0.512177   |
| 36       | 33397   | 0.509598   |
| 39       | 32902   | 0.502045   |
| 41       | 31641   | 0.482803   |
| 6        | 30445   | 0.464554   |
| 2        | 30324   | 0.462708   |
| 42       | 30313   | 0.462540   |
| 34       | 29627   | 0.452072   |

**Figure 10.** As 20 maiores interseções de épocas entre os vizinhos para cada sensor



**Figure 11.** Interseções de épocas entre os vizinhos usados no FI para cada sensor

|                              |                     |
|------------------------------|---------------------|
| Dados Brutos (Média)         | 38.537282627450985  |
| Passo 1 (Média)              | 21.17358688235294   |
| Passo 2 (Média)              | 21.17368988235294   |
| Passo 3 (Média)              | 19.121089607843135  |
| Dados Brutos (Desvio Padrão) | 10.22819222239568   |
| Passo 1 (Desvio Padrão)      | 1.2497044054204085  |
| Passo 2 (Desvio Padrão)      | 1.2031993838631772  |
| Passo 3 (Desvio Padrão)      | 0.6665920370056875  |
| Dados Brutos (Mediana)       | 22.4482             |
| Passo 1 (Mediana)            | 21.3604             |
| Passo 2 (Mediana)            | 21.3604             |
| Passo 3 (Mediana)            | 17.1                |
| Dados Brutos (MAD)           | 9.158628987312573   |
| Passo 1 (MAD)                | 0.38839618685121136 |
| Passo 2 (MAD)                | 0.37891316339869263 |
| Passo 3 (MAD)                | 0.23042622914263755 |

**Figure 12. Informação estatística básica de todos os sensores antes e depois da remoção dos outliers e imputação dos dados nulos (SD=Desvio Padrão, MAD=Desvio médio absoluto sobre a mediana; ARO e DRO=antes e depois da remoção dos outliers, respectivamente)**

| Sensor | Dados Brutos (Tam.) | Passo 1 (Tam.) | Passo 2 (Tam.) | Passo 3 (Tam.) | Dados Brutos (Média) | Passo 1 (Média) | Passo 2 (Média) | Passo 3 (Média) | Dados Brutos (Desvio Padrão) | Passo 1 (Desvio Padrão) | Passo 2 (Desvio Padrão) | Passo 3 (Desvio Padrão) | Dados Brutos (Mediana) | Passo 1 (Mediana) | Passo 2 (Mediana) | Passo 3 (Mediana) | Dados Brutos (MAD) | Passo 1 (MAD) | Passo 2 (MAD) | Passo 3 (MAD) |
|--------|---------------------|----------------|----------------|----------------|----------------------|-----------------|-----------------|-----------------|------------------------------|-------------------------|-------------------------|-------------------------|------------------------|-------------------|-------------------|-------------------|--------------------|---------------|---------------|---------------|
| 1      | 43047               | 33963          | 32795          | 65536          | 35.882437            | 21.619922       | 21.621043       | 19.355837       | 33.655123                    | 1.873518                | 1.880624                | 2.62225                 | 22.1444                | 21.6348           | 21.6446           | 17.1              | 23.397582          | 1.471464      | 1.477494      | 2.365429      |
| 2      | 46915               | 35739          | 35660          | 65536          | 34.574593            | 21.76821        | 21.76986        | 19.610636       | 32.536711                    | 1.630443                | 1.625275                | 2.614852                | 21.968                 | 21.9092           | 21.919            | 19.2436           | 21.845785          | 1.294003      | 1.291603      | 2.406085      |
| 3      | 46915               | 35739          | 35660          | 65536          | 34.574593            | 21.76821        | 21.76986        | 19.610636       | 32.536711                    | 1.630443                | 1.625275                | 2.614852                | 21.968                 | 21.9092           | 21.919            | 19.2436           | 21.845785          | 1.294003      | 1.291603      | 2.406085      |
| 4      | 43793               | 31086          | 31036          | 65536          | 45.46441             | 21.895891       | 21.897452       | 19.371938       | 41.776798                    | 1.752459                | 1.718102                | 2.671292                | 22.605                 | 21.9288           | 21.9288           | 17.1              | 35.146679          | 1.306499      | 1.306499      | 2.446204      |
| 6      | 35666               | 31160          | 30941          | 65536          | 27.561589            | 17.201921       | 17.40031        | 17.292734       | 31.59153                     | 9.682337                | 9.28285                 | 6.277796                | 21.625                 | 21.2722           | 21.282            | 17.1              | 18.133299          | 7.072025      | 6.790397      | 3.241562      |
| 7      | 55361               | 42880          | 42529          | 65536          | 41.937863            | 21.587396       | 21.585565       | 19.997151       | 39.674927                    | 1.833245                | 1.834918                | 2.602306                | 22.1052                | 21.5956           | 21.5956           | 20.1648           | 31.879996          | 1.425963      | 1.427399      | 2.294049      |
| 8      | 16778               | 16751          | 16691          | 65536          | 21.326354            | 21.235066       | 21.235887       | 18.152798       | 21.3506                      | 2.17833                 | 2.212344                | 2.1194                  | 21.3506                | 21.3506           | 21.3506           | 17.1              | 1.882725           | 1.79607       | 1.795186      | 1.593237      |
| 9      | 49890               | 39868          | 39557          | 65536          | 40.226775            | 21.484627       | 21.481689       | 19.733827       | 38.959621                    | 2.050303                | 2.05075                 | 2.671914                | 22.1052                | 21.4878           | 21.478            | 19.3514           | 30.231795          | 1.654633      | 1.65493       | 2.35021       |
| 10     | 47165               | 34856          | 34598          | 65536          | 46.98509             | 21.355438       | 21.350222       | 19.340431       | 43.743121                    | 1.921471                | 1.921471                | 2.538148                | 22.1542                | 21.3604           | 21.3604           | 18.2734           | 37.981664          | 1.539473      | 1.537184      | 2.261404      |
| 11     | 41835               | 33890          | 33675          | 65536          | 38.659552            | 20.814063       | 20.808312       | 19.002857       | 38.403151                    | 2.108741                | 2.10757                 | 2.390732                | 21.6838                | 20.9488           | 20.9488           | 17.1              | 29.125805          | 1.733777      | 1.733373      | 2.1126        |
| 12     | 23078               | 19611          | 19480          | 65536          | 32.905798            | 20.330601       | 20.325919       | 18.056503       | 32.824377                    | 2.373884                | 2.37056                 | 1.959516                | 20.9684                | 20.3804           | 20.3804           | 17.1              | 21.50122           | 1.96041       | 1.95838       | 1.470422      |
| 13     | 31505               | 25115          | 24907          | 65536          | 39.105833            | 20.645832       | 20.646885       | 18.449201       | 38.500415                    | 2.263277                | 2.240381                | 2.201415                | 21.2232                | 20.6352           | 20.6352           | 17.1              | 29.603541          | 1.610155      | 1.603163      | 1.801888      |
| 14     | 31804               | 26360          | 26195          | 65536          | 36.538528            | 21.234938       | 21.23385        | 18.749665       | 35.514003                    | 2.187352                | 2.18612                 | 2.450348                | 21.8504                | 21.3898           | 21.38             | 17.1              | 25.597664          | 1.733408      | 1.732799      | 2.117654      |
| 16     | 34600               | 24910          | 24574          | 65536          | 45.529473            | 20.446843       | 20.438721       | 18.351791       | 43.610766                    | 2.461845                | 2.460814                | 2.20971                 | 21.7622                | 20.6646           | 20.6548           | 17.1              | 37.420202          | 1.984008      | 1.983079      | 1.786066      |
| 17     | 33786               | 26632          | 26456          | 65536          | 38.408482            | 21.37461        | 21.372901       | 18.824516       | 37.493668                    | 2.173808                | 2.173697                | 2.510466                | 22.2522                | 21.6054           | 21.6054           | 17.1              | 27.881832          | 1.74381       | 1.743576      | 2.186949      |
| 18     | 35312               | 23663          | 23600          | 65536          | 47.102248            | 18.935807       | 18.939128       | 17.76223        | 45.137586                    | 7.066411                | 7.061043                | 4.328024                | 23.4478                | 20.9684           | 20.9684           | 17.1              | 39.372971          | 4.86427       | 4.858808      | 2.388224      |

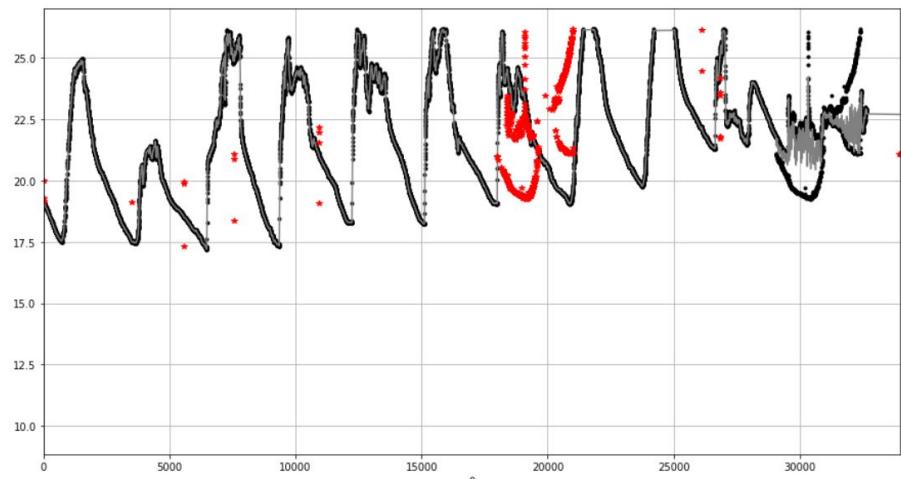
**Figure 13. Série temporal dos sensores 1-18. Informação estatística básica dos dados antes e depois da remoção dos outliers e imputação dos dados nulos (SD=Desvio Padrão, MAD=Desvio médio absoluto sobre a mediana; ARO e DRO=antes e depois da remoção dos outliers, respectivamente)**

| Sensor | Dados Brutos (Tam.) | Passo 1 (Tam.) | Passo 2 (Tam.) | Passo 3 (Tam.) | Dados Brutos (Média) | Passo 1 (Média) | Passo 2 (Média) | Passo 3 (Média) | Dados Brutos (Desvio Padrão) | Passo 1 (Desvio Padrão) | Passo 2 (Desvio Padrão) | Passo 3 (Desvio Padrão) | Dados Brutos (Mediana) | Passo 1 (Mediana) | Passo 2 (Mediana) | Passo 3 (Mediana) | Dados Brutos (MAD) | Passo 1 (MAD) | Passo 2 (MAD) | Passo 3 (MAD) |
|--------|---------------------|----------------|----------------|----------------|----------------------|-----------------|-----------------|-----------------|------------------------------|-------------------------|-------------------------|-------------------------|------------------------|-------------------|-------------------|-------------------|--------------------|---------------|---------------|---------------|
| 19     | 39459               | 25954          | 25777          | 65536          | 47.167461            | 22.006552       | 22.0025         | 19.028098       | 42.763208                    | 2.337638                | 2.336413                | 2.807467                | 23.9084                | 22.0562           | 22.0464           | 17.1              | 36.721548          | 1.930917      | 1.929667      | 2.446816      |
| 20     | 28835               | 25472          | 25428          | 65536          | 22.351076            | 21.645188       | 21.650691       | 18.864875       | 3.985533                     | 2.837276                | 2.763068                | 2.806615                | 22.4482                | 21.9876           | 21.9876           | 17.1              | 2.7067             | 2.35671       | 2.350856      | 2.360625      |
| 21     | 58525               | 37697          | 37259          | 65536          | 37.639654            | 21.769181       | 21.765837       | 19.752492       | 33.65854                     | 2.295683                | 2.296316                | 2.887581                | 23.781                 | 21.7818           | 21.7916           | 18.8418           | 23.378782          | 1.864722      | 1.865333      | 2.594592      |
| 22     | 60165               | 41603          | 41344          | 65536          | 31.319529            | 21.468746       | 21.459558       | 19.794418       | 26.418827                    | 2.428                   | 2.424977                | 2.847928                | 23.0068                | 21.6348           | 21.6348           | 19.0672           | 14.515759          | 1.97435       | 1.972353      | 2.548048      |
| 23     | 62440               | 37832          | 37789          | 65536          | 44.647866            | 21.8248         | 21.855126       | 19.832504       | 40.318131                    | 2.940081                | 2.626079                | 3.079925                | 24.1779                | 21.9092           | 21.9092           | 19.1946           | 32.960926          | 1.867724      | 1.837242      | 2.645573      |
| 24     | 57362               | 29928          | 29581          | 65536          | 49.759221            | 21.497619       | 21.474618       | 19.073982       | 43.102656                    | 2.507693                | 2.500983                | 2.749758                | 25.5254                | 21.7916           | 21.7524           | 17.1              | 37.821871          | 2.053213      | 2.047503      | 2.415269      |
| 25     | 53175               | 29272          | 29181          | 65536          | 41.684587            | 21.184517       | 21.221842       | 18.899722       | 37.256582                    | 2.733089                | 2.449997                | 2.608296                | 24.614                 | 21.2722           | 21.2722           | 17.1              | 28.3206            | 2.069428      | 2.029404      | 2.229431      |
| 26     | 61521               | 35189          | 35134          | 65536          | 50.808895            | 21.369683       | 21.369874       | 19.144398       | 44.252027                    | 2.548792                | 2.527084                | 2.744752                | 24.663                 | 21.3408           | 21.3408           | 17.1              | 39.711039          | 2.093019      | 2.089984      | 2.367083      |
| 27     | 37628               | 22858          | 22552          | 65536          | 47.523634            | 21.945422       | 21.938228       | 18.754142       | 42.223143                    | 2.205723                | 2.204243                | 2.631496                | 24.1436                | 21.9582           | 21.9582           | 17.1              | 36.129209          | 1.783327      | 1.7829        | 2.234313      |
| 29     | 64391               | 40954          | 40285          | 65536          | 48.166542            | 21.346821       | 21.327049       | 19.494429       | 43.556083                    | 2.396856                | 2.398989                | 2.749006                | 23.4576                | 21.3408           | 21.3114           | 18.4008           | 38.016229          | 1.970843      | 1.972181      | 2.427512      |
| 30     | 41129               | 21893          | 21514          | 65536          | 55.795512            | 20.930782       | 20.905934       | 18.313067       | 46.542893                    | 2.47366                 | 2.461798                | 2.250366                | 25.2314                | 20.8116           | 20.7822           | 17.1              | 43.564507          | 2.048868      | 2.037073      | 1.750042      |
| 31     | 65694               | 43125          | 42807          | 65536          | 41.774647            | 21.32703        | 21.319163       | 19.736719       | 38.937329                    | 2.389638                | 2.388252                | 2.790124                | 23.1244                | 21.1448           | 21.135            | 19.1652           | 30.577113          | 1.963964      | 1.96264       | 2.422639      |
| 32     | 43093               | 25068          | 25006          | 65536          | 51.507564            | 20.891485       | 20.892183       | 18.515512       | 45.038235                    | 2.159978                | 2.125033                | 2.247443                | 23.2812                | 20.8018           | 20.8018           | 17.1              | 40.781993          | 1.730159      | 1.726357      | 1.869728      |
| 33     | 35749               | 23167          | 22971          | 65536          | 48.635339            | 21.682197       | 21.672074       | 18.694505       | 43.570363                    | 1.926743                | 1.919755                | 2.456128                | 23.0264                | 21.6936           | 21.6936           | 17.1              | 38.315264          | 1.50922       | 1.503228      | 2.115348      |
| 34     | 48786               | 30727          | 30355          | 65536          | 50.525772            | 21.161829       | 21.157303       | 18.976037       | 44.693619                    | 2.049063                | 2.048232                | 2.433975                | 22.948                 | 21.1154           | 21.1056           | 17.1              | 40.175438          | 1.602168      | 1.601664      | 2.143681      |
| 35     | 51338               | 34453          | 34266          | 65536          | 47.292161            | 21.639631       | 21.637883       | 19.472273       | 42.765895                    | 1.77796                 | 1.777201                | 2.605525                | 22.8892                | 21.8112           | 21.8063           | 18.1166           | 36.861071          | 1.409684      | 1.409317      | 2.408228      |
| 36     | 57743               | 41152          | 40686          | 65536          | 40.865096            | 21.369379       | 21.369379       | 21.369379       | 38.456873                    | 2.090523                | 2.090523                | 2.644491                | 22.4776                | 21.3016           | 21.3016           | 19.4592           | 29.80327           | 1.695508      | 1.692544      | 2.34077       |

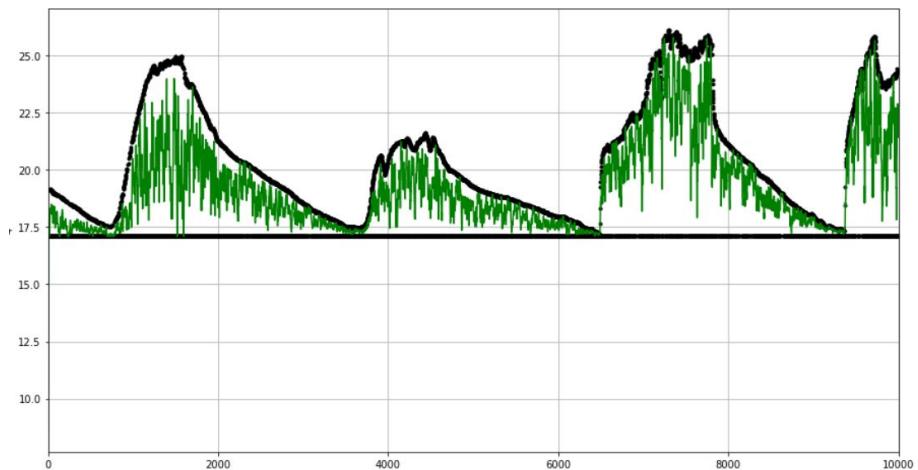
**Figure 14. Série temporal dos sensores 19-36. Informação estatística básica dos dados antes e depois da remoção dos outliers e imputação dos dados nulos (SD=Desvio Padrão, MAD=Desvio médio absoluto sobre a mediana; ARO e DRO=antes e depois da remoção dos outliers, respectivamente)**

| Sensor | Dados Brutos (Tam.) | Passo 1 (Tam.) | Passo 2 (Tam.) | Passo 3 (Tam.) | Dados Brutos (Média) | Passo 1 (Média) | Passo 2 (Média) | Passo 3 (Média) | Dados Brutos (Desvio Padrão) | Passo 1 (Desvio Padrão) | Passo 2 (Desvio Padrão) | Passo 3 (Desvio Padrão) | Dados Brutos (Mediana) | Passo 1 (Mediana) | Passo 2 (Mediana) | Passo 3 (Mediana) | Dados Brutos (MAD) | Passo 1 (MAD) | Passo 2 (MAD) | Passo 3 (MAD) |
|--------|---------------------|----------------|----------------|----------------|----------------------|-----------------|-----------------|-----------------|------------------------------|-------------------------|-------------------------|-------------------------|------------------------|-------------------|-------------------|-------------------|--------------------|---------------|---------------|---------------|
| 37     | 47929               | 31778          | 31513          | 65536          | 49.4906              | 21.985043       | 21.980511       | 19.446146       | 44.144517                    | 1.920885                | 1.921944                | 2.778805                | 23.4772                | 22.1934           | 22.1836           | 17.1              | 39.134108          | 1.56662       | 1.568226      | 2.550153      |
| 38     | 49161               | 30523          | 30141          | 65536          | 44.595351            | 21.593933       | 21.583477       | 19.160918       | 39.58897                     | 2.346463                | 2.346318                | 2.742934                | 23.7712                | 21.478            | 21.4584           | 17.1              | 32.492502          | 1.905707      | 1.906147      | 2.403442      |
| 39     | 32740               | 24285          | 24042          | 65536          | 41.384376            | 21.984395       | 21.982283       | 18.889086       | 38.751849                    | 2.165411                | 2.166471                | 2.693513                | 23.193                 | 22.066            | 22.066            | 17.1              | 30.262676          | 1.790277      | 1.792222      | 2.326731      |
| 40     | 46624               | 34825          | 34378          | 65536          | 36.552531            | 21.634705       | 21.629605       | 19.472837       | 34.172157                    | 2.180979                | 2.179419                | 2.75807                 | 22.8598                | 21.8014           | 21.8014           | 17.6266           | 23.651935          | 1.798405      | 1.797643      | 2.489045      |
| 41     | 40837               | 31398          | 31040          | 65536          | 31.827641            | 21.644312       | 21.634062       | 19.246537       | 28.680629                    | 2.551672                | 2.553115                | 2.865505                | 22.9284                | 21.6544           | 21.6348           | 17.1              | 16.658891          | 2.159147      | 2.160892      | 2.496         |
| 42     | 45019               | 35286          | 34733          | 65536          | 34.394834            | 21.205443       | 21.183017       | 19.253882       | 32.863623                    | 2.545055                | 2.542555                | 2.750906                | 22.4972                | 21.4584           | 21.4388           | 17.1              | 21.577814          | 2.13753       | 2.135349      | 2.430442      |
| 43     | 41050               | 31406          | 31092          | 65536          | 36.701885            | 21.696393       | 21.687307       | 19.27601        | 34.806069                    | 2.271409                | 2.269541                | 2.773091                | 22.8696                | 21.8112           | 21.8112           | 17.1              | 24.370739          | 1.866246      | 1.865238      | 2.487376      |
| 44     | 48632               | 40197          | 39673          | 65536          | 28.359107            | 21.093174       | 21.074618       | 19.504476       | 24.082571                    | 2.55224                 | 2.549359                | 2.77652                 | 21.7132                | 21.086            | 21.0664           | 18.538            | 12.151971          | 2.086081      | 2.082512      | 2.423468      |
| 45     | 53245               | 48818          | 48533          | 65536          | 21.846607            | 21.176797       | 21.170896       | 20.111753       | 32.93981                     | 2.206145                | 2.204307                | 2.605045                | 21.5564                | 21.331            | 21.331            | 20.2653           | 2.269401           | 1.757264      | 1.756129      | 2.264694      |
| 46     | 52988               | 41585          | 41260          | 65536          | 32.594331            | 21.070001       | 21.06054        | 19.593158       | 29.60603                     | 2.132941                | 2.129385                | 2.551994                | 21.8112                | 21.2036           | 21.1938           | 19.5376           | 18.703295          | 1.667195      | 1.664753      | 2.275828      |
| 47     | 57201               | 50584          | 49750          | 65536          | 25.587461            | 20.972032       | 20.956711       | 20.02568        | 19.650956                    | 2.279332                | 2.281443                | 2.583568                | 21.3604                | 20.9194           | 20.8998           | 20.0374           | 8.172135           | 1.841431      | 1.842796      | 2.206385      |
| 48     | 58226               | 53073          | 52772          | 65536          | 25.321222            | 20.920086       | 20.91474        | 20.16805        | 19.787776                    | 2.13349                 | 2.131264                | 2.438539                | 21.1938                | 20.9096           | 20.9096           | 20.2628           | 8.033412           | 1.714935      | 1.713135      | 2.062598      |
| 49     | 41041               | 34388          | 34243          | 65536          | 31.280641            | 20.67848        | 20.695594       | 18.978175       | 29.9747                      | 2.570186                | 2.411614                | 2.500104                | 21.4388                | 20.7136           | 20.7136           | 17.1              | 18.05786           | 2.008424      | 1.988573      | 2.14804       |
| 50     | 15741               | 15173          | 15022          | 65536          | 20.363758            | 20.083302       | 20.077932       | 17.782594       | 3.180016                     | 2.870566                | 2.870261                | 1.858769                | 20.3412                | 20.1354           | 20.1256           | 17.1              | 2.609753           | 2.422296      | 2.423883      | 1.225121      |
| 51     | 42259               | 34098          | 33985          | 65536          | 34.306298            | 21.08421        | 21.082614       | 19.164431       | 33.289384                    | 2.199068                | 2.172151                | 2.530627                | 21.772                 | 21.0762           | 21.0762           | 17.1              | 22.158344          | 1.709392      | 1.705193      | 2.250194      |
| 52     | 34067               | 28270          | 28150          | 65536          | 33.416355            | 21.230853       | 21.230285       | 18.869157       | 31.67973                     | 2.251499                | 2.23403                 | 2.512812                | 22.0758                | 21.6348           | 21.6348           | 17.1              | 20.613796          | 1.779295      | 1.77408       | 2.219806      |
| 53     | 28131               | 23365          | 23133          | 65536          | 33.473715            | 20.610538       | 20.598563       | 18.331255       | 33.034509                    | 2.483034                | 2.47705                 | 2.225964                | 21.5956                | 21.086            | 21.086            | 17.1              | 21.744291          | 2.033914      | 2.030275      | 1.809791      |
| 54     | 35384               | 28431          | 28152          | 65536          | 39.656527            | 20.428997       | 20.417583       | 18.522122       | 39.740315                    | 2.631198                | 2.627372                | 2.378083                | 21.7132                | 20.7528           | 20.743            | 17.1              | 31.027428          | 2.168138      | 2.165031      | 1.981619      |

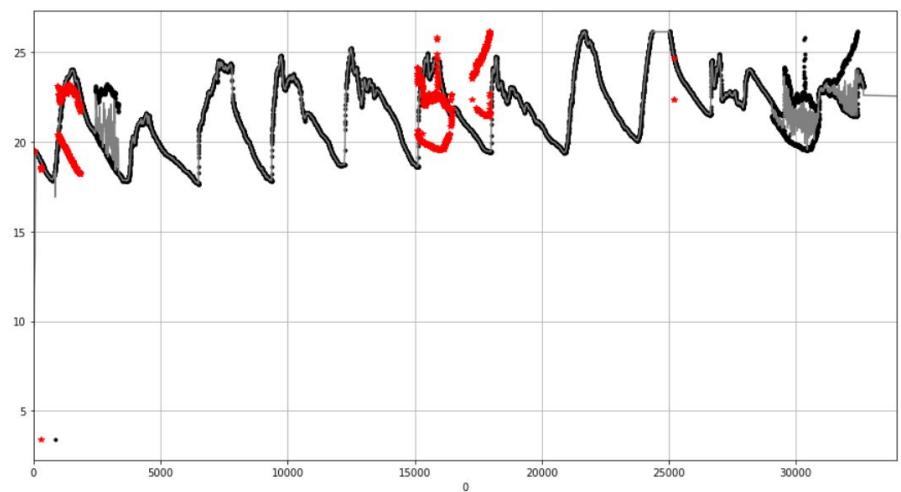
**Figure 15. Série temporal dos sensores 37-54. Informação estatística básica dos dados antes e depois da remoção dos outliers e imputação dos dados nulos (SD=Desvio Padrão, MAD=Desvio médio absoluto sobre a mediana; ARO e DRO=antes e depois da remoção dos outliers, respectivamente)**



**Figure 16.** série temporal do sensor 1 com os dados brutos (passo 1 (ARO))



**Figure 17.** série temporal do sensor 1 após a reconstrução dos dados (passo 3)



**Figure 18.** série temporal do sensor 2 com os dados brutos (passo 1 (ARO))

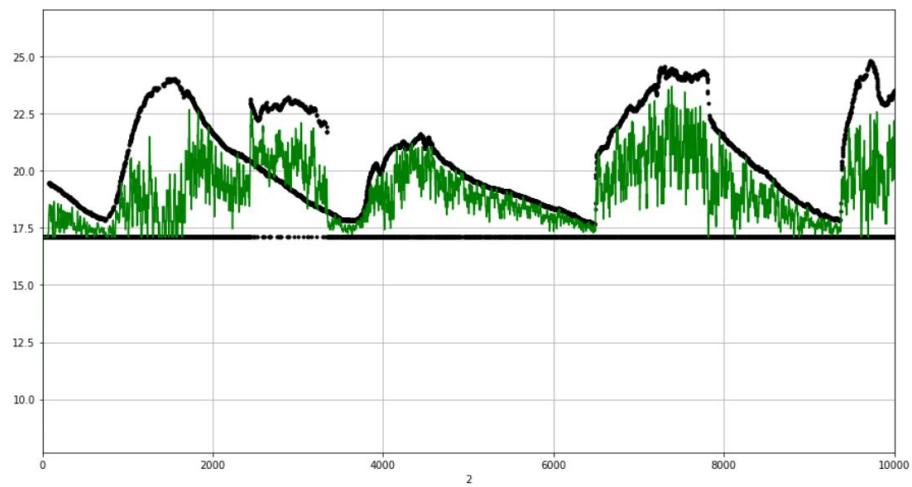


Figure 19. série temporal do sensor 2 após a reconstrução dos dados (passo 3)

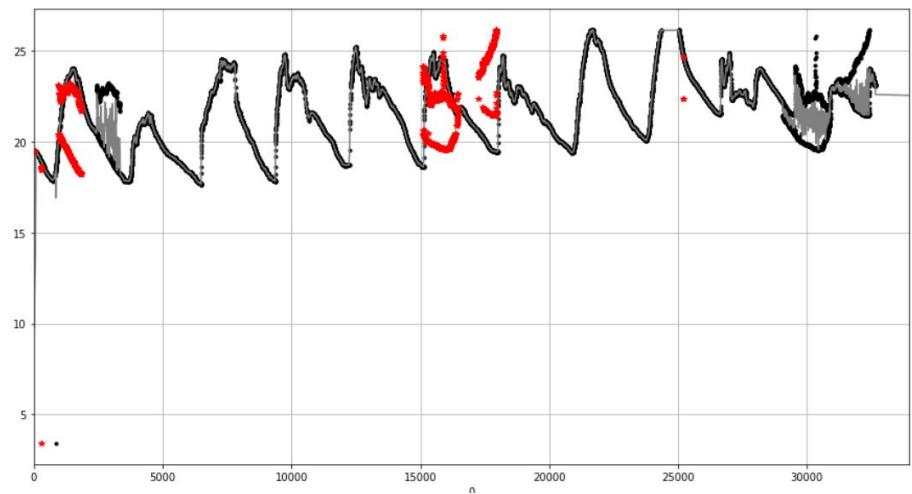


Figure 20. série temporal do sensor 3 com os dados brutos (passo 1 (ARO))

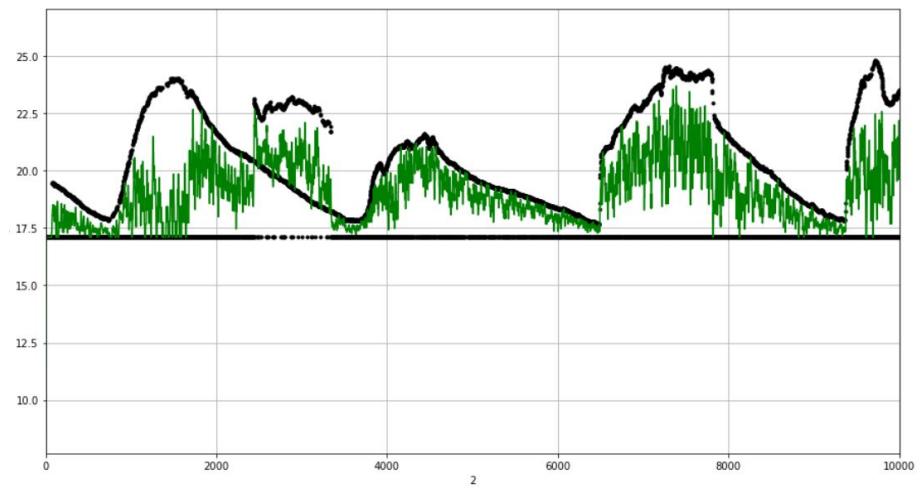
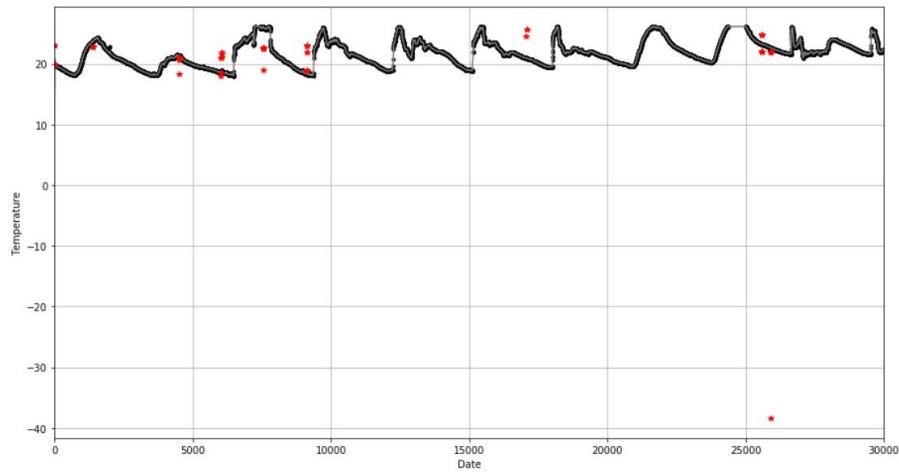
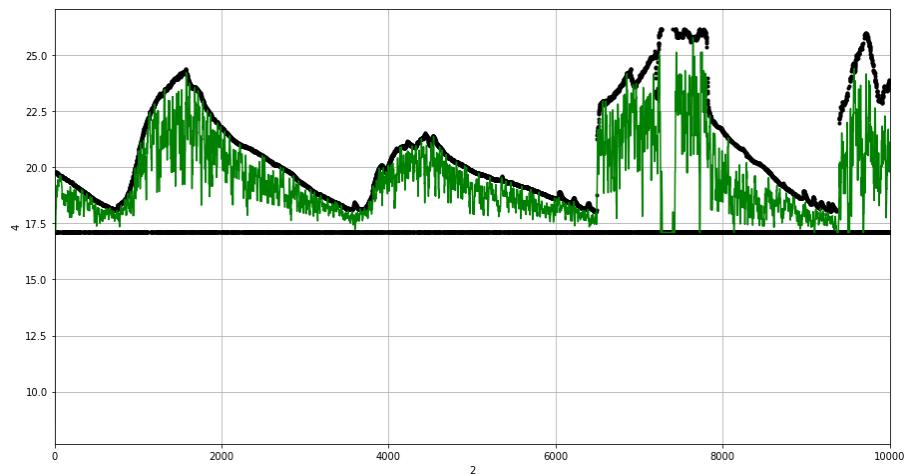


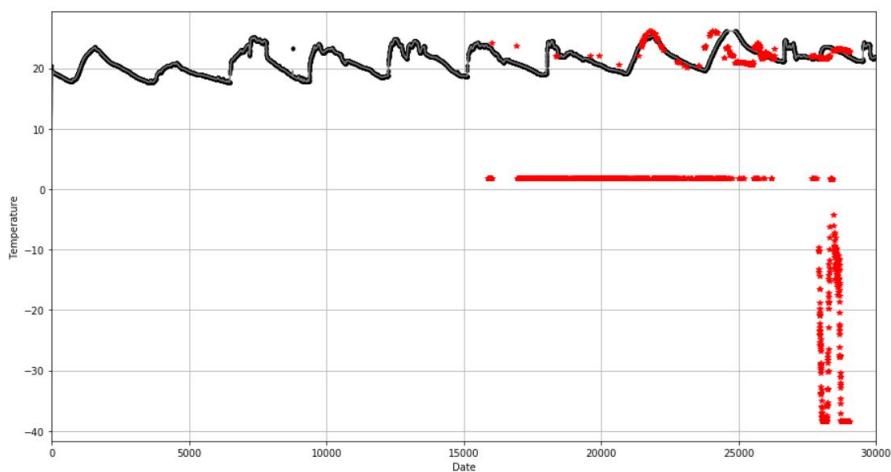
Figure 21. série temporal do sensor 3 após a reconstrução dos dados (passo 3)



**Figure 22. série temporal do sensor 4 com os dados brutos (passo 1 (ARO))**



**Figure 23. série temporal do sensor 4 após a reconstrução dos dados (passo 3)**



**Figure 24. série temporal do sensor 6 com os dados brutos (passo 1 (ARO))**

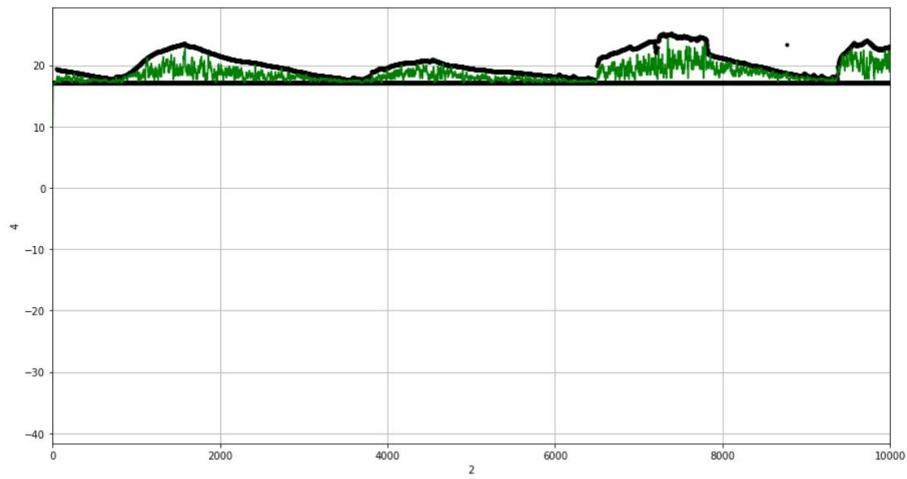


Figure 25. série temporal do sensor 6 após a reconstrução dos dados (passo 3)

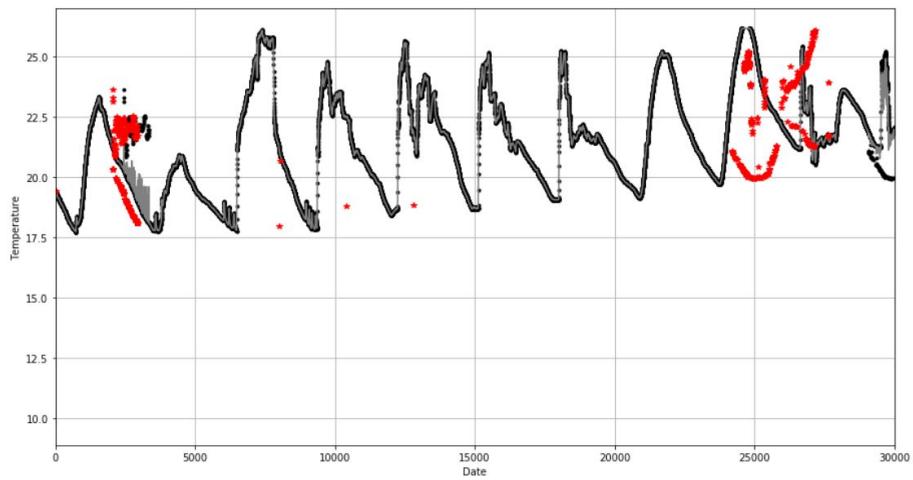


Figure 26. série temporal do sensor 7 com os dados brutos (passo 1 (ARO))

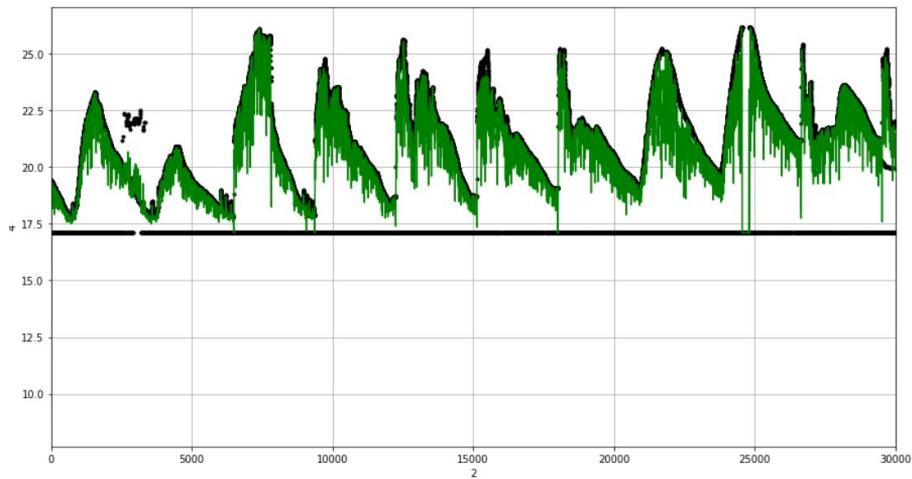
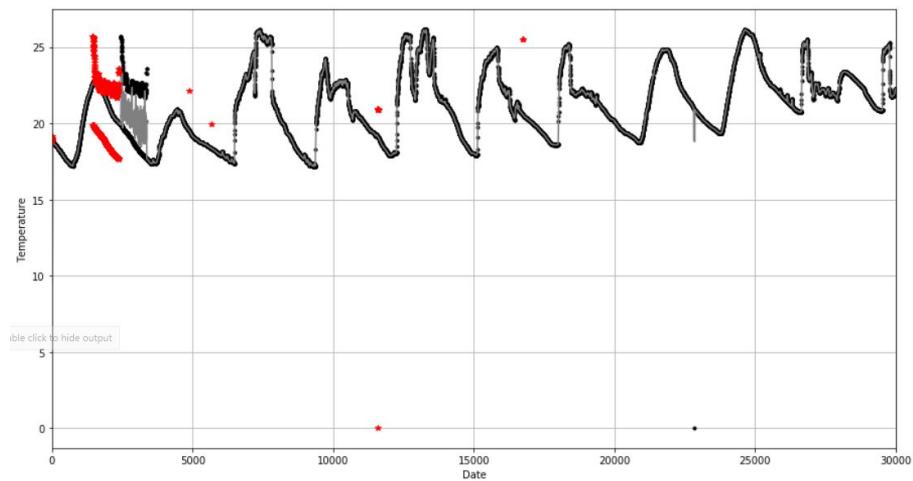
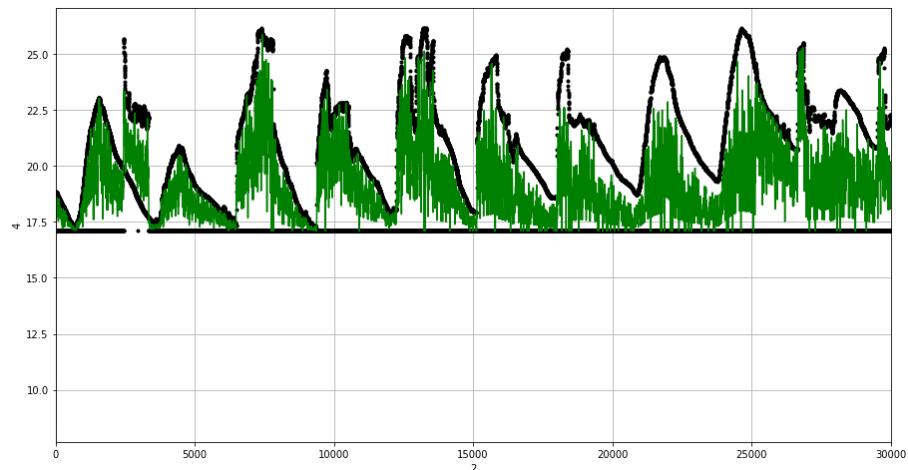


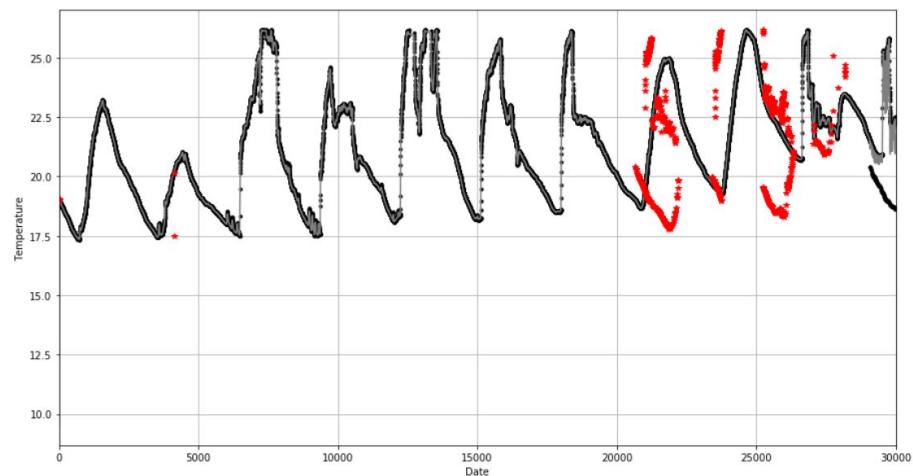
Figure 27. série temporal do sensor 7 após a reconstrução dos dados (passo 3)



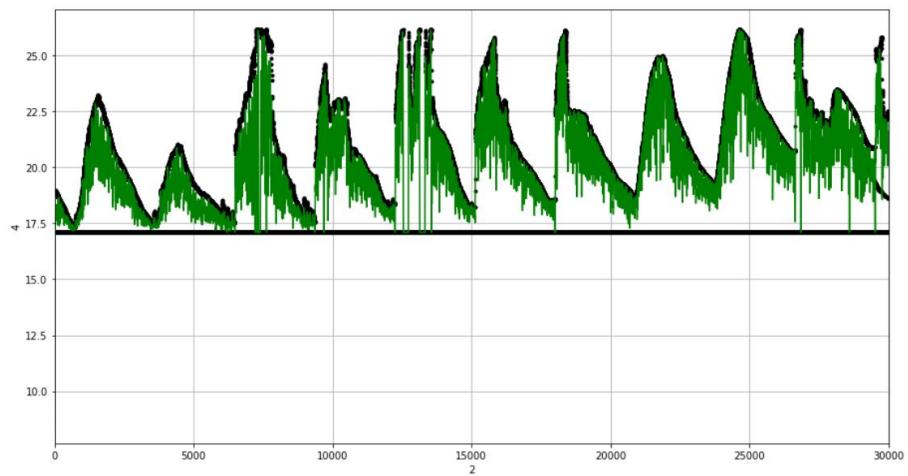
**Figure 28. série temporal do sensor 8 com os dados brutos (passo 1 (ARO))**



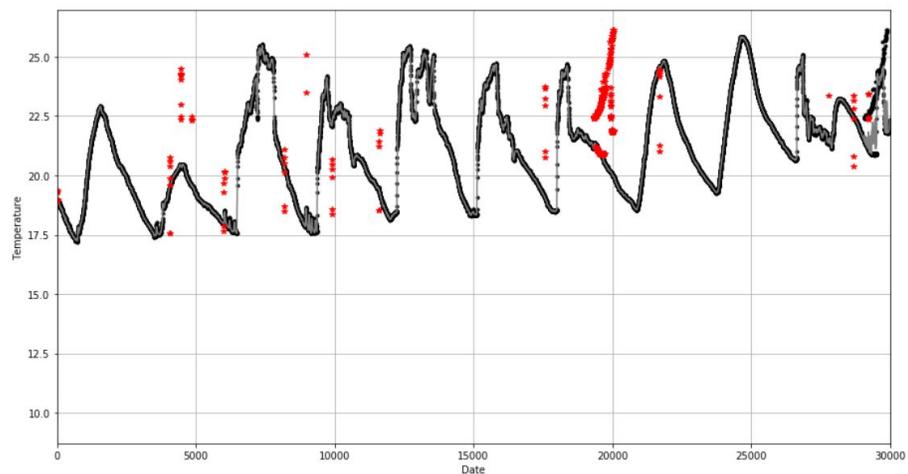
**Figure 29. série temporal do sensor 8 após a reconstrução dos dados (passo 3)**



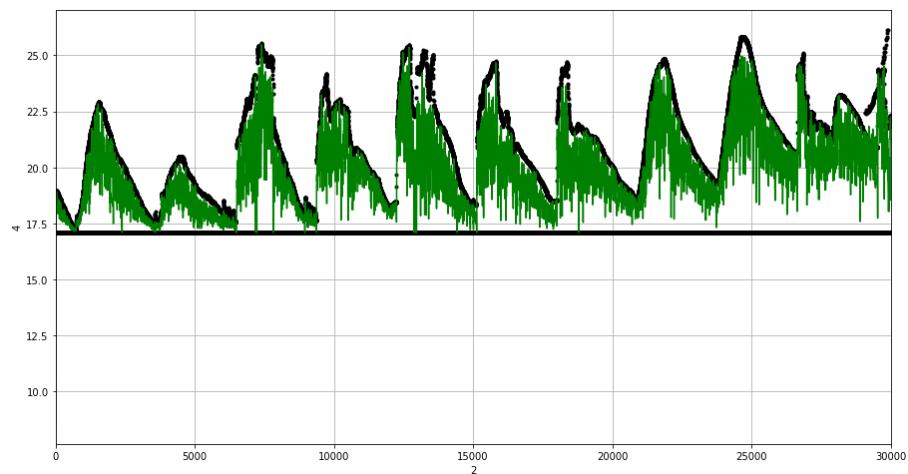
**Figure 30. série temporal do sensor 9 com os dados brutos (passo 1 (ARO))**



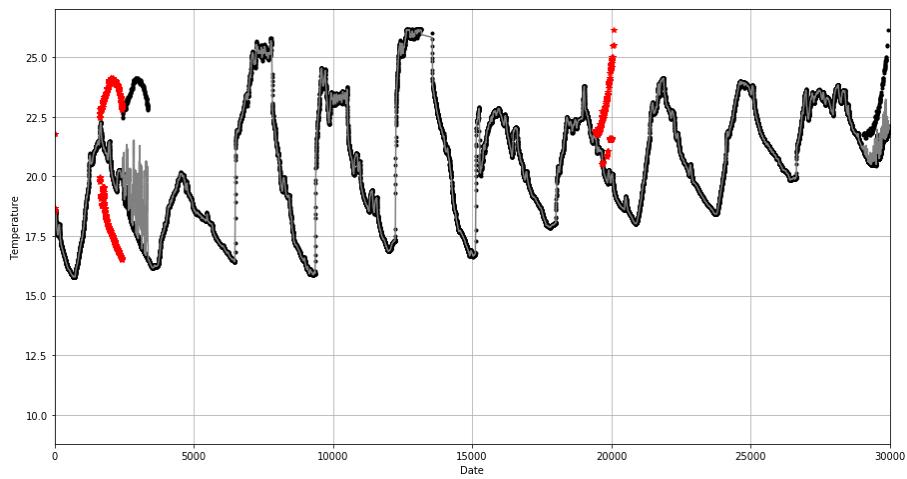
**Figure 31. série temporal do sensor 9 após a reconstrução dos dados (passo 3)**



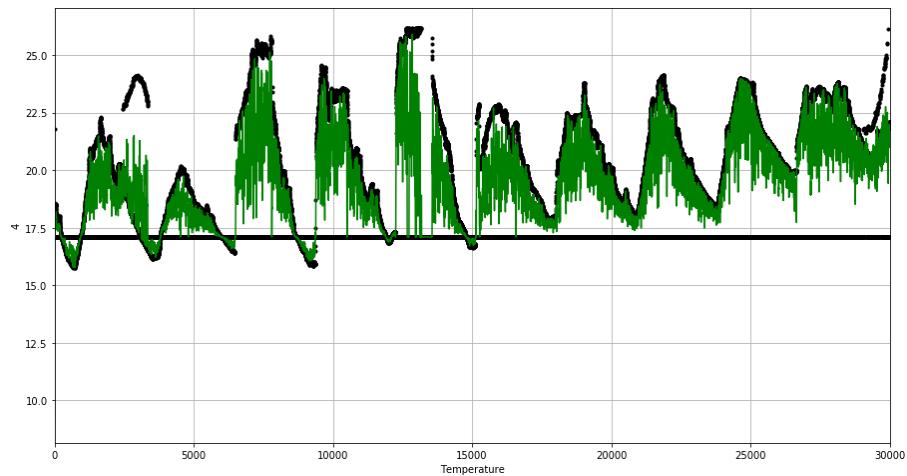
**Figure 32. série temporal do sensor 10 com os dados brutos (passo 1 (ARO))**



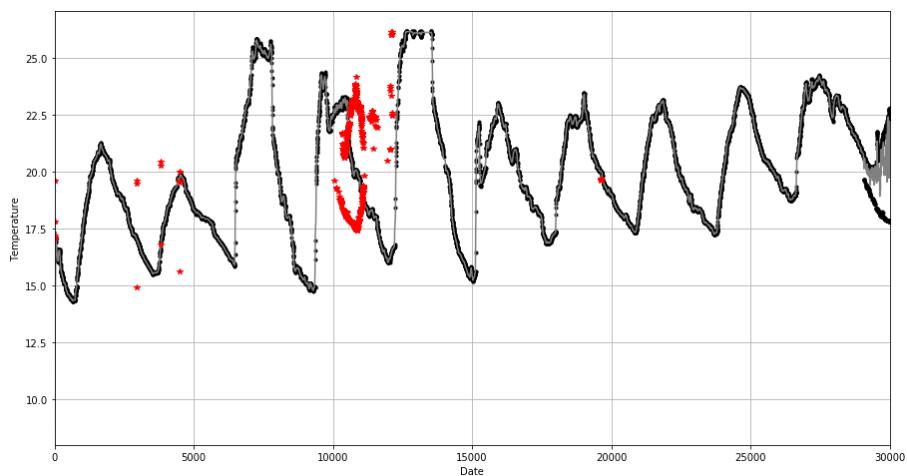
**Figure 33. série temporal do sensor 10 após a reconstrução dos dados (passo 3)**



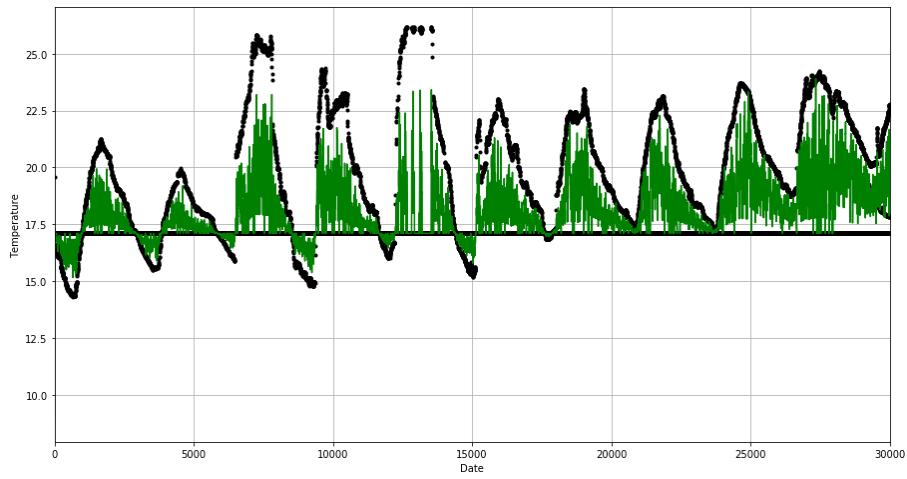
**Figure 34. série temporal do sensor 11 com os dados brutos (passo 1 (ARO))**



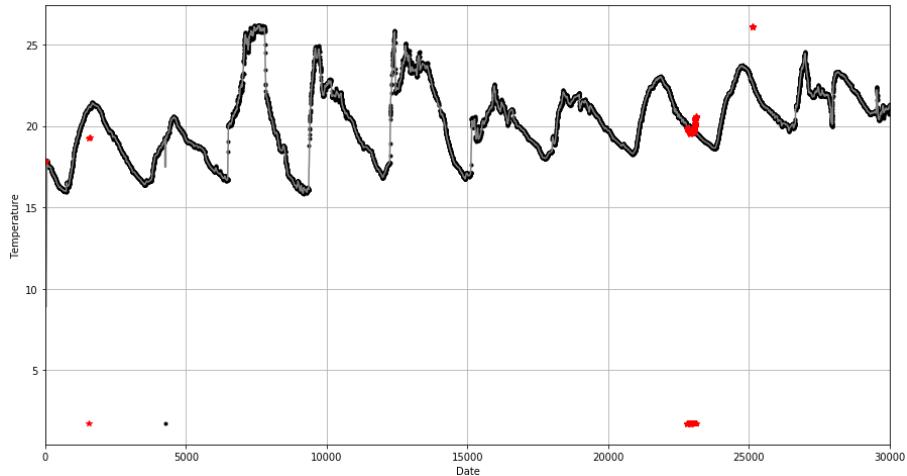
**Figure 35. série temporal do sensor 11 após a reconstrução dos dados (passo 3)**



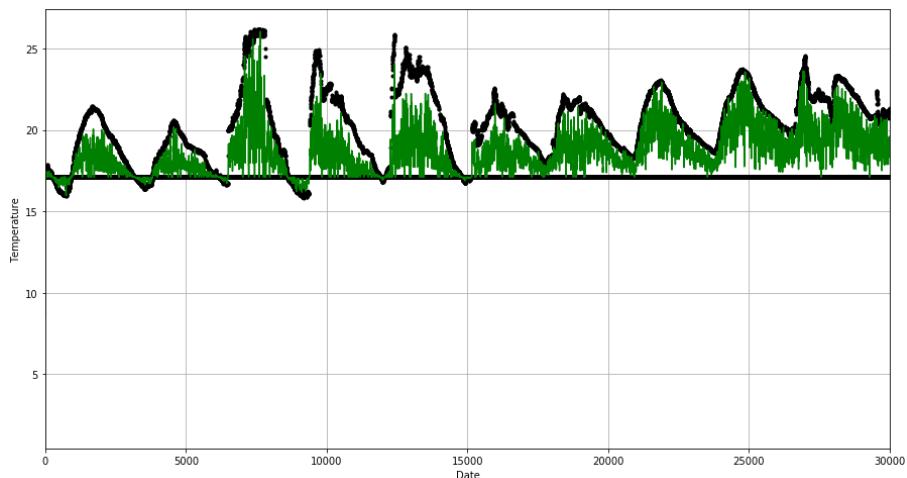
**Figure 36. série temporal do sensor 12 com os dados brutos (passo 1 (ARO))**



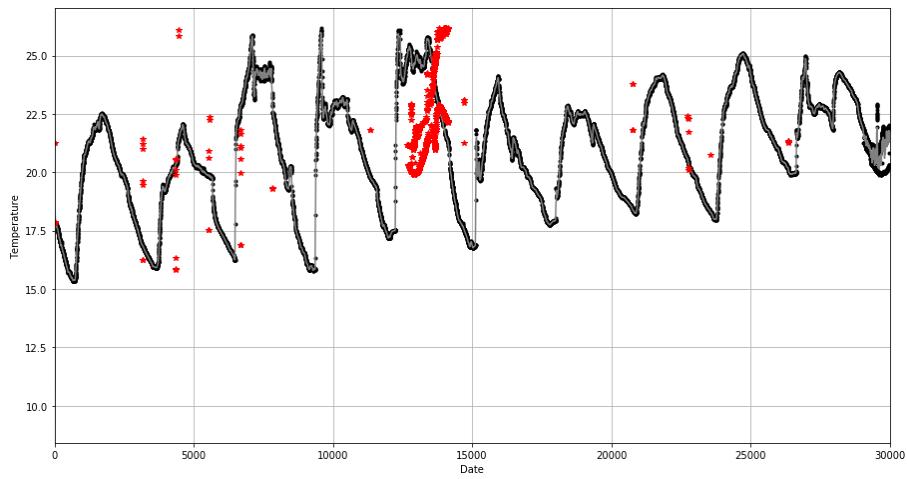
**Figure 37.** série temporal do sensor 12 após a reconstrução dos dados (passo 3)



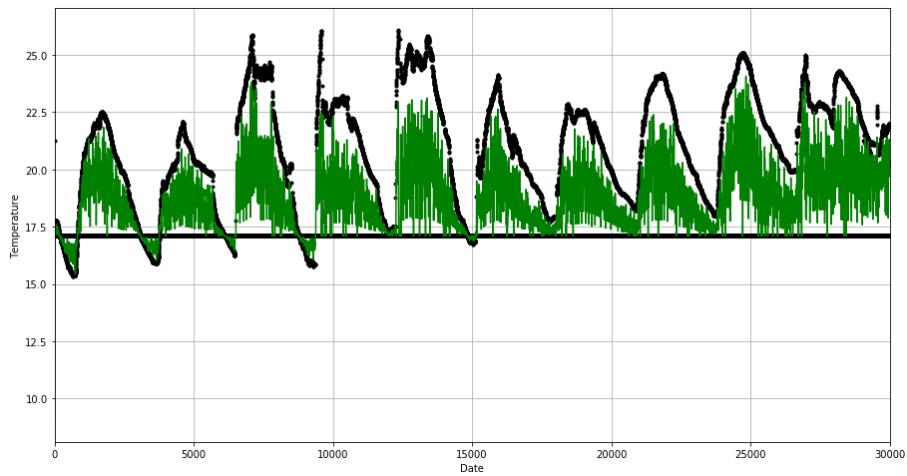
**Figure 38.** série temporal do sensor 13 com os dados brutos (passo 1 (ARO))



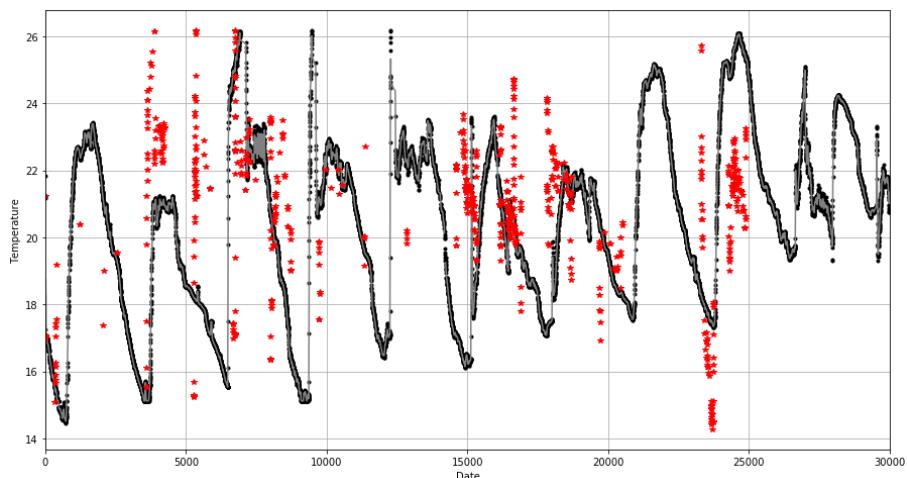
**Figure 39.** série temporal do sensor 13 após a reconstrução dos dados (passo 3)



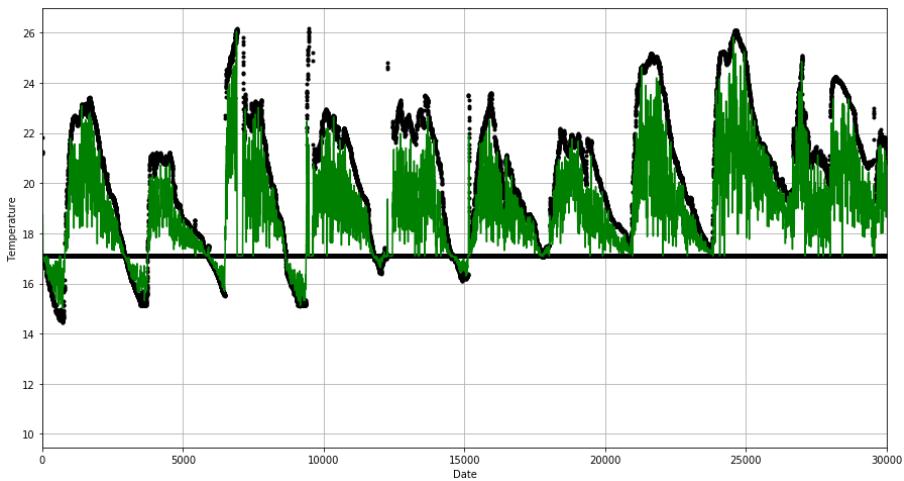
**Figure 40. série temporal do sensor 14 com os dados brutos (passo 1 (ARO))**



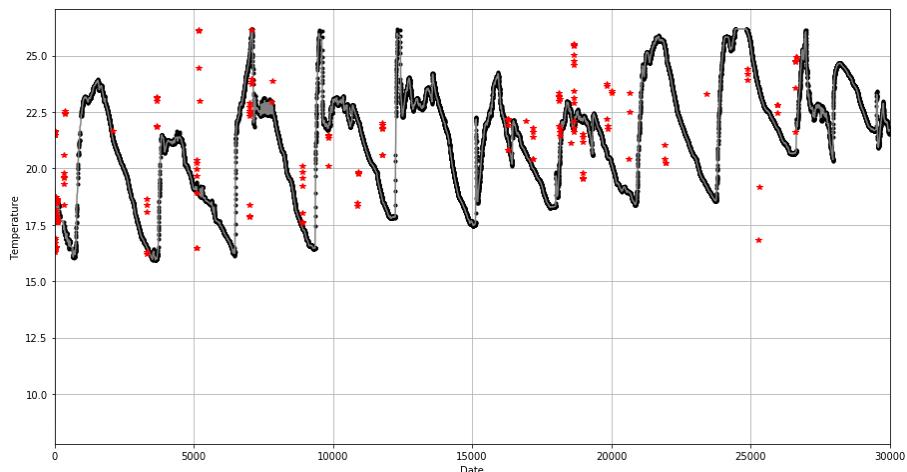
**Figure 41. série temporal do sensor 14 após a reconstrução dos dados (passo 3)**



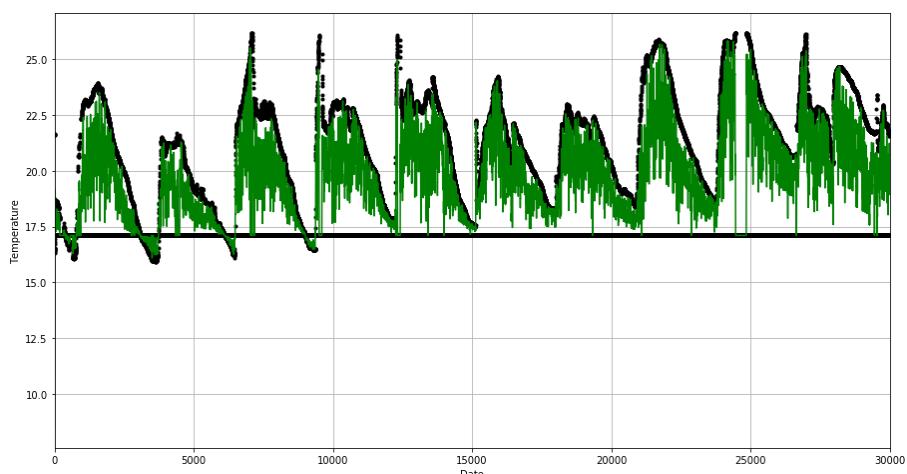
**Figure 42. série temporal do sensor 16 com os dados brutos (passo 1 (ARO))**



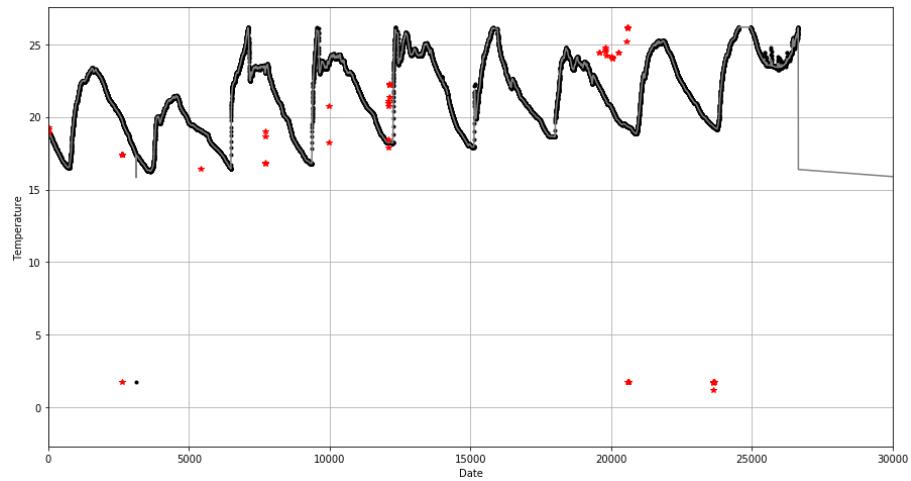
**Figure 43.** série temporal do sensor 16 após a reconstrução dos dados (passo 3)



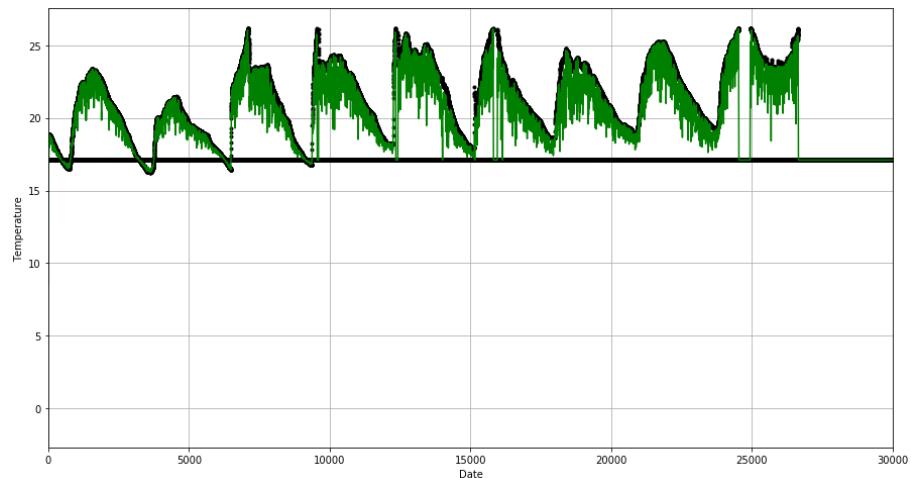
**Figure 44.** série temporal do sensor 17 com os dados brutos (passo 1 (ARO))



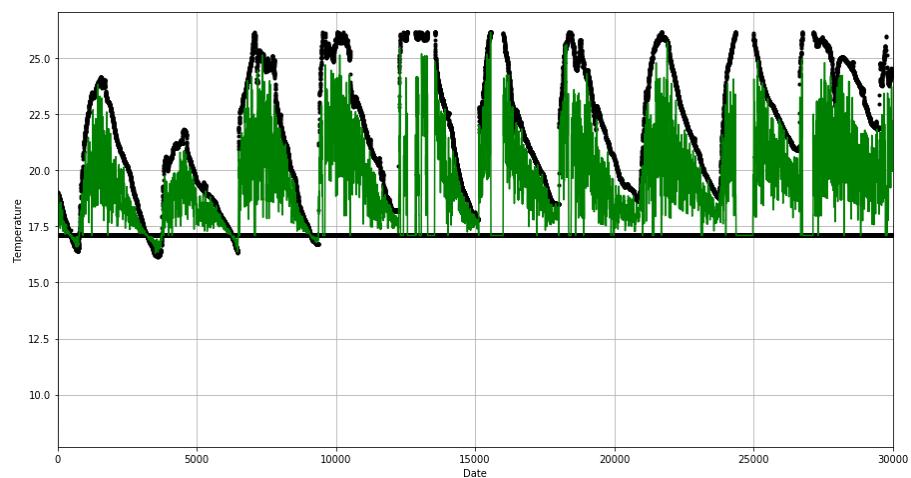
**Figure 45.** série temporal do sensor 17 após a reconstrução dos dados (passo 3)



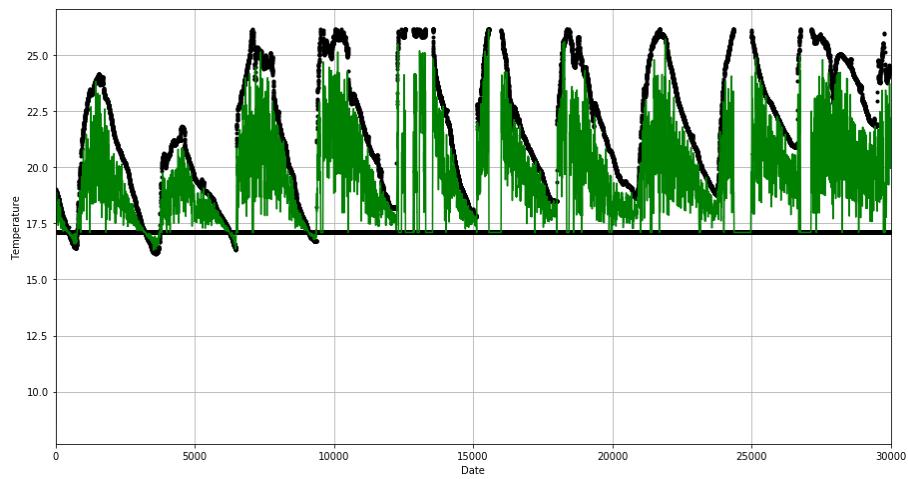
**Figure 46.** série temporal do sensor 18 com os dados brutos (passo 1 (ARO))



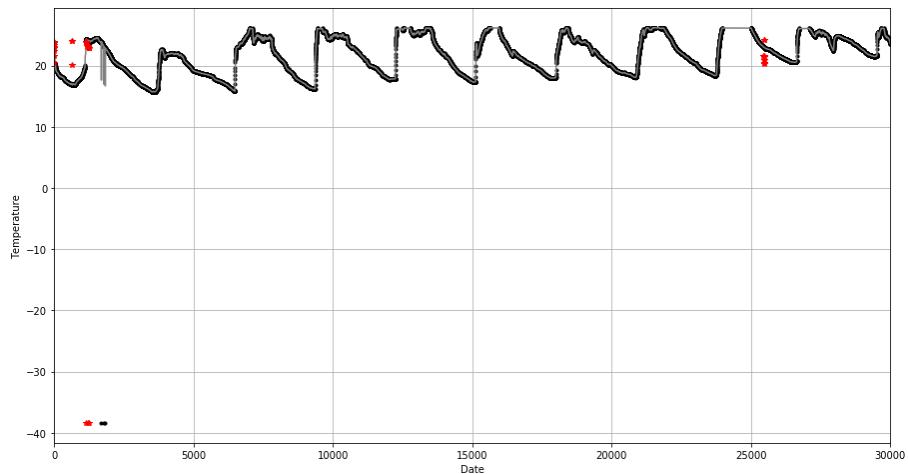
**Figure 47.** série temporal do sensor 18 após a reconstrução dos dados (passo 3)



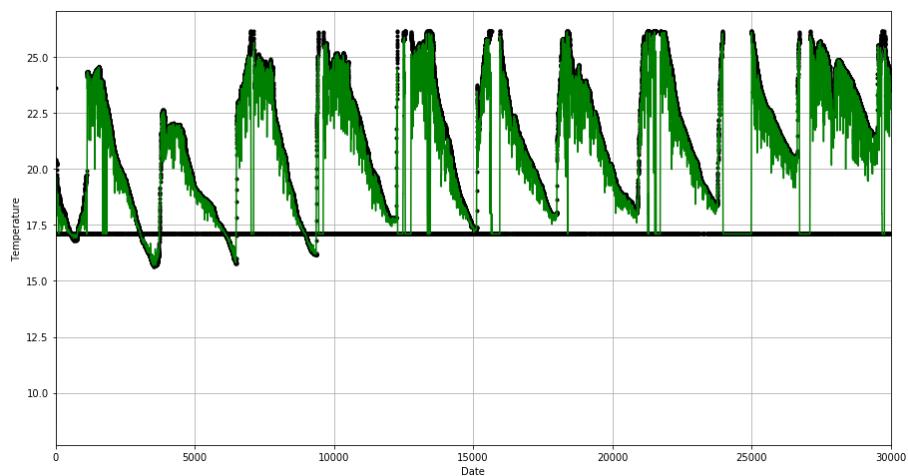
**Figure 48.** série temporal do sensor 19 com os dados brutos (passo 1 (ARO))



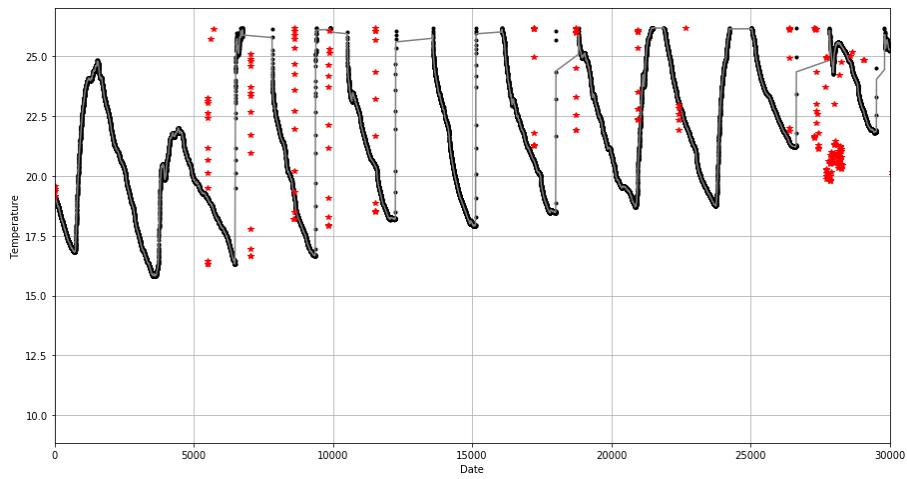
**Figure 49.** série temporal do sensor 19 após a reconstrução dos dados (passo 3)



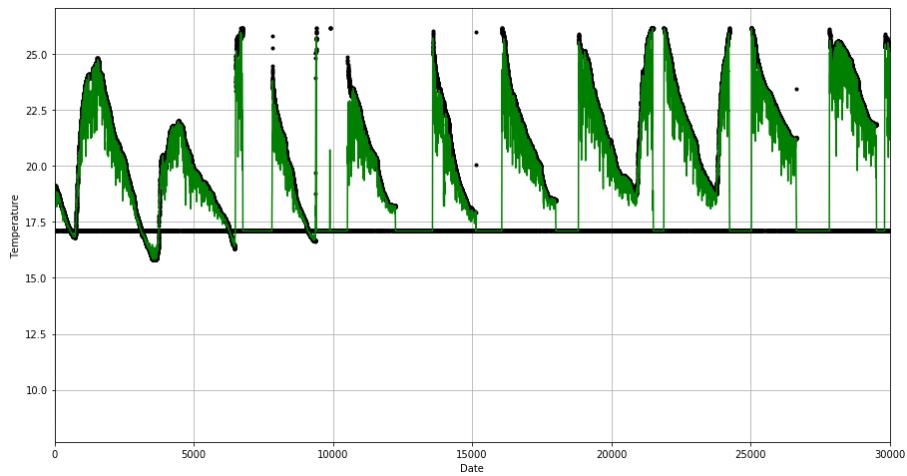
**Figure 50.** série temporal do sensor 20 com os dados brutos (passo 1 (ARO))



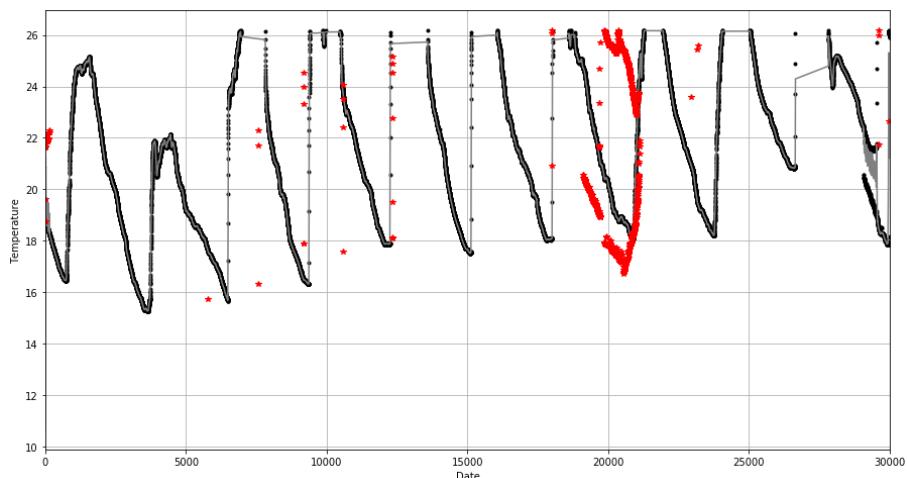
**Figure 51.** série temporal do sensor 20 após a reconstrução dos dados (passo 3)



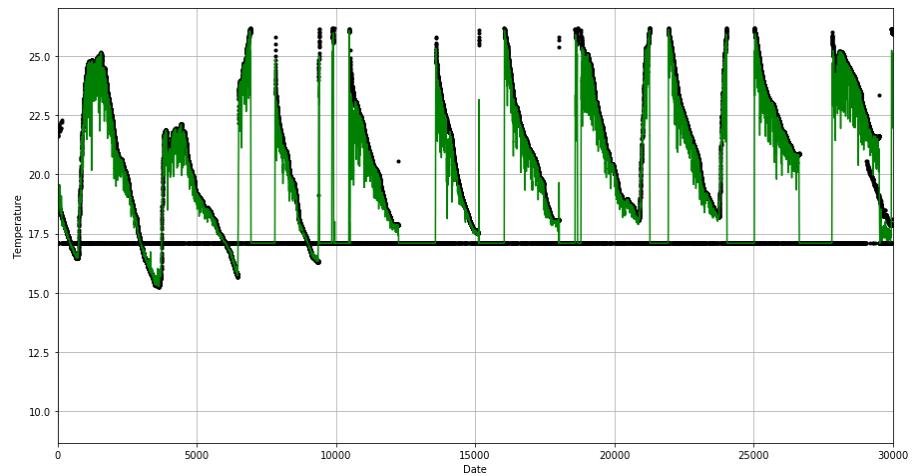
**Figure 52. série temporal do sensor 21 com os dados brutos (passo 1 (ARO))**



**Figure 53. série temporal do sensor 21 após a reconstrução dos dados (passo 3)**



**Figure 54. série temporal do sensor 22 com os dados brutos (passo 1 (ARO))**



**Figure 55. série temporal do sensor 22 após a reconstrução dos dados (passo 3)**