



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CAMPUS TRINDADE  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Felipe Schneider Costa

**Método FOCUSeR para Ranqueamento de Sensores em Ambientes IoT**

Florianópolis  
2022

Felipe Schneider Costa

**Método FOCUSeR para Ranqueamento de Sensores em Ambientes IoT**

Tese submetida ao Programa de Pós-Graduação em  
Ciência da Computação da Universidade Federal de  
Santa Catarina para a obtenção do título de doutor em  
Ciência da Computação.

Orientador: Prof. Mario Antonio Ribeiro Dantas, Dr.  
Coorientadora: Profa. Silvia Modesto Nassar, Dra.

Florianópolis

2022

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Costa, Felipe Schneider

Método FOCUSER para Ranqueamento de Sensores em  
Ambientes IoT / Felipe Schneider Costa ; orientador, Mario  
Antonio Ribeiro Dantas, coorientador, Sílvia Modesto  
Nassar, 2022.

132 p.

Tese (doutorado) - Universidade Federal de Santa  
Catarina, Centro Tecnológico, Programa de Pós-Graduação em  
Ciência da Computação, Florianópolis, 2022.

Inclui referências.

1. Ciência da Computação. 2. sensor ranking. 3. online.  
4. up-to-date. 5. anomaly. I. Dantas, Mario Antonio  
Ribeiro . II. Nassar, Sílvia Modesto. III. Universidade  
Federal de Santa Catarina. Programa de Pós-Graduação em  
Ciência da Computação. IV. Título.

Felipe Schneider Costa

**Método FOCUSER para Ranqueamento de Sensores em Ambientes IoT**

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Edson Roberto De Pieri, Dr.  
Universidade Federal de Santa Catarina

Prof. Mauro Roisenberg, Dr.  
Universidade Federal de Santa Catarina

Profa. Renata de Matos Galante, Dra.  
Universidade Federal do Rio Grande do Sul

Prof. Victor Stroële de Andrade Menezes, Dr.  
Universidade Federal de Juiz de Fora

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de doutor em Ciência da Computação.

---

Coordenação do Programa de Pós-Graduação

---

Prof. Mario Antonio Ribeiro Dantas, Dr.  
Orientador

Florianópolis, 2022.



## AGRADECIMENTOS

Aos meus pais Angelo (*in memoriam*) e Ester pelo apoio, dedicação e pela educação recebida. Agradeço também a toda minha família pelo estímulo e apoio nos momentos difíceis. Às minhas irmãs Rosangela e Denise pelo apoio e momentos de descontração e a minha sobrinha Andressa e ao amigo Rubem pelo apoio recebido.

A minha filha amada, Jeniffer, que me inspira e me dá força diariamente.

A Rosana pelo apoio, compreensão, carinho e principalmente pelo incentivo durante o desenvolvimento da tese.

Ao meu orientador, professor Mario Antonio Ribeiro Dantas, por todo apoio e incentivos recebidos, bem como pelas tantas lições aprendidas que me permitiram chegar a esse momento.

A minha orientadora de mestrado e coorientadora do doutorado, professora Silvia Modesto Nassar, pela amizade, apoio, ensinamentos, conselhos e sugestões recebidas ao longo da minha vida acadêmica, sem os quais com certeza esse momento não seria possível.

A Jacqueline, por todo apoio recebido no início dessa jornada.

Aos grandes amigos e colegas, Alexandre Zammar, Andrés, Delcio e Nilo pelos momentos de descontração e pelas lições aprendidas.

Aos professores da UFSC e das bancas do doutorado pelos comentários e contribuições. Aos técnicos da UFSC, em especial ao Ivan pela excelência no atendimento. Ao Instituto Federal de Santa Catarina (IFSC) pelo apoio concedido, propiciando minha dedicação total à pesquisa. A Universidade Federal de Santa Catarina (UFSC) pela oportunidade e ao projeto FASTEN, representado pelo INESC P&D Brasil pela experiência propiciada.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

"[...] *Things are not the way they used to be* [...]" (MARLEY; WAILERS, 1995).

## RESUMO

Dados obtidos de sensores conectados a *Wireless Sensor Networks* (WSNs) devem ser armazenados e processados para viabilizar ambientes como cidades inteligentes. No entanto, com o crescimento exponencial do número de dispositivos na borda da rede, faz-se necessária a implementação de técnicas robustas, capazes de selecionar fontes de dados confiáveis e atender requisitos de baixa latência, capazes de servir aplicações críticas. Assim, para superar esses desafios, este trabalho de pesquisa apresenta um método para *ranking* de sensores denominado FOCUSER. O método utiliza a avaliação de dados como critério para o *ranking*, permitindo a identificação de ocorrências de falhas em sensores e anomalias em ambientes. Para atender aos requisitos inerentes às WSNs, o método proposto foi projetado para execução em um ambiente típico de *Fog Computing*, utilizando aprendizagem online e atualização constante ao longo do tempo para evitar efeitos como desvio de tempo (*concept drift*). O método também oferece recursos para recuperação de dados oriundos de falhas, buscando dados de sensores similares nas regiões próximas ao sensor em estado de falha. As listas de *ranking* geradas são gerenciadas por meio de tabelas de *hash* distribuídas. Para dar maior confiabilidade aos resultados experimentais, foi desenvolvido um ambiente experimental real (*testbed*). Em adição, utilizando este *testbed*, foram geradas amostras com rótulos, constituindo um novo conjunto de dados, com o intuito de apoiar a avaliação do método desenvolvido. Outros quatro conjuntos de dados reais também foram utilizados, três dos quais rotulados por meio de injeção artificial de falhas. Esses conjuntos de dados foram rotulados em um trabalho relacionado com foco na injeção de falhas artificiais. Os resultados experimentais obtidos indicam que o método proposto pode fornecer confiabilidade no uso dos dados dos sensores, utilizando poucos recursos computacionais e reduzindo a latência no processo de seleção dos sensores. Nos testes realizados, foram obtidas taxas de Precisão de aproximadamente 98% e taxas de Acurácia superiores a 94% em todos os conjuntos de dados. Além disso, as análises realizadas apontam que a Acurácia tem uma taxa crescente à medida que o número de amostras também aumenta. Os resultados obtidos na recuperação de dados de falhas também indicam a viabilidade do método proposto neste recurso, no qual foi obtida uma taxa de recuperação de dados falhos de mais de 87%. Dessa forma as principais contribuições do trabalho consistem em um método capaz de reduzir a latência no uso de dados de sensores e ao mesmo tempo capaz de aprender e manter atualizados, de forma online, os parâmetros necessários para avaliação dos dados produzidos pelos sensores.

**Palavras-chave:** *Ranking* de sensores. *On-line*. *Context-aware*. *Up-to-date*. Anomalia. *Matrix Profile*. Latência. *Fog Computing*.



## ABSTRACT

*Data obtained from sensors connected to Wireless Sensor Networks must be stored and processed to enable environments such as Smart Cities. However, with the exponential growth in the number of devices at the edge of the network, it is necessary to implement robust techniques, capable of selecting reliable data sources and meeting low latency requirements, in order to serve critical applications. Thus, to overcome these challenges, this research presents FOCUSeR, a method for ranking sensors. The method uses the evaluation of data as a criterion for the ranking, allowing the identification of occurrences of failures in sensors and anomalies in environments. In order to meet the requirements inherent to WSNs, the proposed method was developed to run in a Fog Computing environment, using online learning and constant updating over time to avoid effects such as time drift. The method also offers resources for data recovery from failures, searching for data from similar sensors in regions close to the sensor in a failure state. The generated ranking lists are managed through distributed hash tables. To provide reliability to the experimental results, a real experimental environment was developed. Also, using this developed testbed, a dataset with labels was created, to support the evaluation of the method. In addition, four other real datasets were used, three of which were labeled through artificial fault injection. These datasets were labeled in a related work that focused on injecting artificial faults. The experimental results obtained indicate that the proposed approach can provide reliability in the use of sensor data, using low computational resources and reducing latency in the sensor selection process. Precision rates are approximately 98% and accuracy rates greater than 94% across all datasets. In addition, the analyzes carried out show that the accuracy has an increasing rate as the number of samples also increases. Results obtained in the recovery of data from failures also indicate the feasibility of the proposal in this resource in which a rate of recovery of failed data of more than 87% was obtained. Thus, the main contributions of the work consist of a method capable of reducing latency in the use of sensor data and at the same time capable of learning and keeping updated, in an online way, the parameters necessary for evaluating the data produced by the sensors.*

**Keywords:** *Sensor ranking. Online. Context-aware. Up-to-date. Anomaly. Matrix profile. Latency. Fog computing.*

## LISTA DE FIGURAS

Figura 1. Procedimentos metodológicos do método FOCUSeR.....	21
Figura 2. Tecnologias relacionadas ao crescimento dos dados digitais. ....	27
Figura 3. Recursos e distância versus latência. ....	27
Figura 4. Plataforma IIoT FASTEN.....	28
Figura 5. Cenário Embraer. ....	29
Figura 6. Cenário Thyssenkrupp. ....	30
Figura 7. Um <i>geohash</i> 9q9p6, com vários <i>geohashes</i> mais precisos contidos nele. ...	42
Figura 8. FOCUSeR - Visão geral.....	57
Figura 9. Modelo de valores discrepantes/falhas. ....	59
Figura 10. Exemplo de saída produzida de acordo com o modelo de <i>outliers</i> /falhas. .....	60
Figura 11. Divisão de dados em normais, anomalia e falha usando IQR.....	62
Figura 12. FOCUSeR: Ambiente experimental. ....	67
Figura 13. Ambiente experimental - <i>Raspberrys PI</i> (a) e sensores (b). ....	68
Figura 14. Conjunto de dados Intel Lab. ....	70
Figura 15. SensorScope: mapa implantação no <i>Grand Saint-Bernard</i> . ....	70
Figura 16. Acurácia ao longo do tempo. ....	74
Figura 17. Recomposição de dados de falhas.....	75
Figura 18. Evolução do índice de <i>ranking</i> de um sensor em relação ao número de erros real do conjunto de dados. ....	76
Figura 19. Comportamento da métrica IQR utilizando apenas uma fatia de dados (IQR parcial), utilizando todo o conjunto de dados conhecido (IQR total) e utilizando a técnica <i>t-digest</i> . ....	77
Figura 20. Conjunto de dados Intel: Resultados finais para o sensor 4.....	77
Figura 21. Conjunto de dados Intel: resultados <i>matrix profile</i> .....	78
Figura 22. NUMENT A: resultados online, atualizados e de identificação de dados.	79
Figura 23. Medidas de latência.....	80
Figura 24. Exemplo de painel de monitoramento. ....	81
Figura 25. Intel, saída do primeiro nível de avaliação. ....	106
Figura 26. Intel, saída do segundo nível de avaliação.....	106
Figura 27. Intel, saída do terceiro nível de avaliação.....	107

Figura 28. Intel, decisão final. ....	107
Figura 29. Intel, tipos de outlier. ....	108
Figura 30. Intel, saída do algoritmo Matrix Profile. ....	108
Figura 31. Santander, saída do primeiro nível de avaliação. ....	109
Figura 32. Santander, saída do segundo nível de avaliação. ....	109
Figura 33. Santander, saída do terceiro nível de avaliação. ....	110
Figura 34. Santander, decisão final. ....	110
Figura 35. Santander, tipos de outlier. ....	111
Figura 36. Santander, saída do algoritmo Matrix Profile. ....	111
Figura 37. SensorScope, saída do primeiro nível de avaliação. ....	112
Figura 38. SensorScope, saída do segundo nível de avaliação. ....	112
Figura 39. SensorScope, saída do terceiro nível de avaliação. ....	113
Figura 40. SensorScope, decisão final. ....	113
Figura 41. SensorScope, tipos de outlier. ....	114
Figura 42. SensorScope, saída do algoritmo Matrix Profile. ....	114
Figura 43. NUMENTA, saída do primeiro nível de avaliação. ....	115
Figura 44. NUMENTA, saída do segundo nível de avaliação. ....	115
Figura 45. NUMENTA, saída do terceiro nível de avaliação. ....	116
Figura 46. NUMENTA, decisão final. ....	116
Figura 47. NUMENTA, tipos de outlier. ....	117
Figura 48. NUMENTA, saída do algoritmo Matrix Profile. ....	117
Figura 49. FOCUSeR, saída do primeiro nível de avaliação. ....	118
Figura 50. FOCUSeR, saída do segundo nível de avaliação. ....	118
Figura 51. FOCUSeR, saída do terceiro nível de avaliação. ....	119
Figura 52. FOCUSeR, decisão final. ....	119
Figura 53. FOCUSeR, tipos de outlier. ....	120
Figura 54. FOCUSeR, saída do algoritmo Matrix Profile. ....	120

## LISTA DE QUADROS

Quadro 1. Trabalhos relacionados.....	54
Quadro 2. Saída produzida de acordo com modelo valores discrepantes/falhas.....	59

## LISTA DE TABELAS

Tabela 1. Conjuntos de dados.....	69
Tabela 2. Análise de sensibilidade de parâmetros.....	71
Tabela 3. Eficácia do modelo proposto.....	72
Tabela 4. Comparação de Avaliação de Eficácia.....	73
Tabela 5. Teste de sensibilidade de parâmetros – Intel.....	122
Tabela 6. Teste de sensibilidade de parâmetros – SensorScope.....	123
Tabela 7. Teste de sensibilidade de parâmetros – SmartSantander.....	124

## LISTA DE ABREVIATURAS E SIGLAS

FOCUSeR - *Fog Online Context-aware Up-to-date Sensor Ranking*

ARPAnet - *Advanced Research Projects Agency Network*

IoT - *Internet of Things*

WSN - *Wireless Sensor Network*

TI - *Tecnologia da Informação*

NSF - *National Science Foundation*

QoC - *Quality of Context*

R1 - *Requisito 1*

R2 - *Requisito 2*

R3 - *Requisito 3*

R4 - *Requisito 4*

R5 - *Requisito 5*

R6 - *Requisito 6*

R7 - *Requisito 7*

R8 - *Requisito 8*

R9 - *Requisito 9*

IECON - *IEEE Industrial Electronics Society*

IIoT - *Industrial Internet of Things*

FASTEN - *Flexible and Autonomous Manufacturing Systems for Custom-Designed Products*

CLOSER - *Conference on Cloud Computing and Services Science*

3PGCIC - *Advances on P2P, Parallel, Grid, Cloud and Internet Computing*

EMBPT - *Embraer Portugal*

TSK - *Thyssenkrupp*

IQR - *Intervalo Interquartil*

Q1 - *Primeiro quartil*

Q2 - *Segundo quartil*

Q3 - *Terceiro quartil*

FFT - *Fast Fourier Transform*

STAMP - *Scalable Time Series Anytime Matrix Profile*

STAMPI - *STAMP Incremental*

M2M - *Machine-to-machine*

DHT - *Distributed Hash Table*

P2P - *Point-to-point*

QoS - *Quality of Service*

CPU - *Central Process Unit*

RAM - *Random Access Memory*

GPU - *Graphic Process Unit*

SO - *Sistema Operacional*

WoT - *Web of Things*

CoAP - *Constrained Application Protocol*

WSDL - *Web Services Description Language*

AHP - *Analytical Hierarchy Process*

MCDA - *Multi-Criteria Decision Analysis*

LMT - *Logistic Model Tree*

SAW - *Simple Additive Weighting*

API - *Application Programming Interface*

REST - *Representational State Transfer*

RESTful - *Representational State Transfer Web Service*

ES - *Elimination-Selection*

ViSIoT - *Visual Search for Internet of Things*

TOPSIS - *Technique for Order Preference by Similarity to Ideal Solution*

PBWI - *Preference Based Weighted Index*

P-CNP - *Primitive Cognitive Network Process*

UQA - *Avaliação de QoS do Usuário*

CQA - *Avaliação de QoS de Contexto*

CPWI - *Índice Comparativo Baseado em Prioridade Ponderada*

AF - *Anomaly Factor*

FF - *Failure Factor*

C1 - *Custo para calcular IQR para todos os níveis*

C2 - *Custo para calcular estruturas do algoritmo *matrix profile**

C3 - *Custo para atualizar as estruturas do algoritmo *matrix profile**

C4 - *Custo para testes em todos os níveis*

C5 - *Custo para localizar amostras de outros sensores*

MQTT - *Message Queuing Telemetry Transport*

## LISTA DE SÍMBOLOS

1NN Vizinho mais próximo

$\ominus$  Função booleana

$\bowtie$  Conjunto de junção de similaridade



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>15</b>
1.1	CONTEXTUALIZAÇÃO .....	15
1.2	OBJETIVOS .....	17
<b>1.2.1</b>	<b>Objetivo Geral.....</b>	<b>17</b>
<b>1.2.2</b>	<b>Objetivos Específicos .....</b>	<b>17</b>
1.3	ESCOPO E RELEVÂNCIA DO TRABALHO .....	17
1.4	INEDITISMO E CONTRIBUIÇÃO CIENTÍFICA .....	18
1.5	ASPECTOS METODOLÓGICOS .....	20
<b>1.5.1</b>	<b>Classificação da Pesquisa .....</b>	<b>20</b>
<b>1.5.2</b>	<b>Procedimentos Metodológicos .....</b>	<b>20</b>
1.6	PUBLICAÇÕES .....	22
1.7	ORGANIZAÇÃO DO TRABALHO .....	24
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>27</b>
2.1	PROJETO FASTEN .....	28
2.2	PERCEPÇÃO ATIVA.....	30
2.3	DETECÇÃO DE ANOMALIAS.....	31
2.4	INTERVALO INTERQUARTIL .....	32
<b>2.4.1</b>	<b><i>t-digest</i>.....</b>	<b>32</b>
2.5	<i>MATRIX PROFILE</i> .....	33
2.6	MÉTRICAS DE AVALIAÇÃO .....	35
2.7	CONSCIÊNCIA E QUALIDADE DE CONTEXTO.....	35
2.8	SENSORES .....	37
2.9	<i>WIRELESS SENSOR NETWORKS</i> .....	38
2.10	<i>BIG DATA, CLOUD, FOG E EDGE COMPUTING</i> .....	38
2.11	<i>MIDDLEWARE IoT</i> .....	40

2.12	TABELAS DE <i>HASH</i> DISTRIBUÍDAS .....	41
2.13	<i>RANKING</i> DE SENSORES .....	42
<b>3</b>	<b>ESTADO DA ARTE .....</b>	<b>45</b>
<b>4</b>	<b><i>FOG ONLINE CONTEXT-AWARE UP-TO-DATE SENSOR RANKING</i></b> <b>(FOCUSeR) .....</b>	<b>57</b>
4.1	<i>RANKING</i> DE SENSORES INSPIRADO NA PERCEPÇÃO ATIVA .....	57
4.2	MODELO DE VALORES DISCREPANTES/FALHAS .....	58
4.3	<i>BURN-IN</i> .....	62
4.4	PRIMEIRO NÍVEL DE AVALIAÇÃO: PERCEPÇÃO .....	63
4.5	SEGUNDO NÍVEL DE AVALIAÇÃO: PERCEPÇÃO AO LONGO DO TEMPO	63
4.6	TERCEIRO NÍVEL DE AVALIAÇÃO: PERCEPÇÃO ATIVA .....	63
4.7	COMPLEXIDADE DE TEMPO .....	64
4.8	CONSIDERAÇÕES .....	65
<b>5</b>	<b>AMBIENTE EXPERIMENTAL E RESULTADOS .....</b>	<b>67</b>
5.1	AMBIENTE EXPERIMENTAL .....	67
5.2	CONJUNTOS DE DADOS .....	69
5.3	TESTE DE SENSIBILIDADE DE PARÂMETROS.....	71
5.4	AVALIAÇÃO DE ACURÁCIA .....	72
5.5	AVALIAÇÃO DE RECOMPOSIÇÃO DE DADOS.....	74
5.6	RESULTADOS DA ATUALIZAÇÃO ONLINE E AO LONGO DO TEMPO ..	75
5.7	AVALIAÇÃO DE LATÊNCIA .....	79
5.8	MONITORAMENTO NA NUVEM .....	80
5.9	CONSIDERAÇÕES .....	81
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>85</b>
	<b>REFERÊNCIAS.....</b>	<b>89</b>
	<b>APÊNDICE A – Publicações .....</b>	<b>99</b>
	<b>APÊNDICE B – Resultados produzidos pelo método .....</b>	<b>105</b>
	<b>APÊNDICE C – Teste de Sensibilidade de Parâmetros.....</b>	<b>121</b>

<b>APÊNDICE D – Referências utilizadas na criação da Figura 2.....</b>	<b>125</b>
--	------------



# 1 INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO

Após quatro décadas do advento da Internet pela ARPAnet, o termo "Internet" refere-se à vasta categoria de aplicativos e protocolos construídos sobre redes de computadores sofisticados e interconectados, atendendo a bilhões de usuários em todo o mundo de maneira ininterrupta. Atualmente, comunicação e conectividade ubíquas não são mais um sonho nem um desafio. Os esforços migraram para uma integração perfeita de pessoas e dispositivos, visando convergir o mundo físico aos ambientes virtuais produzidos pelo homem (ALFUQAHA *et al.*, 2015; BUYYA; DASTJERDI, 2016).

Um olhar mais atento ao próprio termo revela os dois pilares da Internet das Coisas (IoT): “Internet” e “coisas”. A notação “coisas”, é usada para abranger um conjunto genérico de entidades, que inclui dispositivos inteligentes, sensores, seres humanos e qualquer outro objeto ciente de seu contexto e que seja capaz de se comunicar com outras entidades. Isso implica que os objetos precisam estar acessíveis sem restrições de tempo ou local. A conectividade ubíqua é um requisito crucial na IoT e, para isso, os aplicativos precisam oferecer suporte a um conjunto diversificado de dispositivos e protocolos de comunicação. A integração de dispositivos móveis, como dispositivos de borda (roteadores e hubs inteligentes) também é necessária (BUYYA; DASTJERDI, 2016; PELLE *et al.*, 2022).

Sabe-se que a IoT oferece uma plataforma para sensores e dispositivos se comunicarem perfeitamente em um ambiente inteligente e permite o compartilhamento de informações entre plataformas de maneira conveniente. No paradigma da IoT, sensores em todo o mundo estão coletando e transmitindo dados que devem ser armazenados e processados. Nesse sentido, o avanço da tecnologia sem fio é uma das que mais contribuiu para o desenvolvimento da IoT nos últimos anos, colocando-a como uma tecnologia revolucionária, a partir das oportunidades oferecidas pela Internet. Isso ocorre porque muitos dispositivos de comunicação estão embutidos em sensores no mundo real e assim os dispositivos de coleta de dados detectam e transmitem esses dados usando os dispositivos de comunicação incorporados (MARJANI *et al.*, 2017; MOHINUR RAHAMAN; AZHARUDDIN, 2022; SHAH; YAQOOB, 2016). Isso é possível graças redes de sensores sem fio (WSNs). Uma WSN pode ser descrita como uma rede de nodos que detectam e controlam uma determinada área geográfica para monitorar fenômenos físicos, permitindo a

interação entre pessoas ou computadores e o ambiente circundante. Uma WSN geralmente é composta por nodos sensores, nodos atuadores, *gateways* e clientes (ANASTASI *et al.*, 2009; CAI *et al.*, 2022; YINBIAO, 2014).

Conforme o número de objetos adicionados à IoT cresce de forma exponencial (TURNER, 2014), os dados gerados excedem em muito as capacidades das arquiteturas e infraestruturas de Tecnologia da Informação (TI) existentes. Em cenários onde o requisito de tempo real está presente, o poder de computação disponível precisa ser ainda mais robusto.

É importante reconhecer que grande parte dessa explosão de dados é o resultado de uma explosão da quantidade de dispositivos localizados na borda da rede, incluindo sensores embutidos, smartphones e tablets. Todos os dados desses dispositivos criam novas oportunidades para análise em diversas áreas, como saúde, petróleo e gás, pesquisa, vigilância, finanças e muitas outras (VILLARS; EASTWOOD; OLOFSON, 2011).

Além disso, a complexidade e heterogeneidade dos sensores conectados à rede também se refletem nos dados gerados por eles. Isso faz surgir um conjunto de desafios relacionados ao processamento e uso dessa imensa massa de dados (PATTAR *et al.*, 2018). De acordo com o relatório da *National Science Foundation* (NSF), alguns dos principais desafios desse ambiente incluem suporte a aplicativos em tempo real, desde permitir inteligência artificial incorporada ao controle de *feedback* em sistemas *ciber* físicos, visando reduzir a latência nos processos envolvidos (NSF, 2016).

Para superar esses obstáculos, é fundamental que os processos se tornem inteligentes. Nesse sentido, entender o contexto, ou a situação em que ocorre a captura, utilizando os dados dos sensores e depois agindo de forma autônoma, requer detecção e aprendizado. Isso é chamado de computação sensível ao contexto. A Qualidade do Contexto (QoC) fornece um conjunto de métricas com informações que auxiliam na resolução de conflitos relacionados às informações de contexto (SEZER; DOGDU; OZBAYOGLU, 2018).

Diante das evidências já mencionadas, esta tese busca responder a seguinte questão: é possível atender a requisitos de baixa latência, provendo confiabilidade na atividade de seleção de sensores no ambiente IoT em um processo automático?

Para responder a esta pergunta esta tese propõe um método para *ranking* de sensores. O método proposto baseia-se no fato de que soluções para *ranking* de sensores devem ser viáveis considerando a heterogeneidade, baixa capacidade computacional e alto volume de dispositivos conectados. Além disso, considera que a exigência de tempo real está presente em diversos serviços que fazem uso dos dados de sensores. O intuito é prover uma solução

que permita a seleção de sensores baseada na confiabilidade e ao mesmo tempo atendendo aos recursos de baixa latência.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

Propor um método para *ranking* de sensores, denominado FOCUSeR, que seja capaz de prover confiabilidade, redução da latência, que utilize aprendizagem não supervisionada, e que permita identificar anomalias nos ambientes e falhas em sensores.

### 1.2.2 Objetivos Específicos

Abaixo são listados os objetivos específicos da pesquisa:

- Investigar técnicas para identificação de anomalias e falhas em dados de sensores.
- Investigar técnicas para manipulação de listas ordenadas em ambientes distribuídos.
- Desenvolver o método.
- Desenvolver um ambiente experimental para avaliação do método.
- Avaliar o método desenvolvido.

## 1.3 ESCOPO E RELEVÂNCIA DO TRABALHO

O escopo desta pesquisa envolve o *ranking* de sensores com base na qualidade dos dados gerados por eles. A lista, resultado do processo de *ranking* de sensores, é importante para permitir a utilização dos dados produzidos por sensores consumindo a menor quantidade de recursos computacionais possíveis, permitindo que processos de tomada de decisão ocorram no menor tempo possível, principalmente em cenários nos quais requisitos de baixa latência estejam presentes.

Cabe ressaltar que o método proposto não se destina a resolver problemas no domínio de aplicativos, mas a fornecer informações suficientes para que os aplicativos tomem as decisões corretas com base nos indicadores fornecidos.

As principais atividades de um *middleware* IoT consistem em aquisição, descoberta, indexação, *ranking*, consulta de sensores e dados relacionados a esses sensores e publicação

dos dados/informações. O foco deste trabalho se concentra principalmente na atividade de *ranking* e na publicação dos resultados. As demais atividades podem estar presentes no trabalho apenas como apoio para a etapa de avaliação do método proposto, na ferramenta construída para isso.

A relevância da abordagem do método proposto está relacionada com a capacidade de prover uma forma rápida, e com alto grau de confiabilidade, de seleção de fontes de dados (sensores) (IGGENA *et al.*, 2021; TURNER, 2014; XIAO *et al.*, 2016), para tomada de decisões e controle de ambientes e dispositivos (ALAM *et al.*, 2017; MEKALA; RIZWAN; KHAN, 2021), além de realizar essas atividades de forma distribuída para garantir a escalabilidade, requisito imprescindível no ambiente IoT (NSF, 2016).

O crescimento exponencial do número de sensores no ambiente IoT justifica a necessidade desse tipo de atividade (JAIN *et al.*, 2022; SEZER; DOGDU; OZBAYOGLU, 2018). Da mesma forma, a utilização dos aspectos relacionados com a geração dinâmica de dados (geração de dados ao longo do tempo) para avaliação da qualidade dos dados produzidos por sensores garante uma maior confiabilidade nas respostas produzidas (ZHANG *et al.*, 2016).

#### 1.4 INEDITISMO E CONTRIBUIÇÃO CIENTÍFICA

Muitas pesquisas têm se dedicado à tarefa de aprimorar a atividade de *ranking* de sensores (COSTA; NASSAR; DANTAS, 2021). A partir da análise de trabalhos relacionados realizada, é possível verificar que um método de classificação viável deve atender a diversos requisitos (IGGENA *et al.*, 2021; WANG *et al.*, 2015), para prestar um serviço capaz de satisfazer requisitos de qualidade de serviço quando utilizado em um cenário real.

Os trabalhos relacionados analisados apresentam diferentes abordagens para *ranking* de sensores, indicando a importância da atividade para o uso seguro dos dados produzidos. No entanto, apesar das propostas relacionadas apresentarem resultados promissores, é possível notar alguns desafios e questões que ainda precisam ser superados. Assim, ao considerar esses problemas, é possível elencar um conjunto de requisitos que devem ser atendidos nos processos de *ranking* de sensores. Esses requisitos são listados a seguir:

- Requisito 1 (R1): a lista de ranking deve ser disponibilizada de forma distribuída.



- Requisito 2 (R2): para reduzir a latência nas consultas do usuário final e aplicações, nas quais os requisitos de tempo real estão presentes, é desejável que o serviço de *ranking* seja executado próximo à fonte de dados, ou seja, nas camadas de computação *Edge* ou *Fog*.

- Requisito 3 (R3): para ser viável em termos de processamento, e considerando o R2, é essencial que o algoritmo tenha baixa complexidade computacional de tempo.

- Requisito 4 (R4): com a mesma finalidade do R2, a lista de *ranking* deve estar sempre pronta, de forma a evitar que o usuário tenha que aguardar a geração da lista a cada requisição.

- Requisito 5 (R5): um dos principais desafios de uma WSNs consiste em como entregar eficientemente as medições detectadas com a máxima fidelidade aos dados capturados. Para isso, os fenômenos medidos por sensores requerem uma medida de classificação eficiente para diferenciar valores normais de valores anômalos (GHADDAR; DARWISH; YAMOUT, 2019). Assim, não realizar a avaliação dos dados pode comprometer o desempenho das WSNs e dos serviços e aplicações interessados nesses dados.

- Requisito 6 (R6): considerando que as WSNs são altamente dinâmicas, distribuídas, heterogêneas e com grande número de objetos, a utilização de avaliação de dados (R5) com métodos que requerem treinamento torna-se inviável. Portanto, é essencial que os algoritmos de classificação tenham aprendizagem *online* de parâmetros. Como o método proposto utiliza dados sem rótulos para aprendizagem ele pode ser considerado como um método de aprendizagem não supervisionada.

- Requisito 7 (R7): como os objetos nas WSNs são dinâmicos, e para evitar efeitos como "desvio de conceito" (*concept drift*), é importante utilizar métodos capazes de evitar a defasagem dos modelos (parâmetros).

- Requisito 8 (R8): verificar a viabilidade do algoritmo, realizando os testes em ambientes o mais próximo possível dos ambientes reais de processamento, corrobora os resultados do trabalho.

- Requisito 9 (R9): verificar a viabilidade do algoritmo, realizando testes com dados reais. Aqui a intenção é semelhante à apresentada em R8.

Assim, após apresentar os requisitos considerados importantes para um método de *ranking* de sensores, a seguir são apresentadas as contribuições científicas da tese:

- O processamento ocorre em um ambiente distribuído (R1).
- O método é capaz de atender requisitos de tempo real utilizando poucos recursos computacionais e assim fornecendo baixa latência nas consultas (R2, R3 e R4).

- O método é capaz de entregar, juntamente com os dados obtidos dos sensores, metadados com informações sobre a qualidade, confiabilidade e natureza desses dados e do próprio sensor (R5, R6 e R7).
- O método proposto é livre de treinamento e se mantém atualizado ao longo do tempo (R6 e R7).
- Ao avaliar os resultados experimentais com dados e ambientes reais o método contribui com a área de pesquisa (R8 e R9).
- O método proposto é capaz de recompor dados oriundos de falhas por meio da busca de sensores vizinhos, capazes de fornecer as informações semelhantes.

## 1.5 ASPECTOS METODOLÓGICOS

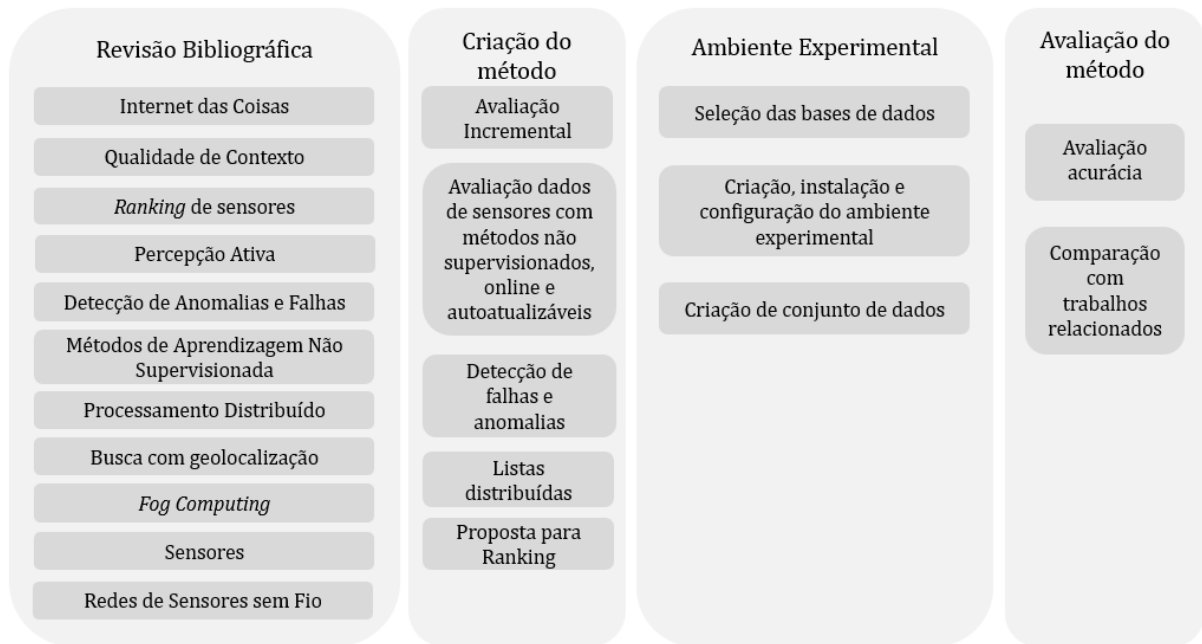
### 1.5.1 Classificação da Pesquisa

A metodologia utilizada nesta pesquisa, quanto a sua natureza, é caracterizada como aplicada, pois a pesquisa aplicada objetiva gerar conhecimentos para aplicação prática e dirigidos à solução de problemas específicos, envolvendo verdades e interesses locais. A pesquisa aplicada ou tecnológica tem como objetivo alcançar a inovação em um produto ou processo, frente a uma demanda ou necessidade preestabelecida. Neste tipo de pesquisa, o resultado a ser medido é a solução concreta do problema proposto, representado por um novo produto ou um novo processo. No que diz respeito aos procedimentos esta pesquisa se classifica como experimental já que o método desenvolvido provoca alterações no ambiente sob estudo de forma a observar se estas intervenções produzem os resultados esperados. Com relação à abordagem do problema é do tipo quantitativa e com relação aos objetivos é exploratória (SILVA; MENEZES, 2005).

### 1.5.2 Procedimentos Metodológicos

Nesta seção são descritos os procedimentos que orientam as etapas desta pesquisa, listados a seguir: revisão bibliográfica, a criação de um método para *ranking* de sensores no ambiente IoT, a implementação desse método em uma ferramenta computacional e finalmente a avaliação do método proposto, conforme pode ser visto na Figura 1. A imagem também mostra as atividades envolvidas em cada etapa.

Figura 1. Procedimentos metodológicos do método FOCUSeR.



Fonte: Elaborada pelo autor (2022).

Para a revisão bibliográfica foram escolhidas as bases *Web of Knowledge* (CLARIVATE, [s. d.]), Scopus (ELSEVIER, [s. d.]) e Google Scholar (GOOGLE, [s. d.]), todas com reconhecimento científico e que serviram como base para a análise do estado da arte do tema. Após obter os primeiros resultados das consultas, foram aplicados filtros para manter apenas as áreas de pesquisa de ciência da computação e engenharia. As buscas foram feitas e atualizadas de forma a incluir novos artigos sobre o tema que viessem a ser publicados. Ao todo, na primeira busca foram encontrados 713 artigos para análise através do título e do resumo. Desta etapa, foram selecionados 44 artigos e após a leitura completa deles, restaram 31 artigos relacionados com o tema de pesquisa deste trabalho.

A expressão de busca utilizada nas consultas continha uma combinação de termos para encontrar trabalhos com referências a sensores, termos para identificar a atividade de *ranking* e termos para identificar o ambiente IoT ou as WSNs. A busca foi realizada nos elementos título, resumo e palavras chaves dos trabalhos. A expressão de busca utilizada foi a seguinte: *sensor AND ranking AND ("internet of things" OR iot) OR ("wireless sensor network" OR wsn)*.

Na etapa de criação do método foram consideradas as lacunas e possibilidades de melhoria nas soluções existentes. Foi possível perceber a importância da utilização do fator tempo na avaliação da qualidade dos dados produzidos por sensores visando permitir a

identificação de anomalias e falhas dentro das faixas de valores válidos dos sensores. Considerou-se também a utilização de um método em que a análise dos dados ocorresse de forma progressiva, onde a cada etapa mais conhecimento fosse adicionado para essa avaliação. Considerando principalmente estes aspectos, optou-se por utilizar a teoria da percepção ativa, que se assemelha ao processo de percepção de seres humanos (BIEL; WIDE, 2000; SCHIFFMAN, 2001).

A implementação do método envolveu a seleção de conjuntos de dados (*datasets*) para a realização dos testes iniciais do método proposto. Foram utilizados três conjuntos de dados disponibilizados por um trabalho com foco na injeção de falhas. A motivação para essa escolha teve como objetivo permitir a comparação do método proposto com trabalhos futuros, já que não foram encontrados conjuntos de dados com o padrão ouro (rótulos de dados) real.

Além desses, foram utilizados mais dois conjuntos: um conjunto de dados de outro trabalho com foco na detecção de anomalias e um conjunto criado durante o desenvolvimento da pesquisa. Esse conjunto foi gerado através da captura de dados de um nodo sensor projetado juntamente com o ambiente experimental desenvolvido. Todos os conjuntos possuem dados meteorológicos de temperatura.

Com exceção do conjunto de dados criado, que foi submetido ao método em tempo real, os dados foram submetidos por meio de scripts. Esses scripts tiveram como função enviar os dados de cada sensor para a aplicação desenvolvida.

A execução de todos os processos foi realizada no ambiente experimental desenvolvido.

## 1.6 PUBLICAÇÕES

Assim que o tema da pesquisa foi definido e a revisão da literatura foi iniciada, o primeiro caminho percorrido tinha como foco o desenvolvimento de um mecanismo que funcionasse como filtro, que a cada etapa do processamento fosse descartando sensores que estivessem fornecendo dados ruidosos. Mas, com o andamento da pesquisa, foi possível perceber que os sensores em perfeito estado de funcionamento são tão importantes quanto aqueles que estivessem apresentando alguma instabilidade. Isso porque o interesse pode estar apenas em detectar e iniciar processos de manutenção desses dispositivos, de forma a não comprometer o monitoramento de ambientes ou para disparar ações que visem minimizar

possíveis prejuízos nos ambientes nos casos em que estão ocorrendo eventos extremos nesses ambientes.

Após avaliar os resultados iniciais foi possível perceber que o treinamento do algoritmo precisava ser realizado para cada sensor dentro de cada conjunto de dados, já que as diferenças entre as faixas de temperatura de um sensor para outro variavam bastante devido a fatores como altitude e outras condições climáticas do local do sensor. Um exemplo disso ocorre no conjunto de dados SensorScope. Como isso seria inviável, o método foi alterado para adotar uma aprendizagem automática de parâmetros.

Entretanto, o dinamismo característico dos dados de sensores (é comum que o estado do ambiente monitorado pelo sensor se altere com o passar do tempo) também comprometeu os resultados obtidos. Assim adotou-se uma forma de manter os parâmetros do modelo atualizados ao longo do tempo.

Pelos motivos expostos, a abordagem de solução foi direcionada para o *ranking* de sensores com base na qualidade dos dados gerados por eles, deixando a cargo das aplicações de cada domínio, realizar a seleção dos sensores pertinentes.

O processo e os resultados iniciais do primeiro estágio da pesquisa foram publicados e apresentados na IECON 2019 (45th *Annual Conference of the IEEE Industrial Electronics Society*) sob o título de "*A three level sensor ranking method based on active perception*".

Durante o desenvolvimento da tese, houve também a participação em um projeto de pesquisa voltado para a Internet das Coisas Industrial (IIoT). O projeto FASTEN (*Flexible and Autonomous Manufacturing Systems for Custom-Designed Products*), financiado pelo programa *Horizon 2020* da União Europeia, teve como objetivo desenvolver um framework aberto e padronizado para a Indústria 4.0. A participação no projeto FASTEN agregou conhecimento à pesquisa, uma vez que por meio da atuação no projeto foi possível conhecer a arquitetura de um *framework* robusto e completo para o processamento de dados de sensores. O conhecimento adquirido em relação a IIoT foi descrito no artigo intitulado "*FASTEN IIoT: An Open Real-Time Platform for Vertical, Horizontal and End-To-End Integration*", publicado no periódico *Sensors*.

Com algumas alterações no método (mudanças nas técnicas de avaliação), um segundo artigo foi publicado no evento CLOSER (*11th International Conference on Cloud Computing and Services Science*).

Outra submissão está em avaliação no periódico Springer Nature - SN Computer Science, intitulado "*A Sensor Ranking Approach for Edge-Fog-Cloud Environments*" na qual

foi utilizada pela primeira vez a técnica de vizinhos mais próximos (kNN) no segundo nível de avaliação do método. Nesta versão do método, foram utilizadas comparações entre subsequências de amostras utilizando a métrica distância euclidiana. Este artigo ainda está em avaliação.

O último esforço descrevendo os avanços da pesquisa foi descrito no artigo “*FOCUSer: a Fog Online Context-aware Up-to-date Sensor Ranking method*” em que são apresentados os resultados atuais da pesquisa, os mesmos apresentados nesta tese. Este artigo foi publicado no periódico *Journal of Sensor and Actuator Networks* (JSAN).

Além do artigo desenvolvido sobre o projeto FASTEN, houve colaboração em outras duas pesquisas relacionadas ao projeto sobre segurança no envio de dados de sensores e sobre presença do requisito de tempo real no uso de dados de sensores. Essas participações foram registradas nos artigos “*Métis - An Approach Utilized as Differentiated Authenticity Tool in an IIoT Infrastructure*” (*Advances on P2P, Parallel, Grid, Cloud and Internet Computing*) e “*A Survey from Real-Time to Near Real-Time Applications in Fog Computing Environments*” (*Telecom*) respectivamente.

## 1.7 ORGANIZAÇÃO DO TRABALHO

Esta seção descreve a organização do trabalho e procura justificar a escrita de cada capítulo. O trabalho está dividido em sete capítulos. O presente capítulo, Introdução, apresenta a contextualização, o problema para o qual se buscou uma solução, os objetivos, a hipótese e a motivação para realização da pesquisa.

O Capítulo 2, Fundamentação Teórica, descreve os conceitos e temas relevantes com este trabalho de pesquisa. Inicialmente é descrito o projeto FASTEN, que proporcionou um aprendizado relacionado ao tema da tese. Em seguida são apresentados os conceitos relacionados à teoria da percepção ativa, que serviu como inspiração para o desenvolvimento do método proposto. Depois disso, são descritos os conceitos relacionados à detecção de anomalias, bem como as métricas e técnicas utilizados neste trabalho para identificação delas. Essas técnicas foram utilizadas na geração das listas de *ranking* de sensores. Logo depois, são apresentados os conceitos relacionados ao *hardware*, à arquitetura de rede, as camadas de computação, os frameworks utilizados para processamento de dados de sensores e a técnica utilizada para manter as listas de *ranking* de forma distribuída. Fechando o capítulo é apresentado o conceito de *ranking* de sensores utilizado neste trabalho de pesquisa.

No Capítulo 3, Estado da Arte, são apresentados os trabalhos relacionados à proposta da tese. Foram considerados os trabalhos com foco em *ranking* de sensores.

O Capítulo 4, *Fog Online Context-aware Up-to-date Sensor Ranking* (FOCUSeR), detalha o método proposto FOCUSeR.

No Capítulo 5, Ambiente Experimental e Resultados, são descritos o ambiente experimental e os resultados obtidos após a implementação e execução dos componentes propostos. Esses resultados são discutidos no Capítulo 6, Discussão, no qual é realizada uma discussão analítica com o objetivo de pontuar desafios encontrados e resultados obtidos.

O Capítulo 7, Conclusões e Trabalhos Futuros, conclui a tese e apresenta os objetivos alcançados com a pesquisa, além de apresentar possibilidades de futuras pesquisas sobre o tema.

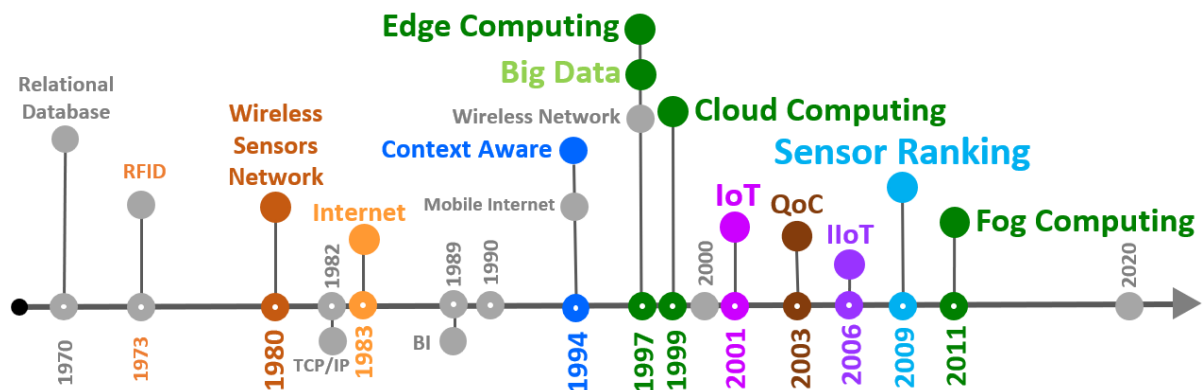




## 2 FUNDAMENTAÇÃO TEÓRICA

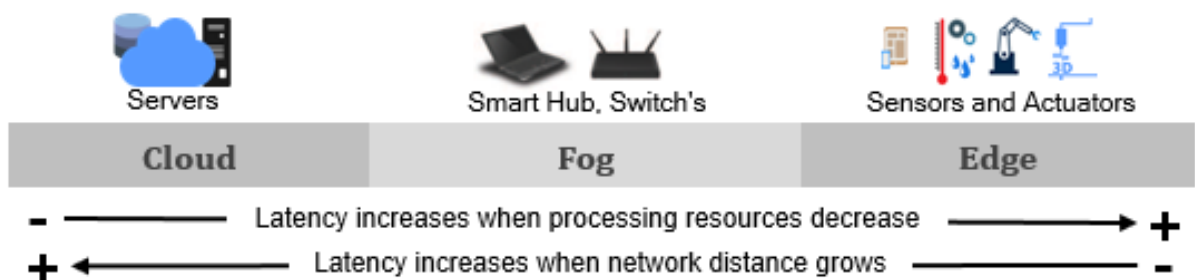
A linha do tempo<sup>1</sup> mostrada na Figura 2 apresenta as diversas tecnologias relacionadas ao crescimento do volume de dados produzidos por sensores, bem como as tecnologias que surgiram para atender as demandas de serviços e aplicações. Analisando a imagem, é possível verificar que há uma tendência na miniaturização dos dispositivos e que a análise e o processamento dos dados têm se movido em direção a esses dispositivos.

Figura 2. Tecnologias relacionadas ao crescimento dos dados digitais.



Fonte: Elaborada pelo autor (2021).

Figura 3. Recursos e distância versus latência.



Fonte: Elaborada pelo autor (2022).

Como pode ser visto na Figura 3, existem dois fatores que afetam diretamente a latência: a distância da rede e o poder de processamento dos dispositivos. Dessa forma, para reduzir ainda mais a latência no uso do serviço, o processamento dos dados do método FOCUSeR é realizado na camada *Fog* da rede. Isso fornece processamento, análise e armazenamento de dados próximos aos usuários finais, permitindo que as ferramentas sejam implantadas fora da nuvem. Isso também favorece o uso de aplicativos de alto desempenho

<sup>1</sup> Ver Apêndice D para referências utilizadas.

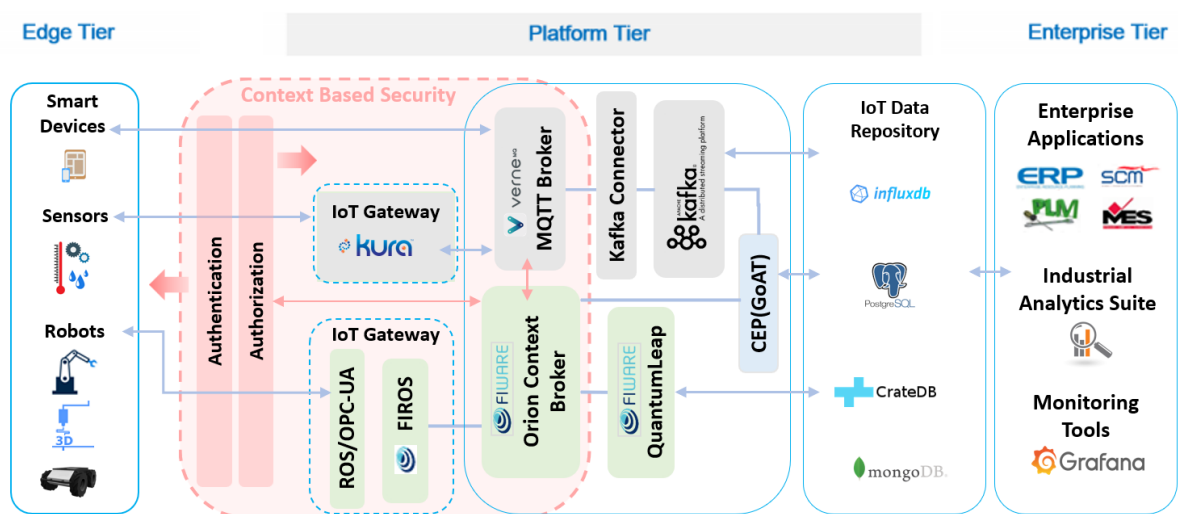
capazes de processar e armazenar dados próximos de onde são gerados, oferecendo latência muito baixa e responsividade inteligente em tempo real (AWS FOR THE EDGE, [s. d.]; MICROSOFT AZURE IOT SUITE, [s. d.]).

Tendo essas questões em mente, as próximas seções descrevem os principais conceitos relacionados com esta tese.

## 2.1 PROJETO FASTEN

*Flexible and Autonomous Manufacturing Systems for Custom-Designed Products* (FASTEN) foi um projeto financiado pelo programa *Horizon 2020* da União Europeia (COSTA *et al.*, 2020). O projeto FASTEN teve como objetivo desenvolver um *framework* aberto e padronizado para produzir e entregar itens projetados sob medida, capaz de ser executado de forma autônoma, além de utilizar manufatura aditiva de baixo custo. Para atingir esse objetivo, foram combinados serviços/produtos digitais integrados a processos de fabricação aditiva, além de ferramentas para descentralizar a tomada de decisões e o intercâmbio de dados. Tecnologias sofisticadas de software para autoaprendizado, auto otimização e controle avançado foram aplicadas para criar um sistema de manufatura aditiva totalmente conectado. Uma visão geral dos componentes de software do *framework* pode ser vista na Figura 4. A avaliação do projeto foi realizada em dois casos piloto: um especificado pela Embraer Portugal (EMBPT), em Portugal, e outro pela Thyssenkrupp (TSK) no Brasil.

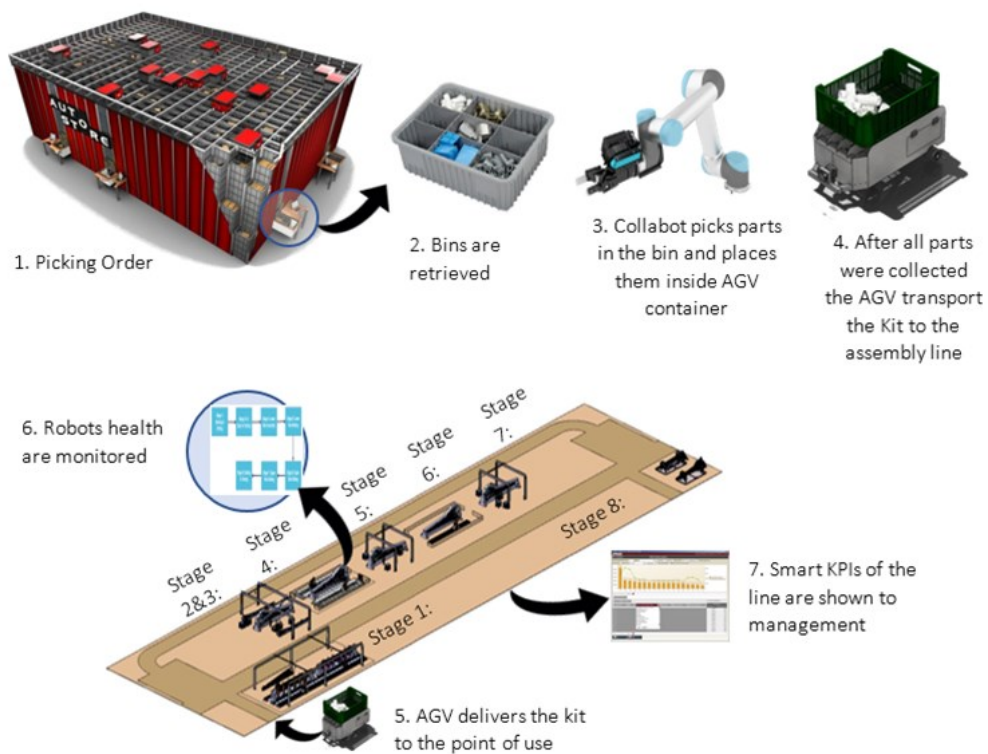
Figura 4. Plataforma IIoT FASTEN.



Fonte: Elaborada pelo autor (2020).

A TSK e a EMBPT buscavam superar permanentemente desafios como os descritos acima. A TSK fornece, do Brasil para países da América do Sul, diversos componentes para serem utilizados durante os serviços de manutenção de elevadores e escadas rolantes. Nesse caso, aspectos como nível de serviço, tempo e custo para fabricar peças exclusivas e custos de estoque podem ser melhorados drasticamente com a incorporação de tecnologias IoT, tanto na fabricação quanto nos processos de manuseio de materiais. O principal objetivo aqui era o de implementar um sistema que proporcionasse disponibilidade imediata de peças, melhorando assim as operações de manutenção.

Figura 5. Cenário Embraer.



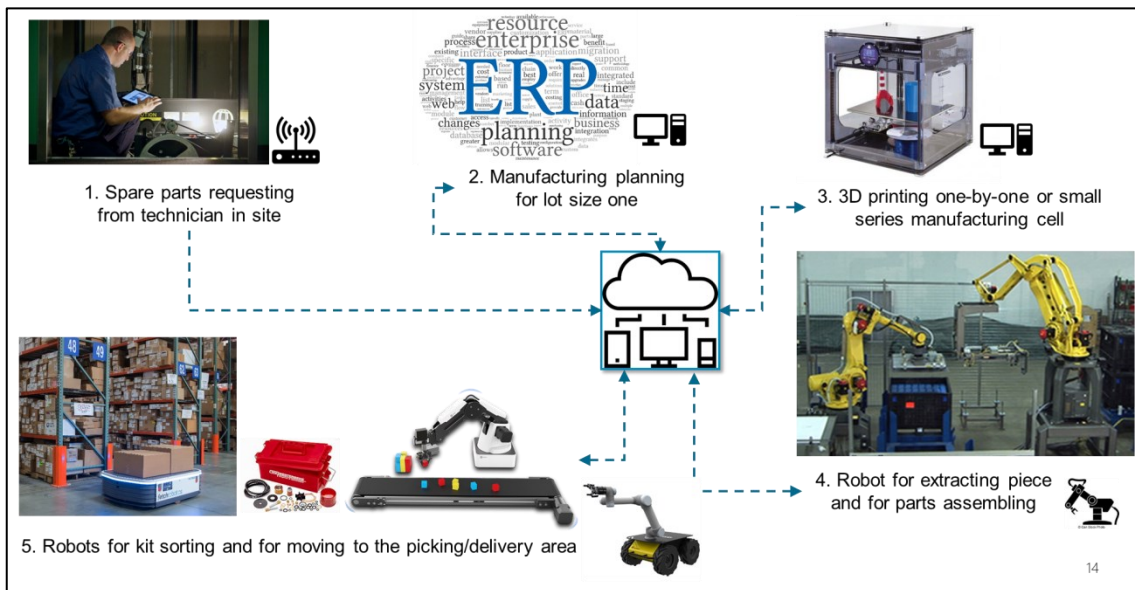
Fonte:(COSTA *et al.*, 2020).

A EMBPT desejava aumentar a sua flexibilidade de armazenamento, para poder lidar com mudanças nas demandas. Isto implicava em uma nova abordagem baseada em soluções inovadoras de robótica e gestão de estoques. Robôs avançados e estratégias de localização mais inteligentes, IoT e tecnologias de aprendizagem de máquina foram utilizados para melhorar o armazenamento, localização, recuperação e agrupamento de materiais, para maximizar a automação, a flexibilidade e capacidade de resposta.

A Figura 5 apresenta a manipulação das peças no estoque desejado pela *Embraer*. A imagem aponta o caminho das peças, selecionadas por robôs nos armazéns e levadas até a linha de produção de asas de avião.

A Figura 6 apresenta o cenário *Thyssenkrupp*, no qual são ilustrados os processos de manufatura aditiva, ou seja, a produção de itens customizados por impressoras 3D, bem como a sua movimentação para o estoque.

Figura 6. Cenário Thyssenkrupp.



Fonte:(COSTA *et al.*, 2020).

A participação no projeto possibilitou um rápido aprendizado sobre os conceitos e arquiteturas de *frameworks* para o ambiente IoT. Através dessa participação foi possível visualizar vários desafios relacionados a manipulação de dados obtidos de sensores.

## 2.2 PERCEPÇÃO ATIVA

A percepção ativa pode ser definida como um processo inteligente de aquisição de dados. Este processo deve definir e medir parâmetros e erros ambientais, que, por sua vez, podem ser retroalimentados para controlar o processo de aquisição de dados. A importância desse processo é que o tempo não é gasto processando e aprimorando dados imperfeitos, mas sim aceitando dados imperfeitos e ruidosos como uma questão de fato e incorporando-os à estratégia geral de processamento (BAJCSY, 1988). Assim, a percepção não é um processo passivo, mas ativo.

Considerando a abordagem humana da percepção, durante o ato de olhar, as pupilas se ajustam ao nível de iluminação, os olhos trazem o mundo em foco, convergem ou divergem, as cabeças se movem para melhorar a visão de algo (BAJCSY, 1988). Analogamente, na abordagem computacional, o problema da percepção ativa pode ser definido como o uso de estratégias de controle aplicadas ao processo de aquisição de dados, que, por sua vez, dependerá da interpretação dos dados e do objetivo da tarefa. Essa adaptabilidade é crucial para a sobrevivência em um mundo incerto (BAJCSY, 1988).

Enquanto a percepção depende do contexto, de outros sentidos e do tempo, a percepção ativa se divide em quatro níveis: sensação, percepção, percepção ao longo do tempo e percepção ativa. A sensação envolve o processo de captura dos dados. Após a captura, a percepção interpreta e dá sentido a esses dados. Observando um processo, ao longo de um período de tempo, é possível melhorar a interpretação desses dados. Esse processo é chamado de percepção ao longo do tempo, indicando que é possível avaliar a percepção em relação a outra percepção. O último nível é a percepção ativa, que inclui raciocínio, decisão e ações baseadas nas informações recebidas (BIEL; WIDE, 2000, 2002; WIDE, 2012). A sensação é realizada pelos sensores. Assim, no primeiro nível de avaliação dos dados no método FOCUSeR, o da percepção, a avaliação é realizada utilizando apenas uma amostra, ou seja, um único dado. No segundo nível, o da percepção ao longo do tempo, a avaliação é realizada utilizando um pequeno conjunto de dados. No terceiro e último nível, o da percepção ativa, são tomadas as decisões, considerando as duas primeiras avaliações, e ações são tomadas nos casos em que são identificadas falhas nos dados. Nesses casos, ocorre uma busca por dados similares em sensores próximos.

### 2.3 DETECÇÃO DE ANOMALIAS

A detecção de anomalias refere-se ao problema de encontrar padrões em dados que não estão de acordo com o comportamento esperado. Esses padrões não conformes são frequentemente referidos como anomalias e *outliers* (CHANDOLA; BANERJEE; KUMAR, 2009). Alguns problemas que ocorrem durante a operação dos sensores podem levar à não confiabilidade dos dados coletados, resultando em resultados imprecisos no processamento (CAI *et al.*, 2022). *Outliers* com origem nas WSNs geralmente são causados por erros ou por eventos (GHORBEL *et al.*, 2013). Um erro é representado por uma mudança significativa na leitura de um sensor ou uma grande diferença entre os dados de um determinado sensor e

outros sensores próximos. Em WSNs, os erros geralmente se originam de leituras instáveis ou imprecisas do sensor causadas por ruído do sensor, desvio de tempo, erro humano etc. Um evento geralmente decorre de mudanças repentinas nas condições do mundo real, como incêndios florestais, poluição do ar ou mudanças ambientais (CAI *et al.*, 2022).

De acordo com a literatura, os *outliers* podem ser classificados em *outliers pontuais*, *outliers contextuais* e *outliers coletivos* (CHANDOLA; BANERJEE; KUMAR, 2009). Se uma instância de dados individual pode ser considerada anômala em relação ao restante dos dados, a instância é denominada como um ponto atípico. Em geral, os pontos atípicos geralmente são causados por erros. Se uma instância de dados for anômala em um contexto específico, mas não em outro, ela será denominada como um *outlier contextual*. Se uma coleção de instâncias de dados relacionadas for anômala em relação a todos os padrões no conjunto de dados, ela será denominada como um *outlier coletivo*. As instâncias de dados individuais em um *outlier coletivo* podem não ser *outliers* por si só, mas sua ocorrência conjunta como uma coleção, sim (CHANDOLA; BANERJEE; KUMAR, 2009). Em geral, os *outliers coletivos* vêm de múltiplos erros de medição contínua do sensor (CAI *et al.*, 2022).

## 2.4 INTERVALO INTERQUARTIL

A métrica Intervalo Interquartil (IQR) é útil para indicar se os valores de um conjunto de dados podem ser considerados *outliers* (WITTEN; FRANK; HALL, 2011). Os quartis são os separadores que dividem um conjunto de dados numéricos em quatro partes iguais. O primeiro quartil (Q1) é o valor do conjunto que delimita os valores 25% menores. O segundo quartil (Q2) é a própria mediana. O terceiro quartil (Q3) é o valor que delimita os valores 25% maiores. Assim, o primeiro intervalo de dados começa no menor valor e vai até o Q1. A segunda faixa, entre 25% e 50%, são valores maiores que Q1 e menores que Q2, e assim por diante (BARBETTA; BORNIA; REIS, 2010). O intervalo interquartil é a diferença entre o Q3 e o Q1. O objetivo é criar limites utilizando o valor do índice IQR para permitir a identificação de valores que estão longe dos valores frequentes do conjunto de dados.

### 2.4.1 *t-digest*

Ao lidar com séries temporais, muitas estatísticas interessantes são difíceis de calcular. Por exemplo, para encontrar o valor mais comum, ou o número de valores distintos,

ou mesmo apenas a mediana sem classificar todos os dados, a abordagem mais popular é calcular um pequeno resumo dos dados, chamado de resumo (*digest*) ou esboço, e em seguida, usar esse resumo para estimar o valor desejado, como uma mediana ou um quartil. O método *t-digest* fornece essas estatísticas com uma precisão de uma ordem de magnitude melhor do que outros algoritmos existentes e com baixíssima complexidade computacional (DUNNING, 2021).

## 2.5 MATRIX PROFILE

Esta seção descreve o algoritmo *matrix profile* (YEH *et al.*, 2016) e algumas de suas variações utilizadas neste trabalho. O algoritmo é utilizado no método para criar junções de subsequências de séries temporais. O algoritmo é simples, rápido, paralelizável e praticamente livre de parâmetros, podendo ser atualizado de forma incremental. Como será ilustrado, o algoritmo é extremamente eficiente para tarefas de descoberta de *discórdias*, que são subsequências maximamente diferentes de todo o restante dos dados (YEH *et al.*, 2016).

Uma série temporal  $T$  é uma sequência finita de amostras  $T = t_1, t_2, \dots, t_n$ , tomadas em instantes crescentes de tempo, com comprimento  $n$ . Uma *subsequência* é um pequeno subconjunto contínuo  $T_{i,m}$  obtido de  $T$ , de comprimento  $m$ , começando em  $i$ .  $T_{i,m} = t_i, t_{i+1}, \dots, t_{i+m-1}$ , onde  $1 \leq i \leq n-m+1$ . Uma *subsequência* refere-se a uma sequência de amostras de séries temporais ordenadas no tempo.

Nos algoritmos que serão apresentados, a métrica de distância euclidiana é utilizada para produzir uma medida de similaridade entre as *subsequências*. A distância zero, ou seja, máxima similaridade, ocorre apenas quando dois segmentos são muito semelhantes em todas as  $n$  amostras. A fórmula da distância euclidiana é apresentada na Equação 1.

$$d(t_1, t_2) = \sqrt{\sum_{i=1}^m (t_1 - t_2)^2} \quad (1)$$

Um *distance profile*  $D$  é um vetor das distâncias euclidianas entre uma determinada consulta e cada *subsequência* em um conjunto de todas as *subsequências*. Quando o conjunto de consulta e todas as *subsequências* pertencem à mesma série temporal, o *distance profile* deve ser zero no local da consulta e próximo de zero antes e logo depois (YEH *et al.*, 2016).

Dados dois conjuntos de todas as *subsequências*  $A$  e  $B$  e duas *subsequências*  $A[i]$  e  $B[j]$ , uma função de junção  $INN-\theta Inn(A[i], B[j])$  é uma função booleana que retorna “verdadeiro” somente se  $B[j]$  é o vizinho mais próximo de  $A[i]$  no conjunto  $B$ .

Dados os conjuntos de todas as *subsequências*  $A$  e  $B$ , um conjunto de junção de similaridade  $J_{AB}$  de  $A$  e  $B$  é um conjunto contendo pares de cada *subsequência* em  $A$  com seu vizinho mais próximo em  $B$ :  $J_{AB} = \{ \langle A[i], B[j] \rangle \mid \theta Inn(A[i], B[j]) \}$ . Isso é formalmente indicado como  $J_{AB} = A \bowtie_{\theta Inn} B$ .

Neste contexto, um *matrix profile*  $P_{AB}$  é um vetor das distâncias euclidianas entre cada *subsequência* em  $A$  e seu vizinho mais próximo em  $B$ . Esse vetor é chamado de *matrix profile* porque uma maneira de calculá-lo seria calcular a matriz de distância completa de todas as *subsequências* em uma série temporal com todas as *subsequências* em outra série temporal e extrair o menor valor em cada linha (o menor valor não diagonal para o caso de auto junção). O *profile* tem uma série de propriedades interessantes e exploráveis. Por exemplo, o ponto mais alto no *profile* corresponde à *discórdia* da série (YEH *et al.*, 2016), os pontos mais baixos correspondem às localizações do padrão mais comum na série temporal (TRUONG; ANH, 2019), e a variância pode ser vista como uma medida da complexidade de  $T$ . Este caso especial do conjunto de junção de similaridade é denominado como *self-similarity join set*. Cada elemento no *matrix profile* nos informa a distância euclidiana até o vizinho mais próximo, no entanto, não nos diz onde esse vizinho está localizado. Essas informações são registradas no elemento *matrix profile index*, um vetor de inteiros que permite recuperar com eficiência o vizinho mais próximo de um elemento acessando a posição indicada por cada inteiro.

O algoritmo de *Mueen* para busca de similaridade (MASS) é um algoritmo de busca de similaridade de distância euclidiana para dados de séries temporais. O algoritmo encontra o vizinho mais próximo de uma consulta e, diferentemente das dezenas de algoritmos de busca *kNN* de séries temporais na literatura (DING *et al.*, 2008), este algoritmo calcula a distância para cada *subsequência*, ou seja, o *distance profile* da série temporal  $T$ . Para isso, utiliza um algoritmo de busca de similaridade rápida sob distância euclidiana como uma sub-rotina, explorando a sobreposição entre *subsequências* usando o algoritmo *Fast Fourier Transform* (FFT) para calcular os produtos escalares entre a consulta e todas as *subsequências* da série temporal (YEH *et al.*, 2016). Como resultado, é produzida uma linha completa da matriz de semelhança de todos os pares. O algoritmo *Scalable Time Series Anytime Matrix Profile* (STAMP) (YEH *et al.*, 2016) é um loop que calcula cada linha completa da matriz de



semelhança de todos os pares e atualiza o perfil atual da matriz, ou o “melhor até agora”, quando necessário.

STAMP é a versão *batch* do algoritmo. *Batch* significa que o algoritmo precisa ver toda a série temporal  $T_A$  e  $T_B$  (ou apenas  $T_A$  se um *self-similarity join matrix profile* estiver sendo calculado) antes de criar o *matrix profile*. Nesta tese o interesse é construir o *matrix profile* de forma incremental. Assim, após o *matrix profile* ser construído utilizando uma abordagem *batch*, quando novos dados chegam, é realizado um ajuste incremental do *profile* atual. Este algoritmo é chamado de algoritmo STAMPI (STAMP Incremental) (YEH *et al.*, 2016).

## 2.6 MÉTRICAS DE AVALIAÇÃO

Para avaliar a eficácia do desempenho de detecção de anomalias foram utilizadas as mesmas métricas presentes nos trabalhos relacionados e que também utilizaram os mesmos conjuntos de dados. Assim foram utilizadas as métricas *Acurácia*, *Precisão*, *Recall* e *F1 Score*, apresentadas nas Equações 2, 3, 4 e 5 respectivamente. Nestas equações, deve-se considerar a seguinte notação: TP = verdadeiro positivo; TN = verdadeiro negativo; FP = falso positivo; e FN = falso negativo. Os resultados são apresentados em porcentagem.

$$Acurácia = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Precisão = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$F1\ Score = 2 * \frac{Recall*Precision}{Recall+Precision} \quad (5)$$

## 2.7 CONSCIÊNCIA E QUALIDADE DE CONTEXTO

O processamento de dados que tenta entender o “contexto” ou o ambiente, situação ou estado usando dados de sensores para então agir de forma autônoma, é chamado de “computação sensível ao contexto”. Essas informações de contexto relativas a entidades do

mesmo tipo e obtidas simultaneamente podem diferir em termos de exatidão, probabilidade de correção, confiabilidade, etc. Essa qualidade do contexto pode ser medida em termos de atributos como acurácia, cobertura, frequência, repetibilidade, resolução e pontualidade. Essas métricas são chamadas de parâmetros da "Qualidade de Contexto" (QoC) (BUCHHOLZ; SCHIFFERS, 2003; SEZER; DOGDU; OZBAYOGLU, 2018). Dessa forma, aplicações sensíveis ao contexto podem se beneficiar da QoC quando são fornecidas métricas de QoC que são avaliadas levando em consideração os requisitos relativos à coleta, processamento e fornecimento de informações de contexto (MANZOOR; TRUONG; DUSTDAR, 2014). Deve-se notar que “dados” são dados brutos (*raw*), retirados diretamente de sua fonte (sensores), e “informações de contexto” são dados processados obtidos pelo processamento de dados *raw* e geralmente com metadados adicionados (PERERA *et al.*, 2014).

Os parâmetros de QoC utilizados neste trabalho são descritos a seguir. O parâmetro *trustworthiness* ( $T_i$ ) da QoC (Equação 6) é utilizado como indicador para gerar o *ranking* de sensores. No parâmetro, o valor 0 (zero) significa que esta fonte não é confiável, e 1 (um) representa confiabilidade total na fonte de contexto (BRINGEL FILHO; AGOULMINE, 2011). Na Equação 6, o parâmetro  $T_i$  é a confiabilidade do sensor, o  $ctx_i$  é o conjunto de elementos de contexto confiáveis para um sensor e  $W_i$  é o número total de elementos de contexto e deve ser maior que zero.

$$T_i = \frac{\text{NumberOfReliableSamples}(ctx_i)}{W_i} \quad (6)$$

O parâmetro *sensor\_location* (MANZOOR; TRUONG; DUSTDAR, 2014), que descreve a localização do sensor no momento em que a informação de contexto foi coletada, é utilizado para possibilitar a recuperação de dados de sensores semelhantes (com as mesmas características) ao redor de algum sensor que está em estado anormal.

Após realizar uma busca utilizando os dados de localização, os dados encontrados são avaliados com os parâmetros de QoC descritos abaixo. Esses parâmetros foram usados para verificar a validade das informações de contexto ao longo do tempo, ou seja, para verificar se os dados não expiraram de acordo com os parâmetros.

$$\text{Age}(ctx_i) = t_{curr} - t_{meas}(ctx_i) \quad (7)$$

$$\text{Timeliness}(ctx_i) = \begin{cases} 1 - \frac{\text{Age}(ctx_i)}{\text{TimePeriod}(ctx_i)} & : \text{ se } \text{Age}(ctx_i) < \text{TimePeriod}(ctx_i) \\ 0 & : \text{ senao} \end{cases} \quad (8)$$

O parâmetro *Age*, idade do objeto de contexto  $ctx_i$  (Equação 7), é calculado tomando a diferença entre o horário atual (*current*),  $t_{curr}$ , e o horário da medição (*measurement*) do objeto de contexto  $ctx_i$ ,  $t_{meas}(ctx_i)$ . O parâmetro *Timeliness* (Equação 8) é uma medida que indica o grau de frescor de um objeto de contexto em um determinado momento (MANZOOR; TRUONG; DUSTDAR, 2014).

Como a situação em ambientes pervasivos muda rapidamente, serviços que usam objetos de contexto sem conhecer esse valor podem tomar decisões erradas causando prejuízos ou desperdício de recursos. Essa característica de um objeto de contexto pode ser medida objetiva e subjetivamente. Para obter a visão subjetiva, considera-se o tempo de validade do objeto de contexto mencionado pelo usuário, independentemente do período de tempo (MANZOOR; TRUONG; DUSTDAR, 2014). No entanto, como a atividade de consulta de sensores está fora do escopo do método, o valor de 0,8 foi definido como limite para a idade dos dados. Esse valor é parametrizável por conjunto de dados. O parâmetro *TimePeriod* indica o intervalo de tempo entre duas medições. O valor de *Timeliness* e a validade do objeto de contexto diminuem à medida que a idade desse objeto de contexto aumenta (MANZOOR; TRUONG; DUSTDAR, 2014).

## 2.8 SENSORES

Os avanços na fabricação de sensores de baixo custo e a miniaturização de sistemas microeletrônicos são os principais responsáveis pelo desenvolvimento das tecnologias de sensoriamento, computação e comunicação, representadas por sensores inteligentes (PATTAR *et al.*, 2018). Um nodo composto por um sensor inteligente é um dispositivo com baixo consumo de energia, equipado com uma fonte de alimentação, normalmente uma bateria, um processador, memória, um rádio e pelo menos um sensor. Eventualmente atuadores podem ser incorporados para permitir ajustes nos parâmetros do sensor, movimentação do sensor ou mesmo monitoramento da energia disponível. Como os sensores geralmente são implantados em locais de difícil acesso, a comunicação precisa ser feita por meio de dispositivos sem fio, como rádios, responsáveis pela transferência de dados para um ponto de acesso em uma infraestrutura fixa (YICK; MUKHERJEE; GHOSAL, 2008).

Esses nodos sensores coletam informações sobre o ambiente ao redor e transmitem esses dados para a Internet. A demanda por monitoramento ambiental, que, por meio do controle de valores críticos, evita falhas e minimiza o impacto de eventos extremos causados

por mudanças climáticas ou intervenções humanas, tem alavancado o desenvolvimento das WSNs. Esse desenvolvimento iniciou por volta da década de 1980, mas foi somente em 2001 que essas redes realmente se tornaram foco de pesquisa e interesse da indústria (YINBIAO, 2014).

## 2.9 WIRELESS SENSOR NETWORKS

Nos anos 2000, empresas começaram a se concentrar em automação residencial e casas inteligentes. Os aplicativos e produtos se basearam principalmente na comunicação com eletrodomésticos e na resposta às pessoas via tecnologia com e sem fio. Por outro lado, as tecnologias M2M (*machine-to-machine*) propuseram soluções mais abrangentes para que as máquinas pudessem se comunicar diretamente, usando tecnologias com ou sem fio, criando as chamadas redes de sensores sem fio (*wireless sensor networks* ou simplesmente WSN). Em uma WSN, diferentes tipos de sensores, como temperatura, pressão e som, cada um se comportando como um único nodo, criam redes para transmitir dados por essa rede. As WSNs podem consistir em centenas ou milhares de nodos (SEZER; DOGDU; OZBAYOGLU, 2018).

Normalmente, uma WSN possui pouca ou nenhuma infraestrutura e consiste em um conjunto de nodos sensores (algumas dezenas a milhares) trabalhando juntos para monitorar uma região em busca de dados sobre o ambiente. Ao contrário das redes tradicionais, uma WSN normalmente possui restrições de recursos que incluem quantidade limitada de energia, curto alcance de comunicação, baixa largura de banda e processamento e armazenamento limitados em cada nodo (DESAI; JAIN; MERCHANT, 2007; YICK; MUKHERJEE; GHOSAL, 2008). No caso da IoT, desenvolvida em paralelo com as WSNs (ASHTON, 2009), não se presume uma tecnologia de comunicação específica (YINBIAO, 2014).

As WSNs são enormes fontes de geração de *big data* devido ao grande número de nodos conectados à rede. Nessas redes, dados em larga escala coletados por sensores podem ser distribuídos, processados e utilizados por diversos serviços e aplicações.

## 2.10 BIG DATA, CLOUD, FOG E EDGE COMPUTING

Nas últimas décadas houve um aumento exponencial do volume de dados nas mais diferentes aplicações e uma crescente demanda de extração de valor desses dados. Para

denominar esse aumento na geração de dados, surge o termo *big data*, utilizado principalmente para descrever enormes conjuntos de dados, normalmente não estruturados (CHEN; MAO; LIU, 2014). Em decorrência disso, armazenar e gerenciar esses enormes conjuntos de dados heterogêneos com requisitos moderados em infraestrutura de *hardware* e *software* tornou-se um grande desafio. Os desafios envolvidos nas atividades de gerenciamento e armazenamento desses dados costumam ser identificados como os “Vs” do *big data*, segundo o modelo proposto pela IBM em 2012 que definiu três dimensões principais relacionadas a *big data*: volume, velocidade e variedade (ZIKOPOULOS *et al.*, 2012). Com o passar do tempo, outras características como valor e veracidade têm sido incorporadas, motivando o surgimento de novas soluções (ZHANG *et al.*, 2017). Trabalhos mais recentes (STOREY; SONG, 2017) citam pelo menos doze dimensões do *big data*.

Assim, como forma de superar os obstáculos introduzidos pelo *big data*, surge o paradigma de computação em nuvem (*Cloud Computing*), que oferece recursos virtualmente ilimitados (escalabilidade), processamento paralelo e a integração de serviços e armazenamento (ZHANG *et al.*, 2017).

O modelo de *Cloud Computing* é uma ótima alternativa para gerenciar esses grandes volumes de dados, proporcionando maior capacidade de processamento sem afetar o desempenho. No entanto, quando se tratam de aplicativos sensíveis à latência, essas características podem não ser suficientes para atender serviços onde os requisitos de atraso estão presentes (GOMES *et al.*, 2021). Esse tipo de serviço exige que os nodos da rede estejam próximos ao usuário final, de modo que o tempo entre a solicitação e a resposta seja o menor possível. As implementações na IoT requerem suporte de mobilidade e distribuição geográfica, além de reconhecimento de localização e baixa latência (BONOMI *et al.*, 2012).

Para superar esses obstáculos, a CISCO propôs a criação da plataforma *Fog Computing* (BONOMI *et al.*, 2012). *Fog Computing* é um paradigma de recursos horizontais, físicos ou virtuais que reside entre os dispositivos finais inteligentes e a nuvem. Esse paradigma oferece suporte a aplicativos sensíveis à latência, fornecendo computação distribuída e escalável, armazenamento e conectividade de rede (IORGA *et al.*, 2018).

Nesta arquitetura, cada nodo sensor inteligente é conectado a um dispositivo *Fog*. Esses dispositivos podem ser interligados e estão vinculados à nuvem (STOJMENOVIC, 2014). É um paradigma no qual os recursos de comunicação, processamento, controle e armazenamento são colocados na borda da Internet, próximos a dispositivos móveis, sensores, atuadores, objetos conectados e usuários finais (NSF, 2016).

Ao permitir o uso de dados produzidos por sensores, essa arquitetura agrega inteligência aos ambientes. Isso é possível acessando as informações necessárias relacionadas aos ambientes, coletando e analisando dados passados e presentes. Assim, esses dados irão apoiar a tomada de decisão ideal sobre as pessoas e seus ambientes, preferencialmente em tempo real (BONOMI *et al.*, 2012).

Esse paradigma permitiu a criação de uma nova geração de aplicativos e serviços. As características definidoras do Fog são: baixa latência e reconhecimento do local; ampla distribuição geográfica; mobilidade; grande número de nodos; interoperabilidade e associação; acesso predominantemente sem fio; forte presença de aplicações de *streaming* e tempo real; e heterogeneidade (IORGA *et al.*, 2018). O *Fog Computing* permite a interação proveitosa entre o *Cloud* e o *Fog*, especialmente quando se trata de gerenciamento e análise de dados (BONOMI *et al.*, 2012).

*Edge Computing*, por outro lado, é um paradigma no qual os recursos de comunicação, computação, controle e armazenamento são colocados na borda da Internet, próximos a dispositivos móveis, sensores, atuadores, coisas conectadas e usuários finais (NSF, 2016).

Os aplicativos executados no *Edge Computing* executarão ações localmente antes de se conectarem à nuvem. No entanto, essas ações são limitadas pelos recursos computacionais escassos, características dos dispositivos desta camada da rede.

## 2.11 MIDDLEWARE IoT

Para permitir o processamento desse grande volume de dados, na camada de software de um sistema computacional, foram adotadas soluções de *middleware* IoT (FATHY; BARNAGHI; TAFAZOLLI, 2018). Um *middleware* IoT é usado para gerenciar dados de sensores, permitindo seu uso por diferentes aplicações, atuando assim como uma *interface* entre o usuário/aplicativo e as redes de sensores e permitindo uma abstração da heterogeneidade inerente às fontes de dados de sensores (PATTAR *et al.*, 2018).

Um *middleware* IoT oferece recursos para aquisição, descoberta, indexação, *ranking* e consulta de sensores e dados relacionados a esses sensores (PATTAR *et al.*, 2018). A indexação consiste em armazenar e indexar dados sobre sensores, para acelerar o processo de seleção de um subconjunto de sensores.

O *ranking* de sensores é realizado priorizando critérios como qualidade dos dados, disponibilidade do dispositivo, eficiência energética e latência da rede. Essa tarefa pode ser realizada com base nos dados gerados pelo sensor (conteúdo) e nas informações disponíveis sobre as condições em que esses dados foram gerados (contexto). Aplicações como indústria e saúde, que exigem confiabilidade associada ao processamento e fornecimento de dados com baixa latência, são exemplos de cenários de uso (KERTIOU *et al.*, 2018).

## 2.12 TABELAS DE *HASH* DISTRIBUÍDAS

*Distributed Hash Tables* (DHTs) são excelentes soluções para gerenciar listas distribuídas. O método tem um alto grau de escalabilidade e suporte flexível para operações de consulta e atualização. O DHT possui uma abordagem descentralizada que fornece mecanismos rápidos para armazenamento, consulta e atualização. São construídas em redes sobrepostas nas quais os objetos da rede são espalhados e identificados com chaves únicas (CHERBAL; BOUKERRAM; BOUBETRA, 2016).

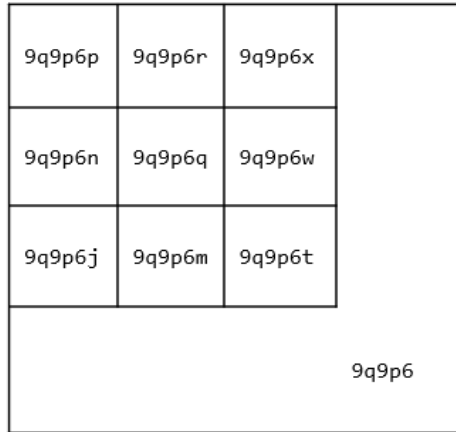
A ferramenta Apache Cassandra implementa um algoritmo da classe *Chord* (APACHE FOUNDATION, [s. d.]). O Cassandra alcança escalabilidade horizontal particionando todos os dados armazenados no sistema usando uma função de *hash* chamada *hash* consistente que é semelhante a um modelo *Chord* (STOICA *et al.*, 2001), e o modelo *Chord* por sua vez é baseado em DHT. Este modelo fornece um protocolo escalável e eficiente para pesquisa dinâmica em sistemas ponto-a-ponto (P2P) com entradas e saídas de nodos frequentes. Ele também fornece mecanismos para facilitar a localização de recursos, novas entradas de nodos e gerenciamento de falhas. Para melhorar a escalabilidade, o nodo *Chord* não precisa de informações de roteamento sobre todos os outros nodos, mas apenas um número de nodos  $O(\log N)$  e, portanto, a pesquisa é feita em no máximo  $O(\log N)$  mensagens.

Para tornar essa busca rápida e direta na lista de sensores, dados geoespaciais são previamente armazenados no Apache Cassandra. Ao armazenar os dados dessa forma, é possível encontrar pontos próximos a um determinado local. O desafio aqui é encontrar uma chave capaz de restringir a lista potencial de localizações e evitar consultar muitas chaves ao mesmo tempo (STRICKLAND, 2014).

Embora haja mais de uma estrutura de dados possível utilizada para este fim, o *geohashing* foi escolhido por oferecer diversos benefícios em relação a outras técnicas. Um

*geohash* é uma representação de base 32 de uma área geográfica, onde cada dígito adicional representa maior precisão.

Figura 7. Um *geohash* 9q9p6, com vários *geohashes* mais precisos contidos nele.



Fonte: Elaborada pelo autor (2021).

A propriedade dos *geohashes* que os torna particularmente adequados para pesquisa geoespacial é que adicionar um nível de precisão a um dado *geohash* resulta em uma área contida no valor de precisão mais baixo (STRICKLAND, 2014). Um exemplo pode ser visto na Figura 7, que mostra um diagrama com o *geohash* 9q9p6, contendo uma série de *geohashes* mais precisos contidos nele. Todos os *geohashes* menores começam com o prefixo 9q9p6.

### 2.13 RANKING DE SENSORES

O principal objetivo do *ranking* é priorizar e ordenar recursos selecionando os mais adequados. O algoritmo PageRank do Google (BRIN; PAGE, 1998) é provavelmente um dos algoritmos de *ranking* mais conhecidos. Ele explora os links entre as páginas da Web para atribuir pontuações aos documentos, que são usados em combinação com as métricas de similaridade de texto no contexto da pesquisa de documentos da Web (IGGENA *et al.*, 2021). Na Web, o Google usa o PageRank para classificar os resultados das consultas de pesquisa. Na IoT, os recursos podem ser relacionados entre si com base em seus tipos de recursos espaciais. No entanto, cada recurso pode ter muitos recursos, bem como diferentes dados de observação e medição. Nesse caso, a classificação de IoT deve ser um processo de tomada de decisão no qual vários critérios devem ser considerados dependendo do domínio da aplicação (FATHY; BARNAGHI; TAFAZOLLI, 2018). O *ranking* de recursos de IoT requer a priorização de critérios como por exemplo, qualidade de dados, disponibilidade de



dispositivos, energia eficiente e largura de banda, especialmente se houver os mesmos tipos de serviços (por exemplo, temperatura) no mesmo local (ELAHI *et al.*, 2009). Aplicativos de saúde, por exemplo, exigem dados confiáveis e de alta qualidade, casos de emergência requerem transmissão e processamento de dados com baixa latência para fornecer comandos on-line e mecanismos de controle (FATHY; BARNAGHI; TAFAZOLLI, 2018).

A ideia é gerar uma lista de sensores de forma que quanto maior o *ranking* de um sensor nessa lista, melhor deve ser o desempenho (funcionamento) desse sensor. Dessa forma, um mecanismo de pesquisa pode processar os sensores na ordem do seu *ranking*, gastando esforço nos sensores com maior probabilidade de corresponder à consulta. Esse esforço inclui, em particular, a leitura do valor de saída atual do sensor para verificar se ele realmente corresponde à consulta. Como esta é uma operação dispendiosa, o *ranking* do sensor ajuda a evitar o esforço inútil em sensores que provavelmente não correspondem à consulta e assim reduzir a latência de produção dos resultados (ELAHI *et al.*, 2009).



### 3 ESTADO DA ARTE

Aplicações como indústria, agricultura e saúde requerem confiabilidade associada ao processamento e publicação de dados em processos que atendam requisitos de baixa latência (FATHY; BARNAGHI; TAFAZOLLI, 2018). Nesse sentido, muitas propostas foram desenvolvidas envolvendo busca, indexação e *ranking* de sensores, a fim de fornecer uma maneira rápida e segura de recuperar dados de dispositivos IoT. Neste capítulo, são apresentados os trabalhos mais relevantes relacionados a *ranking* de sensores.

*IoT-Crawler* (IGGENA *et al.*, 2021) é um framework desenvolvido para a tarefa de busca em ambientes IoT. O *framework* é composto por duas camadas: a Camada de Rastreamento e Processamento que é responsável por integrar constantemente novas fontes de dados ao *framework*; e a camada de pesquisa e orquestração que contém componentes para lidar com solicitações de pesquisa e assinatura provenientes de aplicativos ou usuários IoT. Na Camada de Rastreamento e Processamento, o fluxo de dados é monitorado para permitir a detecção e a recuperação de falhas. Aqui também são gerados os índices de busca. Na Camada de Pesquisa e Orquestração, o Orquestrador é o ponto de entrada para qualquer usuário ou aplicativo que queira pesquisar dispositivos IoT. Essa camada pode fornecer um canal para receber notificações sobre falhas. Nesta camada, o componente de *ranking* usa os índices construídos, determinadas restrições e informações enriquecidas para classificar as fontes de dados encontradas antes de serem enviadas de volta ao usuário ou aplicativo.

Bharti propôs VoISRAM, um mecanismo de classificação de sensores baseado em VoI (BHARTI; PATTANAIK; BELLAVISTA, 2021), que se destina principalmente a explorar os serviços de *gateway* do mecanismo de *ranking* para encontrar um equilíbrio entre os requisitos de Qualidade de Serviço (QoS) específicos da aplicação e o consumo de energia da rede. O método propôs um mecanismo de *ranking* de sensores baseado em valor que modela o *ranking* de um sensor como um valor de atributo de informação considerando seu contexto de uso na aplicação correspondente. Lee (LEE; LEE, 2020) propõe uma técnica de recuperação de informações com reconhecimento de estado de ativos IoT para sistemas de energia elétrica. Em GaaS (ABDELAAL *et al.*, 2019) é proposta uma arquitetura de *gateway* multiplataforma para recuperar dados dos sensores de baixa potência. A arquitetura aproveita o *multithreading* para permitir que o *gateway* se conecte simultaneamente a vários sensores. Para regular as conexões *Bluetooth* de baixa energia, o trabalho desenvolveu dois

escalonadores distintos baseados em prioridades que fazem o *ranking* dos sensores, antes de gerar o escalonamento de conexão.

No trabalho de Nesa e Banerjee (2019) é proposto um algoritmo de ativação de sensor dinâmico, inspirado no algoritmo *PageRank*, chamado *SensorRank*. Ao contrário do algoritmo *PageRank*, que exige apenas os links externos para classificar páginas da Web, o *SensorRank* analisa dinamicamente a topologia da rede em termos de distâncias relativas, qualidades de link e a energia restante dos dispositivos, e utiliza essas informações para classificar os sensores. Um subconjunto ideal de sensores é ativado e os dados produzidos são incluídos em um processo de fusão de dados. Os dados dos sensores inativos (não utilizados) são regenerados utilizando a técnica de compressão de sensores que é uma técnica de amostragem capaz de recuperar dados/sinais utilizando poucas amostras.

Dautov e Distefano (2019) apresentam uma arquitetura descentralizada para agrupar dispositivos heterogêneos e executar fluxos de trabalho IoT, com uso intensivo de dados, no ambiente de *Edge Computing*. A abordagem proposta, inicialmente divide um fluxo de trabalho complexo em tarefas mais simples, em seguida descobre e seleciona dispositivos de borda adequados e, por fim, aloca as tarefas para os nodos selecionados, conectando-os para recompor o fluxo de trabalho original. A descoberta e seleção dos dispositivos de borda adequados é feita com base em requisitos de tarefas funcionais e não funcionais, respectivamente, antes de mapear as tarefas decompostas para os nodos selecionados. Os requisitos funcionais referem-se principalmente à disponibilidade de recursos de *hardware* específicos (CPU, RAM, GPU) e recursos de *software* (por exemplo, uma biblioteca ou um componente do SO). Com relação aos requisitos não funcionais, as tarefas individuais são anotadas com as demandas esperadas de processamento de *big data* que são conceitualmente classificadas usando três métricas: quantidade de dados envolvidos; rapidez com que esses dados serão gerados e processados; e a quantidade de dados heterogêneos. Para cada métrica, um valor numérico correspondente é fornecido por um especialista do domínio. O *ranking* dos dispositivos é calculado na etapa de seleção dos dispositivos utilizando os requisitos não funcionais.

No trabalho de Ruta et al. (2019) é proposto um *framework* que estende o protocolo CoAP (SHELBY; HARTKE; BORMANN, 2014) para compor uma *Web das Coisas* (WoT) semântica. O *framework* propõe um agente móvel capaz de descobrir dispositivos (*smartphones*) e compartilhar os dados dos sensores embarcados utilizando uma comunicação P2P. Cada sensor é descrito através de uma anotação semântica modelando suas capacidades

e características, além de atributos orientados a dados. Os nodos do coletor atuam como centros de cluster para sensores implantados localmente, comunicando-se via CoAP. Eles incorporam servidores CoAP para registrar sensores juntamente com suas descrições semânticas e suportam a descoberta de recursos baseada em lógica utilizando metadados anotados. Os coletores detectam eventos através da coleta e mineração de dados. Os eventos reconhecidos são anotados e armazenados, atualizando os registros de recursos no servidor. Um registro também inclui atributos extra lógicos dependentes do contexto, como registro de data e hora e coordenadas geográficas. Um *gateway* é conectado a vários coletores de forma que clientes que procuram eventos ou dispositivos em uma determinada área, enviam solicitações de descoberta de recursos via CoAP para o *gateway*, que responde em nome dos coletores conectados. Os *gateways* podem se comunicar formando micro redes federadas. O *ranking* dos dispositivos é baseado em uma métrica que calcula a distância semântica entre os requisitos do usuário e a descrição semântica de cada dispositivo. A distância geográfica do sensor com relação ao ponto de referência (parâmetros informados), e normalizada pela distância máxima especificada pelo usuário, é utilizada como fator de ponderação. O resultado do algoritmo é o conjunto de sensores que melhor atende a solicitação do usuário.

Kakunsi e Candra (2018) propõem um *framework* que consiste de módulos para consulta, descoberta, *ranking*, persistência, cache e notificação. O *framework* contempla um modelo para descrição de dispositivos IoT utilizando WSDL. O *framework* possui um recurso para manter consultas feitas pelos usuários, e mais tarde gerar notificações sobre possíveis alterações nessa lista que podem estar relacionadas a dispositivos que não estavam disponíveis inicialmente, disponibilidade ou qualidade da informação e adição de novos dispositivos na rede. O *ranking* dos dispositivos, assim como a qualidade e a consideração dos aspectos dinâmicos dos dispositivos IoT são calculados utilizando quatro atributos: tempo de resposta, cobertura, precisão e custo. O score final do dispositivo é gerado considerando pesos informados pelo usuário. Esses pesos são multiplicados pelos respectivos valores de cada atributo e ao final somados entre si. Por fim, a técnica *Analytical Hierarchy Process* (AHP) é utilizada para calcular a relevância entre os dispositivos e a solicitação do usuário, comparando cada candidato.

Em Kertiou et al. (KERTIOU *et al.*, 2018) os autores usam informações de contexto de sensores com operador *skyline* dinâmico para reduzir o espaço de pesquisa e selecionar os melhores sensores seguindo os requisitos do usuário. Para fornecer escalabilidade, os autores propõem o uso de *gateways* distribuídos conectados a um servidor, cada *gateway* respondendo

às solicitações locais dos usuários. O operador *skyline* é utilizado nos *gateways* de forma a prover escalabilidade para a solução e em seguida os dados são enviados para um servidor central, onde são agregados.

Em Dilli et al. (DILLI *et al.*, 2018), é apresentado um modelo para IoT que utiliza ontologias para descrever recursos e parâmetros de Qualidade de Serviço (QoS) como indicadores da qualidade. O algoritmo de *Multiple Criteria Decision Analysis* (MCDA) considera as preferências do cliente e é utilizada em conjunto com lógica *fuzzy* para reduzir os custos computacionais e tratar a incerteza na definição de pesos de atributos. A proposta executa uma pré-classificação de recursos por meio do algoritmo de aprendizagem de máquina *Logistic Model Tree* (LMT). O algoritmo MCDA refere-se à tomada de decisão na presença de critérios conflitantes e procura auxiliar na tomada de decisão usando um conjunto de objetivos e critérios, estimando a importância relativa através do uso de pesos. Na proposta, o *ranking* é calculado pelo algoritmo *Simple Additive Weighting* (SAW) que usa a pontuação obtida pelos valores de critérios normalizados, multiplicada pelos pesos correspondentes. Em seguida as opções são classificadas em ordem decrescente de acordo com a pontuação final, que é a soma das pontuações de cada critério. Os autores argumentam que a definição do grau de importância dos atributos de QoS pelo usuário e pelos administradores da infraestrutura computacional é uma atividade que depende da experiência e do conhecimento de cada um e por isso propõem o uso da lógica *fuzzy* na especificação dos parâmetros ideais de QoS.

No trabalho de Hussain, Anees e AzeemUllah (2018) é desenvolvido um mecanismo de pesquisa para a IoT. O trabalho compara diferentes modelos de descrição de sensores para encontrar campos semelhantes entre diferentes modelos e descrições de objetos inteligentes e os utiliza para criar um índice de pesquisa. A solução é fornecida na forma de uma API RESTful aberta visando permitir indexar e pesquisar objetos inteligentes. O *ranking* dos dispositivos é calculado com base na frequência com que cada dispositivo aparece nos resultados de uma consulta (quanto mais um dispositivo aparece na lista de sensores candidatos, maior sua pontuação).

Considerando que o processo de busca e descoberta pode ser dividido nas etapas de uso de uma consulta para encontrar recursos e na etapa seguinte, aplicar critérios de decisão para filtrar e classificar os dados de resposta, Nunes et al. (2016) propõem um algoritmo de Seleção-Eliminação (ES) para ser usado na segunda etapa. Os autores apresentam um estudo de caso agrícola utilizando o *Visual Search for Internet of Things* (ViSIoT) (NUNES, L. *et al.*,

2016) para comparar, em termos de qualidade e tempo de resposta, o algoritmo *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS), usado para pesquisa e descoberta de recursos, com o novo algoritmo ES.

Em Kang et al. (KANG *et al.*, 2016), os autores propõem um método para aumentar a colaboração entre os dispositivos IoT, em uma comunicação do tipo M2M (máquina a máquina). O método usa como base elementos como a frequência com que os usuários usam itens individuais, as distâncias entre usuários e dispositivos (serviços), o QoS da rede, o desempenho dos dispositivos e seus históricos de operações e a prioridade estabelecida pelo usuário. O objetivo é que as informações resultantes possam ser usadas por uma plataforma ou dispositivo, baseado na nuvem, para calcular as informações de *ranking* do dispositivo. Os dados mantidos por esse método, chamado Dispositivo-Rank, são acumulados e atualizados por dispositivos e em seguida, compartilhados entre eles para que a colaboração aconteça de forma mais ativa, visando melhorar o *ranking* dos dispositivos com o passar do tempo.

Em Zhang et al. (ZHANG *et al.*, 2016), é proposto o modelo *Low-Overhead and High-Precision Prediction* (LHPM) de forma a melhorar a pesquisa de sensores usando um método para reduzir o custo de energia na geração de dados. Quanto a abordagem distribuída, utiliza *gateways* locais que estão em contato com os sensores e geram os dados de previsão, e um servidor central, que recebe informações dos *gateways* e disponibiliza as respostas para os usuários. O trabalho também propõe um método de previsão para estimar os valores do sensor. Esses valores são utilizados para gerar o *ranking* de sensores, no qual são avaliadas as probabilidades de correspondência entre os sensores e os parâmetros informados pelo usuário. A lista gerada pelo processo de *ranking* é utilizada para selecionar apenas os sensores relevantes, reduzindo assim o consumo de energia, quando a comunicação (transferência de dados) desnecessária entre sensor e *gateway* (comunicação no processo de busca) é evitada. O método de previsão utilizado para definir qual o valor mais provável a ser gerado pelo sensor considera o aspecto dinâmico da geração de dados de sensores. Para isso, utiliza dados históricos no treinamento dos modelos através de um modelo adaptado a partir da técnica de mínimos quadrados com máquinas de vetor de suporte. Utilizando esses valores e os parâmetros da consulta do usuário o método faz um cálculo de distância para verificar o quanto cada sensor atende o escopo delimitado pela consulta. Como resultado desse processo, os sensores são ordenados em uma lista decrescente (*ranking*).

Neha e Saxena (2016) trabalham com pesos para importância de cada propriedade de contexto utilizada (disponibilidade, acurácia, confiabilidade, frequência, ...) e calculam o

valor do escore (*ranking*) de cada sensor utilizando uma métrica denominada *Preference Based Weighted Index* (PBWI).

No trabalho de Wang et al. (WANG *et al.*, 2015) é proposto um mecanismo de *ranking* de sensores baseado no custo de acesso ao serviço (dispositivo). O custo é calculado utilizando propriedades do sensor e informações de contexto. O método se propõe, além de oferecer uma solução com boa performance, a ser capaz de preservar os recursos de energia do sensor. Para cálculo do custo de acesso ao sensor é proposta uma função que estima o custo em relação às consultas e propriedades dos nodos de sensores relevantes (por exemplo, importância, energia e qualidade do *link*). Para isso, é proposto um método probabilístico para derivar importância de cada sensor com base na topologia de sobreposição da WSN. O *ranking* de cada sensor é calculado executando análises no processo de acesso ao serviço de sensores usando informações contextuais coletadas da WSN. O *framework* proposto pelo trabalho utiliza uma abordagem distribuída de quatro camadas: sensores, *gateways* responsáveis por indexar e calcular o *ranking* dos sensores associados, um servidor central no qual são agrupados os índices e respectivas informações de *ranking* de cada sensor e finalmente uma camada de consulta (acesso de clientes).

No trabalho de Yuen et al. (2014), é utilizado o método *Primitive Cognitive Network Process* (P-CNP) para derivar a importância dos nodos dos sensores com base em dois tipos de parâmetros de QoS: baseados na rede (latência, largura de banda e vazão) e baseados no sensor (acurácia, confiabilidade e custo). Os parâmetros recebem pesos de acordo com sua importância para o usuário e utilizando os valores dos atributos e pesos é gerado o conjunto de valores que irão permitir a geração do *ranking*. Os valores são dispostos em uma matriz e podem, de acordo com as prioridades estabelecidas pelo usuário, ser ordenados. Na avaliação do método foram utilizados apenas seis sensores.

No trabalho de Cabral, Compton e Müller (2014), os autores propõem uma estrutura baseada em ontologia para encontrar sensores que atendam aos requisitos do usuário. O esforço no desenvolvimento da ontologia se concentra em descrever os sensores (serviços) considerando a natureza dinâmica dos mesmos. Na proposta o *ranking* dos sensores é realizado tendo como base em critérios como localização e evento de calibração mais recente (dados armazenados na ontologia), e é administrada por uma função que mantém a lista de sensores e a qualidade dos dados por um determinado período. Na proposta, os parâmetros são classificados em propriedades necessárias e propriedades desejadas, semelhante as categorias negociáveis e não negociáveis do trabalho de Perera et al. (2014). Com relação ao



aspecto dinâmico da geração de dados de sensores, a proposta afirma que o uso do fator tempo é importante e pode afetar as decisões tomadas com base em dados gerados por sensores que nem sempre apresentam um bom comportamento, mas a abordagem de solução utilizada é feita utilizando dados de calibragem do sensor. Para isso, a proposta utiliza dois parâmetros da calibragem do sensor: a data da calibragem e o período que essa calibragem garante a confiabilidade dos dados do sensor. Esta abordagem, apesar de interessante, só tem sentido em ambientes controlados, ou seja, sensores que disponibilizam essa informação e que são periodicamente verificados. O algoritmo apresenta ainda uma função para o uso de dados com características de séries temporais. Essa função pode, ao ser solicitada pelo usuário, calcular algum valor consolidado (média por exemplo) sobre o conjunto de dados disponibilizado e isso pode levar a perda de informação relacionada a geração de dados ao longo do tempo. Outra desvantagem é que este cálculo só é executado após receber os parâmetros do usuário, ou seja, precisa ser calculado a cada solicitação e para cada sensor presente na lista inicial (candidatos) retornada pela consulta.

Em Niu et al. (2014), para ajustar as características do ambiente dinâmico da WSN, os autores propõem uma abordagem de classificação de serviço com reconhecimento de contexto, agregando a classificação do usuário e o contexto de serviço da WSN. Duas métricas de *ranking* são propostas: Avaliação de QoS do Usuário (UQA) e Avaliação de QoS de Contexto (CQA) e, através da influência do desempenho no serviço WSN pelas variações em seu contexto, um mecanismo *fuzzy* é desenvolvido para agregar o UQA e o CQA.

Um *middleware* IoT é proposto no trabalho de Perera et al. (2014). O *middleware* apresenta uma ferramenta de pesquisa na qual se estabelecem duas classes de parâmetros do usuário (negociáveis e não negociáveis). Por exemplo, se um usuário deseja medir a temperatura em um determinado local, o resultado (a lista de sensores), deve conter apenas sensores que possam medir a temperatura. O usuário não aceita qualquer outro tipo de sensor (por exemplo, sensores de pressão). Da mesma forma, a localização também não é negociável. Os parâmetros que não possuem essa restrição podem ser atendidos por aproximação da solução desejada. Por exemplo, se o usuário deseja receber como resposta sensores que apresentam uma acurácia próxima dos 90% o sistema pode retornar sensores com essas características mesmo se o custo ou o tempo para obter esses dados forem mais altos. O *middleware* possui uma arquitetura que pode ser executada com técnicas baseadas em *MapReduce* para aumentar a escalabilidade da solução. Os autores também desenvolveram uma estrutura de contexto baseada em ontologia para sensores na IoT, que permite capturar e

modelar propriedades de contexto relacionadas a sensores. Essas informações permitem que os usuários pesquisem os sensores com base no contexto. Para indexar e ordenar os sensores, as preferências do usuário são usadas com uma técnica de indexação utilizando a métrica de distância euclidiana ponderada chamada *Comparative Priority Based Weighted Index* (CPWI). Para o cálculo do *ranking*, cada sensor é plotado em um espaço multidimensional, onde cada propriedade de contexto é representada por uma dimensão. Em seguida, um sensor ideal é plotado no espaço multidimensional a partir das informações do usuário (propriedades de contexto). No passo seguinte, os pesos são calculados de forma comparativa, o indicador CPWI é calculado e finalmente os sensores são ordenados em ordem decrescente. Apesar de apresentar recursos interessantes, este trabalho utilizou como base o banco de dados MySQL para todos os testes da ferramenta, o que dificulta uma avaliação de comportamento da ferramenta em relação a um ambiente real IoT.

O trabalho de Truong, Römer e Chen (2012) apresenta uma proposta de pesquisa de sensores por similaridade. Na proposta, um usuário fornece um sensor e uma fração de sua saída passada como exemplo e solicita sensores que produziram saída semelhante no passado. No trabalho, esse serviço de pesquisa é denominado similaridade de sensores. A proposta utiliza como arquitetura um modelo com sensores conectados a *gateways* e esses *gateways* conectados a um servidor central. Com relação ao fator tempo, a proposta utiliza séries temporais nas buscas de sensores, quando são realizadas as comparações por similaridade. A comparação dos sensores é feita utilizando lógica *fuzzy* na comparação de cada um dos sensores candidatos com o sensor fornecido pelo usuário. O resultado da métrica de similaridade é utilizado para construir o *ranking* de sensores.

O framework *MediaSense* apresenta uma proposta de *ranking* de sensores distribuída (WALTERS; ÖSTERBERG; KANTER, 2011). Para manter as listas de sensores de forma distribuída, os autores utilizam P-Grid (ABERER *et al.*, 2003), que é um sistema de pesquisa P2P baseado em uma árvore de pesquisa distribuída virtual, estruturada de maneira semelhante às tabelas de *hash* distribuídas (ROCHA; BRANDÃO, 2019). Para construção do *ranking* os autores introduzem inicialmente o conceito de "Esquema de Contexto", definido como a coleção de pontos de informação associados (sensores) e contribuindo para o contexto atual de uma presença (um sensor em determinado local). Os objetos de esquema que contêm as relações entidade e contexto (*presentidade*) permitem a criação de uma coleção de sensores, que contribuem e expressam o contexto sobre uma *presentidade*. Essa abordagem é utilizada para obter conhecimento local de todas as instâncias em que um sensor *s* foi

utilizado por uma presença  $P$  na construção do esquema. Da mesma forma, é possível conhecer os esquemas que foram usados por essa presença  $P$  e, portanto, também é possível derivar alguma representação da importância de  $s$  em relação a  $P$ . Esse valor representa o *ranking* local de  $s$  e, como tal, o nodo onde  $s$  reside é, portanto, capaz de coletar e agregar esses valores, formando assim um valor que, ao final, representará um *ranking* geral de  $s$ . A geração do *ranking* global ocorre através da troca de informações entre os pontos combinada com os parâmetros informados pelo usuário. Basicamente, o cálculo do *ranking* é feito com base nas informações de contexto (utilização prévia do sensor). Com relação ao uso do tempo, o algoritmo permite que o *ranking* de determinado sensor seja calculado para um tempo específico  $t$ , mas sem alterações no cálculo, ou seja, o valor resultante da métrica aplicada irá desconsiderar os valores que foram gerados após o tempo  $t$ .

Snoogle (WANG; TAN; LI, 2010) é um *framework* com recursos para busca, transmissão de dados, recursos de segurança e consulta e recursos para geração de dados de sensores em larga escala. Para transmissão de dados, foram utilizadas técnicas de compactação que geram uma quantidade menor de informações a ser transmitida pela rede, mas gera incerteza sobre as informações originais, o que pode resultar em falsos positivos durante o processo de pesquisa. Para reduzir o custo de comunicação em consultas distribuídas, o trabalho apresenta um algoritmo de consulta distribuída. Com relação aos aspectos de segurança, é proposta uma estrutura de segurança e privacidade mais flexível que permite ao usuário realizar pesquisas em uma rede de objetos com sensores. O *framework* também apresenta um protótipo de recuperação de informações para redes de sensores. Para verificar a escalabilidade da ferramenta os autores desenvolveram um simulador capaz de gerar grandes volumes de dados. O *ranking* dos objetos retornados por uma consulta é realizado utilizando a relevância da descrição do objeto aos parâmetros informados na consulta, sem considerar aspectos relacionados a qualidade das informações geradas pelos sensores.

Elahi et al. (2009) foram os primeiros a utilizar o termo *ranking* de sensores. O trabalho se concentra no uso de dados produzidos por sensores centrados nas pessoas, considerando que os hábitos podem indicar o comportamento futuro, e assim utilizam os dados gerados por esses sensores para criar modelos de previsão. Essa abordagem, além de não atender a situações em que requisitos de baixa latência estão presentes, por ser muito específica, tem a necessidade do desenvolvimento (treinamento) de um modelo de previsão

para cada conjunto de dados, tornando a intervenção de um especialista obrigatória e necessitando de um longo tempo de preparação para uso.

Quadro 1. Trabalhos relacionados.

Trabalho	Características dos Recursos										
	Distribuído	<i>Fog/Edge</i>	Baixa Complexidade	Lista de <i>Ranking</i> Pronta	Avaliação de Dados	Tipo de <i>outlier</i>	<i>Online</i>	<i>Up-to-date</i>	Recomposição de Dados	Ambiente de Teste Real	Dados Reais
<b>FOCUSeR (2022)</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Mekala (2021)	✓	✓	-	-	✓	-	-	-	-	-	-
Iggena (2021)	✓	-	✓	-	✓	✓	-	-	✓	-	✓
Costa (2021)	-	✓	✓	✓	✓	-	-	-	-	-	✓
Bharti (2021)	✓	✓	✓	✓	-	-	-	-	-	-	-
Liu (2020)	-	-	-	-	-	-	-	-	-	-	✓
Lee (2020)	-	-	✓	-	-	-	-	-	-	-	-
Sundar (2019)	-	-	-	-	-	-	-	-	-	-	-
Biró (2019)	-	-	✓	-	-	-	-	-	-	-	-
Abdelaal (2019)	-	✓	✓	-	-	-	-	-	-	✓	✓
Costa (2019)	-	✓	✓	✓	✓	-	-	-	-	-	✓
Nesa (2019)	✓	-	-	✓	-	-	-	-	✓	-	✓
Dautov (2019)	✓	-	-	✓	-	-	-	-	-	✓	✓
Ruta (2019)	✓	✓	✓	-	✓	✓	-	-	-	✓	✓
Kakunsi (2018)	-	-	✓	-	-	-	-	-	-	-	-
Kertiou (2018)	✓	-	✓	-	-	-	-	-	-	✓	✓
Dilli (2018)	-	-	✓	✓	-	-	-	-	-	-	-
Hussain (2018)	-	-	✓	✓	-	-	-	-	-	-	-
Nunes (2018)	-	-	✓	-	-	-	-	-	-	-	✓
Kang (2016)	✓	✓	✓	✓	-	-	-	-	-	-	-
Zhang (2016)	✓	-	-	-	-	-	-	-	-	-	✓
Neha (2016)	-	-	✓	-	-	-	-	-	-	-	-
Wang (2015)	✓	-	✓	-	-	-	-	-	-	-	-
Yuen (2014)	-	-	✓	-	✓	✓	-	-	-	-	-
Cabral (2014)	✓	-	✓	✓	-	-	-	-	-	-	✓
Niu (2014)	-	-	-	-	-	-	-	-	-	-	-
Perera (2014)	✓	-	✓	✓	-	-	-	-	-	-	✓
Truong (2012)	✓	-	✓	✓	-	-	-	-	-	✓	✓
Walters (2011)	✓	✓	✓	-	-	-	-	-	-	-	-
Wang (2010)	✓	-	✓	✓	-	-	-	-	-	✓	-
Ostermaier (2010)	✓	-	✓	-	-	-	-	-	-	✓	✓
Elahi (2009)	✓	✓	-	✓	-	-	-	-	-	-	✓

Fonte: Elaborada pelo autor (2022).

O Quadro 1 apresenta a lista dos trabalhos relacionados apresentados. A simbologia da tabela utiliza um traço se a obra não atende ao recurso apresentado e um sinal especial se atende. As funcionalidades referem-se, nesta ordem, a: se o processamento de dados da proposta é realizado totalmente em ambiente de rede distribuído; em qual camada de computação (*Cloud, Fog ou Edge*) o algoritmo é executado; se o algoritmo possui baixa complexidade computacional; se a lista de classificação está sempre pronta (não há necessidade de gerá-la a cada consulta); se o trabalho considera a avaliação dos dados gerados pelo sensor; se a proposta identifica os tipos de *outliers*; se o processo de avaliação de dados requer treinamento; se o processo de avaliação de dados se mantém atualizado ao longo do tempo; se a proposta faz uso de recuperação de dados de outros sensores quando ocorrem falhas no sensor atual; se o ambiente experimental utilizado na avaliação da proposta se aproxima de um ambiente real de processamento; e finalmente se dados reais foram usados na avaliação da proposta.

De acordo com a pesquisa bibliográfica realizada, é possível verificar que poucos trabalhos abordam a avaliação de dados, e dentre estes, nenhuma das propostas utiliza treinamento online ou se mantém atualizado ao longo do tempo.

Os poucos trabalhos que consideram a utilização de séries temporais na avaliação da qualidade dos dados gerados por sensores sofrem com complexidades computacionais altas (CABRAL; COMPTON; MÜLLER, 2014, 2014; ELAHI *et al.*, 2009; TRUONG; RÖMER; CHEN, 2012). No método FOCUSeR são mantidas e utilizadas séries temporais geradas por cada sensor durante um determinado período (fatias de tempo) para melhorar a qualidade da avaliação sobre a qualidade dos dados. O uso de algoritmos computacionalmente leves para avaliação dos dados é considerado um requisito imprescindível para o ambiente IoT.

Outro problema das propostas existentes é a alta latência para disponibilização do *ranking* de sensores. Grande parte dos trabalhos relacionados faz a seleção dos sensores após receber os parâmetros do usuário, o que aumenta o tempo entre a consulta e a resposta na busca da lista de sensores. No método proposto, os sensores são monitorados a cada geração de dados e a lista é atualizada automaticamente com as informações sobre cada sensor. Uma consulta baseada nos parâmetros do usuário pode ser feita apenas na lista de melhores ou piores sensores, conforme o caso, o que reduz em muito o tempo de processamento da tarefa, garantindo a baixa latência no processo.

Os trabalhos pesquisados também não têm a preocupação de disponibilizar informações sobre sensores em estado de falha ou que estão indicando anomalias nos

ambientes. A abordagem utilizada no FOCUSeR se propõe a identificar os padrões de normalidade na geração dos dados bem como identificar quando isto não ocorre.

Com relação à arquitetura, alguns trabalhos também não apresentam propostas distribuídas, o que para garantir a escalabilidade, também é um requisito imprescindível em aplicações IoT. O método FOCUSeR considera o uso de *gateways*, nos quais são processadas as informações e avaliados os dados de cada sensor. O gerenciamento das listas de *ranking* distribuídas é feito utilizando a técnica *Distributed Hash Table* (DHT) que possui alto grau de escalabilidade e o suporte flexível para operações de consulta e atualização.

Nenhum dos trabalhos relacionados é capaz de dispensar totalmente o treinamento prévio de parâmetros, assim como não têm recursos para manter esses parâmetros atualizados. Por outro lado, muitos dos trabalhos não utilizam ambientes nem conjuntos de dados reais nos processos de avaliação dos resultados experimentais, fato esse que impede uma avaliação mais precisa dos resultados em ambientes experimentais.

#### 4 FOG ONLINE CONTEXT-AWARE UP-TO-DATE SENSOR RANKING (FOCUSeR)

Este capítulo descreve o método proposto para *ranking* de sensores, denominado FOCUSeR. Inicialmente é apresentada uma visão geral e antes de detalhar cada nível de processamento do método, o modelo de avaliação de valores discrepantes é apresentado. Este modelo é apresentado previamente para facilitar o entendimento dos critérios de avaliação de dados utilizados no método FOCUSeR. Além disso, os conjuntos de dados utilizados na avaliação do método, bem como a plataforma computacional experimental utilizada nos testes, também são descritos.

##### 4.1 RANKING DE SENSORES INSPIRADO NA PERCEPÇÃO ATIVA

O método de *ranking* proposto neste trabalho é inspirado na teoria da percepção ativa. Assim, nesta tese, a tarefa de ranquear os sensores é realizada por meio de uma análise que permite diferentes níveis de raciocínio de acordo com o contexto encontrado. Além disso, caracteriza-se como ativa pois pode buscar alternativas aos problemas encontrados consultando fontes de dados alternativas e confiáveis.

A Figura 8 mostra uma visão geral de todas as etapas envolvidas na tese. Na parte superior da figura, as camadas de computação (*Edge, Fog e Cloud*) são identificadas. Abaixo dessas camadas, são apresentados (da esquerda para a direita): os níveis de aquisição, raciocínio e monitoramento; as funções padrão (texto em negrito), comumente encontradas em *middlewares* para *ranking* de sensores; e as ferramentas utilizadas para realizar cada função no ambiente experimental desenvolvido para avaliar o método proposto FOCUSeR.

Figura 8. FOCUSeR - Visão geral.



Fonte: Elaborada pelo autor (2022).

As caixas na parte inferior da figura apresentam os níveis do processo de *ranking*, detalhando o fluxo da informação no método proposto FOCUSeR. Esta parte da figura apresenta os níveis da percepção ativa (caixas com a cor cinza claro no fundo), iniciando pela atividade de sensação, ou seja, a coleta de dados no ambiente, e seguindo a direção da seta vermelha, percepção, percepção ao longo do tempo e percepção ativa. Nesse fluxo também são apresentadas as atividades de *burn-in* e armazenamento (caixas com fundo cinza mais escuro).

O primeiro nível de percepção ativa, sensação, é realizado pelos sensores e representa a captura dos dados produzidos por eles. Na atividade de *burn-in* são criados os indicadores utilizados pela métrica IQR. Após o *burn-in*, aparecem os níveis da percepção ativa.

O fluxo termina com a atividade de persistência de dados, usada para ilustrar exemplos de monitoramento. A seta vermelha na imagem informa que o fluxo retorna ao ambiente, indicando a execução das ações a serem tomadas definidas no nível de percepção ativa. Nas subseções a seguir, as implementações de cada nível serão detalhadas.

#### 4.2 MODELO DE VALORES DISCREPANTES/FALHAS

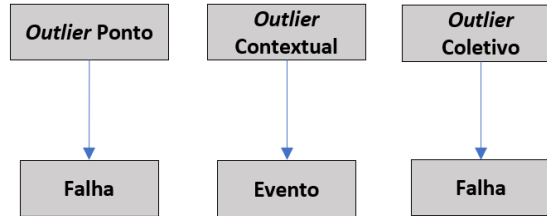
Falhas na captura e transmissão de dados, obtidos de sensores, normalmente ocorrem em uma captura e normalizam na próxima (*outlier* pontual). A razão é que os distúrbios relacionados a esses problemas são em sua maioria transitórios, causados por desequilíbrios que levam a instabilidades momentâneas (BEVRANI, 2009). Assim, quando ocorre interferência em uma única medição, gerando um único valor incorreto, há uma grande probabilidade de que esses dados sejam mal interpretados. Isso pode ser evitado analisando fatias de tempo de dados (*subsequências*).

Nesta tese, os valores são primeiramente analisados individualmente e, em seguida, avaliados por meio de comparações entre *subsequências* de dados. Os valores mais distantes dos valores frequentes (*outliers*) são considerados como falhas e o restante dos dados são considerados como eventos extremos no ambiente. No caso de um *outlier* pontual, a avaliação é simples. Ao considerar *subsequências* de dados, se a *subsequência* for considerada *outlier* em relação as demais *subsequências* já vistas, ela pode ser considerada como uma falha (*outlier* coletivo) ou como um evento (*outlier* contextual). O critério adotado para identificar



o tipo de subsequência se baseia na intensidade da diferença da *subsequência* em relação as demais. Esse modelo é apresentado na Figura 9.

Figura 9. Modelo de valores discrepantes/falhas.



Fonte: Elaborada pelo autor (2022).

Quando ocorrem valores anormais, o método proposto FOCUSeR também entrega, juntamente com as demais variáveis de saída, uma classificação para o tipo de *outlier*, com base nas informações apresentadas no modelo de falha/*outlier* da Figura 9 e no esquema apresentado no Quadro 2.

Quadro 2. Saída produzida de acordo com modelo valores discrepantes/falhas.

		Primeiro Nível		
		Normal	Anomalia	Falha
Segundo Nível	Rótulo de Saída			
	Normal	Normal	<i>Outlier</i> Pontual	<i>Outlier</i> Pontual
	Anomalia	<i>Outlier</i> Contextual	<i>Outlier</i> Contextual	<i>Outlier</i> Contextual
Falha	<i>Outlier</i> Coletivo	<i>Outlier</i> Coletivo	<i>Outlier</i> Coletivo	

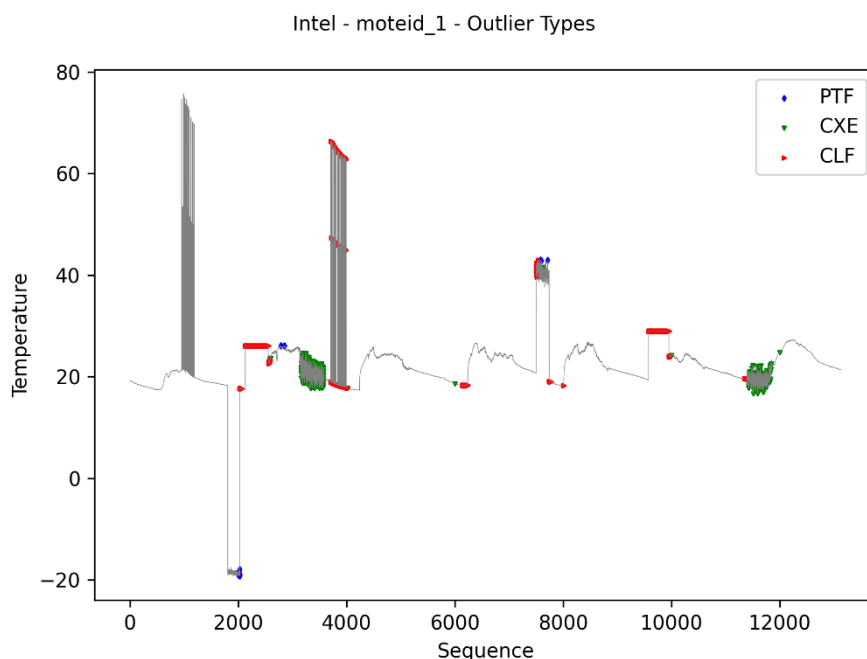
Fonte: Elaborada pelo autor (2022).

Os critérios adotados para a decisão consideram que falhas identificadas no primeiro nível, ou seja, na análise individual (pontual) da amostra, têm alta probabilidade de representarem falhas transitórias no sensor. Assim, quando apenas o primeiro nível falha, o erro é considerado como um *outlier* pontual. Se apenas no segundo nível, ou seja, na análise de uma subsequência de dados, for identificado um *outlier*, os dados são considerados como um evento, ou seja, um *outlier* contextual, e não como uma falha. Entretanto, se uma falha for identificada também no primeiro nível, os dados serão considerados como uma falha, e nesse caso, como um *outlier* coletivo.

A Figura 10 apresenta um exemplo da classificação realizada pelo método proposto FOCUSeR. A legenda da imagem apresenta os três tipos de *outliers* encontrados nos dados,

ou seja, *outlier* pontual (PTF), *outlier* contextual (CXE) e *outlier* coletivo (CLF). O eixo horizontal do gráfico apresenta a ordenação dos dados no tempo (sequência de geração) e o eixo vertical apresenta o valor da temperatura lido pelo sensor. É possível verificar na imagem que nenhum dado foi considerado como anormal pelo método antes da sequência 2000, justamente porque esse foi o valor limite definido para a etapa de *burn-in* (nesta etapa, como será visto a seguir, o método apenas coleta amostras para gerar os parâmetros a serem utilizados na métrica de avaliação de dados).

Figura 10. Exemplo de saída produzida de acordo com o modelo de *outliers*/falhas.



Fonte: Elaborada pelo autor (2022).

No método proposto FOCUSER, os dados gerados pelos sensores são classificados em normal, anomalia (evento) e falha. Os dados considerados normais são aqueles que indicam que tanto o sensor quanto o ambiente em que a medição está sendo feita estão estáveis (sem anomalias) e o processo de leitura não foi afetado por interferências durante a captura e transmissão dos dados. Os dados neste intervalo de valores representam os dados gerados com mais frequência pelo sensor.

Quando uma medição afetada por uma falha é analisada como uma série temporal (fatia), a *subsequência* correspondente a essa falha será diferente da tendência normal de toda série temporal. No entanto, isso só será verdade se o desvio da normalidade for grande o suficiente para influenciar a medição de todo o segmento, o que ocorre em falhas persistentes. No caso de falhas transitórias, um único ponto de dados anormal, a influência na medição em

série é menos significativa. Assim, quando analisado em um segmento de dados, o distúrbio causado é mitigado devido à presença de vários outros valores normais.

Nesta segunda avaliação, os dados também podem ser considerados como uma anomalia, indicando a ocorrência de eventos extremos no ambiente. Isso ocorre quando vários dos valores no segmento de dados estão relativamente distantes do intervalo normal, mas dentro de um intervalo de valores possíveis para o domínio do ambiente (*outlier* contextual). Esse tipo de dado não acontece com frequência.

Por fim, os subconjuntos de dados considerados como falhas são aqueles que ocorrem devido a falhas persistentes nos dispositivos, afetando a captura ou transmissão de dados (*outlier* coletivo). Nesse caso, a faixa de valores está mais distante dos demais valores, extrapolando os limites de valores válidos do domínio.

A métrica IQR é aplicada para obter os indicadores a serem utilizados na avaliação dos dados. No caso de avaliar amostras de dados individuais, o IQR é aplicado aos dados produzidos pelo sensor. No caso de avaliação de *subsequências* de dados, o IQR é aplicado ao *perfil de distância* gerado pelo algoritmo de *perfil de matriz*.

$$(Q1 - FF * IQR) < fi > (Q3 + FF * IQR) \quad (9)$$

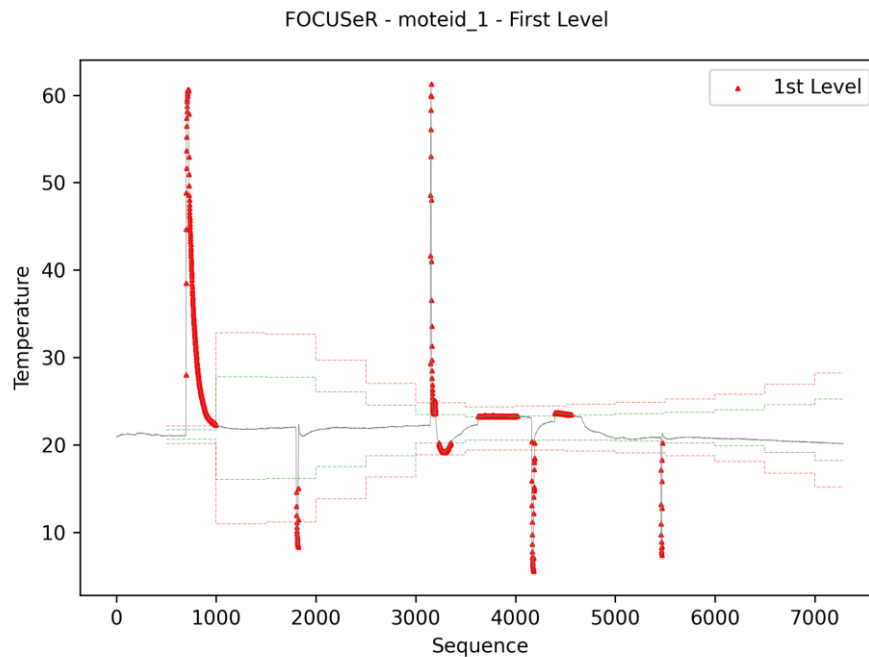
$$(Q1 - FF * IQR) < ai > (Q1 - AF * IQR) \text{ or}$$

$$(Q3 + AF * IQR) < ai > (Q3 + FF * IQR) \quad (10)$$

As Equações 9 e 10 usam o *índice de anomalia* (*ai*) e o *índice de falha* (*fi*) para rotular as amostras. A Equação 9 é utilizada para verificar se um valor é considerado uma falha e a Equação 10 para valores considerados anomalias. O *fator de anomalia* (*AF*) é definido com um valor 3 (três) e o *fator de falha* (*FF*) definido como  $AF * 2$ . Normalmente, ao calcular o corte de *outliers* usando a métrica IQR, o valor para o fator (anomalia e falha) geralmente adotado é igual a 1,5. Entretanto, estudos preliminares realizados no desenvolvimento do método proposto, esse coeficiente mostrou-se ineficiente como limiar para definir anomalias e falhas relacionadas aos dados de sensores.

A Figura 11 mostra um exemplo de aplicação desta métrica no primeiro nível de avaliação do algoritmo proposto: as amostras localizadas entre as linhas verdes representam os dados considerados normais; as amostras entre as linhas verde e vermelha representam dados anômalos (eventos) e os pontos acima da linha vermelha representam dados de falha.

Figura 11. Divisão de dados em normais, anomalia e falha usando IQR.



Fonte: Elaborada pelo autor (2022).

#### 4.3 BURN-IN

Devido à heterogeneidade dos ambientes monitorados por sensores, se torna inviável realizar o ajuste parâmetros (valores limites para anomalia e falha) de forma manual para cada sensor. Como forma de superar esta limitação, o método FOCUSeR utiliza a abordagem de aprendizagem online de parâmetros. Na etapa de *burn-in*, as amostras inicialmente obtidas de cada sensor são usadas para gerar os índices de anomalia e falha apresentados nas Equações 9 e 10. O número de amostras utilizado para gerar os índices é um parâmetro de entrada do algoritmo do método FOCUSeR. Nos testes realizados foram utilizadas as primeiras 2000 (duas mil) amostras, com exceção dos conjuntos de dados *FOCUSeR* e *SmartSantander* em que foram utilizadas as primeiras 500 (quinhentas) devido ao baixo número de amostras de cada sensor. Estes valores foram escolhidos após uma análise de sensibilidade de parâmetros, conforme será descrito na seção de avaliação de resultados.

A atualização do índice de anomalia e falha (Equações 9 e 10), de acordo com o valor definido no parâmetro *burn-in*, mostrou ser eficaz, o que se verificou através de um aumento nas taxas de Acurácia, conforme será apresentado.

A estrutura de dados *t-digest* foi utilizada para permitir a atualização desses índices sem a necessidade de manter na memória todo o conjunto de dados já visto.

#### 4.4 PRIMEIRO NÍVEL DE AVALIAÇÃO: PERCEPÇÃO

No primeiro nível de processamento do método FOCUCeR, ou seja, percepção, os índices de anomalia e falha (Equações 9 e 10) são aplicados às amostras que chegam ao sistema. Como resultado dessa avaliação, a amostra é classificada em normal, anomalia ou falha.

#### 4.5 SEGUNDO NÍVEL DE AVALIAÇÃO: PERCEPÇÃO AO LONGO DO TEMPO

Este nível utiliza o tempo para avaliar os dados. Para implementar este conceito, foi escolhido o algoritmo *matrix profile*. O algoritmo é executado inicialmente com os dados de amostra salvos (*batch*), coletados durante a etapa de *burn-in*. Uma vez gerada a estrutura *distance profile*, a métrica IQR é aplicada para gerar os índices de anomalia e falha dos segmentos. Após a etapa de *burn-in*, a estrutura *matrix profile* é atualizada a cada novo dado e esses dados são avaliados por meio dos índices gerados, a fim de identificar as *subsequências* que divergem das demais *subsequências* do conjunto de dados. Da mesma forma que no nível anterior, as *subsequências* também são classificadas como normal, anomalia e falha.

#### 4.6 TERCEIRO NÍVEL DE AVALIAÇÃO: PERCEPÇÃO ATIVA

No nível final, ou seja, o nível de percepção ativa, utilizando os rótulos obtidos nos dois primeiros níveis, identifica-se o tipo de outlier de acordo com os valores apresentados no Quadro 2. O índice do sensor também é atualizado com as novas informações. Para atualizar o *ranking*, é utilizado o parâmetro *trustworthiness* da QoC.

Quando uma falha é identificada nos dados do sensor, a função de percepção ativa é ativada para verificar se existe um sensor próximo o suficiente para fornecer a medição do ambiente que possa substituir a medição fornecida pelo sensor em estado de falha. Para esta tarefa, os parâmetros *Timeliness* e *Age* da QoC são utilizados. Além de verificar o quão atual é o dado capturado, o valor do parâmetro de confiabilidade do sensor que está fornecendo as informações é testado e aceito se estiver acima de 0,9. Esse valor é parametrizável por conjunto de dados e sua configuração será definida de acordo com restrições de cada domínio.

---

**ALGORITHM 1:** Fog Online Context-aware Up-to-date Sensor Ranking (FOCUSeR)

---

**Input:** *parameters* (*p*trustworthiness=0.9, *p*timeliness=0.5, *p*time\_period, *p*burn\_in=3000, *p*window\_size=30, *p*iqr\_refresh=3000), *sensor id* (*id*), *sensor value* (*val*), *sensor\_latitude*, *sensor\_longitude*, *sensor\_dt\_measurement*

**Output:** *trustworthiness* (*tt*), *fault type* (*ft*), *label* (*lba*), *from sensor id* (*fsid*), *from sensor value* (*fsv*)

---

**focuser** (*par*, *id*, *val*, *lat*, *lon*, *dt*): *tt*, *ft*, *lb*, *fsid*, *vf*

*s1* := [ ]

*ct* := 0

**repeat**

*ct* := *ct* + 1

*s1*[*ct*] = *val*

**if** *par*.*burnin*

**if** *par*.*burnin\_size* < *ct* **then**

*build\_iqr*()

*par*.*burnin* := *false*

*ct* := 0

**end if**

**else**

**if** *par*.*iqr\_refresh* < *ct* **then**

*build\_iqr*()

*ct* := 0

**end if**

**end if**

*lbp* := *perception*(*val*, *iqr*)

*lbt* := *time\_perception*(*s1*, *lbp*)

*lba*, *ft*, *tt*, *fsid*, *fsv* := *active\_perception*(*lbp*, *lbt*)

*update\_rank*(*par*, *id*, *val*, *lat*, *lon*, *dt*)

**end repeat**

---

O algoritmo também alimenta informações sobre a última amostra analisada nas tabelas de *hash* distribuídas, como a classificação dos dados em normal, anomalia e falha e uma segunda categoria identificando a amostra como normal, *outlier* pontual (falha), *outlier* contextual (anomalia) ou *outlier* coletivo (falha). O Algoritmo 1 apresenta todo o processo.

#### 4.7 COMPLEXIDADE DE TEMPO

Como forma de não impactar a latência, é importante que o algoritmo tenha baixa complexidade computacional de tempo. Na equação de complexidade de tempo, foi utilizada a seguinte notação:

*n*: o número de amostras para classificação

*C1*: custo para calcular IQR para todos os níveis

*C2*: custo para calcular estruturas do algoritmo *matrix profile*

*C3*: custo para atualizar as estruturas do algoritmo *matrix profile*

*C4*: custo para testes em todos os níveis

*C5*: custo para localizar amostras de outros sensores

O custo total do algoritmo (Equação 11) é o resultado da soma dos custos da etapa de *burn-in* e da operação normal do algoritmo. No caso da etapa de *burn-in*, o custo é representado pelo cálculo dos indicadores IQR ( $C1$ ) e pela execução inicial (*batch*) do algoritmo *matrix profile* ( $C2$ ). Na execução, após a etapa de *burn-in* do algoritmo, o custo é representado pela soma dos custos dos testes de cada nível ( $C4$ ) com a soma do custo de atualização das estruturas do algoritmo *matrix profile* ( $C3$ ) e quando ocorre uma anormalidade, pelo custo de busca, nas tabelas de *hash* distribuído, de uma amostra de sensor em estado normal ( $C5$ ).

$$C_{total} = C1 + C2 + C3 + C4 + C5 \quad (11)$$

O cálculo do IQR ( $C1$ ) dos dados no valor de quartil fornecido tem complexidade de tempo  $O(n \log n)$ . A complexidade geral do algoritmo *matrix profile* ( $C2$ ) é  $O(n^2 \log n)$ . A complexidade de tempo do algoritmo STAMPI ( $C3$ ) é  $O(n \log n)$ . O custo dos testes em cada nível é igual a  $O(n)$ . E o custo para procurar amostras de outros sensores é  $O(\log n)$ . Assim, a complexidade de tempo total do algoritmo é  $O(n^2 \log n)$ . No entanto, considerando que a etapa de *burn-in* (custos  $C1$  e  $C2$ ) ocorre apenas no momento inicial da execução do algoritmo, pode-se dizer que o custo computacional do algoritmo é  $O(n \log n)$ .

#### 4.8 CONSIDERAÇÕES

O método FOCUSeR apresentado neste capítulo procura fornecer uma solução viável, respeitando as características presentes em ambientes como as WSNs, de forma segura e eficiente. O método é capaz de criar e gerenciar listas de classificação de sensores, previamente criadas, de forma distribuída, para que as informações estejam disponíveis para serviços e aplicações no menor tempo possível.

O método proposto também mantém atualizados os parâmetros obtidos no treinamento online, para evitar que fiquem desatualizados devido a alterações nos dados ao longo do tempo. Para prover confiabilidade às respostas, o método utiliza múltiplas técnicas de avaliação de dados e dados de contexto visando reduzir a incerteza contida nos dados e ainda assim atendendo aos requisitos de baixa latência.



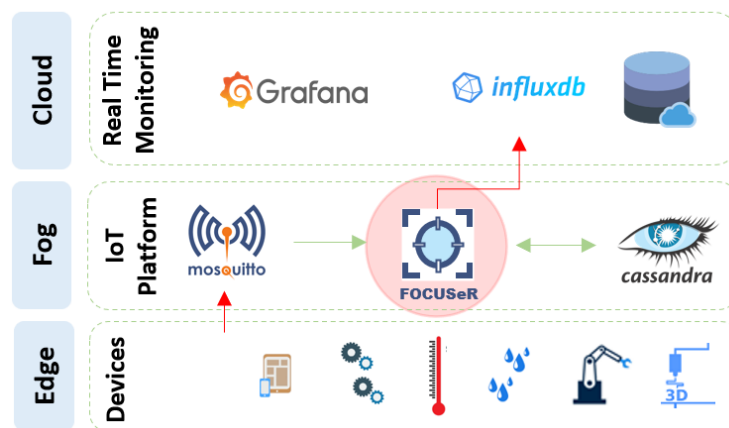


## 5 AMBIENTE EXPERIMENTAL E RESULTADOS

### 5.1 AMBIENTE EXPERIMENTAL

Dada a natureza distribuída dos dispositivos IoT, é necessário adotar uma arquitetura distribuída capaz de garantir a escalabilidade do método FOCUSeR. O crescimento exponencial do número de sensores no ambiente IoT justifica a necessidade deste tipo de abordagem. Assim, o método proposto utiliza uma abordagem distribuída em que o processamento ocorre na camada de *Fog Computing*. Este trabalho considera o uso de *gateways*, nos quais os dados de cada sensor são processados e avaliados. Os *gateways* trocam informações usando um modelo de gerenciamento de listas distribuídas disponível no Apache Cassandra.

Figura 12. FOCUSeR: Ambiente experimental.



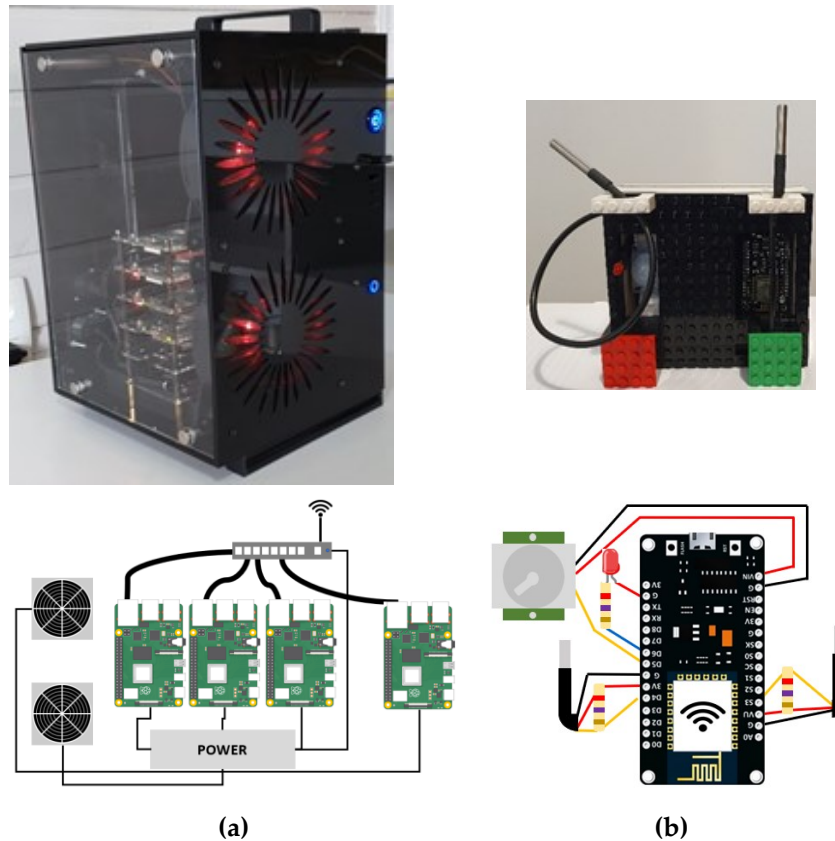
Fonte: Elaborada pelo autor (2022).

A Figura 12 apresenta a arquitetura experimental desenvolvida. A caixa inferior na Figura 12 representa os sensores (dispositivos) na camada de *Edge Computing*. Os dados desses sensores são enviados para a plataforma usando o protocolo *Message Queuing Telemetry Transport* (MQTT). Para desempenhar essa função, foi escolhida a ferramenta Mosquitto (LIGHT, 2017). Um conector monitora continuamente tópicos (como uma fila de mensagens) no MQTT e, assim que uma mensagem chega, ela é automaticamente transferida para o FOCUSeR, na camada de *Fog Computing*.

Após o processamento pelo FOCUSeR, as informações sobre os sensores são mantidas na ferramenta Apache Cassandra. Para visualizar os resultados do processamento, os

dados de saída do algoritmo são persistidos no banco de dados InfluxDB, que é usado como fonte de dados pela ferramenta Grafana, ambos executando na camada de *Cloud Computing*. Para os testes, foi utilizada uma rede com três nodos. Desta forma, esta arquitetura permite o processamento paralelo, reforçando também a escalabilidade do método FOCUSeR.

Figura 13. Ambiente experimental - *Raspberrys PI* (a) e sensores (b).



Fonte: Elaborada pelo autor (2022).

Para avaliar o método proposto, foi desenvolvido um ambiente experimental composto por quatro nodos de processamento e dois sensores (Figura 13). Como *hardware* para cada nodo, foram utilizadas placas *Raspberry PI 4* com 4 GB de RAM cada (Figura 13a). Três dos quatro nodos foram usados para executar o FOCUSeR e a ferramenta Apache Cassandra. O outro nodo, uma placa *Raspberry PI 2*, executou o MQTT *Broker*.

Foram utilizados dois sensores de temperatura DS18B20, um sensor infravermelho passivo e um *led*, todos conectados a um módulo ESP8266 12E (Figura 13b). Foram utilizados dois sensores visando fornecer medidas de comparação entre os dados produzidos pelos sensores. O sensor normal, identificado pela cor verde, não é manipulado e assim apenas captura os dados de temperatura do ambiente. O sensor identificado pela cor vermelha é manipulado em determinados momentos (inserido em copos com água quente ou fria), de

forma a gerar alterações bruscas no conjunto de dados. No caso do sensor infravermelho o intuito foi o de auxiliar a identificação dos momentos em que o sensor vermelho foi manipulado, já que os dados de presença também foram registrados no arquivo do conjunto de dados. As ferramentas Grafana e InfluxDB (*Cloud Computing*) foram executadas em um computador com processador Intel Core i7 com 8 GB de RAM e um SSD de 500 GB.

## 5.2 CONJUNTOS DE DADOS

Cinco conjuntos de dados reais foram selecionados para avaliar o método proposto, conforme mostrado na Tabela 1<sup>2</sup>. O conjunto de dados FOCUSeR foi criado neste trabalho de pesquisa. Dois sensores de temperatura DS18B20 foram conectados a uma placa ESP8266 Figura 13b. As capturas foram feitas a cada dois (2) segundos. Para criar dados anormais no conjunto, em momentos aleatórios, um dos sensores (identificado na imagem pelos blocos vermelhos) foi mergulhado alternadamente em um copo com água quente ou fria. Os momentos de cada mudança foram gerados aleatoriamente, assim como a duração de cada evento e o tipo de mudança (frio ou quente). O outro sensor não foi manipulado, permanecendo com as leituras normais da temperatura do ambiente.

Tabela 1. Conjuntos de dados.

Conjunto de dados	Amostras (milhões)	Estações
Dados FOCUSeR	0,014	2
Intel	2.300,000	54
NUMENTA	0,022	1
SensorScope	0,418	23
SmartSantander	0,044	16

Fonte: Elaborada pelo autor (2022).

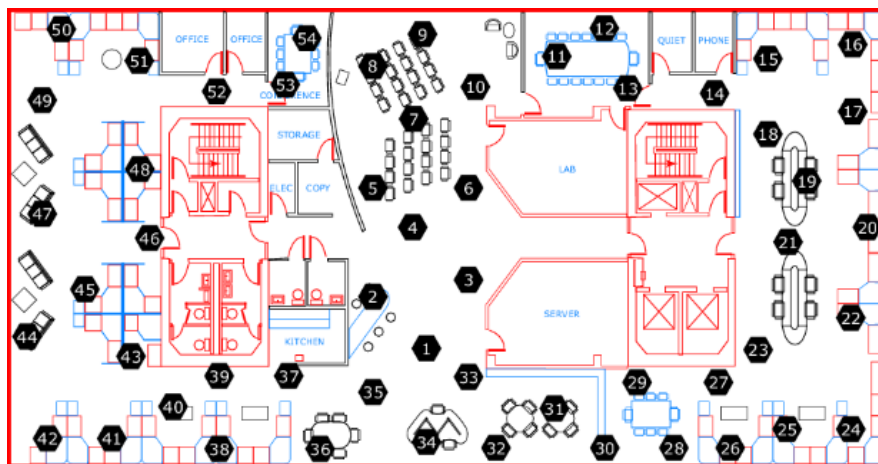
Como forma de auxiliar o processo de criação de rótulos para os dados, junto ao sensor manipulado, foi instalado um sensor infravermelho para detectar movimento. Aproximadamente 7.000 (sete mil) amostras foram geradas a partir de cada um dos sensores.

O conjunto de dados *Intel Lab* (INTEL LAB DATA, [s. d.]) contém informações sobre dados de 54 sensores com placas meteorológicas que coletaram informações de umidade, temperatura, luz e tensão a cada 31 segundos. Os sensores, implantados no *Intel Berkeley Research Lab* foram dispostos no laboratório conforme a Figura 14.

<sup>2</sup> Disponíveis em [https://github.com/felipekosta/FOCUSeR\\_datasets](https://github.com/felipekosta/FOCUSeR_datasets)

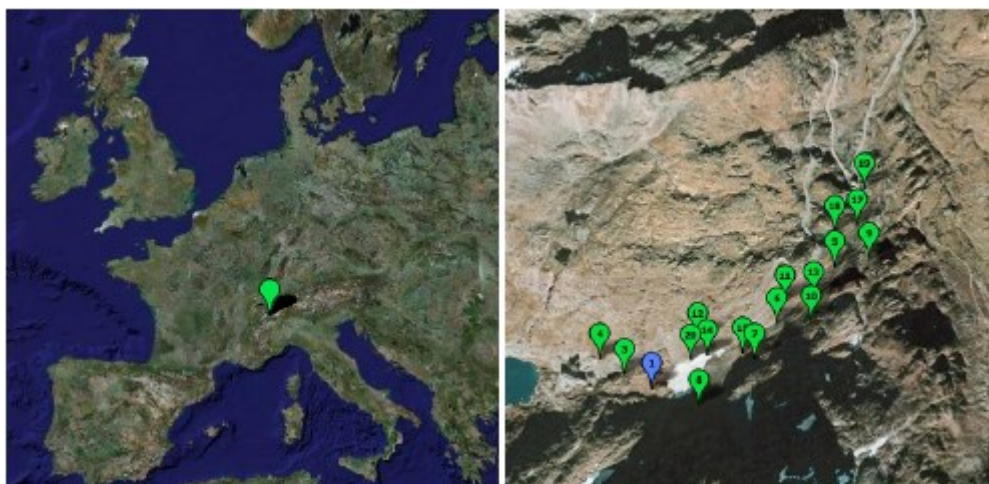
A coleção *Numenta Anomaly Benchmark* (NAB) é constituída de conjuntos de dados do mundo real sobre domínios diversos. O conjunto de dados "*machine\_temperature\_system\_failure*" contém dados do sensor de temperatura de um componente interno de uma máquina industrial. O arquivo apresenta três anomalias: a primeira refere-se a uma parada planejada do equipamento monitorado; a segunda, indica o início do mal funcionamento do equipamento causando a terceira e última falha, uma parada total do equipamento. (NUMENTA ANOMALY BENCHMARK (NAB), 2021).

Figura 14. Conjunto de dados Intel Lab.



Fonte: (INTEL LAB DATA, [s. d.])

Figura 15. SensorScope: mapa implantação no *Grand Saint-Bernard*.



Fonte: (INGELREST *et al.*, 2010).

O conjunto de dados *SmartSantander* consiste em medições de temperatura de 16 sensores externos que fazem parte do projeto *SmartSantander* (SMARTSANTANDER, [s. d.]). O último conjunto de dados utilizado possui dados de temperatura externa coletados de

23 estações do projeto *SensorScope* (INGELREST *et al.*, 2010). A Figura 15 apresenta a disposição dos sensores no continente europeu (esquerda) e na região monitorada (direita).

A combinação dos conjuntos, nos quais os dados foram gerados em diferentes condições, contribuiu para a avaliação do método FOCUSeR. Além disso, os dados referentes aos conjuntos *Intel Lab*, *SmartSantander* e *SensorScope* foram recuperados de Bruijn (2016). O trabalho descrito no artigo fornece uma estrutura para a preparação de conjuntos de dados anotados com falhas injetadas artificialmente, adequadas para avaliar métodos de detecção de falhas. A motivação para a utilização desses arquivos é permitir uma avaliação do método de tratamento de anomalias e falhas.

### 5.3 TESTE DE SENSIBILIDADE DE PARÂMETROS

A Tabela 2 apresenta os resultados da análise de sensibilidade dos parâmetros de *burn-in* e tamanho da *subsequência* (tamanho da janela). O parâmetro *IQR Refresh* herda o valor do parâmetro *burn-in*, ou seja, a atualização dos indicadores da métrica ocorre em intervalos iguais ao valor definido no parâmetro *burn-in*.

Tabela 2. Análise de sensibilidade de parâmetros.

Conjunto de dados	<i>Burn-in</i>	Subsequência	Acurácia	Precisão	Recall	F1 Score
<b>Intel</b>	<b>5000</b>	<b>30</b>	<b>94.59</b>	96.16	97.68	<b>96.91</b>
	3000	30	94.19	<b>98.56</b>	94.51	96.49
	4000	270	89.47	89.35	<b>99.99</b>	94.37
	2000	30	91.49	95.03	94.55	94.79
<b>SmartSantander</b>	<b>2000</b>	<b>30</b>	<b>94.83</b>	97.06	<b>96.80</b>	<b>96.93</b>
	1000	30	92.47	<b>98.10</b>	92.72	95.33
	3000	30	91.19	97.14	91.31	94.14
	1000	150	82.68	90.74	88.10	89.40
<b>SensorScope</b>	<b>5000</b>	<b>30</b>	<b>95.73</b>	<b>99.00</b>	95.86	<b>97.40</b>
	4000	30	95.50	98.29	96.17	97.22
	3000	30	94.74	97.25	96.34	96.79
	4000	270	85.22	84.72	<b>99.98</b>	91.72

Fonte: Elaborada pelo autor (2022).

Essa abordagem foi adotada porque a taxa de atualização de indicadores depende do tipo de ambiente monitorado e assim pode ser customizada no momento da inclusão do sensor na lista de sensores controlados pelo método. Os testes foram realizados com os valores 1000, 2000, 3000, 4000 e 5000 para o parâmetro *burn-in* e com os valores 30, 90, 150, 210 e 270 para o parâmetro *subsequência*, ou seja, 25 combinações diferentes para cada conjunto de dados. A Tabela 2 mostra apenas quatro resultados de cada conjunto. Os melhores valores

para cada métrica estão destacados em negrito. Os resultados completos podem ser vistos no Apêndice C.

Embora nos conjuntos de dados Intel e SensorScope os valores com tamanho de *burn-in* igual a 5000 tenham apresentados os melhores resultados, optou-se por utilizar o valor 3000 como padrão. Essa escolha se justifica porque valores de *burn-in* mais altos “gastam” mais amostras na operação do método para iniciar efetivamente o processamento. Além disso, eles são apenas levemente melhores do que aqueles com valores mais baixos nesses parâmetros. No caso do conjunto de dados FOCUSeR e *SmartSantander*, o valor 500 foi escolhido pelo fato de os arquivos possuírem poucas amostras (3000 amostras por arquivo, em média no caso do conjunto *SmartSantander* e 7000 no caso do conjunto FOCUSeR).

No caso do parâmetro tamanho da *subsequência*, o melhor resultado é obtido quando o parâmetro é definido com o valor 30.

#### 5.4 AVALIAÇÃO DE ACURÁCIA

Para verificar a eficácia do método proposto FOCUSeR, foram considerados apenas os conjuntos de dados com rótulos, ou seja, aqueles rotulados em Bruijn (2016). Por estar presente em todos os conjuntos de dados escolhidos, a variável temperatura foi selecionada para avaliação.

Como pode ser visto na Tabela 3, os quatro conjuntos de dados submetidos ao método FOCUSeR, obtiveram Acurácia em torno de 95% para todos os conjuntos e aproximadamente 97% para as demais métricas.

Tabela 3. Eficácia do modelo proposto.

<b>Conjunto de dados</b>	<b>Acurácia (%)</b>	<b>Precisão (%)</b>	<b>Recall (%)</b>	<b>F1 (%)</b>
Dados FOCUSeR	98,4	98,1	100,0	99,0
Intel	94,6	98,5	100,0	96,9
SmartSantander	94,8	98,1	96,8	96,9
SensorScope	95,7	99,0	99,9	97,4

Fonte: Elaborada pelo autor (2022).

Como forma de avaliar o método proposto FOCUSeR em comparação com outros trabalhos, a Tabela 4 apresenta os valores obtidos nas métricas Acurácia, Precisão, *Recall* e *F1 Score*, respectivamente. Cabe ressaltar que essa avaliação serve apenas como base de comparação, pois as abordagens e propósitos normalmente são diferentes em relação a

aspectos como a aprendizagem online de parâmetros, ambiente de execução e versão dos conjuntos de dados utilizados.

Tabela 4. Comparação de Avaliação de Eficácia.

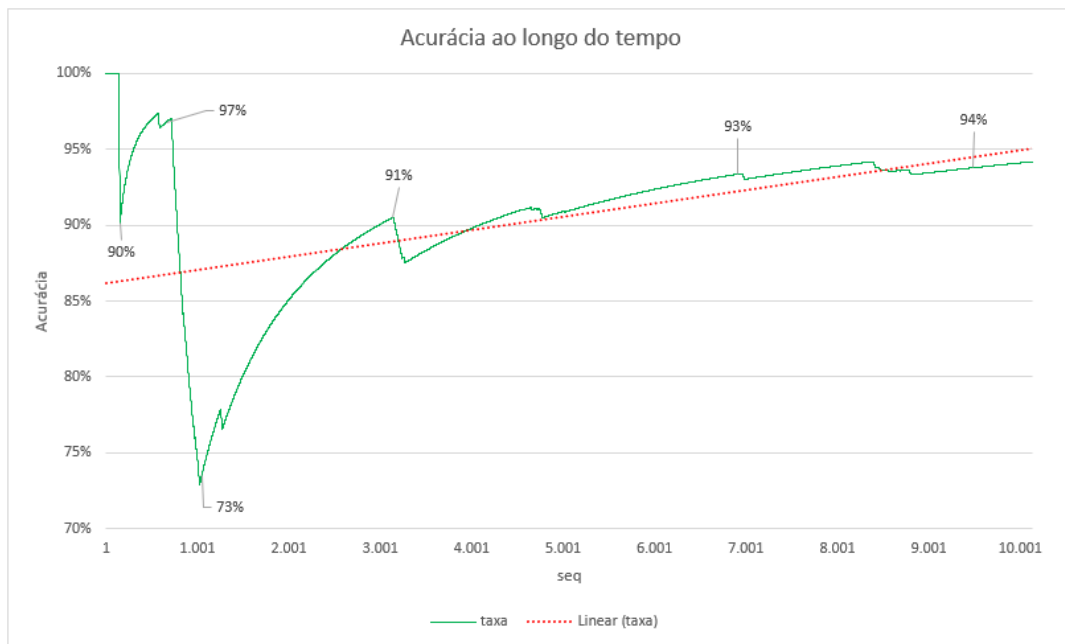
Trabalho	Conjunto de Dados	Acurácia (%)	Precisão (%)	Recall (%)	F1 (%)
<b>Método FOCUSeR</b>	Intel	94,59	<b>98,56</b>	<b>100,0</b>	<b>96,91</b>
	SensorScope	95,73	<b>99,00</b>	<b>99,99</b>	<b>97,40</b>
(ZAMRY; ZAINAL; RASSAM, 2018)	Intel	98,40	-	-	-
	SensorScope	-	-	-	-
(RASSAM; MAAROF; ZAINAL, 2014)	Intel	98,50	-	-	-
	SensorScope	<b>99,10</b>	-	-	-
(ABID; KACHOURI; MAHFOUDHI, 2016)	Intel	95,00	-	100,0	-
	SensorScope	-	-	-	-
(BOSMAN <i>et al.</i> , 2015) <sup>1</sup>	Intel	-	53,32	94,61	68,20
	SensorScope	-	76,04	92,97	83,66
(BOSMAN <i>et al.</i> , 2017) <sup>2</sup>	Intel	-	49,23	95,30	64,92
	SensorScope	-	-	-	-
(KUO; LI; KIFER, 2018)	Intel	-	94,00	89,00	91,43
	SensorScope	-	-	-	-
(ABID <i>et al.</i> , 2017)	Intel	-	83,00	94,00	88,16
	SensorScope	-	-	-	-
(VAMSI; CHAHUAN, 2019)	Intel	-	94,86	-	-
	SensorScope	-	-	-	-
(AL-SHABI; ABUHAMDAH, 2022) <sup>3</sup>	Intel	<b>99,86</b>	-	-	51,94
	SensorScope	-	-	-	-

Fonte: Elaborada pelo autor (2022).

Na Tabela 4 foram considerados apenas os conjuntos de dados Intel e SensorScope, pois são os únicos utilizados nos outros trabalhos. Nos casos em que os conjuntos de dados ou métrica não foram usados no trabalho, um traço foi colocado na célula. Como pode ser visto na tabela, em três das quatro métricas utilizadas, o método FOCUSeR apresenta o melhor desempenho (valores destacados em negrito) em pelo menos um dos dois conjuntos de dados considerados. No caso da Acurácia, onde o método FOCUSeR não obteve o melhor resultado, os valores estão próximos dos valores mais altos.

Além disso, como FOCUSeR usa uma abordagem para manter seu modelo atualizado ao longo do tempo, à medida que o número de amostras aumenta, os parâmetros da métrica IQR são atualizados. Como resultado dessa abordagem, ao analisar o crescimento da taxa de Acurácia ao longo do tempo, que apesar de apresentar uma variação maior nas primeiras 3000 mil amostras, nas quais os indicadores da métrica IQR ainda não estão tão ajustados aos dados, é possível verificar que há uma tendência de crescimento dessa taxa (Figura 16) indicando uma melhora nos resultados ao longo do tempo.

Figura 16. Acurácia ao longo do tempo.



Fonte: Elaborada pelo autor (2022).

A variação mais acentuada nas primeiras amostras ocorre devido a presença de erros de classificação do método (região próxima à sequência 2000 na Figura 17) em um momento no qual ainda existem poucas amostras, e dessa forma, a presença de erros de classificação do método impacta de forma mais intensa no crescimento da taxa de erros.

## 5.5 AVALIAÇÃO DE RECOMPOSIÇÃO DE DADOS

Ao usar a abordagem de *ranking* ativo, o algoritmo procura sensores próximos com valores válidos (normais) se o valor lido do sensor atual for considerado anormal. O cenário deste teste utilizou o processamento de dados em múltiplos nodos (*raspberrys*). Cada nodo escutando um sensor específico. Dessa forma, na busca por sensores vizinhos, os dados são selecionados se atendem aos requisitos de localização geográfica e idade da mensagem, ou seja, os parâmetros QoC.

A Figura 17 apresenta um cenário de recuperação de dados de falhas. Este gráfico foi gerado usando o arquivo de dados do sensor 1 do conjunto Intel, disponível no trabalho de Bruijn (2016). Três conjuntos de dados foram usados para construir o gráfico: dados sem erros (cinza), ou seja, dados originais do conjunto de dados Intel, dados com erros injetados (vermelho) e dados recuperados pelo FOCUSeR (violeta). Conforme mostra o gráfico, a



maioria das amostras identificadas como anomalias ou falhas pode ser recuperada de sensores próximos, mais precisamente 87,31% do total de amostras anormais.

Figura 17. Recomposição de dados de falhas



Fonte: Elaborada pelo autor (2022).

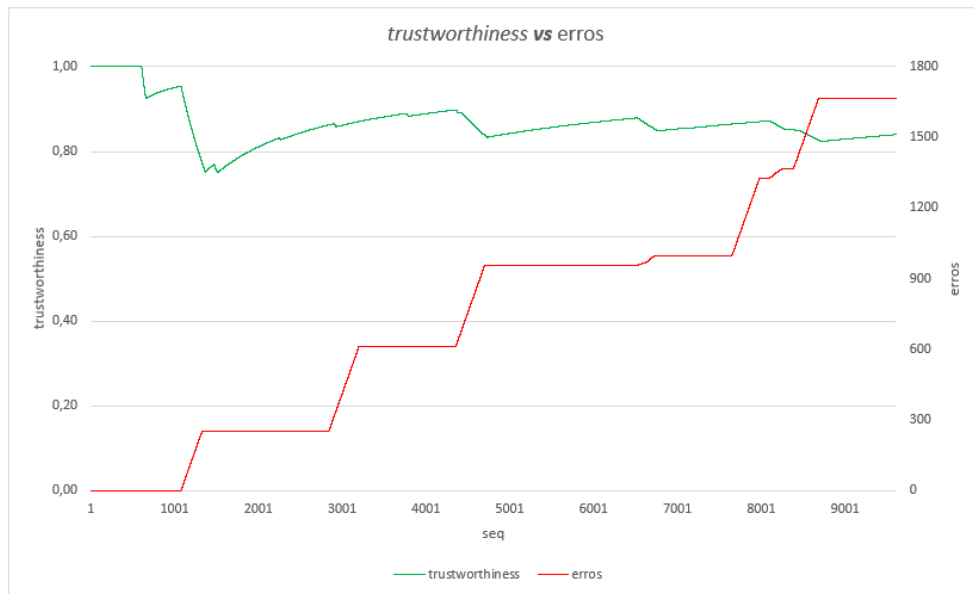
## 5.6 RESULTADOS DA ATUALIZAÇÃO ONLINE E AO LONGO DO TEMPO

A Figura 18 apresenta as alterações dos valores do parâmetro *trustworthiness* e da quantidade de erros (rótulo do conjunto de dados) de um sensor (sensor 4 do conjunto Intel). Na imagem é possível perceber que o valor do parâmetro *trustworthiness*, utilizado como base para cálculo do *ranking* dos sensores, diminui conforme o número de erros do sensor aumenta. Isso significa que a confiabilidade do sensor é inversamente proporcional à quantidade de erros apresentada pelo sensor, pois quanto maior o valor do parâmetro *trustworthiness* (o valor varia entre 0 e 1), maior a confiabilidade.

A Figura 19 apresenta a comparação de valores da métrica IQR utilizando três abordagens diferentes: utilizando apenas uma fatia dos dados e aplicando o cálculo normal para obtenção do valor da métrica IQR (IQR parcial), utilizando todo o conjunto de dados conhecido e aplicando o cálculo normal para obtenção do IQR (IQR total), ou seja, todos os dados já vistos até o momento e utilizando apenas uma fatia dos dados e a técnica *t-digest*

(IQR *t-digest*). Os valores foram gerados utilizando o conjunto de dados NUMENTA. Cabe ressaltar que o método FOCUSeR utiliza o conceito de janela deslizante, ou seja, mantém em memória um número limitado de amostras. Quando uma nova amostra é gerada, a amostra mais antiga é desprezada.

Figura 18. Evolução do índice de *ranking* de um sensor em relação ao número de erros real do conjunto de dados.



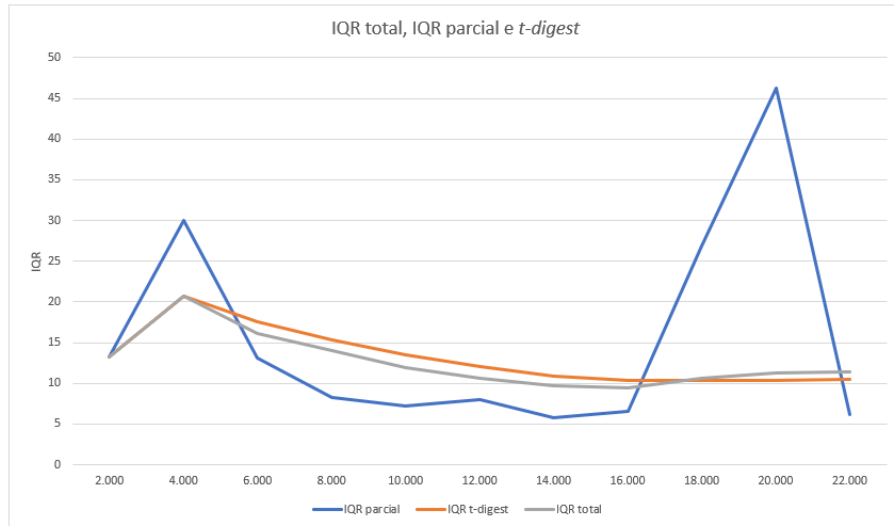
Fonte: Elaborada pelo autor (2022).

Essa abordagem é utilizada porque o ambiente de processamento utilizado (*Fog Computing*) é um ambiente com recursos de memória limitados. Na Figura 19, os valores representados pela linha “IQR parcial” representam os valores da métrica IQR gerados utilizando apenas as amostras presentes no buffer da janela deslizante. Os valores representados pela linha “IQR total” foram gerados utilizando todo o conjunto de dados. E finalmente, os valores representados pela linha “IQR *t-digest*” foram gerados utilizando os valores já gerados pelo algoritmo da métrica *t-digest* passando como parâmetro as amostras presentes no buffer da janela deslizante. Como é possível ver na imagem, os valores referentes a métrica IQR parcial ficam muito distantes dos valores reais da métrica para todo o conjunto de dados, ou seja, IQR total. Por outro lado, os valores gerados utilizando a técnica IQR *t-digest* ficam muito próximos dos valores reais.

As Figura 20 a Figura 22 ilustram a abordagem de treinamento online do método FOCUSeR, bem como a capacidade de manter os índices atualizados. A Figura 20 apresenta os resultados da classificação dos dados em anomalia, normal e falha. Na imagem, os dados

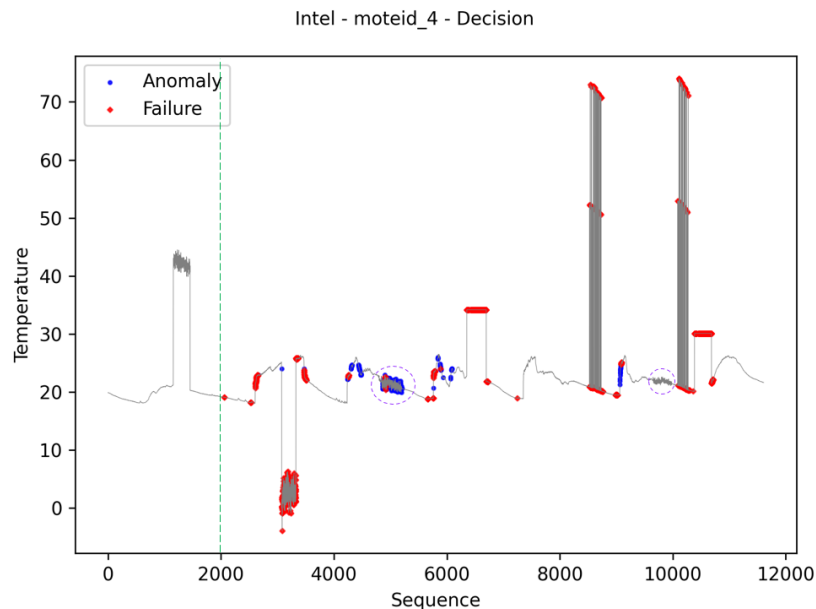
iniciais foram utilizados na etapa de *burn-in* para gerar as métricas a partir da métrica IQR (dados à esquerda da linha tracejada verde).

Figura 19. Comportamento da métrica IQR utilizando apenas uma fatia de dados (IQR parcial), utilizando todo o conjunto de dados conhecido (IQR total) e utilizando a técnica *t-digest*.



Fonte: Elaborada pelo autor (2022).

Figura 20. Conjunto de dados Intel: Resultados finais para o sensor 4.

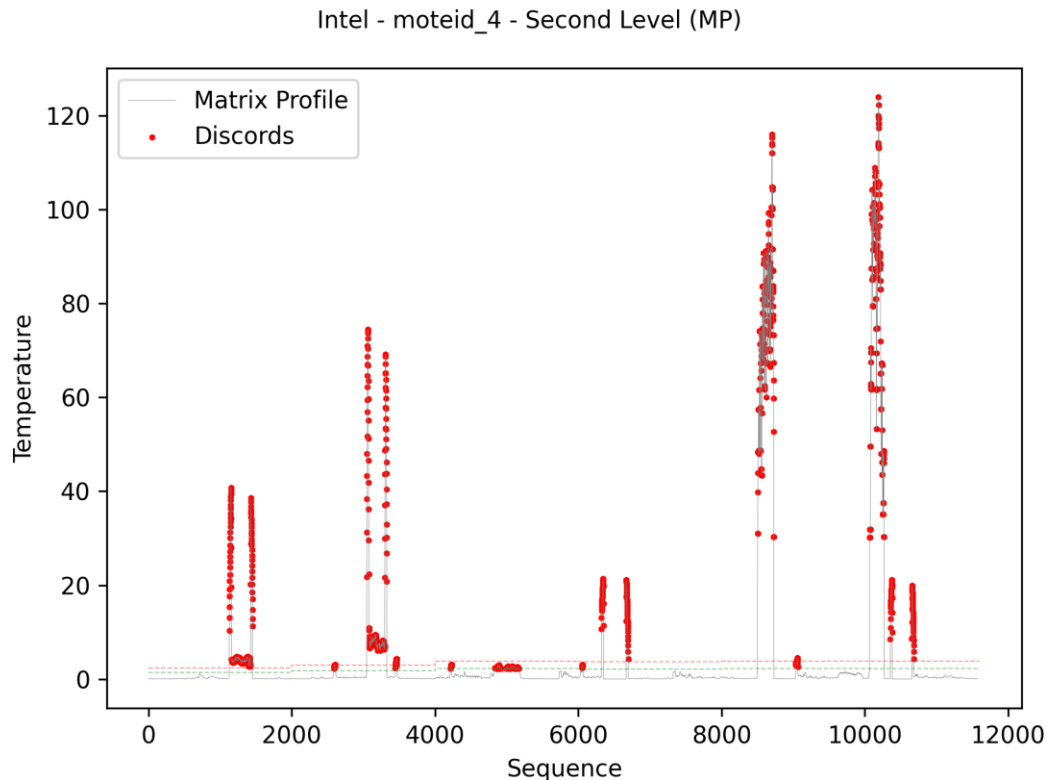


Fonte: Elaborada pelo autor (2022).

A Figura 21 apresenta o resultado da avaliação referente ao segundo nível do algoritmo. Essa avaliação foi realizada utilizando o algoritmo *matrix profile*. As linhas verde e vermelha (Figura 21) representam os limiares utilizados na avaliação dos dados. É possível

ver que as linhas têm quebras suaves em intervalos de comprimento fixo relacionados ao valor do parâmetro de *IQR Refresh*. Essas quebras representam a atualização dos indicadores da métrica IQR. A estrutura *t-digest* foi utilizada para possibilitar a atualização desses indicadores.

Figura 21. Conjunto de dados Intel: resultados *matrix profile*.

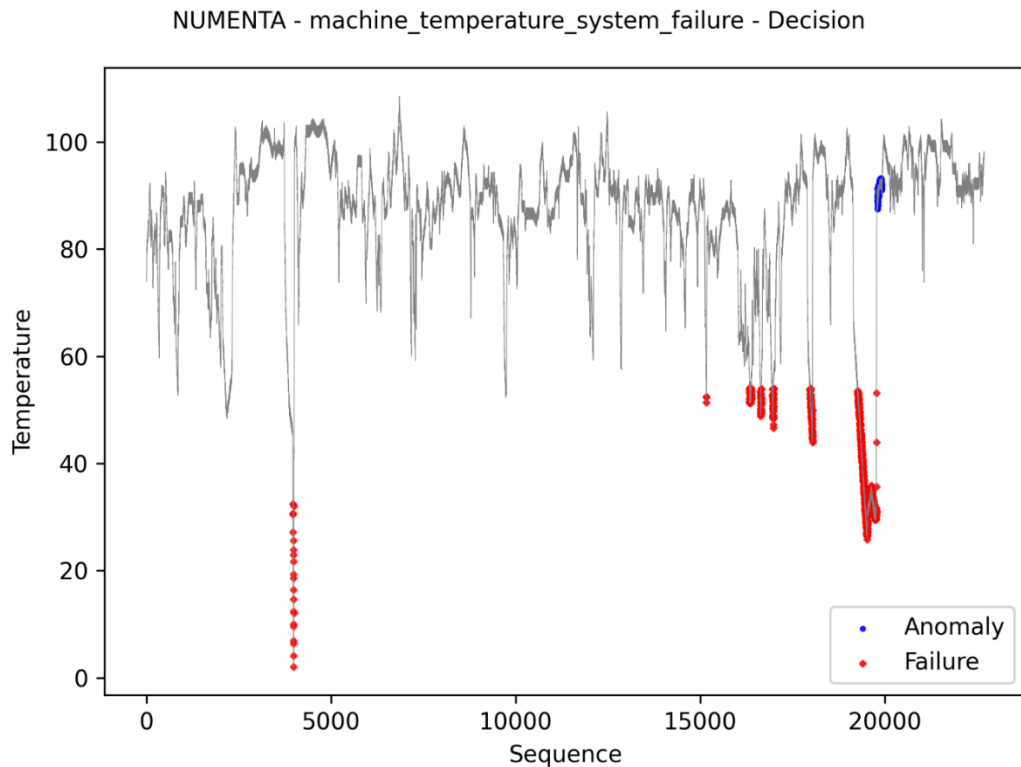


Fonte: Elaborada pelo autor (2022).

A Figura 22 apresenta os resultados referentes ao conjunto de dados *NUMENTA*. Apesar da ausência de rótulos, e por este motivo não ter sido incluído na avaliação de *Acurácia*, alguns trabalhos (AHMAD *et al.*, 2017; LAVIN; AHMAD, 2015) mencionam a presença de três anomalias neste conjunto de dados. Comparando os resultados do gráfico gerado com o do trabalho NUMENTA (2021) é possível ver que a primeira e a última anomalias foram identificadas nos mesmos pontos e a segunda em uma localização próxima, mas com um ponto adicional (próximo a sequência 15000). Como o trabalho afirma, a segunda anomalia é difícil de identificar exatamente. Na imagem, a segunda anomalia é identificada aproximadamente próximo a amostra 18000.

Além de classificar os dados em normal, anomalia e falha, o método FOCUSer entrega uma classificação do tipo outlier, conforme exibido na Figura 10.

Figura 22. NUMENTA: resultados online, atualizados e de identificação de dados.



Fonte: Elaborada pelo autor (2022).

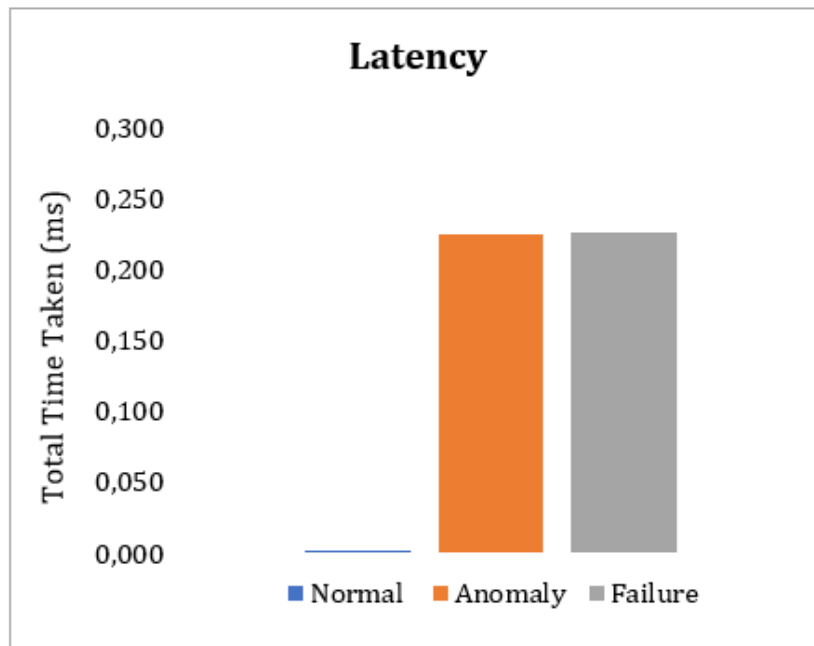
## 5.7 AVALIAÇÃO DE LATÊNCIA

Para avaliação de latência, o conceito de latência foi considerado como a diferença de tempo entre a hora de entrada e a hora de saída da mensagem no sistema. Neste experimento as mensagens foram processadas nos nodos *raspberrypi* configurados no ambiente experimental desenvolvido. Conforme mostrado na Figura 23, a latência média adicionada pelo método FOCUSeR é de aproximadamente 0,0016 milissegundos por transação. Quando são identificadas falhas ou anomalias nas leituras, o método proposto FOCUSeR busca dados normais em sensores vizinhos. Nesses casos, a latência média adicionada é de aproximadamente 0,225 milissegundos por transação.

No caso da latência média para consulta à lista de *ranking*, os tempos são semelhantes aos valores das consultas aos dados dos sensores próximos, pois ambas as consultas são realizadas nas listas de *hash* distribuídas.

O desempenho em relação à latência não foi comparado com outros trabalhos devido à diversidade de abordagens em relação às métricas utilizadas. A maioria dos trabalhos utiliza como métrica os tempos de consulta da lista de *ranking* e, nestes casos, a maior parte do tempo é gasto com as operações de seleção do sensor de acordo com os critérios de consulta. Além disso, aspectos como velocidade de transmissão da rede e *hardware* utilizado nos testes impactam drasticamente os resultados.

Figura 23. Medidas de latência.



Fonte: Elaborada pelo autor (2022).

## 5.8 MONITORAMENTO NA NUVEM

Por fim, a Figura 24 apresenta um exemplo de monitoramento de ambientes utilizando os resultados produzidos pelo método FOCUSeR.

A imagem foi gerada utilizando dados fictícios. O painel de monitoramento mostrado na imagem inclui o monitoramento do valor do sensor (canto superior esquerdo), no caso, o valor da temperatura monitorada; os resultados de saída da avaliação nos diferentes níveis do método FOCUSeR (direita) nos quais seriam exibidos apenas os pontos referentes as amostras identificadas como anormais; e um trecho da lista de *ranking* de sensores (canto inferior esquerdo).

Figura 24. Exemplo de painel de monitoramento.



Fonte: Elaborada pelo autor (2022).

## 5.9 CONSIDERAÇÕES

Os resultados numéricos obtidos nos testes indicam a viabilidade do método proposto. Entretanto, índices obtidos com a métrica *Acurácia* são um pouco inferiores em comparação com alguns trabalhos relacionados. No cálculo da métrica, a maior incidência ocorre devido aos valores de Falso Negativo, ou seja, valores considerados normais quando deveriam ser considerados como anomalias ou falhas. É importante lembrar que os conjuntos de dados em que isso ocorre são conjuntos de dados com falhas inseridas artificialmente (BRUIJN *et al.*, 2016). Analisando mais detalhadamente os rótulos gerados erroneamente, a maior incidência desse tipo de falha ocorre em falhas consideradas como "*malfunction*" no trabalho em que os dados foram obtidos. Nesse tipo de falha artificial, os dados são gerados com uma frequência maior do que a esperada. Existem duas ocorrências desse tipo de falha na imagem apresentada na Figura 20. A primeira falha está destacada pelo primeiro (da esquerda para a direita) círculo pontilhado roxo na imagem, em que as anormalidades foram identificadas corretamente. No caso da segunda falha (segundo círculo tracejado), o método proposto não foi capaz de identificar o erro. Isso se deve à amplitude dos valores inseridos, que no caso da segunda região, é muito pequena, impedindo assim a identificação pelo método *matrix profile*. Para ser capaz de identificar esse tipo de situação o método precisaria utilizar técnicas de reconhecimento de anomalias através de *shapelets* por exemplo, em que as características da forma são utilizadas para identificação dos pontos de interesse. No caso

específico da segunda falha destacada na imagem, apesar de a variação (amplitude) ser pequena, ela ocorre com uma frequência grande, ou seja, aumenta ou diminui entre um intervalo pequeno de amostras, gerando uma forma com vários e pequenos picos e vales.

Também seria possível resolver os problemas encontrados caso fosse adotada uma abordagem supervisionada (*off-line*) como forma de definição dos parâmetros do método (limites da métrica IQR). Com esse tipo de abordagem seria possível treinar modelos mais específicos para cada conjunto de dados. Entretanto, o custo de implantação seria significativamente maior. No início do desenvolvimento do método proposto foram utilizados algoritmos supervisionados para identificação de anomalias e falhas. Nesse estágio da pesquisa, durante a etapa de treinamento do modelo, foram obtidas taxas muito próximas a 100%. Apesar disso, na etapa de testes, os modelos mostravam-se frágeis a novos conjuntos de dados. Um exemplo dessa situação ocorreu no conjunto de dados SensorScope. Lembrando que esse conjunto possui dados de sensores instalados em montanhas nos alpes suíços. Cada arquivo contém dados de um único sensor, e cada sensor está instalado em pontos distintos, com características ambientais distintas, como é o caso, por exemplo, das diferentes altitudes. Dessa forma, o impacto das condições do ambiente nos conjuntos de dados dificulta a reutilização de um mesmo modelo para dois sensores diferentes. Essas condições implicam na necessidade de criação de um novo modelo para cada sensor, o que torna o processo inviável em vários aspectos, como tempo e custo, dado o enorme volume de sensores existente atualmente. Justamente pelos motivos expostos, optou-se pela utilização de um método capaz de aprender, de forma automática, um conjunto de parâmetros capazes de refletir as condições do ambiente. Assim, o método FOCUSeR aprende o conjunto de parâmetros para cada sensor, durante o início do processamento dos dados do sensor (etapa de *burn-in*).

Quanto ao recurso de recomposição de dados quando ocorre uma falha ou uma anomalia (Figura 17), na qual foi obtida a taxa de aproximadamente 87% de recuperação de dados, cabe destacar que essa taxa é diretamente proporcional à existência de dados de outros sensores próximos no espaço e no tempo. Isso quer dizer que se para todo dado de falha ou anomalia existir um outro sensor, semelhante, que esteja próximo geograficamente e que tenha gerado dados recentemente, essa taxa será de 100%. A recuperação e distribuição desse tipo de informação juntamente com as respostas produzidas pelo método FOCUSeR contribui para a redução da incerteza, já que esses dados só são recuperados de sensores com altos valores no parâmetro *trustworthiness*, que indica a confiabilidade de cada sensor, e assim fornecem valores de comparação com os valores produzidos pelo sensor atual.



Em relação à latência, considerando os conjuntos de dados reais e ambiente experimental utilizados nos testes, os tempos de resposta obtidos são promissores e indicam a robustez do método FOCUSER. Cabe ressaltar que a maior parte dos trabalhos relacionados utilizou ambientes extremamente distintos para realização dos testes, como acontece por exemplo em um dos trabalhos que realiza toda a etapa de avaliação utilizando apenas um gerenciador de banco de dados. Ainda assim esses trabalhos medem os tempos dos seus algoritmos. Esse tipo de medida tem pouca utilidade científica, já que não fornece uma medida útil sobre como seriam esses tempos em um ambiente real.



## 6 CONCLUSÕES E TRABALHOS FUTUROS

Esta tese propõe um método para *ranking* de sensores que utiliza a avaliação de dados como critério. O método proposto, denominado FOCUSeR, oferece uma solução real para as WSNs, capaz de executar na camada *Fog Computing*, gerando as listas de *ranking* de forma *online* e atualizadas ao longo do tempo, e disponibilizando essas listas de forma distribuída. Segundo a revisão da literatura realizada nesta tese, este é o primeiro método de *ranking* de sensores que aborda todos esses recursos.

O método foi avaliado em um ambiente experimental real, desenvolvido com nodos *Fog* e sensores de temperatura.

A opção pelo desenvolvimento de um ambiente experimental, abrangendo *hardware* e *software*, e criação de um conjunto de dados para validar o método proposto se deve ao fato de que em muitos dos trabalhos relacionados as medidas apresentadas nos resultados são passivas de questionamentos, devido justamente ao ambiente utilizado na avaliação dessas propostas ser muito distinto dos ambientes reais. Assim utilizar um ambiente o mais próximo possível de um ambiente real sustenta os resultados apresentados neste trabalho de pesquisa, contemplando assim os Requisitos 8 e 9 apresentados na seção de contribuições do método FOCUSeR.

A motivação para o desenvolvimento do método FOCUSeR baseia-se no fato de que, embora os trabalhos relacionados nesta área apresentem resultados promissores, ainda existem desafios a serem superados na área de *ranking* de sensores. Dado o crescimento exponencial do volume de dispositivos conectados, a grande heterogeneidade desses dispositivos e dos dados gerados, e considerando que um dos principais desafios é entregar de forma eficiente as medições detectadas, com a máxima fidelidade. Para isso se faz necessária a utilização de técnicas de avaliação de dados, visando fornecer, juntamente com os dados produzidos pelos sensores, métricas sobre a qualidade desses dados. Assim, ao utilizar avaliação de dados individuais e coletivos (primeiro e segundo níveis do método FOCUSeR) o método fornece confiabilidade às informações produzidas, atendendo ao Requisito 5. Com o mesmo intuito, a utilização de dados de contexto também contribui para a redução da incerteza nas respostas.

É razoável supor que os métodos que utilizem aprendizagem supervisionada para avaliação de dados, serão extremamente trabalhosos, senão inviáveis, quando usados em ambientes reais. Isso ocorre porque o ambiente de cada sensor é frequentemente distinto da maior parte dos demais sensores, impossibilitando a reutilização de um mesmo modelo de

dados para dois sensores diferentes. Nesse sentido, mesmo que fosse possível utilizar o mesmo modelo para vários sensores, haveria a necessidade de reavaliar esse modelo periodicamente, dada a natureza dinâmica dos ambientes monitorados por sensores.

Assim, embora os métodos com essa abordagem possam apresentar resultados ligeiramente superiores aos do método proposto, o método FOCUSeR pode ser considerado superior, já que é livre de treinamento (Requisito 6) e possui recursos para manter seus parâmetros de avaliação de dados atualizados ao longo do tempo. Por isso, o método FOCUSeR garante a aderência dos parâmetros às características do ambiente monitorado fazendo uma atualização desses parâmetros ao longo do tempo, atendendo assim ao Requisito 7.

Uma vez obtida a avaliação do método para os dados produzidos pelo sensor, é preciso manter as listas de *ranking*. Devido à natureza distribuída dos sensores se faz necessário adotar um método capaz de manter essas listas também de forma distribuída, mas permitindo que a busca possa ser facilmente realizada de forma mais ou menos abrangente, dependendo da necessidade do usuário. Para isso, o método FOCUSeR utiliza tabelas de *hash* distribuídas para a manutenção das listas de *ranking* e *geohashes* como argumentos de busca. Esse conjunto de recursos atende ao Requisito 1.

Por fim, já que muitas das aplicações que utilizam dados de sensores possuem requisitos de tempo real, o método precisa atender as requisições no menor tempo possível. Para isso, deve atender aos requisitos de baixa latência. Dessa forma, considerando as diferentes bandas de transmissão de dados disponíveis presentes nas WSNs, o processamento dos dados do sensor precisa acontecer o mais próximo possível da solicitação ou do usuário final. Assim, esse processamento deve ocorrer nas camadas de *Fog* ou *Edge* da rede. Entretanto como na camada de *Edge* os recursos computacionais são extremamente limitados, o ambiente de *Fog Computing* se torna ideal para execução dessa tarefa. Assim, para atender ao Requisito 2, o processamento do método é realizado na camada de *Fog Computing*, ou seja, próximo, em relação à rede, das fontes de dados (sensores). Por outro lado, a camada de *Fog Computing* também possui limitação de recursos computacionais, restringindo assim a utilização de métodos que não possuam baixa complexidade de tempo. Conforme demonstrado, o método possui complexidade de tempo igual a  $O(n \log n)$ , atendendo assim ao Requisito 3. No mesmo sentido, as listas de ranking são atualizadas sempre que uma nova amostra entra no sistema, evitando que essas listas precisem ser geradas a cada nova

requisição (consulta) de usuários ou serviços. Esse recurso atende ao Requisito 4, já que reduz a latência nas requisições.

Como trabalhos futuros indicados para continuidade da pesquisa com melhoria nos resultados obtidos, considera-se a utilização de dados de sensores relacionados espacial e temporalmente com o sensor em avaliação para decisão sobre a anormalidade dos dados. A adoção dessa abordagem pode reduzir a incerteza em relação ao estado do sensor. Com o mesmo intuito, uma possível melhora nos resultados pode ser obtida através da utilização de uma abordagem multivariada, através da observação de outras variáveis para decisão sobre a anormalidade dos dados, que também pode contribuir para a redução da incerteza.

Acrescentar ao método recursos para detectar a convergência entre os parâmetros da métrica de avaliação de dados (IQR) e o conjunto de dados pode contribuir bastante com a precisão das respostas fornecidas, já que o funcionamento normal do método só iniciaria quando os parâmetros estivessem aderentes ao conjunto de dados.

Para superar os problemas enfrentados com a fragilidade do método em detectar erros em que ocorre variação em uma grande quantidade de amostras subsequentes, mas com diferenças de pequena amplitude entre cada amostra (falhas do tipo “*malfunction*” já discutidas e apresentadas na Figura 20) poderiam ser implementados recursos para reconhecimento de características representativas sobre a forma de uma série temporal e dessa forma aprender a detectar séries temporais anormais. Esse recurso poderia ser implementado através do uso de *shapelets* (ALSHAER; GARCIA-RODRIGUEZ; GOUY-PAILLER, 2020).

E finalmente, a implementação de recursos para o uso racional de sensores, visando proporcionar um mecanismo de economia de energia, pode ser extremamente útil, além de minimizar o uso de sensores em estado de falha.



## REFERÊNCIAS

- ABDELAAL, M. *et al.* GaaS: Adaptive Cross-Platform Gateway for IoT Applications. *Em: GAAS*, 2019, Monterey, CA, USA. **16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)**. Monterey, CA, USA: IEEE, 2019. p. 217–226.
- ABERER, K. *et al.* P-Grid: a self-organizing structured P2P system. **ACM SIGMOD Record**, [s. l.], v. 32, n. 3, p. 29, 2003.
- ABID, A. *et al.* Outlier Detection in Wireless Sensor Networks Based on OPTICS Method for Events and Errors Identification. **Wireless Personal Communications**, [s. l.], v. 97, n. 1, p. 1503–1515, 2017.
- ABID, A.; KACHOURI, A.; MAHFOUDHI, A. Anomaly detection through outlier and neighborhood data in Wireless Sensor Networks. *Em: ATSIP*, 2016, Monastir, Tunisia. **2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)**. Monastir, Tunisia: IEEE, 2016. p. 26–30.
- AHMAD, S. *et al.* Unsupervised real-time anomaly detection for streaming data. **Neurocomputing**, [s. l.], v. 262, Online Real-Time Learning Strategies for Data Streams, p. 134–147, 2017.
- ALAM, F. *et al.* Data Fusion and IoT for Smart Ubiquitous Environments: A Survey. **IEEE Access**, [s. l.], v. 5, p. 9533–9554, 2017.
- AL-FUQAHA, A. *et al.* Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. **IEEE Communications Surveys Tutorials**, [s. l.], v. 17, n. 4, p. 2347–2376, 2015.
- AL-SHABI, M.; ABUHAMDIAH, A. Using deep learning to detecting abnormal behavior in internet of things. **International Journal of Electrical and Computer Engineering (IJECE)**, [s. l.], v. 12, n. 2, p. 2108–2120, 2022.
- ALSHAER, M.; GARCIA-RODRIGUEZ, S.; GOUY-PAILLER, C. Detecting Anomalies from Streaming Time Series using Matrix Profile and Shapelets Learning. *Em: ICTAI*, 2020, Baltimore, USA. **32nd International Conference on Tools with Artificial Intelligence (ICTAI)**. Baltimore, USA: IEEE, 2020. p. 376–383. Disponível em: <https://www.computer.org/csdl/proceedings-article/ictai/2020/922800a376/1pP3tGGten6>. Acesso em: 23 set. 2021.
- ANASTASI, G. *et al.* Energy conservation in wireless sensor networks: A survey. **Ad Hoc Networks**, [s. l.], v. 7, n. 3, p. 537–568, 2009.
- APACHE FOUNDATION. **Apache Cassandra Project**. [S. l.], [s. d.]. Disponível em: <http://cassandra.apache.org/>. Acesso em: 21 set. 2020.
- ASHTON, K. **That “Internet of Things” Thing: In the Real World Things Matter More than Ideas**. [S. l.], 2009. Disponível em: <https://www.rfidjournal.com/articles/view?4986>. Acesso em: 12 set. 2018.

AWS FOR THE EDGE. [S. l.], [s. d.]. Disponível em: <https://aws.amazon.com/edge/>. Acesso em: 2 abr. 2022.

BAJCSY, R. Active perception. **Proceedings of the IEEE**, [s. l.], v. 76, n. 8, p. 10, 1988.

BARBETTA, P. A.; BORNIA, A. C.; REIS, M. M. **Estatística Para Cursos De Engenharia E Informática**. 3rd ed.ed. São Paulo, SP, Brazil: Atlas, 2010.

BEVRANI, H. **Robust Power System Frequency Control**. New York, NY, USA: Springer New York, NY, 2009. (Power Electronics and Power Systems). *E-book*. Disponível em: <https://www.springer.com/gp/book/9781441946614>. Acesso em: 14 set. 2021.

BHARTI, S.; PATTANAIK, K. K.; BELLAVISTA, P. Value of Information Based Sensor Ranking for Efficient Sensor Service Allocation in Service Oriented Wireless Sensor Networks. **IEEE Transactions on Emerging Topics in Computing**, [s. l.], v. 9, n. 2, p. 823–838, 2021.

BIEL, L.; WIDE, P. Active perception for autonomous sensor systems. **Instrumentation Measurement Magazine**, [s. l.], v. 3, n. 4, IMM, p. 4, 2000.

BIEL, L.; WIDE, P. Active perception in a sensor fusion model. *Em*: SPIE, 2002, Orlando, FL, USA. **Sensor Fusion: Architectures, Algorithms, and Applications VI**. Orlando, FL, USA: [s. n.], 2002. p. 13. Disponível em: <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/4731/0000/Active-perception-in-a-sensor-fusion-model/10.1117/12.458381.short>. Acesso em: 9 nov. 2018.

BONOMI, F. *et al.* Fog Computing and Its Role in the Internet of Things. *Em*: MCC, 2012, Helsinki, Finland. **Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing**. Helsinki, Finland: ACM, 2012. p. 13–16. Disponível em: <http://doi.acm.org/10.1145/2342509.2342513>. Acesso em: 7 mar. 2018.

BOSMAN, H. H. W. J. *et al.* Ensembles of incremental learners to detect anomalies in ad hoc sensor networks. **Ad Hoc Networks**, [s. l.], v. 35, Special Issue on Big Data Inspired Data Sensing, Processing and Networking Technologies, p. 14–36, 2015.

BOSMAN, H. H. *et al.* Spatial anomaly detection in sensor networks using neighborhood information. **Information Fusion**, [s. l.], v. 33, p. 41–56, 2017.

BRIN, S.; PAGE, L. The Anatomy of a Large-Scale Hypertextual Web Search Engine. **Computer Networks**, [s. l.], v. 30, p. 107–117, 1998.

BRINGEL FILHO, J.; AGOULMINE, N. A Quality-Aware Approach for Resolving Context Conflicts in Context-Aware Systems. *Em*: IFIP, 2011, Melbourne, VIC, Australia. **9th International Conference on Embedded and Ubiquitous Computing**. Melbourne, VIC, Australia: [s. n.], 2011. p. 229–236.

BRUIJN, B. de *et al.* Benchmark Datasets for Fault Detection and Classification in Sensor Data. *Em*: ICSN, 2016, Rome, Italy. **Proceedings of the 5th International Conference on Sensor Networks**. Rome, Italy: [s. n.], 2016. p. 185–195. Disponível em: <https://research.utwente.nl/en/publications/benchmark-datasets-for-fault-detection-and-classification-in-sens>. Acesso em: 21 jul. 2020.



BUCHHOLZ, T.; SCHIFFERS, M. Quality of Context: What It Is And Why We Need It. *Em: QUALITY OF CONTEXT*, 2003. In **Proceedings of the 10th Workshop of the OpenView University Association: OVUA'03**. [S. l.: s. n.], 2003.

BUYYA, R.; DASTJERDI, A. V. **Internet of Things: Principles and Paradigms**. 1st. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016.

CABRAL, L.; COMPTON, M.; MÜLLER, H. A Use Case in Semantic Modelling and Ranking for the Sensor Web. *Em: ISWC*, 2014, Riva del Garda, Italy. (P. Mika et al., Org.) **13th International Semantic Web Conference**. Riva del Garda, Italy: Springer International Publishing, 2014. p. 15.

CAI, S. *et al.* An Efficient Outlier Detection Approach for Streaming Sensor Data Based on Neighbor Difference and Clustering. **Security and Communication Networks**, [s. l.], v. 2022, p. e3062541, 2022.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM Computing Surveys**, [s. l.], v. 41, n. 3, p. 1–58, 2009.

CHARITH PERERA. **Sensing as a Service for Internet of Things - A Roadmap**. [S. l.: s. n.], 2017. Disponível em: [https://www.researchgate.net/profile/Charith\\_Perera2/publication/309585306\\_Sensing\\_as\\_a\\_Service\\_for\\_Internet\\_of\\_Things\\_A\\_Roadmap/links/5b08844a0f7e9b1ed7f56468/Sensing-as-a-Service-for-Internet-of-Things-A-Roadmap.pdf](https://www.researchgate.net/profile/Charith_Perera2/publication/309585306_Sensing_as_a_Service_for_Internet_of_Things_A_Roadmap/links/5b08844a0f7e9b1ed7f56468/Sensing-as-a-Service-for-Internet-of-Things-A-Roadmap.pdf). Acesso em: 7 dez. 2018.

CHEN, M.; MAO, S.; LIU, Y. Big Data: A Survey. **Mobile Networks and Applications**, [s. l.], v. 19, n. 2, p. 171–209, 2014.

CHERBAL, S.; BOUKERRAM, A.; BOUBETRA, A. A survey of DHT solutions in fixed and mobile networks. **Int. J. of Commun. Netw. Distrib. Syst.**, [s. l.], v. 17, n. 1, p. 14, 2016.

CLARIVATE. **Web of Science**. [S. l.], [s. d.]. Disponível em: <https://clarivate.com>. Acesso em: 7 abr. 2022.

COSTA, F. S. *et al.* FASTEN IIoT: An Open Real-Time Platform for Vertical, Horizontal and End-To-End Integration. **Sensors**, [s. l.], v. 20, n. 19, p. 5499, 2020.

COSTA, F. S.; NASSAR, S. M.; DANTAS, M. A. R. FOCUSeR: A Fog Online Context-Aware Up-to-Date Sensor Ranking Method. **Journal of Sensor and Actuator Networks**, [s. l.], v. 11, n. 2, p. 25, 2022.

COSTA, F.; NASSAR, S.; DANTAS, M. GoAT: A Sensor Ranking Approach for IoT Environments. *Em: GOAT*, 2021, Florida, USA. **Proceedings of the 11th International Conference on Cloud Computing and Services Science - CLOSER**. Florida, USA: [s. n.], 2021. p. 169–177. Disponível em: <https://www.scitepress.org/Link.aspx?doi=10.5220/0010403801690177>. Acesso em: 2 set. 2021.

- COX, M.; ELLSWORTH, D. Application-controlled demand paging for out-of-core visualization. *Em: IEEE*, 1997, Phoenix, Arizona, USA. **Proceedings of the IEEE Visualization Conference**. Phoenix, Arizona, USA: [s. n.], 1997. p. 235–244.
- DAUTOV, R.; DISTEFANO, S. Automating IoT Data-Intensive Application Allocation in Clustered Edge Computing. **IEEE Transactions on Knowledge and Data Engineering**, [s. l.], p. 14, 2019.
- DESAI, U. B.; JAIN, B. N.; MERCHANT, S. N. **Wireless Sensor Networks: Technology Roadmap**. Mumbai, India: [s. n.], 2007.
- DILLI, R. *et al.* Fuzzy Logic and MCDA in IoT Resources Classification. *Em: SAC*, 2018, New York, NY, USA. **Proceedings of the 33rd Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2018. p. 6. Disponível em: <http://doi.acm.org/10.1145/3167132.3167216>. Acesso em: 16 fev. 2019.
- DING, H. *et al.* Querying and mining of time series data: experimental comparison of representations and distance measures. **Proceedings of the VLDB Endowment**, [s. l.], v. 1, n. 2, p. 1542–1552, 2008.
- DUNNING, T. The t-digest: Efficient estimates of distributions. **Software Impacts**, [s. l.], v. 7, p. 100049, 2021.
- ELAHI, B. M. *et al.* Sensor ranking: A primitive for efficient content-based sensor search. *Em: SENSOR RANKING*, 2009, San Francisco, CA, USA. **IEEE**. San Francisco, CA, USA: [s. n.], 2009. p. 12.
- ELSEVIER. **Scopus**. [S. l.], [s. d.]. Disponível em: <https://www.scopus.com/search/>. Acesso em: 5 out. 2018.
- F. CODD, E. A Relational Model for Large Shared Data Banks. **Communications of the ACM**, [s. l.], v. 13, p. 377, 1970.
- FATHY, Y.; BARNAGHI, P.; TAFAZOLLI, R. Large-Scale Indexing, Discovery, and Ranking for the Internet of Things (IoT). **ACM Comput. Surv.**, [s. l.], v. 51, n. 2, p. 53, 2018.
- GHADDAR, A.; DARWISH, L.; YAMOUT, F. Identifying Mass-based local anomalies using Binary Space Partitioning. *Em: ,* 2019, Barcelona, Spain. **IEEE**. Barcelona, Spain: [s. n.], 2019. p. 183–190.
- GHORBEL, O. *et al.* An overview of outlier detection technique developed for wireless sensor networks. *Em: ,* 2013, Hammamet, Tunisia. **IEEE**. Hammamet, Tunisia: [s. n.], 2013. p. 1–6.
- GOMES, E. *et al.* A Survey from Real-Time to Near Real-Time Applications in Fog Computing Environments. **Telecom**, [s. l.], v. 2, n. 4, p. 489–517, 2021.
- GOOGLE. **Google Acadêmico**. [S. l.], [s. d.]. Disponível em: <https://scholar.google.com/>. Acesso em: 7 abr. 2022.

- HAUBEN, R. **From the ARPANET to the Internet Ronda Hauben**. [S. l.], 1998. Disponível em: [http://www.columbia.edu/~rh120/other/tcpdigest\\_paper.txt](http://www.columbia.edu/~rh120/other/tcpdigest_paper.txt). Acesso em: 18 set. 2018.
- HUSSAIN, N.; ANEES, T.; AZEEMULLAH. Development of a Novel Approach to Search Resources in IoT. **International Journal of Advanced Computer Science and Applications (IJACSA)**, [s. l.], v. 9, n. 9, p. 14, 2018.
- IGGENA, T. *et al.* IoT-Crawler: Challenges and Solutions for Searching the Internet of Things. **Sensors**, [s. l.], v. 21, n. 5, p. 1559, 2021.
- INGELREST, F. *et al.* SensorScope: Application-specific sensor network for environmental monitoring. **ACM Transactions on Sensor Networks**, [s. l.], v. 6, n. 2, p. 17:1-17:32, 2010.
- INTEL LAB DATA. [S. l.], [s. d.]. Disponível em: <http://db.csail.mit.edu/labdata/labdata.html>. Acesso em: 20 dez. 2018.
- IORGA, M. *et al.* **NIST: Fog computing conceptual model**. Gaithersburg, USA: National Institute of Standards and Technology, 2018. Disponível em: <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-325.pdf>. Acesso em: 19 abr. 2018.
- JAIN, S. *et al.* Fog computing in enabling 5G-driven emerging technologies for development of sustainable smart city infrastructures. **Cluster Computing**, [s. l.], v. 25, n. 2, p. 1111–1154, 2022.
- JINDAL, V.; COLLEGE, D. A. V. History and Architecture of Wireless Sensor Networks for Ubiquitous Computing. [s. l.], v. 7, n. 2, p. 4, 2018.
- KAKUNSI, D. C.; CANDRA, M. Z. C. SDRank: An Adaptable Service Selection for IoT Based on Ranking. *Em: SDRANK*, 2018, Mataram, Indonesia. **IEEE**. Mataram, Indonesia: [s. n.], 2018. p. 6.
- KANG, H. *et al.* A conceptual device-rank based resource sharing and collaboration of smart things. **Multimedia Tools and Applications**, [s. l.], v. 75, n. 22, p. 13, 2016.
- KAUFMAN, L. M. Data Security in the World of Cloud Computing. **IEEE Security & Privacy Magazine**, [s. l.], v. 7, n. 4, p. 61–64, 2009.
- KERTIOU, I. *et al.* A dynamic skyline technique for a context-aware selection of the best sensors in an IoT architecture. **Ad Hoc Networks**, [s. l.], v. 81, p. 14, 2018.
- KUO, Y.-H.; LI, Z.; KIFER, D. Detecting Outliers in Data with Correlated Measures. *Em:* , 2018, Atlanta, GA, USA. **ACM**. Atlanta, GA, USA: ACM, 2018. p. 287–296. Disponível em: <http://doi.acm.org/10.1145/3269206.3271798>. Acesso em: 25 fev. 2019.
- LAVIN, A.; AHMAD, S. Evaluating Real-time Anomaly Detection Algorithms - the Numenta Anomaly Benchmark. *Em: ICMLA*, 2015, Miami, FL, USA. **IEEE**. Miami, FL, USA: [s. n.], 2015. p. 38–44. Disponível em: <http://arxiv.org/abs/1510.03336>. Acesso em: 1 abr. 2022.

- LEE, H.; LEE, B. The Development of a State-Aware Equipment Maintenance Application Using Sensor Data Ranking Techniques. **Sensors**, [s. l.], v. 20, n. 11, p. 3038, 2020.
- LEINER, B. M. *et al.* A brief history of the internet. **ACM SIGCOMM Computer Communication Review**, [s. l.], v. 39, n. 5, p. 22–31, 2009.
- LIGHT, R. A. Mosquitto: server and client implementation of the MQTT protocol. **Journal of Open Source Software**, [s. l.], v. 2, n. 13, p. 265, 2017.
- MANZOOR, A.; TRUONG, H.-L.; DUSTDAR, S. Quality of Context: models and applications for context-aware systems in pervasive environments. **The Knowledge Engineering Review**, [s. l.], v. 29, n. 2, p. 16, 2014.
- MARJANI, M. *et al.* Big IoT Data Analytics: Architecture, Opportunities, and Open Research Challenges. **IEEE Access**, [s. l.], v. 5, p. 5247–5261, 2017.
- MARLEY, B.; WAILERS, T. **Natural Mystic**. 1995.
- MCCANN, S.; ASHLEY, A. **IEEE 802.11, The Working Group Setting the Standards for Wireless LANs**. [S. l.], 2018. Disponível em: [http://grouper.ieee.org/groups/802/11/Reports/802.11\\_Timelines.htm](http://grouper.ieee.org/groups/802/11/Reports/802.11_Timelines.htm). Acesso em: 18 set. 2018.
- MEKALA, M. S.; RIZWAN, P.; KHAN, M. S. Computational Intelligent Sensor-rank Consolidation Approach for Industrial Internet of Things (IIoT). **IEEE Internet of Things Journal**, [s. l.], p. 1–1, 2021.
- MICROSOFT AZURE IOT SUITE. [S. l.], [s. d.]. Disponível em: <https://azure.microsoft.com/pt-br/blog/microsoft-azure-iot-suite-connecting-your-things-to-the-cloud/>. Acesso em: 2 abr. 2022.
- MOHINUR RAHAMAN, M.; AZHARUDDIN, M. Wireless sensor networks in agriculture through machine learning: A survey. **Computers and Electronics in Agriculture**, [s. l.], v. 197, p. 106928, 2022.
- MUNIRATHINAM, S. Chapter Six - Industry 4.0: Industrial Internet of Things (IIOT). *Em*: RAJ, P.; EVANGELINE, P. (org.). **Advances in Computers**. [S. l.]: Elsevier, 2020. (The Digital Twin Paradigm for Smarter Systems and Environments: The Industry Use Cases). v. 117, p. 129–164. *E-book*. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0065245819300634>. Acesso em: 18 abr. 2022.
- NEHA; SAXENA, S. Vector method for ranking of sensors in IoT. *Em*: , 2016, Coimbatore, India. **IEEE**. Coimbatore, India: [s. n.], 2016. p. 5.
- NESA, N.; BANERJEE, I. SensorRank: An Energy Efficient Sensor Activation Algorithm for Sensor Data Fusion in Wireless Networks. **IEEE Internet of Things Journal**, [s. l.], v. 6, n. 2, p. 2532–2539, 2019.
- NIU, W. *et al.* Context-Aware Service Ranking in Wireless Sensor Networks. **Journal of Network and Systems Management**, [s. l.], v. 22, n. 1, p. 25, 2014.

NOBLE, B. D. *et al.* Agile application-aware adaptation for mobility. *Em: SOSP*, 1997, New York, NY, USA. **ACM**. New York, NY, USA: Association for Computing Machinery, 1997. p. 276–287. Disponível em: <https://doi.org/10.1145/268998.266708>. Acesso em: 10 fev. 2022.

NSF. **NSF Edge Workshop Report**. Washington,DC, USA: NATIONAL SCIENCE FOUNDATION, 2016.

NUMENTA ANOMALY BENCHMARK (NAB). Redwood City, CA, USA, 2021. Disponível em: <https://numenta.com/machine-intelligence-technology/numenta-anomaly-benchmark/>. Acesso em: 28 mar. 2022.

NUNES, L. *et al.* A Distributed Sensor Data Search Platform for Internet of Things Environments. [*s. l.*], 2016. Disponível em: <https://arxiv.org/abs/1606.07932v1>. Acesso em: 24 fev. 2019.

NUNES, L. H. *et al.* Multi-criteria IoT resource discovery: a comparative analysis. **Software: Practice and Experience**, [*s. l.*], v. 47, n. 10, p. 16, 2016.

NYLUND, A. L. **Tracing the BI Family Tree**. [*S. l.*], 1999. Disponível em: [http://www.escholar.com/documents/dw\\_family\\_tree.pdf](http://www.escholar.com/documents/dw_family_tree.pdf). Acesso em: 18 set. 2018.

PATIL, C. S.; R.R.KARHE; AHER, M. A. Development of Mobile Technology: A Survey. **International Journal of Advanced Research in Electrical, Electronics and Instrumentation Energy**, [*s. l.*], v. 1, n. 5, p. 374–379, 2012.

PATTAR, S. *et al.* Searching for the IoT Resources: Fundamentals, Requirements, Comprehensive Review, and Future Directions. **IEEE Communications Surveys Tutorials**, [*s. l.*], v. 20, n. 3, p. 31, 2018.

PELLE, I. *et al.* Cost and Latency Optimized Edge Computing Platform. **Electronics**, [*s. l.*], v. 11, n. 4, p. 561, 2022.

PERERA, C. *et al.* Sensor Search Techniques for Sensing as a Service Architecture for the Internet of Things. **IEEE Sensors Journal**, [*s. l.*], v. 14, n. 2, p. 406–420, 2014.

RASSAM, M. A.; MAAROF, M. A.; ZAINAL, A. Adaptive and online data anomaly detection for wireless sensor systems. **Knowledge-Based Systems**, [*s. l.*], v. 60, p. 44–57, 2014.

ROCHA, V.; BRANDÃO, A. A. F. A scalable multiagent architecture for monitoring IoT devices. **Journal of Network and Computer Applications**, [*s. l.*], v. 139, p. 1–14, 2019.

RUTA, M. *et al.* CoAP-based collaborative sensor networks in the Semantic Web of Things. **Journal of Ambient Intelligence and Humanized Computing**, [*s. l.*], v. 10, n. 7, p. 18, 2019.

SCHIFFMAN, H. R. Sensation and Perception: An Integrated Approach. *Em: SENSATION AND PERCEPTION: AN INTEGRATED APPROACH*. Edição: 5thed. New York: John Wiley & Sons, 2001. p. 12.

SCHILIT, B. N.; THEIMER, M. M. Disseminating active map information to mobile hosts. **IEEE Network**, [s. l.], v. 8, n. 5, p. 22–32, 1994.

SEZER, O. B.; DOGDU, E.; OZBAYOGLU, A. M. Context-Aware Computing, Learning, and Big Data in Internet of Things: A Survey. **IEEE Internet of Things Journal**, [s. l.], v. 5, n. 1, p. 27, 2018.

SHAH, S. H.; YAQOOB, I. A survey: Internet of Things (IOT) technologies, applications and challenges. *Em: A SURVEY*, 2016. **2016 IEEE Smart Energy Grid Engineering (SEGE)**. [S. l.: s. n.], 2016. p. 381–385.

SHELBY, Z.; HARTKE, K.; BORMANN, C. **The Constrained Application Protocol (CoAP)**. [S. l.], 2014. Disponível em: <https://www.rfc-editor.org/rfc/rfc7252.txt>. Acesso em: 22 fev. 2020.

SILVA, E. L.; MENEZES, E. M. **Metodologia da pesquisa e elaboração de dissertação**. Florianópolis, SC, Brasil: UFSC, 2005. Disponível em: <http://www.portaldeconhecimentos.org.br/index.php/por/Conteudo/Metodologia-da-pesquisa-e-elaboracao-de-dissertacao>. Acesso em: 7 abr. 2022.

SMARTSANTANDER. [S. l.], [s. d.]. Disponível em: <http://www.smartsantander.eu/>. Acesso em: 20 dez. 2018.

STOICA, I. *et al.* Chord: A scalable peer-to-peer lookup service for internet applications. *Em: CHORD*, 2001, New York, NY, USA. **Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '01**. New York, NY, USA: ACM Press, 2001. p. 149–160. Disponível em: <http://portal.acm.org/citation.cfm?doid=383059.383071>. Acesso em: 17 nov. 2019.

STOJMENOVIC, I. Machine-to-Machine Communications With In-Network Data Aggregation, Processing, and Actuation for Large-Scale Cyber-Physical Systems. **IEEE Internet of Things Journal**, [s. l.], v. 1, n. 2, p. 122–128, 2014.

STOREY, V. C.; SONG, I.-Y. Big data technologies and Management: What conceptual modeling can do. **Data & Knowledge Engineering**, [s. l.], v. 108, p. 50–67, 2017.

STRICKLAND, R. **Cassandra High Availability**. Birmingham, UK: Packt Publishing, 2014.

TRUONG, C. D.; ANH, D. T. A survey on time series motif discovery. **International Journal of Business Intelligence and Data Mining**, [s. l.], v. 15, n. 2, p. 204–227, 2019.

TRUONG, C.; RÖMER, K.; CHEN, K. Fuzzy-based sensor search in the Web of Things. *Em:*, 2012, Wuxi, China. **IEEE**. Wuxi, China: [s. n.], 2012. p. 8.

TURNER, V. **The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things**. [S. l.], 2014. Disponível em: <https://www.emc.com/leadership/digital-universe/2014iview/executive-summary.htm>. Acesso em: 12 set. 2018.

UCKELMANN, D.; HARRISON, M.; MICHAHELLES, F. (org.). **Architecting the Internet of Things**. Berlin Heidelberg: Springer-Verlag, 2011. *E-book*. Disponível em: [//www.springer.com/la/book/9783642191565](http://www.springer.com/la/book/9783642191565). Acesso em: 18 set. 2018.

VAMSI, P.; CHAHUAN, A. Machine Learning Based Hybrid Model for Fault Detection in Wireless Sensors Data. **EAI Endorsed Transactions on Scalable Information Systems**, [s. l.], v. 7, n. 24, p. 1–8, 2019.

VILLARS, R. L.; EASTWOOD, M.; OLOFSON, C. W. Big Data: What It Is and Why You Should Care. [s. l.], p. 14, 2011.

WALTERS, J.; ÖSTERBERG, P.; KANTER, T. The MediaSense Framework: Ranking Sensors in a Distributed Architecture. *Em: THE MEDIASENSE FRAMEWORK*, 2011, Athens, Greece. **Proceedings of Fifth International Conference on Digital Telecommunications, The Updated MediaSense Framework**. Athens, Greece: IEEE conference proceedings, 2011. p. 5. Disponível em: <http://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-13265>. Acesso em: 5 abr. 2019.

WANG, W. *et al.* A ranking method for sensor services based on estimation of service access cost. **Information Sciences**, [s. l.], v. 319, Energy Efficient Data, Services and Memory Management in Big Data Information Systems, p. 17, 2015.

WANG, H.; TAN, C. C.; LI, Q. Snoogle: A Search Engine for Pervasive Environments. **IEEE Transactions on Parallel and Distributed Systems**, [s. l.], v. 21, n. 8, p. 15, 2010.

WIDE, P. **Artificial human sensors: science and applications**. Singapore: Pan Stanford Publishing, 2012.

WITTEN, I. H.; FRANK, E.; HALL, M. A. **Data mining: practical machine learning tools and techniques**. 3rd eded. Burlington, MA, USA: Morgan Kaufmann, 2011. (Morgan Kaufmann series in data management systems).

XIAO, B. *et al.* Intelligent data-intensive IoT: A survey. *Em: ICC*, 2016, Chengdu, China. **Proceedings of 2nd IEEE International Conference on Computer and Communications (ICC)**. Chengdu, China: [s. n.], 2016. p. 2362–2368.

YEH, C. M. *et al.* Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. *Em: ICDM*, 2016, Barcelona, Spain. **Proceedings of 16th International Conference on Data Mining (ICDM)**. Barcelona, Spain: [s. n.], 2016. p. 1317–1322.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. **Computer Networks**, [s. l.], v. 52, n. 12, p. 2292–2330, 2008.

YINBIAO, S. **Internet of Things - Wireless Sensor Networks**: Market Strategy Board. Geneva, Switzerland: IEC - International Electrotechnical Commission, 2014. White Paper. Disponível em: <https://www.iec.ch/basecamp/internet-things-wireless-sensor-networks>. Acesso em: 10 fev. 2022.

YUEN, K. K. F.; WANG, W. Towards a ranking approach for sensor services using primitive cognitive network process. *Em: CYBER*, 2014, Hong Kong, China. **The 4th Annual IEEE International Conference on Cyber Technology in Automation, Control and Intelligent**. Hong Kong, China: [s. n.], 2014. p. 5.

ZAMRY, N. M.; ZAINAL, A.; RASSAM, M. Unsupervised Anomaly Detection for Unlabelled Wireless Sensor Networks Data. **International Journal of Advances in Soft Computing and its Applications**, [s. l.], v. 10, 2, p. 172–191, 2018.

ZHANG, Y. *et al.* A Survey on Emerging Computing Paradigms for Big Data. **Chinese Journal of Electronics**, [s. l.], v. 26, n. 1, p. 1–12, 2017.

ZHANG, P. *et al.* Low-Overhead and High-Precision Prediction Model for Content-Based Sensor Search in the Internet of Things. **IEEE Communications Letters**, [s. l.], v. 20, n. 4, p. 4, 2016.

ZIKOPOULOS, P. C. *et al.* **Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data**. [S. l.], 2012. Disponível em: <http://freecomputerbooks.com/Understanding-Big-Data.html>. Acesso em: 11 set. 2018.



## **APÊNDICE A – Publicações**

## A.1 Periódicos

**Título:** *FASTEN IIoT: An Open Real-Time Platform for Vertical, Horizontal and End-To-End Integration*

**Autores:** Felipe S. Costa; Silvia M. Nassar, Sergio Gusmeroli, Ralph Schultz, André G. S. Conceição, Miguel Xavier; Fabiano Hessel; Mario A. R. Dantas

**Periódico:** Sensors

**Ano:** 2020

**Estrato QUALIS:** A1

**Resumo:** *The Industry 4.0 paradigm, since its initial conception in Germany in 2011, has extended its scope and adoption to a broader set of technologies. It is being considered as the most vital mechanism in the production systems lifecycle. It is the key element in the digital transformation of manufacturing industry all over the world. This scenario imposes a set of major unprecedented challenges which require to be overcome. In order to enable integration in horizontal, vertical, and end-to-end formats, one of the most critical aspects of this digital transformation process consists of effectively coupling digital integrated service/products business models with additive manufacturing processes. This integration is based upon advanced AI-based tools for decentralized decision-making and for secure and trusted data sharing in the global value. This paper presents the FASTEN IIoT Platform, which targets to provide a flexible, configurable, and open solution. The platform acts as an interface between the shop floor and the industry 4.0 advanced applications and solutions. Examples of these efforts comprise management, forecasting, optimization, and simulation, by harmonizing the heterogeneous characteristics of the data sources involved while meeting real-time requirements.*

**Título:** *A Survey from Real-Time to Near Real-Time Applications in Fog Computing Environments*

**Autores:** Eliza Gomes; Felipe Costa; Carlos De Rolt; Patricia Plentz; Mario Dantas

**Periódico:** Telecom

**Ano:** 2021

**Estrato QUALIS:** -

**Resumo:** *In this article, we present a comprehensive survey on time-sensitive applications implemented in fog computing environments. The goal is to research what applications are*

being implemented in fog computing architectures and how the temporal requirements of these applications are being addressed. We also carried out a comprehensive analysis of the articles surveyed and separate them into categories, according to a pattern found in them. Our research is important for the area of real-time systems since the concept of systems that respond in real time has presented various understandings and concepts. This variability of concept has been due to the growing requirements for fast data communication and processing. Therefore, we present different concepts of real-time and near real-time systems found in the literature and currently accepted by the academic-scientific community. Finally, we conduct an analytical discussion of the characteristics and proposal of articles.

**Título:** *FOCUSeR: a Fog Online Context-aware Up-to-date Sensor Ranking method*

**Autores:** Felipe S. Costa; Silvia M. Nassar; Mario Dantas

**Periódico:** *Journal of Sensor and Actuator Networks*

**Ano:** 2022

**Estrato QUALIS:** A1

**Resumo:** *This research work presents FOCUSeR, a method for ranking sensors. The method uses the evaluation of data as a criterion for the ranking, allowing the identification of occurrences of failures in sensors and anomalies in environments. In order to meet the requirements inherent to WSNs, the proposed method was developed to run in a Fog Computing environment, using online learning and constant updating over time to avoid effects such as time drift. The generated ranking lists are managed through distributed hash tables. To provide reliability to the experimental results, a real experimental environment was developed. Also, using this developed testbed, a dataset with labels was created, to support the evaluation of the method. In addition, four other real datasets were used, three of which were labeled through artificial fault injection. These datasets were labeled in a related work that focused on injecting artificial faults. The experimental results obtained indicate that the proposed approach can provide reliability in the use of sensor data, using low computational resources and reducing latency in the sensor selection process. Precision rates are approximately 98% and accuracy rates greater than 94% across all datasets. In addition, the analyzes carried out show that the accuracy has an increasing rate as the number of samples also increases. Results obtained in the failure data recovery also reveal the feasibility of the proposal in this resource.*

**Em avaliação:**

**Título:** *A Sensor Ranking Approach for Edge-Fog-Cloud Environments*

**Autores:** Felipe Costa; Silvia M. Nassar; Mario Dantas

**Periódico:** SN Computer Science

**Ano:** 2022

**Estrato QUALIS: -**

**Resumo:** *This research paper presents a method for ranking sensors using the data produced by them. The method classifies the data, identifying the occurrence of failures in sensors and anomalies in the environments, aiming to maintain a ranking list of sensors. To generate the ranking list, overcoming the challenges implicit in this activity, the method adopts the theory of active perception as a basis. This approach divides the perception activity into levels that progressively add more sense to the information generated by the sensor, thus providing more reliability to the task of classifying the data generated by the sensors, without spending resources. This step allows you to create the ranking list in which the most reliable sensors will be at the top of the list. This list is managed through a distributed hash table to meet the distributed requirement of the Internet of Things (IoT) environment. The proposal was evaluated using four real data sets. The results of this research denote that the proposed approach can provide high reliability in the use of sensor data, using low computational resources and, thus, reducing latency in the processes of selection and use of sensors.*

## A.2 Conferências

**Título:** *A three level sensor ranking method based on active perception*

**Autores:** Felipe Costa; Silvia M. Nassar; Mario Dantas

**Evento:** IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society

**Ano:** 2019

**Estrato QUALIS:** A2

**Resumo:** *It can be said that the growth of cloud computing and Internet of Things (IoT), further drive the growth of data volume. This is due to large part to the development of a set of technologies, each allowing the development or creation of the other. OoT provides an infrastructure for physical and virtual things to communicate seamlessly in an intelligent environment. In the IoT paradigm, sensors are collecting and transmitting data that must be stored and processed. Thus, many studies have been dedicated to the task of searching, indexing and ranking sensors in the IoT environment. This paper presents a method for ranking sensors using the theory of active perception. The solution was evaluated using four data sets (real and synthetic). The results show that the proposed solution can provide reliability in the use of sensor data, with a minimum processing cost. To the best of our knowledge active perception has never been used for sensor ranking as it has never been used for the treatment of anomalies.*

**Título:** *GoAT: A Sensor Ranking Approach for IoT Environments*

**Autores:** Felipe Costa; Silvia M. Nassar; Mario Dantas

**Evento:** CLOSER 2021 - 11th International Conference on Cloud Computing and Services Science

**Ano:** 2021

**Estrato QUALIS:** A4

**Resumo:** *The data collected and transmitted by the sensors, in the Internet of Things environment, must be stored and processed in order to enable Smart Cities and Industry 4.0. However, due to the growth of number of devices, it becomes necessary to implement techniques to select most suitable sensors for each task. This approach is important to make possible to execute applications, where low latency requirements are present. Thus, several works were dedicated to the study on how to search, index, and rank sensors to overcome these challenges. A method, called GoAT, is presented in this paper to rank sensors based on*

*the theory of active perception. The solution was evaluated using four real datasets. Our results successfully indicate that the proposal solution can provide an interesting level of reliability of the utilization of sensor data. Furthermore, GoAT requires a low computational resource, and at the same time, reduces latency in the sensor selection process.*

**Título:** *Métis - An Approach Utilized as Differentiated Authenticity Tool in an IIoT Infrastructure*

**Autores:** Felipe S. Costa; Mario A. R. Dantas; José M. N. David; Regina M. M. B. Villela; Matheus S. Santos

**Evento:** 3PGCIC 2022 - *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*

**Ano:** 2022

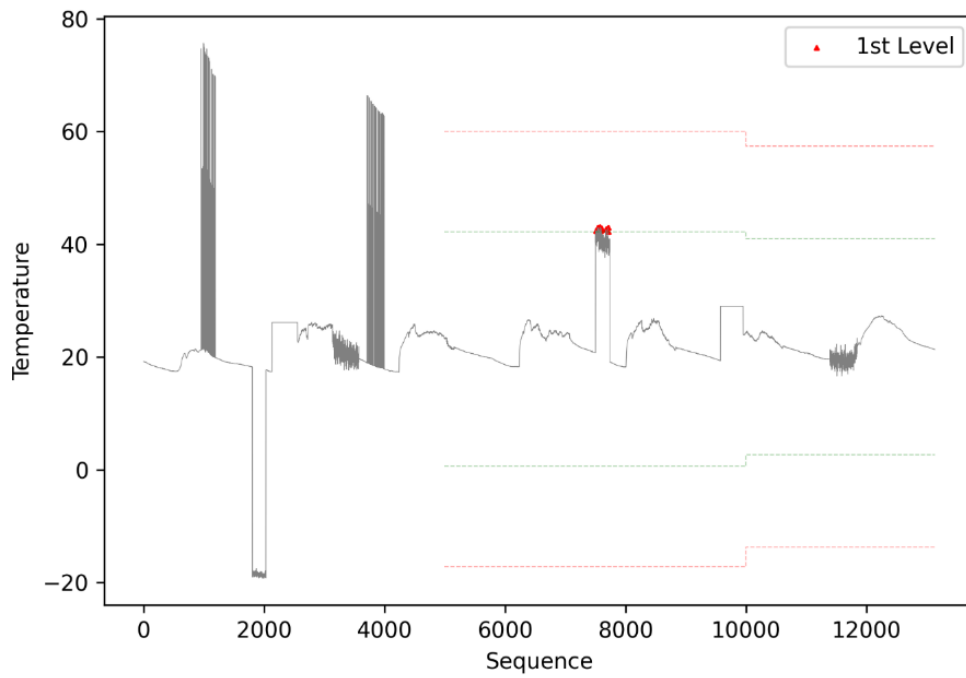
**Estrato QUALIS:** B1

**Resumo:** *Security in industrial environments is a growing concern with the integration of Industrial IoT (IIoT). The communication between devices, diverse users, and the volume of digital data transferred increase the vulnerability. Aiming to tackle this challenge, we developed studies related to the application of smart contracts with blockchain support to guarantee the integrity of identity authenticity of the digital data that travels within the Industrial IoT (IIoT) environment. Therefore, in this paper, we present the Métis proposal, which represents a differentiated authenticity approach and which was tested through simulations to provide a security landscape to a real industrial 4.0 project.*

**APÊNDICE B – Resultados produzidos pelo método**

Figura 25. Intel, saída do primeiro nível de avaliação.

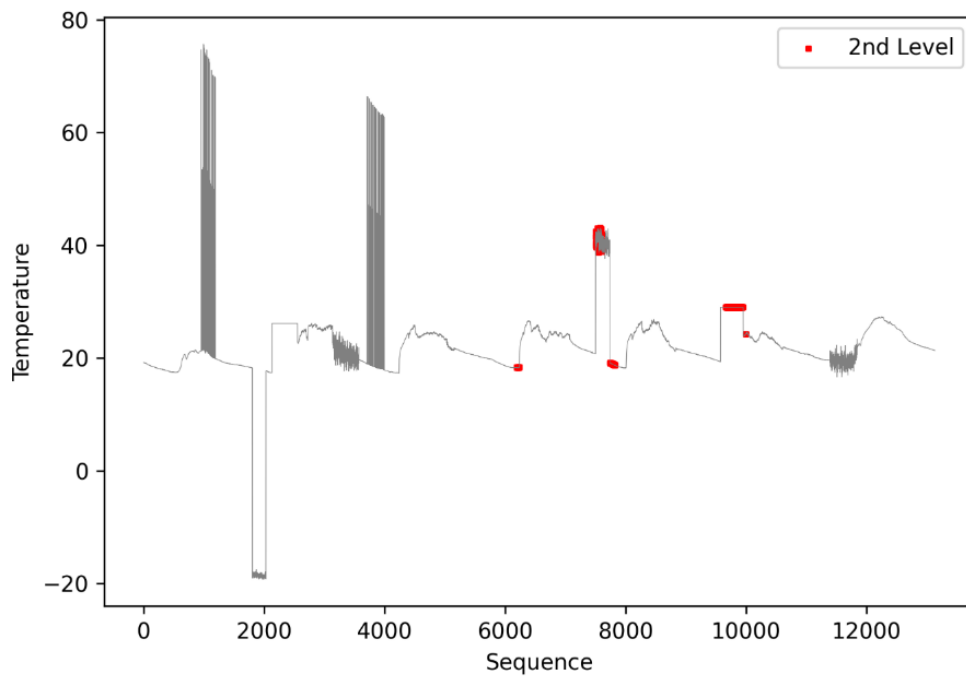
Intel - moteid\_1 - First Level



Fonte: Elaborada pelo autor (2022).

Figura 26. Intel, saída do segundo nível de avaliação.

Intel - moteid\_1 - Second Level

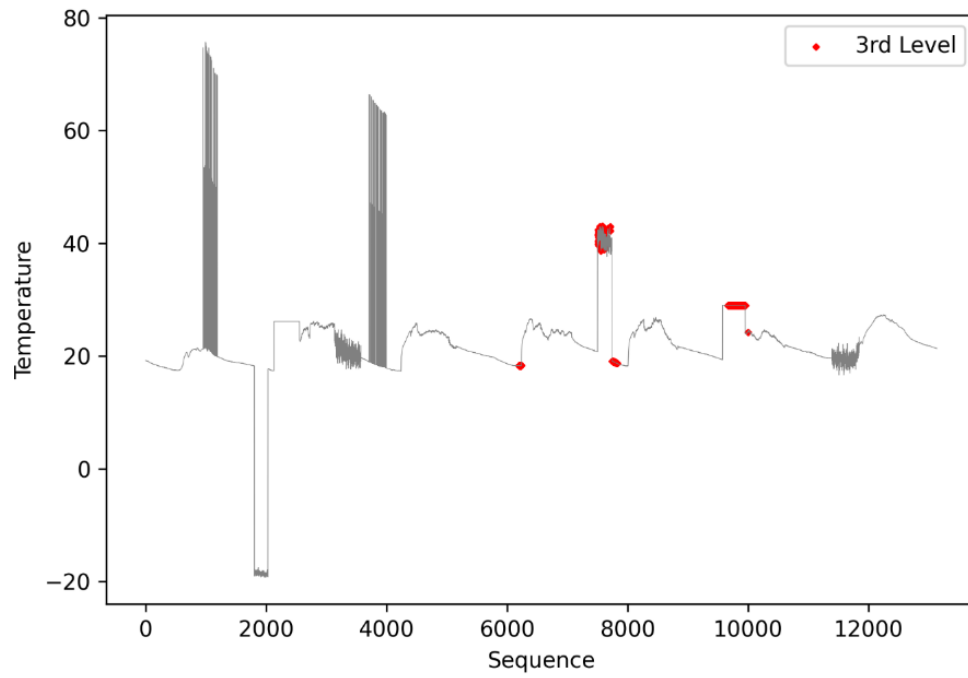


Fonte: Elaborada pelo autor (2022).



Figura 27. Intel, saída do terceiro nível de avaliação.

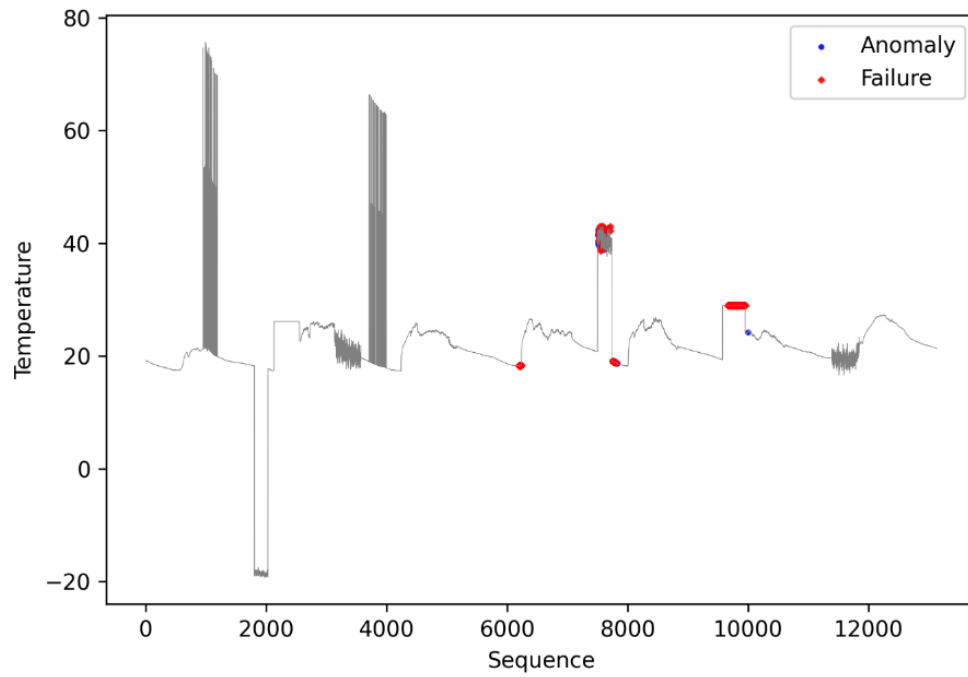
Intel - moteid\_1 - Third Level



Fonte: Elaborada pelo autor (2022).

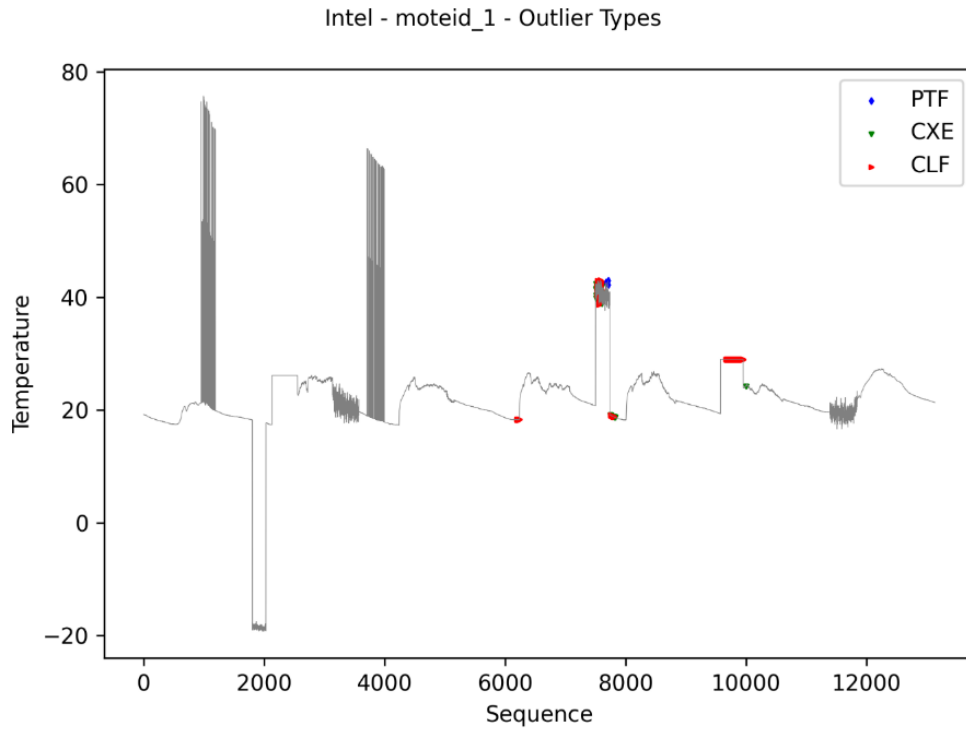
Figura 28. Intel, decisão final.

Intel - moteid\_1 - Decision



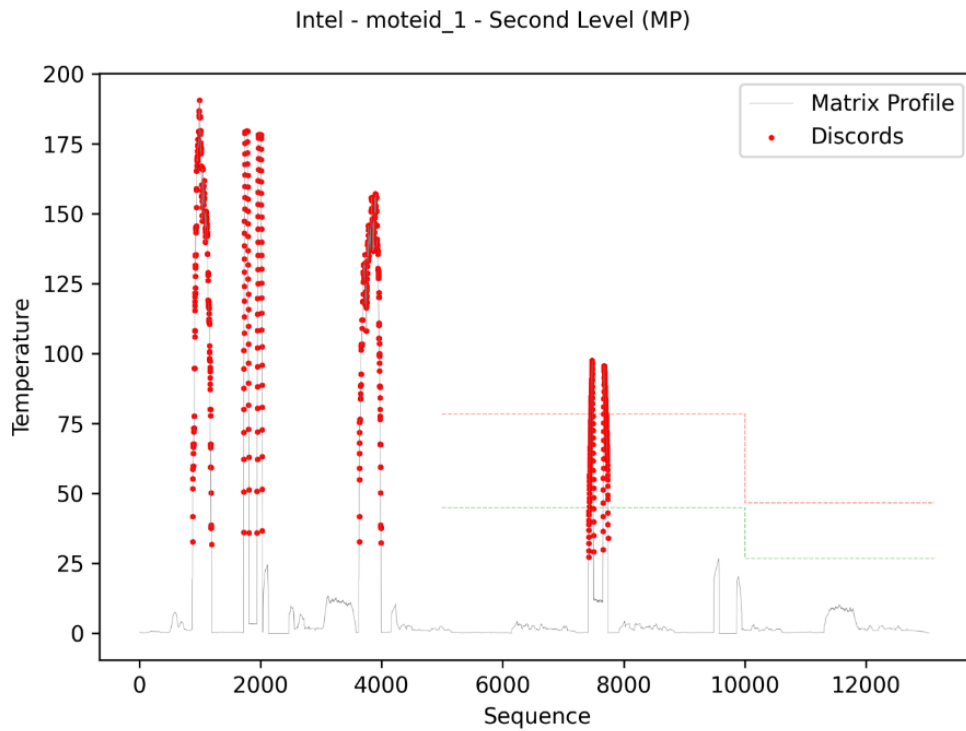
Fonte: Elaborada pelo autor (2022).

Figura 29. Intel, tipos de outlier.



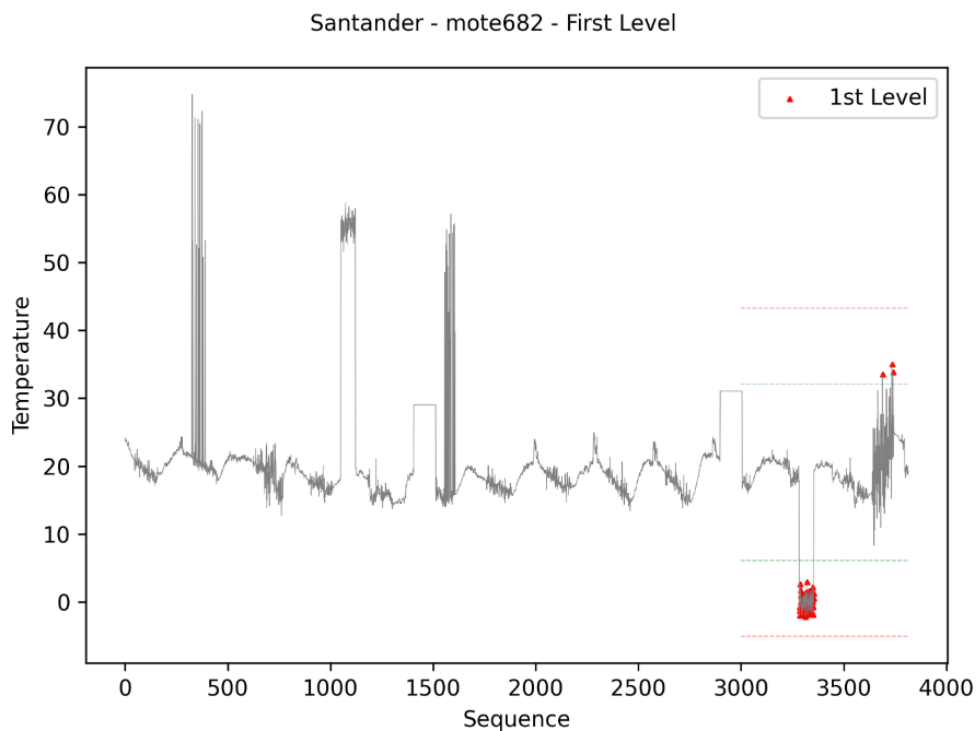
Fonte: Elaborada pelo autor (2022).

Figura 30. Intel, saída do algoritmo Matrix Profile.



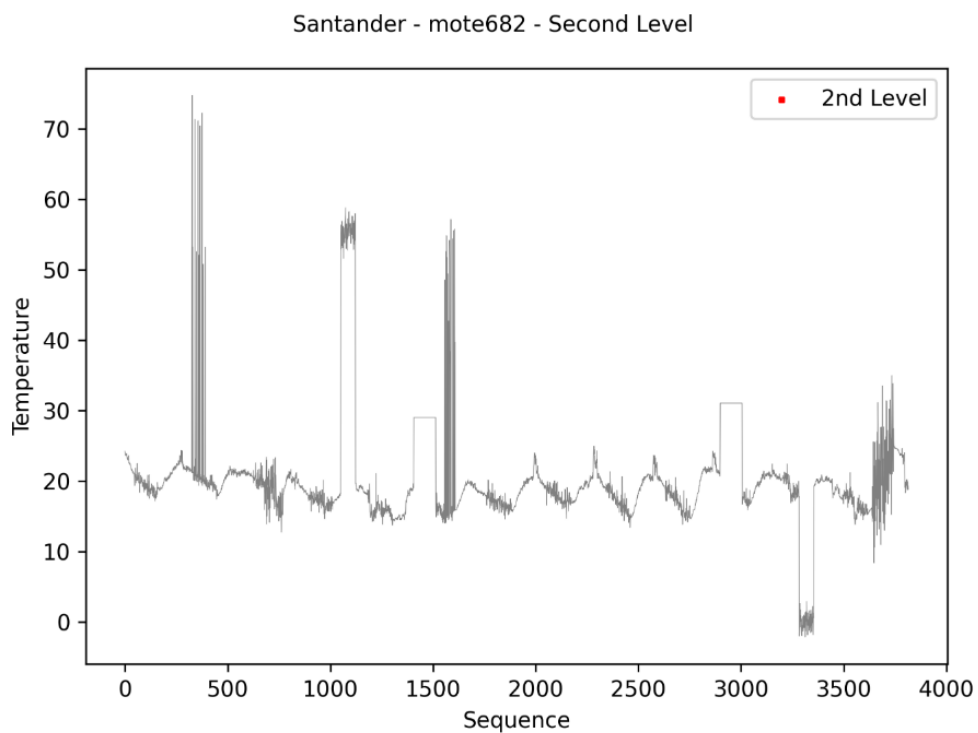
Fonte: Elaborada pelo autor (2022).

Figura 31. Santander, saída do primeiro nível de avaliação.



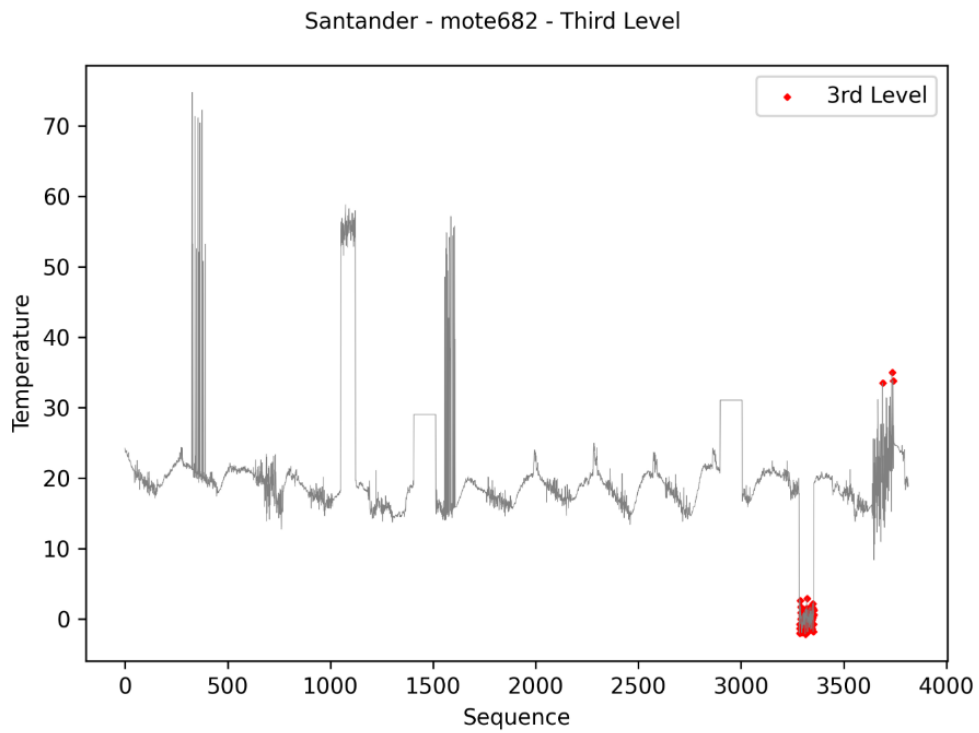
Fonte: Elaborada pelo autor (2022).

Figura 32. Santander, saída do segundo nível de avaliação.



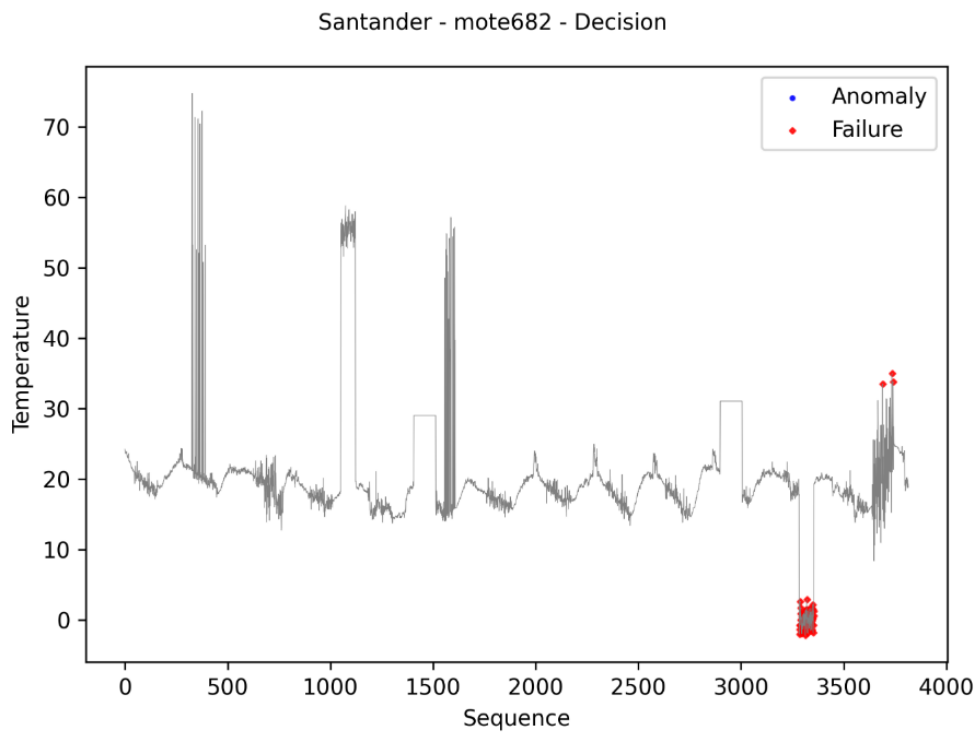
Fonte: Elaborada pelo autor (2022).

Figura 33. Santander, saída do terceiro nível de avaliação.



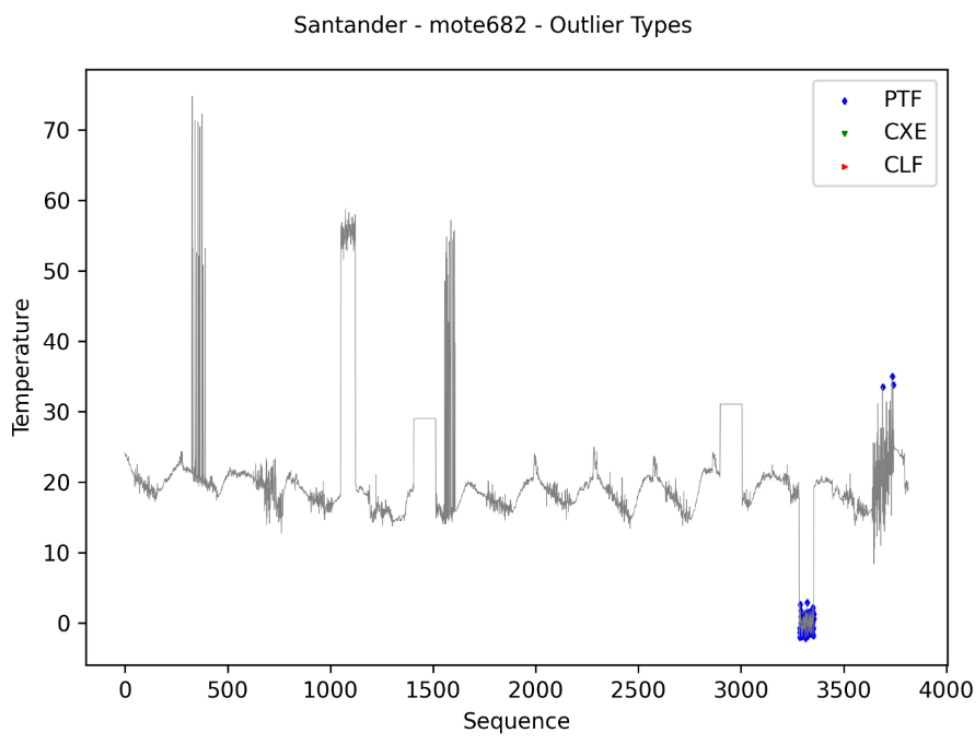
Fonte: Elaborada pelo autor (2022).

Figura 34. Santander, decisão final.



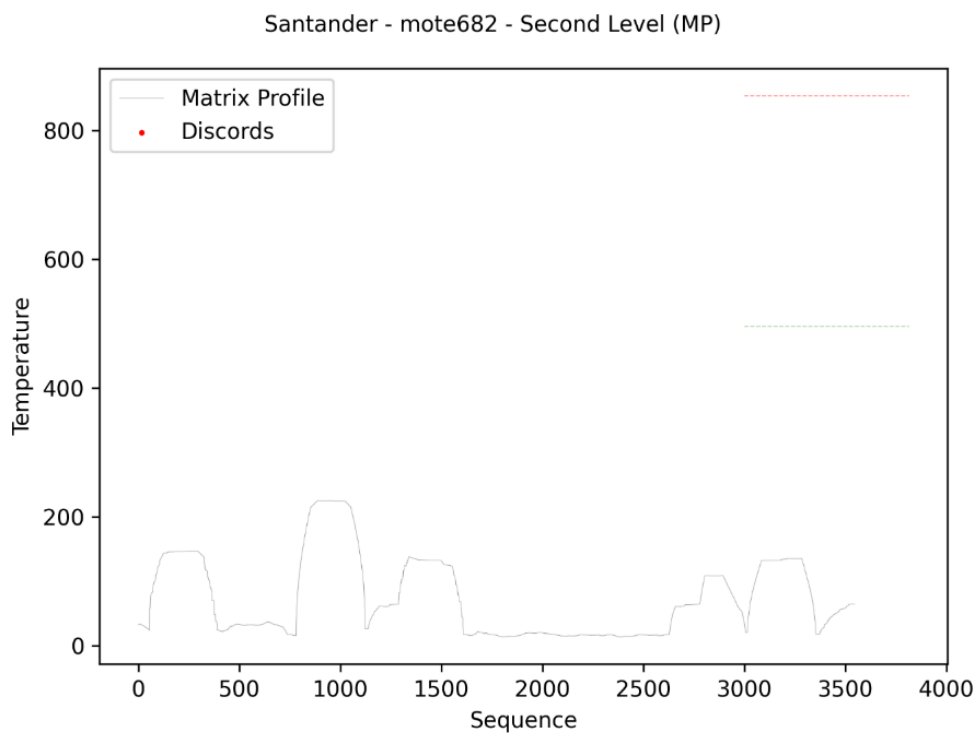
Fonte: Elaborada pelo autor (2022).

Figura 35. Santander, tipos de outlier.



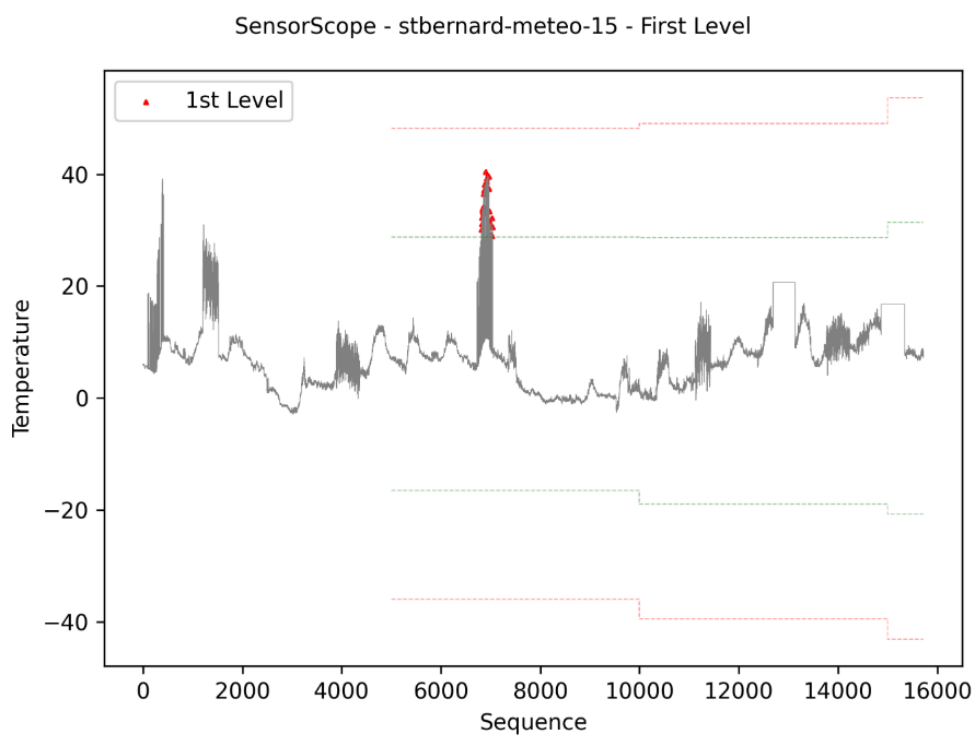
Fonte: Elaborada pelo autor (2022).

Figura 36. Santander, saída do algoritmo Matrix Profile.



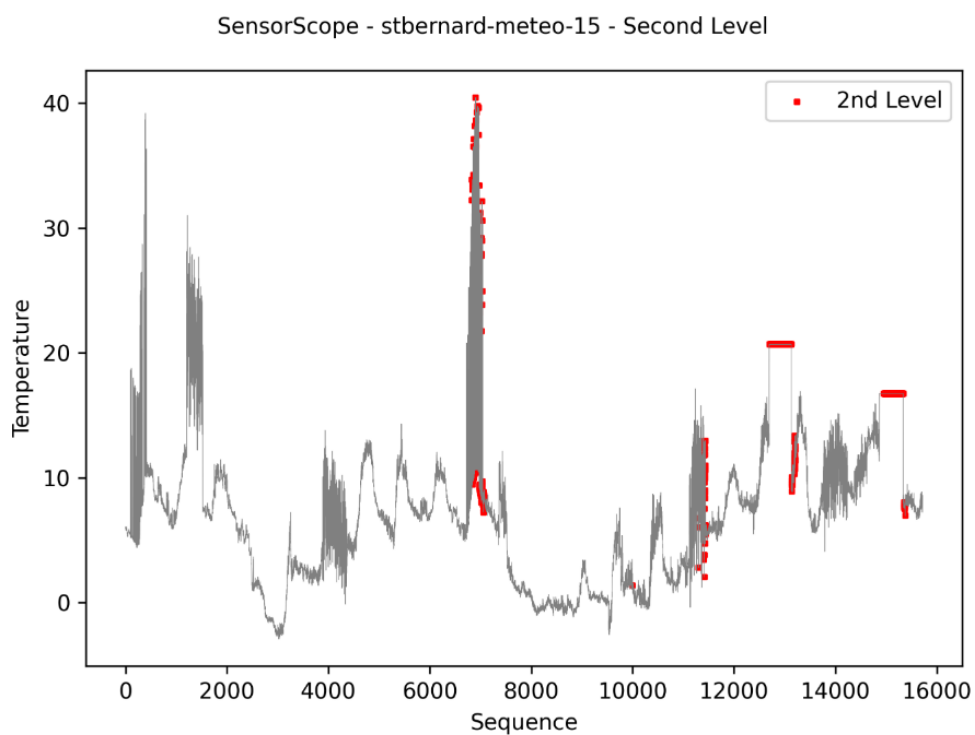
Fonte: Elaborada pelo autor (2022).

Figura 37. SensorScope, saída do primeiro nível de avaliação.



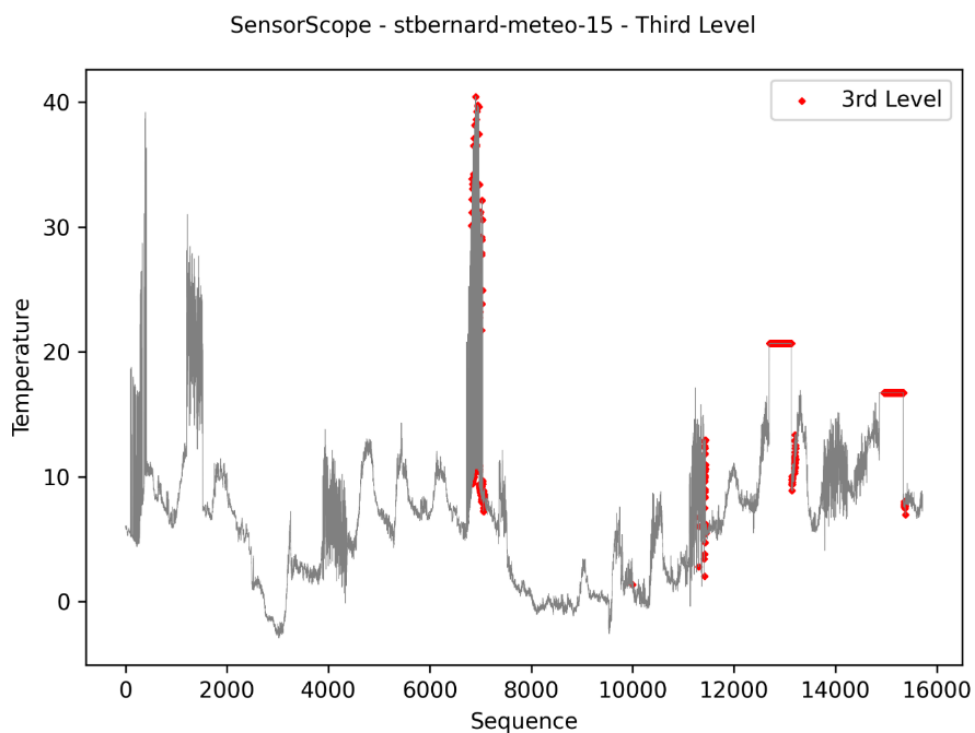
Fonte: Elaborada pelo autor (2022).

Figura 38. SensorScope, saída do segundo nível de avaliação.



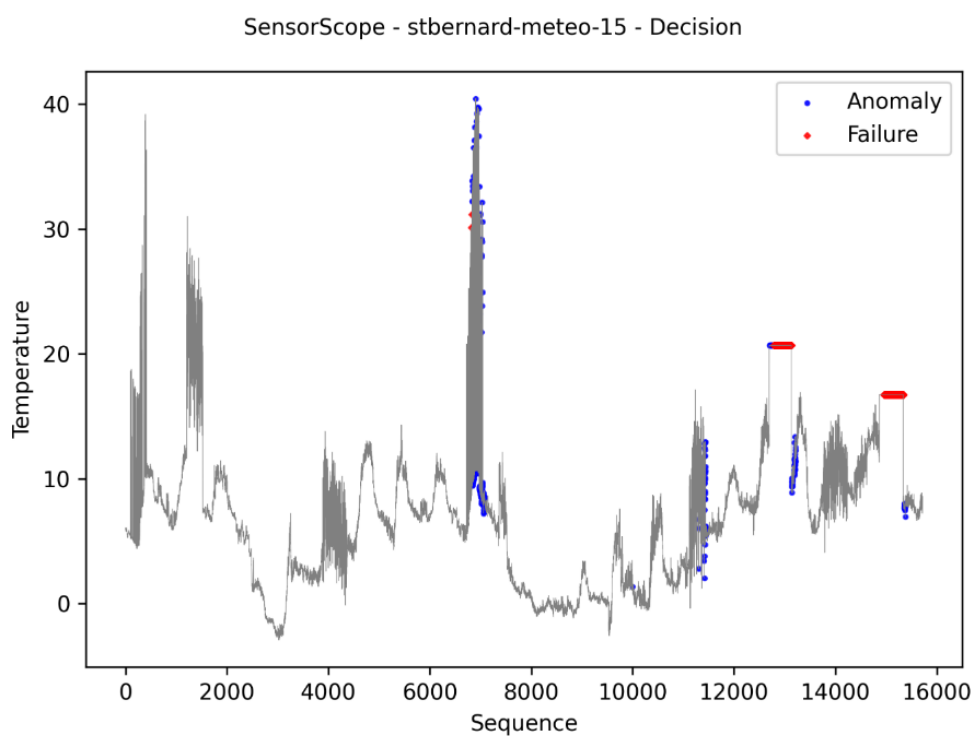
Fonte: Elaborada pelo autor (2022).

Figura 39. SensorScope, saída do terceiro nível de avaliação.



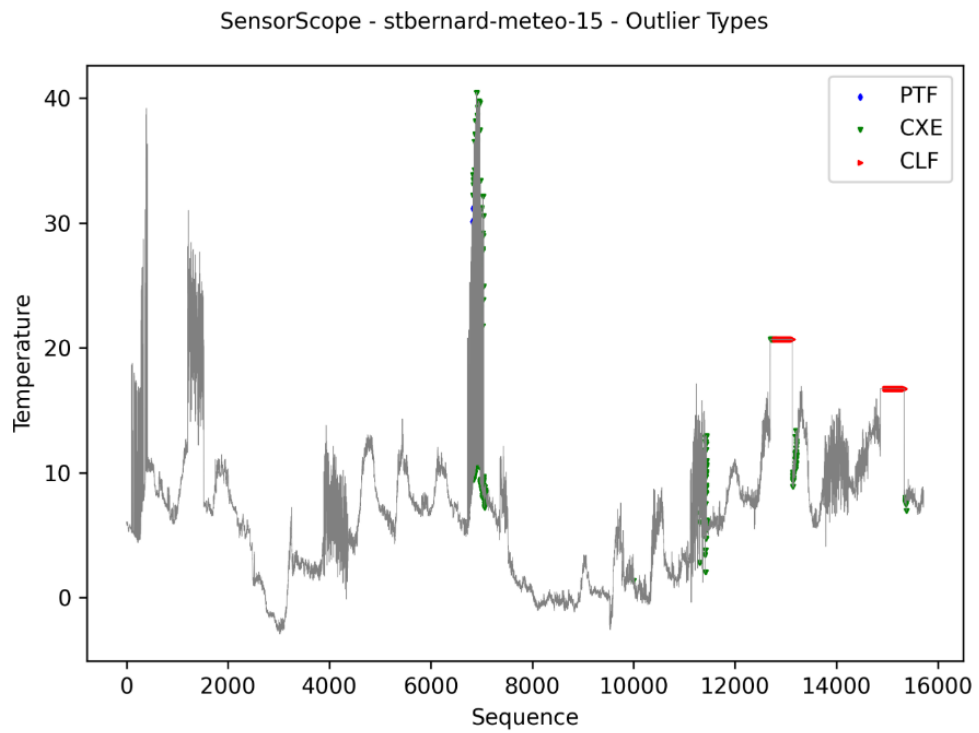
Fonte: Elaborada pelo autor (2022).

Figura 40. SensorScope, decisão final.



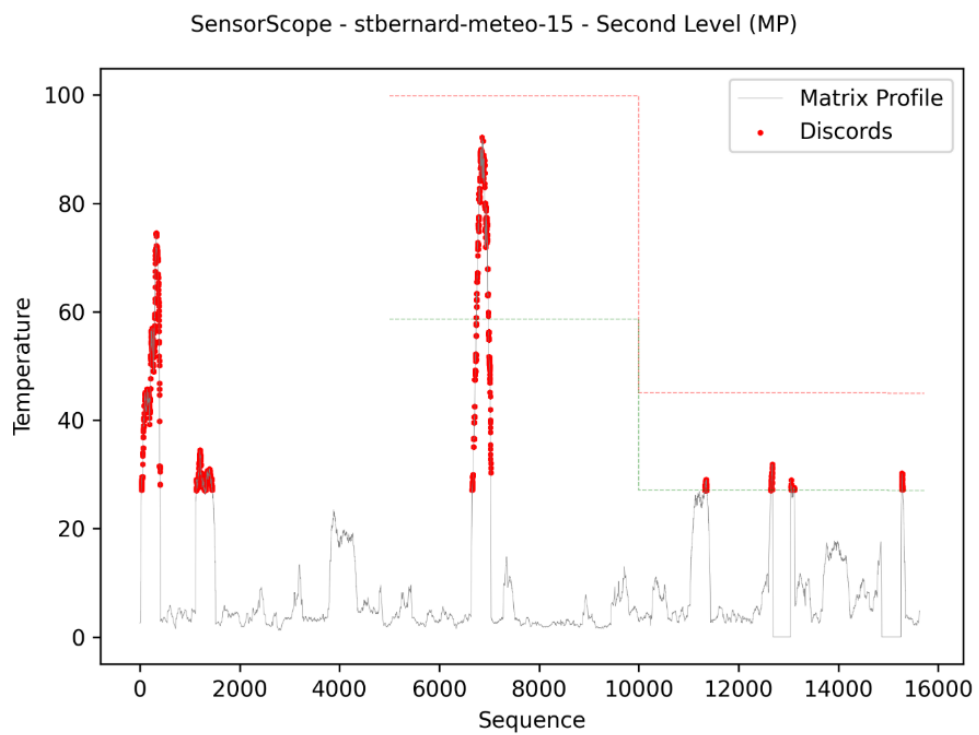
Fonte: Elaborada pelo autor (2022).

Figura 41. SensorScope, tipos de outlier.



Fonte: Elaborada pelo autor (2022).

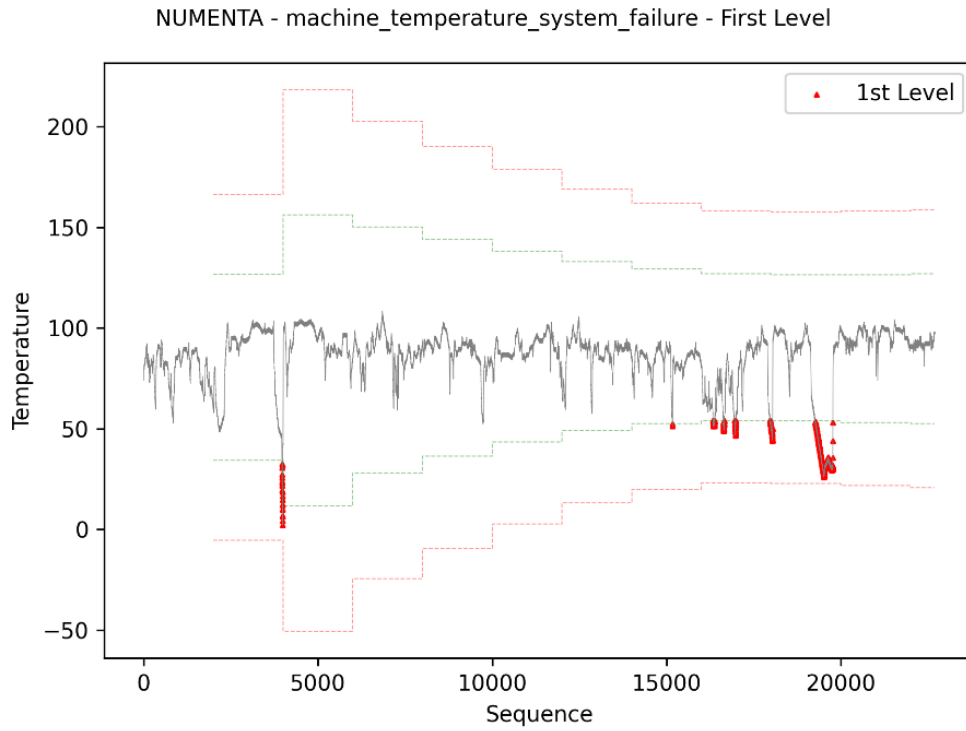
Figura 42. SensorScope, saída do algoritmo Matrix Profile.



Fonte: Elaborada pelo autor (2022).

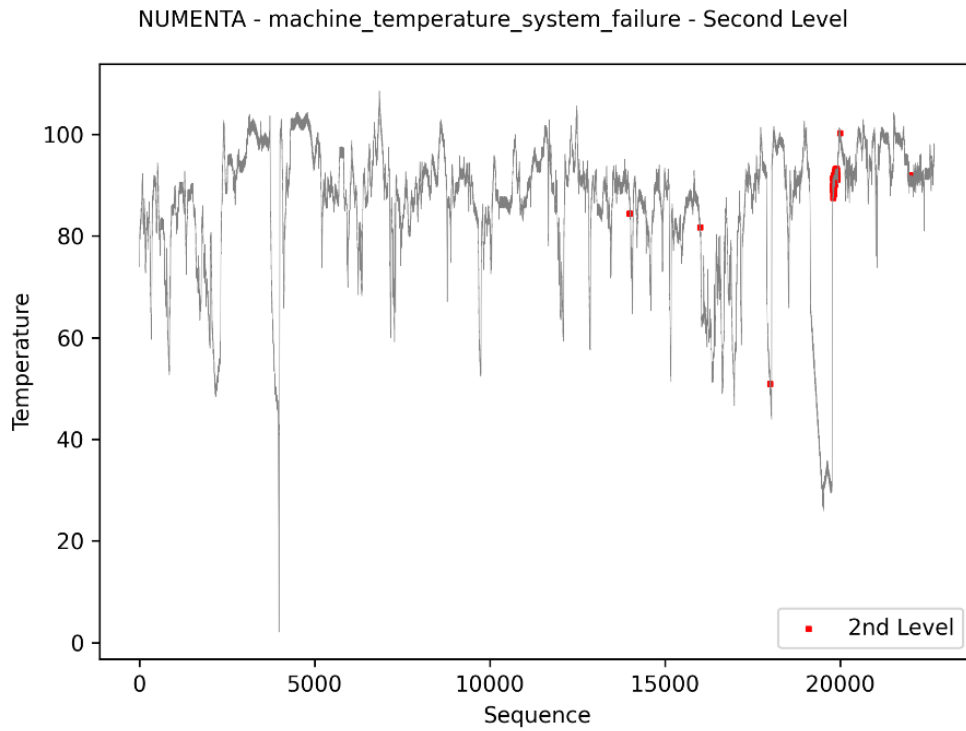


Figura 43. NUMENTA, saída do primeiro nível de avaliação.



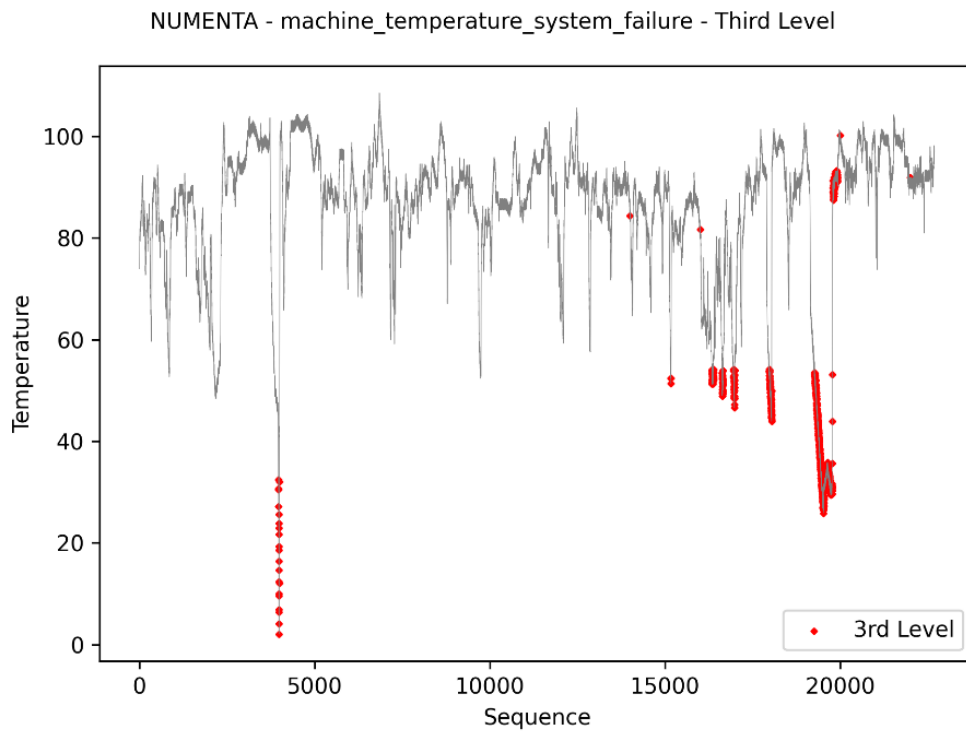
Fonte: Elaborada pelo autor (2022).

Figura 44. NUMENTA, saída do segundo nível de avaliação.



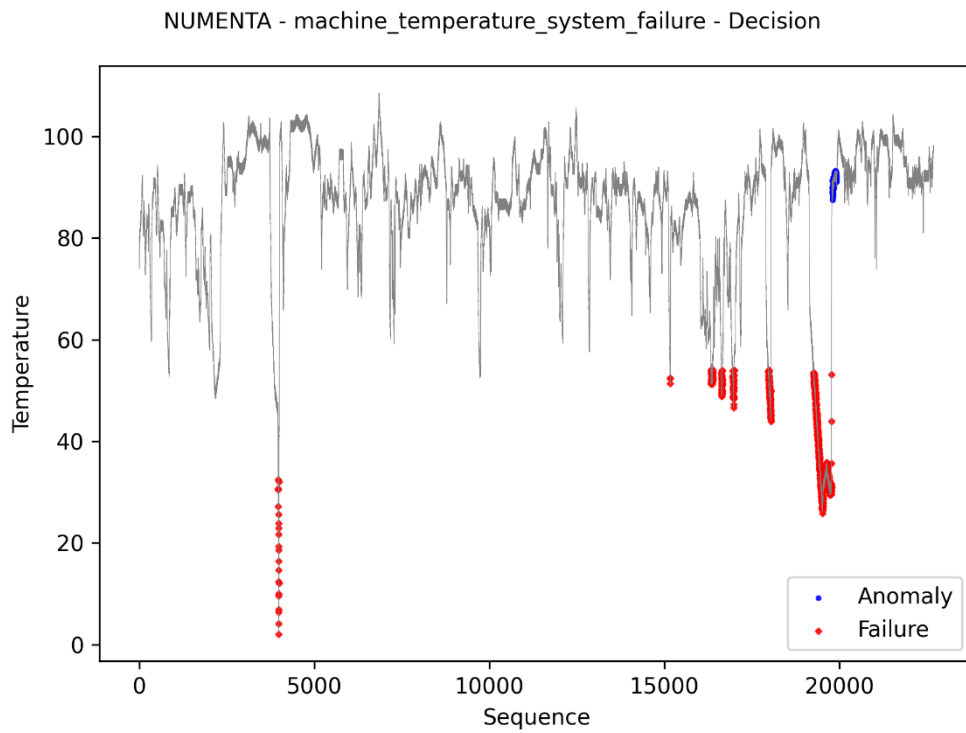
Fonte: Elaborada pelo autor (2022).

Figura 45. NUMENTA, saída do terceiro nível de avaliação.



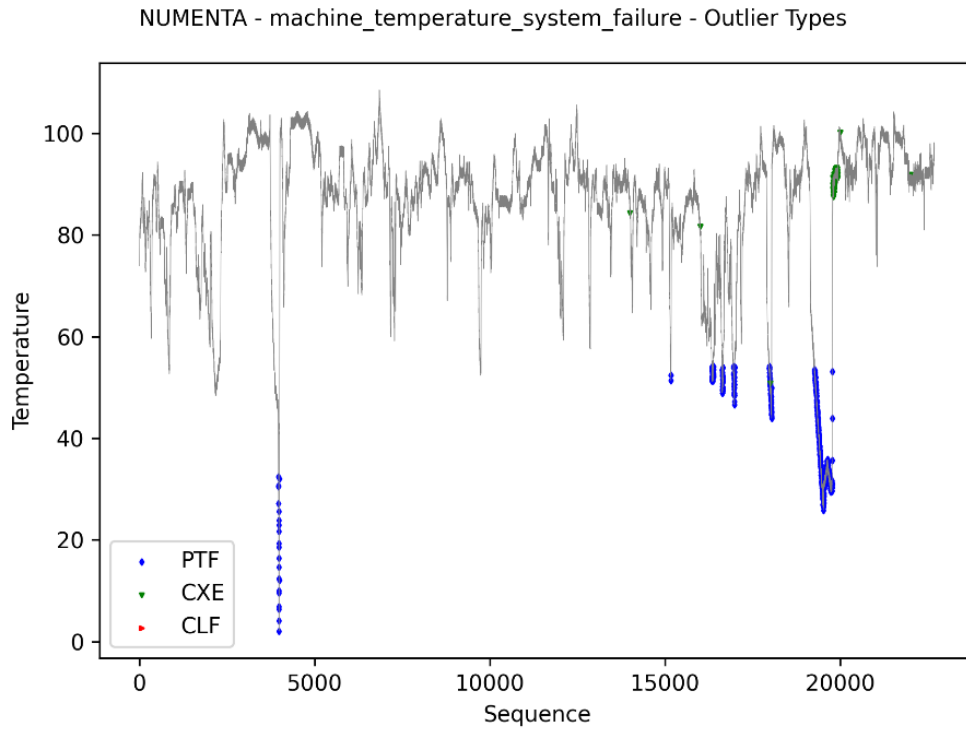
Fonte: Elaborada pelo autor (2022).

Figura 46. NUMENTA, decisão final.



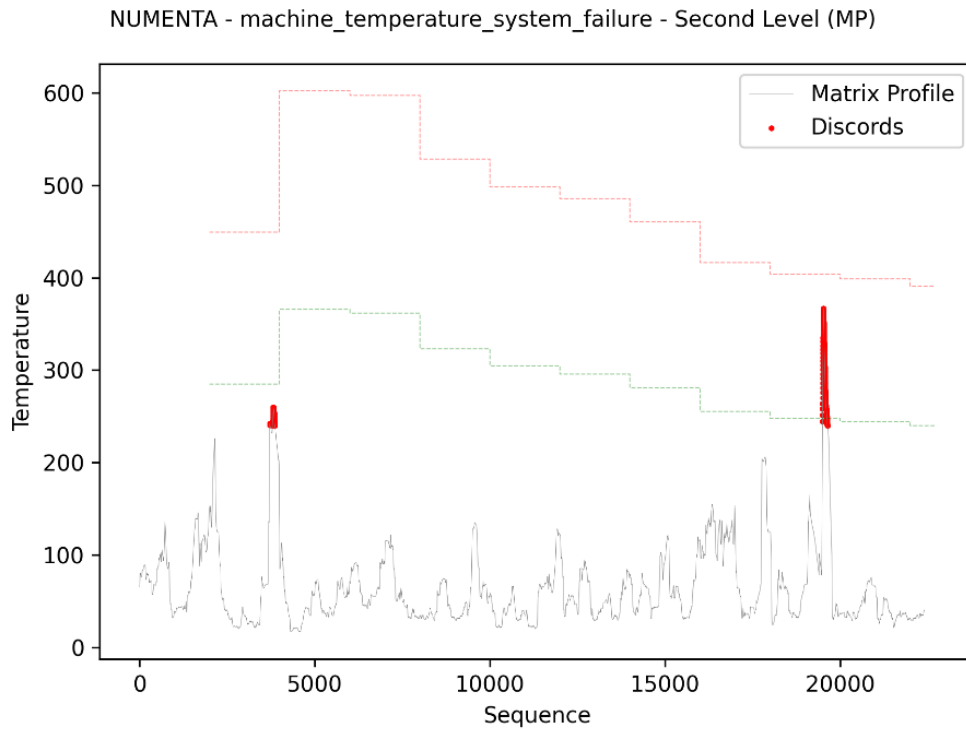
Fonte: Elaborada pelo autor (2022).

Figura 47. NUMENTA, tipos de outlier.



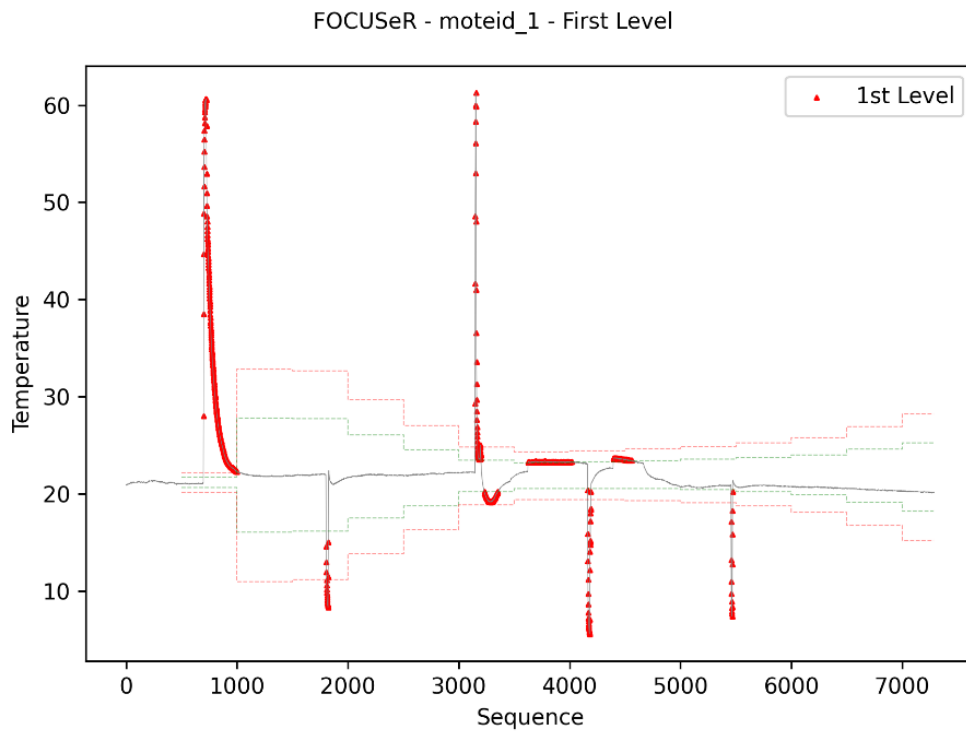
Fonte: Elaborada pelo autor (2022).

Figura 48. NUMENTA, saída do algoritmo Matrix Profile.



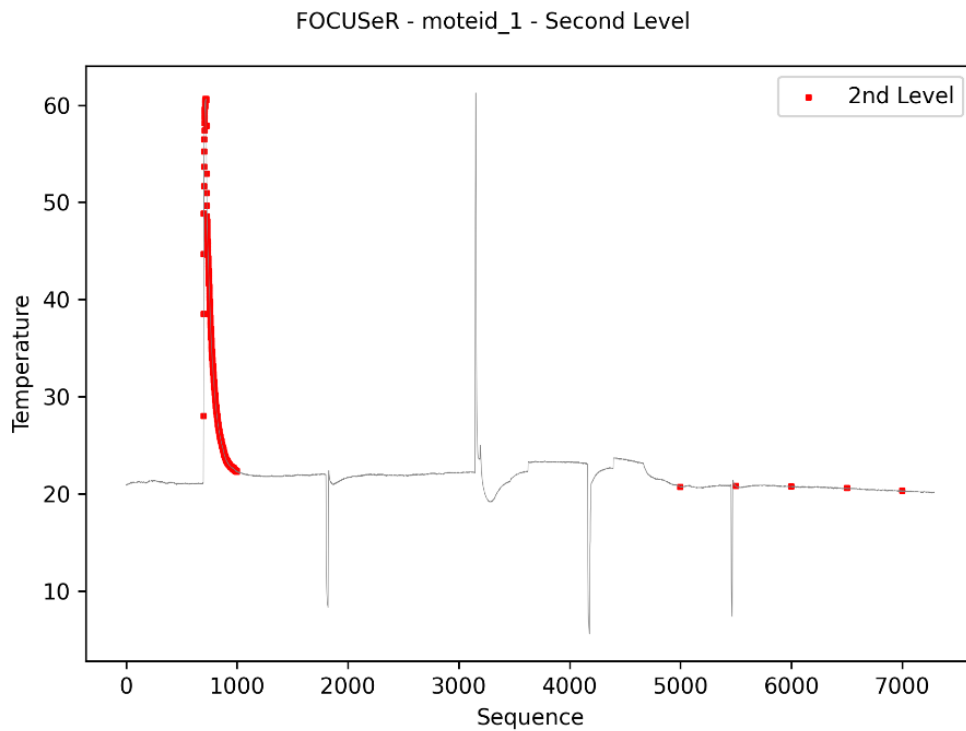
Fonte: Elaborada pelo autor (2022).

Figura 49. FOCUSeR, saída do primeiro nível de avaliação.



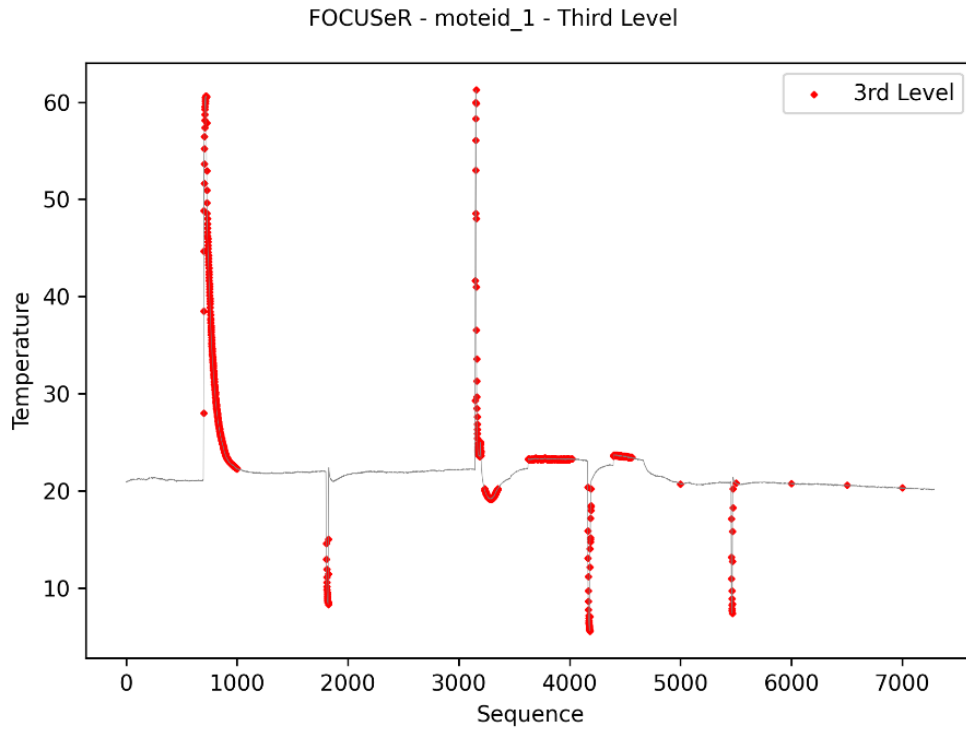
Fonte: Elaborada pelo autor (2022).

Figura 50. FOCUSeR, saída do segundo nível de avaliação.



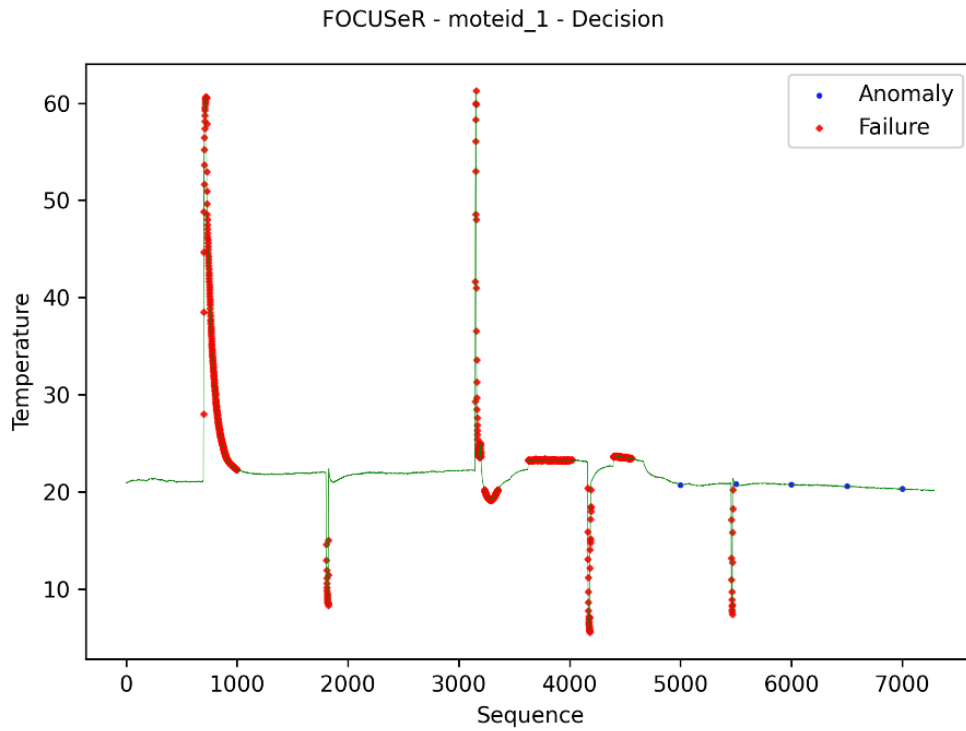
Fonte: Elaborada pelo autor (2022).

Figura 51. FOCUSeR, saída do terceiro nível de avaliação.



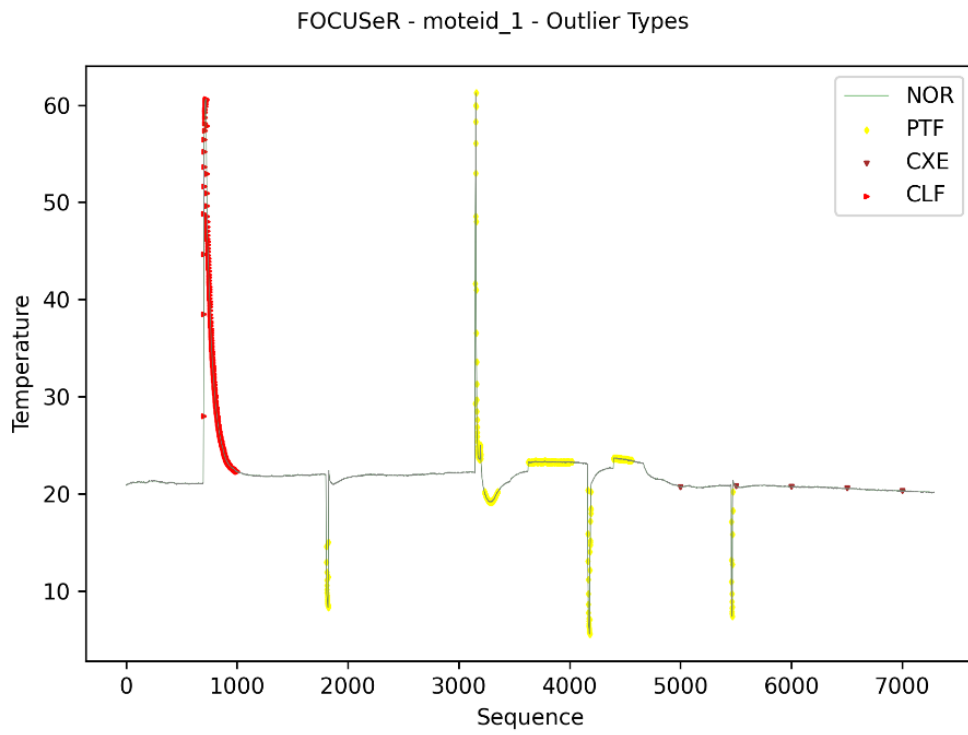
Fonte: Elaborada pelo autor (2022).

Figura 52. FOCUSeR, decisão final.



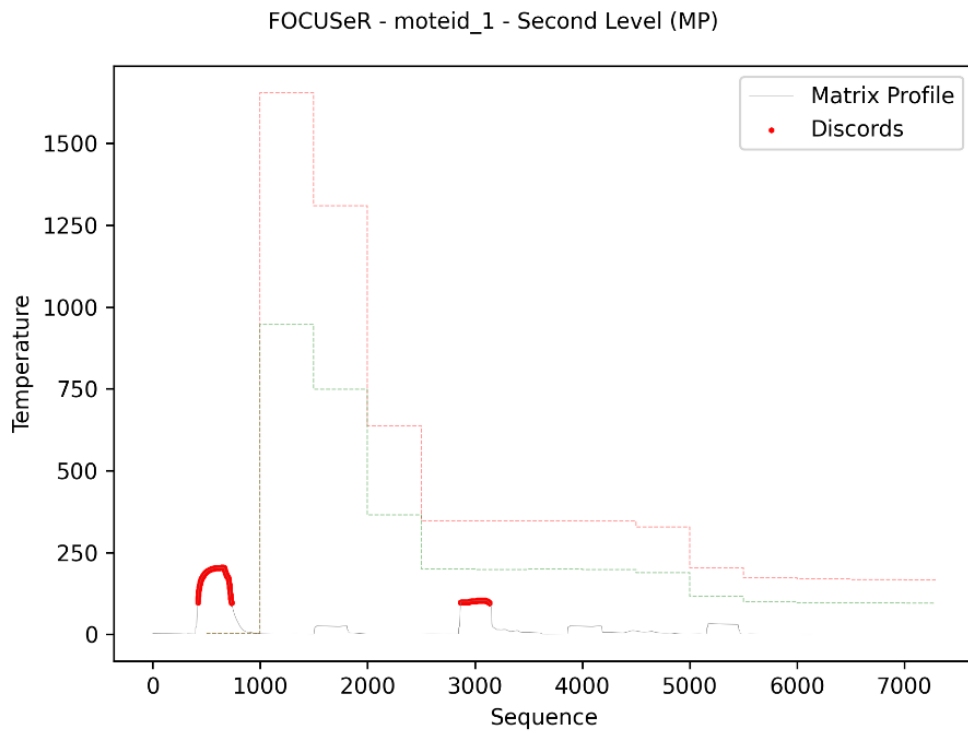
Fonte: Elaborada pelo autor (2022).

Figura 53. FOCUSeR, tipos de outlier.



Fonte: Elaborada pelo autor (2022).

Figura 54. FOCUSeR, saída do algoritmo Matrix Profile.



Fonte: Elaborada pelo autor (2022).

**APÊNDICE C – Teste de Sensibilidade de Parâmetros**

Tabela 5. Teste de sensibilidade de parâmetros – Intel.

<b>Conjunto de Dados</b>	<b><i>Burn-in</i></b>	<b>Subsequência</b>	<b>Acurácia</b>	<b>Precisão</b>	<b><i>Recall</i></b>	<b><i>F1 Score</i></b>
Intel	4000	270	89,47	89,36	99,99	94,37
Intel	5000	210	89,12	88,88	99,99	94,11
Intel	5000	270	88,38	88,22	99,99	93,73
Intel	3000	270	87,61	87,24	99,98	93,17
Intel	4000	210	90,12	89,95	99,98	94,70
Intel	3000	210	88,19	87,77	99,97	93,47
Intel	4000	150	90,74	90,56	99,94	95,02
Intel	2000	210	87,91	87,22	99,88	93,12
Intel	2000	270	87,37	86,72	99,87	92,83
Intel	2000	150	88,75	88,82	98,68	93,49
Intel	5000	150	90,19	90,90	98,59	94,59
Intel	5000	90	90,29	91,17	98,35	94,62
Intel	4000	30	90,88	92,43	97,68	94,98
Intel	5000	30	94,59	96,16	97,68	96,91
Intel	4000	90	90,50	92,07	97,66	94,78
Intel	3000	150	87,28	88,65	97,45	92,84
Intel	1000	270	85,13	87,25	95,72	91,29
Intel	1000	210	85,22	87,68	95,22	91,30
Intel	3000	90	85,77	88,94	95,00	91,87
Intel	2000	30	91,49	95,03	94,55	94,79
Intel	3000	30	94,19	98,56	94,51	96,49
Intel	2000	90	85,35	88,53	94,33	91,34
Intel	1000	90	84,67	89,18	92,37	90,75
Intel	1000	30	91,69	97,72	91,93	94,74
Intel	1000	150	84,08	89,10	91,66	90,36

Fonte: Elaborada pelo autor (2022).



Tabela 6. Teste de sensibilidade de parâmetros – SensorScope.

<b>Conjunto de Dados</b>	<b><i>Burn-in</i></b>	<b>Subsequência</b>	<b>Acurácia</b>	<b>Precisão</b>	<b><i>Recall</i></b>	<b><i>F1 Score</i></b>
SensorScope	4000	270	85,22	84,72	99,98	91,72
SensorScope	2000	270	86,66	86,27	99,97	92,62
SensorScope	1000	270	86,12	85,94	99,47	92,22
SensorScope	4000	210	85,37	85,56	98,80	91,70
SensorScope	1000	210	86,37	86,66	98,70	92,29
SensorScope	2000	210	86,47	86,94	98,66	92,43
SensorScope	3000	270	84,40	85,00	98,44	91,22
SensorScope	1000	150	86,93	87,41	98,35	92,56
SensorScope	2000	150	87,04	87,70	98,29	92,70
SensorScope	4000	150	85,71	86,44	97,90	91,81
SensorScope	3000	210	84,87	85,77	97,87	91,42
SensorScope	3000	150	85,75	86,63	97,80	91,87
SensorScope	5000	270	85,42	86,57	97,70	91,80
SensorScope	5000	150	87,69	88,74	97,65	92,98
SensorScope	1000	90	87,68	88,64	97,60	92,90
SensorScope	5000	210	86,46	87,64	97,54	92,33
SensorScope	2000	90	87,80	88,98	97,50	93,04
SensorScope	5000	90	89,27	90,75	97,04	93,79
SensorScope	1000	30	92,54	94,15	97,00	95,55
SensorScope	2000	30	92,87	94,75	96,85	95,79
SensorScope	3000	90	86,23	88,03	96,39	92,02
SensorScope	3000	30	94,74	97,25	96,34	96,79
SensorScope	4000	90	86,20	87,97	96,31	91,95
SensorScope	4000	30	95,50	98,29	96,17	97,22
SensorScope	5000	30	95,73	99,00	95,86	97,40

Fonte: Elaborada pelo autor (2022).

Tabela 7. Teste de sensibilidade de parâmetros – SmartSantander.

<b>Conjunto de Dados</b>	<b><i>Burn-in</i></b>	<b>Subsequência</b>	<b>Acurácia</b>	<b>Precisão</b>	<b><i>Recall</i></b>	<b><i>F1 Score</i></b>
Santander	2000	30	94,83	97,06	96,80	96,93
Santander	1000	30	92,47	98,10	92,72	95,33
Santander	3000	30	91,19	97,14	91,31	94,14
Santander	1000	150	82,68	90,74	88,10	89,40
Santander	3000	90	77,11	83,48	87,84	85,60
Santander	1000	90	82,36	91,29	87,03	89,11
Santander	2000	150	50,00	84,37	50,00	62,79
Santander	2000	210	50,00	84,37	50,00	62,79
Santander	2000	270	50,00	84,37	50,00	62,79
Santander	2000	90	50,00	84,37	50,00	62,79
Santander	1000	210	50,00	82,93	50,00	62,38
Santander	1000	270	50,00	82,93	50,00	62,38
Santander	3000	150	50,00	77,48	50,00	60,78
Santander	3000	210	50,00	77,48	50,00	60,78
Santander	3000	270	50,00	77,48	50,00	60,78

Fonte: Elaborada pelo autor (2022).

**APÊNDICE D – Referências utilizadas na criação da Figura 2.**

<b>Termo</b>	<b>Referência</b>
<i>Relational Database</i>	(F. CODD, 1970)
<i>RFID</i>	(CHARITH PERERA, 2017)
<i>WSN</i>	(JINDAL; COLLEGE, 2018)
<i>TCP/IP</i>	(HAUBEN, 1998)
<i>Internet</i>	(LEINER <i>et al.</i> , 2009)
<i>(BI) Business Intelligence</i>	(NYLUND, 1999)
<i>Mobile Internet</i>	(PATIL; R.R.KARHE; AHER, 2012)
<i>Context Aware</i>	(SCHILIT; THEIMER, 1994)
<i>Wireless Network</i>	(MCCANN; ASHLEY, 2018)
<i>Big Data</i>	(COX; ELLSWORTH, 1997)
<i>Edge Computing</i>	(NOBLE <i>et al.</i> , 1997)
<i>Cloud Computing</i>	(KAUFMAN, 2009)
<i>IoT (Internet of Things)</i>	(UCKELMANN; HARRISON; MICHAHELLES, 2011)
<i>Quality of Context</i>	(BUCHHOLZ; SCHIFFERS, 2003)
<i>IIoT (Industrial Internet of Things)</i>	(MUNIRATHINAM, 2020)
<i>Sensor Ranking</i>	(ELAHI <i>et al.</i> , 2009)
<i>Fog Computing</i>	(BONOMI <i>et al.</i> , 2012)

Fonte: Elaborada pelo autor (2022).