

Universidade do Minho

Escola de Engenharia

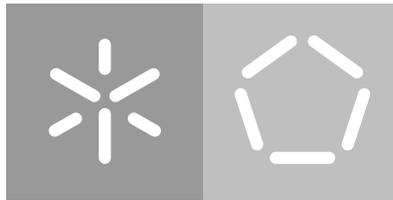
Departamento de Informática

Filipe Vieira Fortunato

Eficiência energética em Redes de Sensores Sem Fios

**Medição adaptativa num Sistema de
Rega Inteligente usando o CupCarbon**

February 2021



Universidade do Minho

Escola de Engenharia

Departamento de Informática

Filipe Vieira Fortunato

Eficiência energética em Redes de Sensores Sem Fios

**Medição adaptativa num Sistema de
Rega Inteligente usando o CupCarbon**

Master dissertation

Integrated Master's in Informatics Engineering

Dissertation supervised by

Paulo Manuel Martins de Carvalho

Maria Solange P. F. Rito Lima

February 2021

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença abaixo indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



**Creative Commons Atribuição-NãoComercial-Compartilhalgal 4.0 Internacional
CC BY-NC-SA 4.0**

<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.pt>

Despacho RT - 31 /2019 - Anexo 4**Declaração a incluir na Tese de Doutoramento (ou equivalente) ou no trabalho de Mestrado**

(Escolher uma das versões, consoante a língua de redação do trabalho)

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

ABSTRACT

Today, there are many cities that offer to citizens smart solutions to make their daily lives easier so that the available resources can be better managed and the global quality of life improved. These solutions generally rely on a variety of Wireless Sensor Networks (WSN), which are applied in a wide range of scenarios. Most of these solutions work without human intervention, therefore, there has been a lot of interest in increasing the longevity of these sensor networks.

In this context, the main purpose of this work is to study and optimize an adaptive, energy-aware sensing algorithm for WSNs, e-LiteSense [11], which is an algorithm capable of auto-regulate how data is sensed, adjusting it to each applicational scenario. This work, resorts to a simulation scenario representing a case in real life, namely, an Intelligent Irrigation system. In this study, CupCarbon is used as a simulation tool to implement WSN-based system and the e-LiteSense algorithm. The aim is to adapt the number of measurement events of environmental parameters so that the energy consumption of the different nodes of the network can be reduced while maintaining the correct evaluation of the measurement data and increasing the lifetime of the sensor network. The versatility of the algorithm in relation to its effectiveness and ability to self-configure in different types of sensing scenarios is also evaluated.

Key Words: Wireless Sensor Networks; Adaptive sensing; Energy-aware sensing; Intelligent Irrigation System; CupCarbon; Energy Consumption.

RESUMO

Atualmente, são várias as cidades que disponibilizam aos seus cidadãos soluções inteligentes para facilitar o seu dia a dia, de maneira a haver uma melhor gestão dos recursos existentes e uma melhoria da qualidade de vida proporcionada aos seus habitantes. Muitas destas soluções recorrem geralmente a uma série de redes de sensores sem fios (RSSF), sendo que estas são aplicadas a uma grande variedade de cenários. A maioria destes soluções funcionam sem intervenção humana, havendo assim cada vez mais interesse em aumentar a longevidade destas redes de sensores.

Neste contexto, o principal objetivo deste trabalho é estudar e otimizar uma solução de *sensing* adaptativo e de eficiência energética para RSSF's, o e-LiteSense [11]. Este algoritmo é capaz de regular automaticamente a forma como os dados são detetados, adaptando-se a cada cenário aplicacional. Este trabalho recorre a um cenário de simulação que representa um caso da vida real, nomeadamente, um sistema de Rega Inteligente. Neste estudo, a ferramenta de simulação CupCarbon é usada para implementar esse sistema, baseado em RSSF, e o algoritmo e-LiteSense. O objetivo é adaptar o número de eventos de medição de parâmetros ambientais para que o consumo energético dos diferentes nós da rede possa ser reduzido, mantendo a avaliação correta dos dados de medição e aumentando a vida útil da rede de sensores. A versatilidade do algoritmo relativamente à sua eficácia e capacidade de auto-configuração em diferentes tipos de sensing cenários também será avaliada.

Palavras Chave: Redes de Sensores Sem Fios; Sensing adaptativo; Eficiência energética; Sistema de Rega Inteligente; CupCarbon; Consumo energético.

CONTEÚDO

Abstract	iv
Resumo	v
Conteúdo	v
Lista de Figuras	viii
Lista de Tabelas	xi
Lista de Acrónimos	xii
1 INTRODUÇÃO	1
1.1 Enquadramento	1
1.2 Motivação e Objetivos	2
1.3 Estrutura do Documento	3
2 TRABALHO RELACIONADO	4
2.1 Projetos Cidades Inteligentes	4
2.2 Eficiência Energética e Medição Adaptativa em RSSF	6
2.3 Plataforma de Simulação CupCarbon	8
2.4 Resumo	12
3 E-LITESENSE: PRINCÍPIOS, ARQUITETURA E ALGORITMOS	13
3.1 Princípios de Conceção primários	13
3.2 Arquitetura	15
3.3 Algoritmos de <i>sensing</i>	17
3.4 Resumo	19
4 AJUSTE DO ALGORITMO E-LITESENSE	21
4.1 Cenário de teste	21
4.2 Estudo da variação da escala de reatividade em situação de <i>stress</i>	25
4.2.1 Nível de <i>stress</i> baixo ou pontual	25
4.2.2 Nível de <i>stress</i> médio	28
4.2.3 Nível de <i>stress</i> alto	32
4.3 Estudo da versatilidade do algoritmo	37
4.4 Análise dos Resultados	38
4.5 Resumo	41

5	PROVA DE CONCEITO	44
5.1	Sistema de Rega Inteligente	44
5.2	Cenário de Teste	46
5.3	Avaliação de Desempenho	48
5.4	Resumo	52
6	CONCLUSÕES	53
6.1	Conclusões	53
6.2	Trabalho Futuro	55
A	GRÁFICOS E TABELAS EXTRA	56
A.1	Nível de <i>stress</i> baixo ou pontual	56
A.2	Nível de <i>stress</i> Médio	59
A.3	Nível de <i>stress</i> Alto	65
A.4	Rácio Peak to Mean - Nível de <i>stress</i> médio	72
A.5	Rácio Peak to Mean - nível de <i>stress</i> alto	73
A.6	Versatilidade - Medições	74
A.7	Versatilidade - Consumo Energético	76
	Bibliografia	78

LISTA DE FIGURAS

Figura 1	Vertentes aplicacionais das Cidades Inteligentes	5
Figura 2	Interface de utilizador CupCarbon	9
Figura 3	Sensor e informações acerca do mesmo	10
Figura 4	Sensor a correr com um <i>script</i>	10
Figura 5	Sensores direcionais	10
Figura 6	Base station	11
Figura 7	Dispositivos móveis	11
Figura 8	Gerador de eventos analógicos	11
Figura 9	Esquema de amostragem e-LiteSense	13
Figura 10	Arquitetura do e-LiteSense	15
Figura 11	Arquitetura do e-LiteSense	19
Figura 12	Cenário de simulação dos testes	22
Figura 13	Número de eventos de <i>sensing</i> variando o gamma	26
Figura 14	Comparação das leituras para gamma igual a 0.5	27
Figura 15	Consumo energético para os diferentes valores de gamma	28
Figura 16	Número de eventos de <i>sensing</i> em função do parâmetro gamma	29
Figura 17	Comparação das leituras para gamma igual a 1.5	31
Figura 18	Consumo energético para os diferentes valores de gamma	32
Figura 19	Número de eventos de <i>sensing</i> variando o gamma	33
Figura 20	Comparação das leituras para gamma igual a 3.5	35
Figura 21	Consumo energético para os diferentes valores de gamma	36
Figura 22	Redução do número de eventos de <i>sensing</i> para os diferentes cenários	39
Figura 23	Comparação das leituras para o dataset da Voltagem	40
Figura 24	Comparação das leituras para o dataset de Monóxido de Carbono	40
Figura 25	Comparação do consumo energético para o dataset de Latitude	42
Figura 26	Comparação do consumo energético para o dataset da Voltagem	42

Figura 27	Algoritmo de controlo da irrigação [10]	45
Figura 28	Cenário de Simulação	46
Figura 29	Redução do número de eventos de <i>sensing</i> para os diferentes tipos de sensores	49
Figura 30	Comparação das leituras para o sensor da Temperatura	50
Figura 31	Comparação do consumo energético para o sensor da Temperatura	51
Figura 32	Medições para gamma igual a 0.005	56
Figura 33	Medições para gamma igual a 0.01	57
Figura 34	Medições para gamma igual a 0.1	57
Figura 35	Medições para gamma igual a 0.2	58
Figura 36	Medições para gamma igual a 0.3	58
Figura 37	Medições para gamma igual a 0.005	59
Figura 38	Medições para gamma igual a 0.01	60
Figura 39	Medições para gamma igual a 0.1	60
Figura 40	Medições para gamma igual a 0.2	61
Figura 41	Medições para gamma igual a 0.3	61
Figura 42	Medições para gamma igual a 0.4	62
Figura 43	Medições para gamma igual a 0.5	62
Figura 44	Medições para gamma igual a 1	63
Figura 45	Medições para gamma igual a 2	63
Figura 46	Medições para gamma igual a 2.5	64
Figura 47	Medições para gamma igual a 3.5	64
Figura 48	Medições para gamma igual a 5	65
Figura 49	Medições para gamma igual a 0.005	65
Figura 50	Medições para gamma igual a 0.01	66
Figura 51	Medições para gamma igual a 0.1	66
Figura 52	Medições para gamma igual a 0.2	67
Figura 53	Medições para gamma igual a 0.3	67
Figura 54	Medições para gamma igual a 0.4	68
Figura 55	Medições para gamma igual a 0.5	68
Figura 56	Medições para gamma igual a 1	69
Figura 57	Medições para gamma igual a 1.5	69
Figura 58	Medições para gamma igual a 2	70
Figura 59	Medições para gamma igual a 2.5	70
Figura 60	Medições para gamma igual a 5	71

Figura 61	Comparação das leituras para o dataset da Latitude	74
Figura 62	Comparação das leituras para o dataset da Longitude	74
Figura 63	Comparação das leituras para o dataset da Velocidade do Vento	75
Figura 64	Comparação das leituras para o dataset da Ozono	75
Figura 65	Comparação do consumo energético para o dataset da Longitude	76
Figura 66	Comparação do consumo energético para o dataset da Vel. Vento	76
Figura 67	Comparação do consumo energético para o dataset de Ozono	77

LISTA DE TABELAS

Tabela 1	Descrição da variáveis presentes nos algoritmos 1 e 2	17
Tabela 2	Precisão das medições - nível de <i>stress</i> baixo ou pontual	26
Tabela 3	Precisão medições - nível de <i>stress</i> médio	29
Tabela 4	Métricas de <i>burstiness</i> - nível de <i>stress</i> médio	30
Tabela 5	Precisão das medições - nível de <i>stress</i> alto	34
Tabela 6	Métricas de <i>burstiness</i> - nível de <i>stress</i> alto	34
Tabela 7	Identificação e descrição dos datasets usados	37
Tabela 8	Precisão das medições - versatilidade	41
Tabela 9	Descrição dos datasets usados na prova de conceito	47
Tabela 10	Precisão das medições - sistema de rega	50
Tabela 11	Poupança de energia para os diferentes sensores de cada zona	52

LISTA DE ACRÓNIMOS

WSN	Wireless Sensor Network
RSSF	Rede de Sensores sem Fios
IoT	Internet of Things
SC	Sensing Compressivo
CI	Cidade Inteligentes
CPU	Unidade Central de Processamento
TCP	Transmission Control Protocol
RTT	Round Trip Time
SDN	Software-defined networking
GPS	Sistema de Posicionamento Global
EQM	Erro Quadrático Médio
CV	Coefficiente de Variação
IDC	Index of dispersion for counts

INTRODUÇÃO

1.1 ENQUADRAMENTO

Com o exponencial aumento da população nas cidades mundiais surgiram diversos problemas de difícil resolução, levando a que as cidades sofressem uma reestruturação na maneira como são geridas. Nesse contexto surgiu o conceito de *Cidade Inteligentes* e *Internet of Things* (IoT) que têm como principais objetivos aumentar a sustentabilidade das cidades e facilitar o acesso a informação útil por parte dos seus cidadãos.

Esta reestruturação envolve a criação de diversos projetos tecnológicos baseados em Redes de Sensores Sem Fios (RSSF) para recolherem diversos dados do meio. As RSSF têm diversos objetivos, que vão desde a medição de parâmetros ambientais ao suporte de serviços de manutenção das cidades. Os seus usos são imensos e estão a ser incorporados a cada momento que passa, sendo que muitas destas aplicações não requerem intervenção humana [1].

Esta versatilidade faz com que cada vez mais seja necessário encontrar mecanismos que tenham a capacidade de, qualquer que seja o cenário em questão, se adaptarem ao mesmo sendo capazes de otimizar o tempo de vida da RSSF, pois a energia consumida por cada um dos sensores tem um papel fundamental na longevidade desse nó e da rede em geral.

Considerando que a frequência de eventos de medição de parâmetros da rede de sensores sem fios (isto é, temperatura, humidade, pressão, luminosidade, etc.) está diretamente relacionada com o número de operações executadas e com a quantidade de dados armazenados e transmitidos, podemos concluir que, para conseguirmos aumentar o tempo de vida útil de um sensor e, conseqüentemente, da rede de sensores sem fios, é necessário reduzir o número de eventos de medição e comunicação sem, contudo, comprometer a avaliação dos parâmetros medidos. É de salientar, que existem outras formas de chegar a esta redução de consumo.

Nesse sentido, surgiu a ideia de estudar soluções que adaptem a frequência das medições dos diversos parâmetros em estudo, de forma a permitir uma redução da energia consumida pelos sensores e, como resultado, um aumento da eficiência energética da RSSF.

1.2 MOTIVAÇÃO E OBJETIVOS

Com um papel determinante na IoT e nas diversas áreas envolvendo Cidades Inteligentes, as Redes de Sensores Sem Fios (RSSF) são uma tecnologia que atualmente tem despertado um interesse substancial, tanto na comunidade acadêmica como na indústria. Estas RSSF são aplicadas em diversas áreas havendo assim a necessidade de encontrar mecanismos de gestão de consumo energético que sejam capazes de se adaptarem aos diversos cenários aplicativos de uma forma autônoma, otimizando assim o tempo de vida da RSSF.

Sendo que a redução do consumo energético dos sensores destas RSSFs tem vindo a ser um tópico de grande investigação nos últimos tempos, esta dissertação tem como motivação melhorar a capacidade das RSSF de se adaptarem autônoma e eficientemente a ambientes operacionais distintos. Posto isto, o principal objetivo desta dissertação passa por, usando um simulador de redes de sensores e um cenário representando um Sistema de Rega inteligente, aplicar o algoritmo *e-LiteSense* [11] nos nós da rede de sensores desse sistema, com o objetivo de avaliar se este algoritmo é ou não capaz de aumentar a longevidade da RSSF, mantendo a precisão das medições obtidas.

Para além de aplicar o algoritmo *e-LiteSense* nos diferentes sensores do sistema de rega, vão ser também realizados testes de ajuste do algoritmo. Estes testes servem para percebermos o impacto da alteração de algumas das variáveis de configuração do algoritmo, bem como entender a forma como o algoritmo se comporta em diferentes cenários. Os testes serão realizados num cenário simplificado, sendo este constituído apenas por 2 sensores, um com o algoritmo *e-LiteSense* e outro sem o algoritmo.

Em estudo anteriores de RSSF, este algoritmo demonstrou ter a capacidade de efetuar uma medição adaptativa dos diferentes parâmetros que estão a ser captados pelos sensores da rede, mantendo a correta recolha dos diferentes dados e aumentando a longevidade dos sensores e, conseqüentemente, da RSSF. Como referido, neste projeto de mestrado a versatilidade, eficiência e capacidade de auto-adaptação deste algoritmo vão ser alvo de análise detalhada.

A ferramenta de simulação escolhida foi o CupCarbon, uma vez que esta ferramenta foi especialmente concebida para simular cenários de Cidades Inteligentes e IoT, sendo uma ferramenta de fácil adaptação e também gratuita.

1.3 ESTRUTURA DO DOCUMENTO

Neste primeiro capítulo é feita uma introdução ao tema que esta dissertação aborda, explicando de que forma este tema se insere na realidade atual, sendo apresentados os motivos que levaram à proposta de desenvolvimento deste tema, bem como os objetivos que se pretendem alcançar.

No segundo capítulo são apresentados projetos relevantes na área de Cidades Inteligentes, mais concretamente na área de *Smart Environment*, projetos estes usando RSSF. São ainda apresentados trabalhos dentro da área da eficiência energética em RSSF. Adicionalmente, é feita uma pequena introdução à ferramenta CupCarbon apresentando algumas das funcionalidades presentes na ferramenta.

No terceiro capítulo são apresentados os princípios primários que levaram à estruturação do algoritmo e-LiteSense, bem como a sua arquitetura e os sub-algoritmos que estipulam o seu comportamento.

No quarto capítulo são apresentados os testes de ajuste que foram realizados ao algoritmo e-LiteSense, começando por apresentar um estudo à alteração do threshold de ajuste da escala de reatividade (γ) e também um estudo à capacidade do algoritmo de funcionar corretamente em cenários heterogéneos.

No quinto capítulo é realizada a demonstração do algoritmo num cenário mais próximo de um exemplo de vida real, aplicando por isso o algoritmo e-LiteSense num Sistema de Rega Inteligente, sistema este desenvolvido usando a ferramenta CupCarbon. São realizados alguns testes para verificar se o algoritmo aumenta eficientemente e de forma precisa, o tempo de vida da RSSF.

No sexto e último capítulo é feito um resumo das conclusões obtidas na realização deste trabalho, bem como são apresentadas algumas sugestões para trabalho futuro.

TRABALHO RELACIONADO

Neste capítulo apresentam-se projetos realizados para Cidades Inteligentes na área da gestão dos recursos de uma cidade e serão, também, apresentados trabalhos relacionados com a área da eficiência energética de Redes de Sensores Sem Fios. De seguida, apresentam-se algumas das funcionalidades da ferramenta CupCarbon, ferramenta esta usada na simulação do Sistema de Rega utilizado como caso de estudo.

2.1 PROJETOS CIDADES INTELIGENTES

Nos tempos atuais, o mundo está a enfrentar uma crescente urbanização e, simultaneamente, as grandes cidades tornam-se um motor para o crescimento económico. Com o crescimento da urbanização, cada vez mais as cidades adoptam soluções tecnológicas já que estas têm o potencial de resolver grandes desafios metropolitanos. Como consequência do aumento do uso de tecnologia, essas cidades transformam-se em 'Cidades Inteligentes'.

Dentro do tema das Cidades Inteligentes existem várias vertentes de aplicação, como podemos ver pela Figura 1, sendo estas *Smart Economy*, *Smart Governance*, *Smart Mobility*, *Smart Population*, *Smart Living* e *Smart Environment* [23], sendo que é na última componente que este trabalho de pesquisa se foca, já que é nela que se enquadra o sistema de rega inteligente usado como caso de estudo.

O *Smart Environment* pode ser dividido em 2 categorias principais: urbanismo sustentável e gestão eficiente de recursos. Como existem diversos projetos nestas duas áreas, são apenas apresentados projetos se enquadram com o tema presente nesta dissertação. Sendo assim, em [5] foi criado um sistema inteligente de controlo das luzes de rua usando redes de sensores em fios, sistema este que permite economizar não só energia eléctrica, mas também custos de manutenção. Este sistema usa sensores que permitem a deteção de veículos e, quando um veículo é detectado, o aumento da intensidade luminosa das lâmpadas para um nível predefinido, de modo a não afetar a

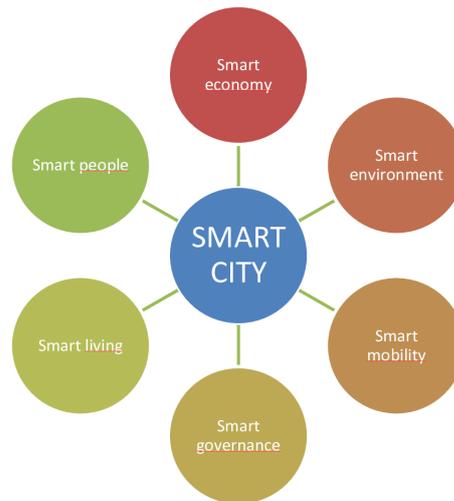


Figura 1: Vertentes aplicacionais das Cidades Inteligentes

segurança do tráfego rodoviário, e uma redução da luminosidade, no caso oposto. Além disso, o sistema utiliza sensores de corrente para permitir a identificação de possíveis avarias e, assim, facilitar o processo de manutenção.

Em [6] na cidade de Lucknow na Índia foi criado um modelo conceptual para melhorar a gestão da água nos sistemas de rega, que eram originalmente controlados de forma manual o que levava a um desperdício de água. Este modelo usa uma RSSF para controlar as lâmpadas de algumas ruas da cidade, usando informação meteorológica (visibilidade, nevoeiro, etc.), levando assim a uma melhor gestão dos recursos energéticos.

Existem também alguns projetos relacionados mais diretamente com sistemas de rega. Nomeadamente, um sistema de controlo e monitorização de rega agrícola [8], no qual é permitido controlar todo o sistema de rega agrícola, usando ainda Cloud Computing para tratar da grande quantidade de dados gerados pela rede de sensores sem fio.

Em Lima, no Peru foi também realizado um estudo de caso para um sistema de rega inteligente [7] para otimizar o uso de água nos parques da cidade, sendo que neste sistema a rede de sensores é encarregue de medir a humidade e temperatura. Em complemento, é usada a previsão da meteorologia para que o sistema seja capaz de tomar a decisão de ligar ou desligar o sistema de rega.

Em Portugal, existe um sistema de rega inteligente em Castelo Branco que funciona da seguinte forma: o Município instalou vários pluviômetros em locais específicos que, quando detectam chuva, desligam automaticamente a irrigação. O sistema permite,

para além disso, ativar ou desativar o sistema de irrigação remotamente e emitir alertas em caso de mau funcionamento dos equipamentos [9].

Para além dos projetos supramencionados, foi desenvolvido um projeto na Universidade do Minho, cujo objetivo é o desenvolvimento de um Sistema de Rega Inteligente. Este projeto, tenta otimizar a manutenção de espaços verdes a dois níveis : (i) desenvolvendo uma aplicação web de forma a controlar o sistema de irrigação dos espaços verdes; e (ii) apresentando um prototipo de simulação de um sistema de rega inteligente, ajudando a conceber soluções de sensoriamento adaptativo para a medição de parâmetros atmosféricos e do solo [10].

2.2 EFICIÊNCIA ENERGÉTICA E MEDIÇÃO ADAPTATIVA EM RSSF

Embora a medição adaptativa tenha sido usada com sucesso em redes de computadores convencionais, as soluções disponíveis dificilmente podem ser aplicadas em redes de sensores com recursos limitados, como as RSSF. De facto, ao não considerar convenientemente o parâmetro energético no processo de medição adaptativa este não reflete o equilíbrio necessário entre precisão e o tempo de vida útil da RSSF.

A definição do tempo de vida útil de uma RSSF, está diretamente relacionado com a sua bateria, mas esta pode variar consoante as funcionalidades da aplicação, função, objetivo e a topologia de rede a ser usada. Em [12] são apresentadas as técnicas de otimização da vida de uma RSSF mais usadas, sendo estes algoritmos classificados como alocação de recursos; escalonamento *sleep-wake*; *mobile relays* e *sinks*; recolha de dados; codificação da rede; transmissões oportunistas; *clustering*; correlação dos dados; *routing*; colheita de energia e *beamforming*.

O algoritmo usado neste trabalho, *e-LiteSense* [11] é um algoritmo que realiza o seu *sensing* de uma forma adaptativa, baseando as suas medições no nível da bateria dos sensores da rede e nas variações dos parâmetros em medição. Este algoritmo é combinado com outras técnicas, como por exemplo a correlação dos dados, já que a correlação de dados reduz a quantidade de transmissão dos dados. Ao combinarmos estas técnicas com medição adaptativa a recolha de dados e transmissão dos mesmos são reduzidas substancialmente, aumentando o tempo de vida da RSSF. No terceiro capítulo será também feita uma breve introdução a este algoritmo.

Ao longo dos tempos alguns estudos foram feitos para RSSF, tendo estes em consideração as restrições de energia e processamento destas redes. Em [13] usam *sensing* compressivo (SC) para reduzir a comunicação de dados na rede, consistindo este processo em recolher informação baseada na escassez de sinal, na realização de análise

estatística, etc. Contudo, e por norma, o uso de SC mostra-se ineficiente, sendo que em [14] os autores referem que o processo de descompressão dos dados em sensores é ineficiente, podendo a escassez dos dados do sensor mudar significativamente devido à sua natureza variável no tempo, afetando a recuperação do sinal medido.

Em [16], os autores desenvolvem uma framework para recolha de dados de RSSF baseadas em *sensing* compressivo adaptativo. Esta proposta tenta reduzir o consumo energético e a quantidade de informação na fase do *sensing*, adaptando a frequência de transmissão com a utilização dum algoritmo para obter as aproximações das medições, consumindo a menor energia possível e mantendo a precisão.

Em [17], é realizado um estudo analítico acerca do processo de *sensing*, compressão e recuperação do sinal para datasets grandes. A análise usa relações probabilísticas entre as variáveis envolvidas nesses processos, de maneira a capturar as correlações de espaço e tempo dos sinais de interesse. A proposta diminui o número de transmissões e, conseqüentemente, o consumo de energia nos nós dos sensores. Contudo, este estudo está focado principalmente na redução da taxa de transmissão, desconsiderando a precisão do processo de *sensing*.

Em [18] é proposta uma framework de seleção de sensores adaptativos baseados na aprendizagem Bayesiana para aumentar a vida útil de um RSSF que monitoriza o sinal variável espacio-temporal. A escolha do sensor leva em consideração a relação entre qualidade da detecção e o nível de energia. A abordagem de otimização analítica é difícil de implementar num ambiente real e requer um conhecimento topológico da RSSF, o que por si só é uma limitação para a implementação de grandes cenários de RSSF.

Em [20] é proposto um sistema para calcular, em tempo real, o consumo de energia de uma RSSF. Os autores basearam a sua avaliação em diferentes partes do hardware de maneira isolada, como por exemplo: leds; transmissão de dados; potência de transmissão; microcontrolador e instruções da CPU. Em [19], os autores estimam a duração da bateria usando um modelo de previsão. A metodologia adotada baseia-se na decomposição das operações de hardware em períodos de tempo, usando datasheets de sensores como referência.

Em [15] é proposta uma política de escalonamento de *sensing* para um sistema equipado com uma bateria finita. Essa abordagem tem como objetivo selecionar os melhores períodos de *sensing* para que o desempenho médio seja otimizado. Os autores assumem que o desempenho depende da duração do tempo entre dois períodos consecutivos de *sensing* e o algoritmo desenvolvido tem em conta os níveis de energia da bateria dos sensores para regular a taxa de *sensing*.

Os diferentes trabalhos referidos anteriormente destacam a necessidade de estudar e melhorar o trade-off entre a precisão da medição e os eventos de detecção para diferentes cenários envolvendo RSSF, incluindo cenários reais, e tendo em conta energia dos sensores da rede.

2.3 PLATAFORMA DE SIMULAÇÃO CUPCARBON

O aumento do uso de sistemas de comunicação via rádio e os avanços significativos das tecnologias de dispositivos, fazem com que este tipo de tecnologias se aproximem da Internet of Things (IoT) e abram novos horizontes para as aplicações em Cidades Inteligentes.

Essa evolução torna-se essencial para melhorar a qualidade dos serviços urbanos, reduzir custos e envolver os cidadãos de maneira mais ativa. Nesse contexto, são necessárias novas ferramentas de simulação para preparar os projetos de futuras infraestruturas de IoT em larga escala para Cidades Inteligentes, nas melhores condições em termos de confiabilidade, consumo de energia e custo.

Dentro dos simuladores disponíveis nesta área de intervenção, a plataforma de simulação selecionada para este trabalho foi o CupCarbon. Este simulador foi escolhido em detrimento de outros simuladores pois, primeiramente, é um simulador gratuito e *Open Source* com o qual já tinha experiência prévia o que facilita a familiarização com a plataforma. Para além disso, este simulador permite o desenvolvimento simplificado de diversos cenários, implementação de diferentes algoritmos e funcionalidades que ajudam a recolher informações acerca dos eventos de *sensing* e do consumo energético dos diferentes sensores.

Em mais detalhe, a ferramenta CupCarbon [4] é um simulador de Wireless Sensor Network para Cidades Inteligentes e IoT. O seu principal objetivo é projetar, visualizar, corrigir e validar algoritmos para monitorização e recolha de dados ambientais. Esta ferramenta pode não só ajudar a explicar visualmente os conceitos básicos de funcionamento das redes de sensores, mas também ajudar no contexto científico para testar topologias, protocolos de rede, etc.

Esta ferramenta oferece dois tipos de simulação:

- um ambiente multi-agent [2], que possibilita a conceção de design de cenários de mobilidade, como, por exemplo, de veículos e objetos voadores (ou seja, UAVs, insetos, etc.) [3] e a geração de eventos naturais;

- um simulador de eventos discretos de RSSF que leva em consideração o cenário projetado com base no ambiente anterior;

Essas redes podem ser desenvolvidas usando uma interface ergonômica e de fácil uso, utilizando a framework OpenStreetMap (OSM) para colocar os sensores diretamente no mapa. Inclui também uma linguagem para implementar *script's* chamada SenScript, que nos permite programar e configurar cada sensor da rede individualmente.

Uma simulação do CupCarbon é baseada na camada aplicacional dos nodos, o que faz com que esta seja um bom complemento em relação aos restantes simuladores. As diferentes simulações de algoritmos e cenários podem ser realizadas em diferentes passos e o consumo de energia da rede de sensores pode ser calculado através de funções que o CupCarbon fornece, permitindo assim clarificar a estrutura, viabilidade e implementação de uma rede antes do desenvolvimento real.

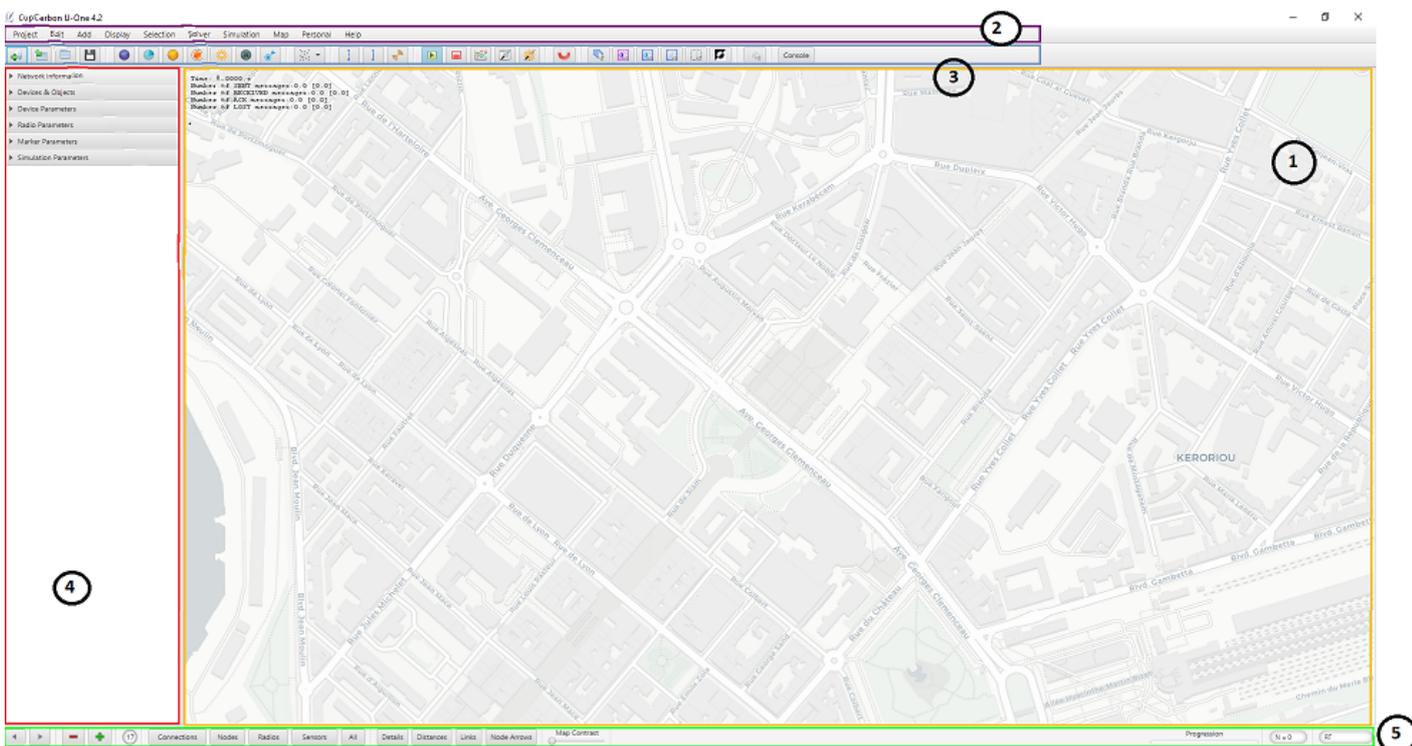


Figura 2: Interface de utilizador CupCarbon

Na Figura 2 é apresentada a Interface Gráfica do Utilizador, contendo esta 5 partes principais: (1) o mapa; (2) a barra do menu; (3) a barra de ferramentas; (4) o painel dos parâmetros; (5) a barra de estado;

Dentro da seleção dos parâmetros de rádio, a ferramenta permite ao utilizador escolher entre 3 protocolos de comunicação: ZigBee [33], Wifi [34] e Lora [35]. Por padrão, o protocolo ZigBee é automaticamente definido para cada novo sensor adicionado.

Uma das características mais interessantes deste simulador é o facto de este nos possibilitar desenvolver a rede em diferentes mapas, usando o Google Maps, de forma a podermos obter uma simulação mais realista. Os seguintes dispositivos podem ser colocados nos diferentes mapas:

- **Sensores:** dispositivos cuja função é a de detetar as variáveis ambiente em observação.

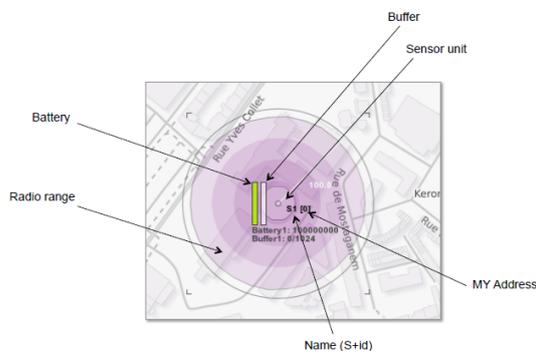


Figura 3: Sensor e informações acerca do mesmo

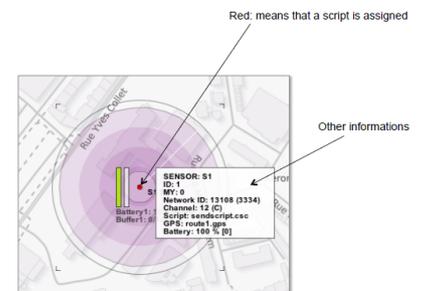


Figura 4: Sensor a correr com um *script*

- **Sensores direcionais:** dispositivos capazes de detetar as variáveis ambiente, tendo em conta uma direção específica.

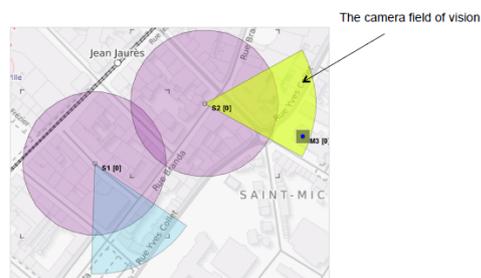


Figura 5: Sensores direcionais

- **Base Station (Sink):** possui as mesmas funcionalidades de um sensor, mas com a característica de a sua bateria ser infinita.

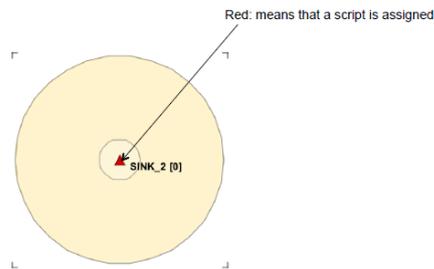


Figura 6: Base station

- **Dispositivo Móvel:** dispositivo que pode percorrer uma rota criada com marcadores.

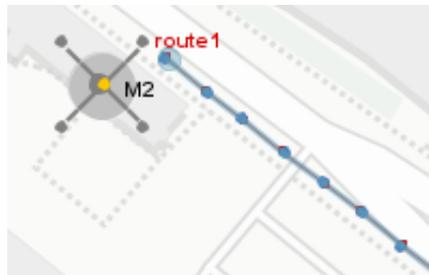


Figura 7: Dispositivos móveis

- **Marcadores:** estes marcadores podem ser usados em diferentes tarefas:
 - adicionar sensores de uma forma aleatória numa área;
 - criar rotas;
 - adicionar ou desenhar edifícios.
- **Gerador de eventos analógicos:** usado para gerar eventos analógicos a fim de simular fenómenos ambientais, tais como a temperatura, humidade, etc.

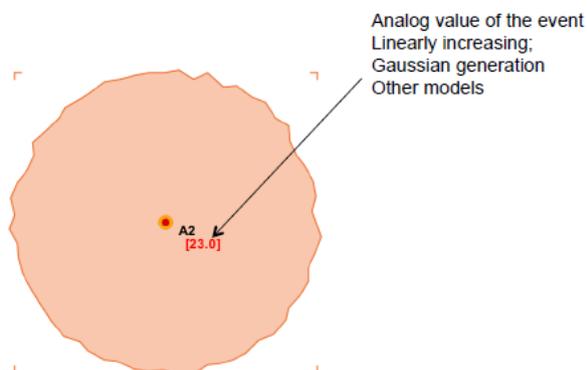


Figura 8: Gerador de eventos analógicos

2.4 RESUMO

Neste capítulo são apresentados alguns projetos relacionados com Cidades Inteligentes, mais concretamente na vertente do *Smart Environment* pois é nesta que se enquadra o sistema de rega analisado neste trabalho. Para além disso, são também apresentados projetos relacionados com o algoritmo utilizado, sendo que os trabalhos apresentados se focam em estudos acerca da medição adaptativa em RSSF. Para finalizar, é feita uma pequena introdução à ferramenta de simulação usada nesta dissertação, apresentando as suas principais características e funcionalidades. No próximo capítulo será feita uma introdução mais detalhada ao algoritmo e-LiteSense, apresentando os princípios na base do mesmo, a sua arquitetura e os algoritmos que modelam o comportamento do algoritmo e-LiteSense.

E-LITESENSE: PRINCÍPIOS, ARQUITETURA E ALGORITMOS

Inicialmente, são identificados os princípios primários na base do conceito do algoritmo E-LiteSense [11]. De seguida, é apresentada a arquitetura do algoritmo de forma a fornecer um enquadramento geral sobre o mesmo e, para terminar, serão explicados os algoritmos que definem o comportamento do e-LiteSense [11].

3.1 PRINCÍPIOS DE CONCEÇÃO PRIMÁRIOS

O objetivo principal a ser alcançado é o de detetar parâmetros físicos efetuando amostragem de forma adaptativa, avaliando corretamente os seus valores e variações com o menor impacto possível sobre os recursos limitados que os sensores têm disponíveis. Posto isto, a idealização e arquitetura do e-LiteSense devem seguir diversos princípios de conceção primários, que são apresentados na Figura 9 e descritos abaixo.

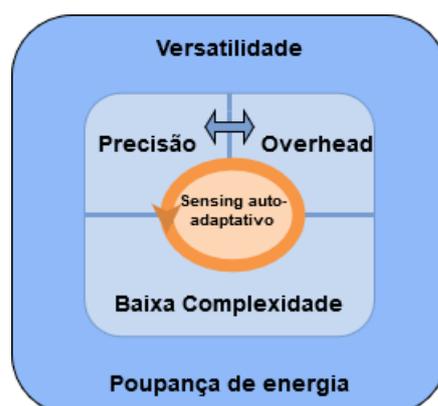


Figura 9: Esquema de amostragem e-LiteSense

A um nível mais alto, o e-LiteSense deve ser uma solução *versátil*. É bem sabido que as RSSFs são dependentes da aplicação e podem incluir diversos sensores que detetam múltiplos fenómenos físicos, portanto, o esquema de *sensing* deve ser capaz de funcionar

corretamente em contextos de RSSFs heterogêneos, com diversos requisitos de medição e parâmetros correspondentes. Outro aspecto fundamental é a consciência energética, a fim de permitir uma economia de energia eficaz. Assim, o mecanismo de *sensing* deve capturar os parâmetros físicos em consideração, usando uma abordagem de conservação de energia para que os componentes de *sensing*, processamento e comunicação poupem energia.

Estes dois princípios, *versatilidade* e *consciência energética*, fazem surgir um grande argumento do e-LiteSense, que é o *sensing* auto-adaptativo. O esquema de amostragem deve ser capaz de auto-regular os eventos de *sensing* de acordo com as variações dos parâmetros detectados e dos níveis de energia disponíveis. Para tal ser possível, é necessário encontrar um equilíbrio entre a precisão, a sobrecarga e a complexidade do processo de *sensing*. Ou seja, mais detalhadamente, *precisão* significa que o mecanismo de *sensing* deve ser capaz de capturar o comportamento das variáveis físicas, captando os seus valores para cada cenário aplicacional de RSSF. "Comportamento correto" é na verdade um "comportamento esperado" pois a aplicação pode necessitar de medições mais precisas ou mais grosseiras. Uma *sobrecarga baixa* significa que o processo de *sensing* deve minimizar o impacto do mesmo, mantendo a precisão dentro de uma faixa aceitável. *Baixa complexidade* significa que o comportamento auto-adaptativo esperado pelo mecanismo de *sensing* deve ser de fácil implementação e com um consumo de recursos mínimo.

A *versatilidade* do mecanismo de *sensing* é entendida como a capacidade de autorregulação dos eventos de *sensing* em função do comportamento intrínseco dos parâmetros sob controlo, tal como foi referido anteriormente. Isto implica que o algoritmo seja capaz de captar as variações dos parâmetros em medição, ajustando a frequência dos eventos de *sensing* conforme seja necessário. A *Precisão* na medição envolve detetar tanto o comportamento contínuo do parâmetro em questão como eventos que sejam críticos ou esporádicos. Mais abaixo neste documento, a precisão das medições obtidas através do esquema e-LiteSense são avaliadas comparando, visual e estatisticamente, as séries temporais envolvidas na amostragem sistemática e auto-adaptativa. Apesar de ser um objetivo comum de conceção, um processo com *sobrecarga baixa* assume um papel ainda mais relevante na presença de dispositivos com recursos limitados.

O esquema de *sensing* auto-adaptativo é, assim, baseado num processo aritmético simples, que consiste numa versão simplificada do mecanismo que o Protocolo de Controlo de Transmissão (Transmission Control Protocol (TCP)) usa para estimar os tempo de ida e volta (Round Trip Time (RTT)) [21], contribuindo desta forma para um processamento de *baixa complexidade*. Resumindo, as medidas que contribuem

diretamente para a *poupança de energia* são as seguintes: (i) reduzir o número de eventos de *sensing* necessários para capturar o comportamento de cada parâmetro; (ii) ajustar a frequência dos eventos de *sensing* através de algoritmos de baixo custo energético e (iii) a redução do número de transmissões entre os diferentes sensores.

3.2 ARQUITETURA

De forma a entender o papel e posicionamento do e-LiteSense no desenvolvimento de um cenário de RSSFs, uma arquitetura tri-dimensional é idealizada na Figura 10 e explicada de seguida.

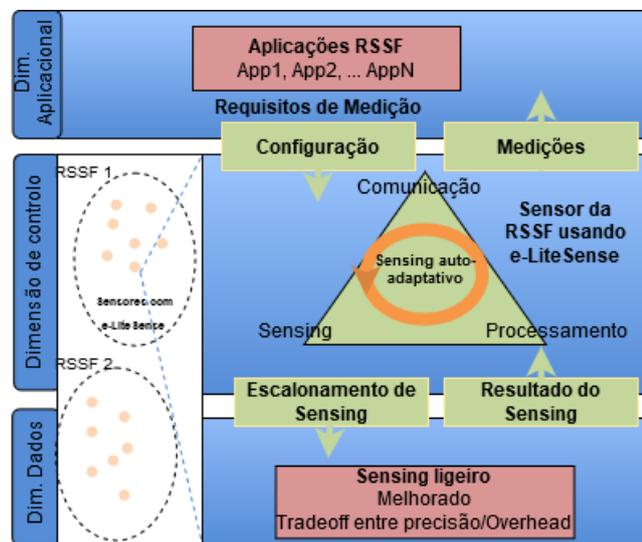


Figura 10: Arquitetura do e-LiteSense

- *Dimensão aplicacional* - Como já referido, a dimensão aplicacional pode abranger uma miríade de áreas aplicacionais de RSSF. De acordo com a taxonomia proposta em [21], aplicações que envolvem RSSFs enquadram-se em 8 áreas distintas, que vão desde a medicina e a saúde até à monitorização ambiental. Consequentemente, diferentes requisitos de medição têm de ser especificados dependendo das funcionalidades necessárias que cada aplicação pretende oferecer, que por sua vez determinam a configuração dos diferentes sensores. A uma determinada altura, o plano aplicacional deverá receber as medições necessárias resultantes de um ou mais sensores pertencentes à RSSF, refletindo, por exemplo, a variação espacial ou temporal das variáveis físicas sob controlo. No momento da receção dos resultados das medições da dimensão de controlo, a dimensão aplicacional

- pode também incluir o processamento e as unidades de interface do utilizador, se aplicável.
- *Dimensão de Controlo* - Desenvolvida diretamente nos sensores da RSSFs, é nesta dimensão que o e-LiteSense está implementado, seguindo os princípios apresentados na secção anterior. Como será explicado com mais detalhe na secção seguinte, o e-LiteSense determina a ocorrência de eventos de *sensing* de forma autónoma, recorrendo a algoritmos de baixa complexidade, que visam a poupança de energia nos subsistemas de comunicação, de processamento e de *sensing* (vértices do triângulo da Figura 10). De acordo com o que já foi explicado e em seguimento disso, entende-se que os algoritmos de *sensing* auto-adaptativos propostos visam obter leituras dos sensores com precisão e com a menor sobrecarga possível. Equilibrando o compromisso entre a precisão-sobrecarga, o escalonamento de eventos de *sensing* é configurado em tempo real dependendo da variação das medições ao longo do tempo, atribuindo uma natureza auto-adaptativa ao processo. Por outro lado, a dimensão de controlo também processa o resultado das medições recebidas da dimensão de dados efetuando, por exemplo, uma agregação de dados antes de enviar os resultados para a dimensão aplicacional. Em prática, as funções da dimensão de controlo podem ser diretamente instanciadas nos sensores (ver Figura 10) ou em qualquer componente externo, como por exemplo *Software Defined Networks* (SDN) [22] ou princípios de *Edge Computing* [30]. Se algum componente externo, tal como um controlador SDN, estiver instalado, será esse controlador responsável por configurar, remotamente, os sensores com o algoritmo e-LiteSense.
 - *Dimensão de dados* - Os dados de *sensing* são recolhidos dentro da dimensão de dados. Tendo em consideração a frequência de detecção calculada pelo algoritmo, as leituras do sensor são recolhidas e enviadas para a dimensão de controle. Lembre-se de que o e-LiteSense, como um mecanismo de *sensing* flexível, permite determinar o próximo evento de detecção de acordo com a dinâmica da aplicação e requisitos de medição.

3.3 ALGORITMOS DE *sensing*

Tabela 1: Descrição da variáveis presentes nos algoritmos 1 e 2

S	- Valor detetado do parâmetro sob observação
\bar{X}_i	- Média móvel do parâmetro detetado da leitura atual
\bar{X}_{i-1}	- Média móvel do parâmetro detetado da leitura anterior
\bar{V}_i	- Desvio médio atual dos valores detectados
α	- Peso de S na média móvel \bar{X}_i
\bar{V}_{i-1}	- Desvio médio da leitura anterior
β	- O peso do desvio atual $ \bar{X}_i - S $
ΔT_{next}	- Intervalo de tempo até ao próximo evento de <i>sensing</i>
$\Delta T_{min} \Delta T_{max}$	- Limite inferior e superior para o ΔT
$getScale()$	- Função de ajuste da escala de reatividade
$\epsilon_{min} \epsilon_{max}$	- Limites ao fator da escala de reatividade
γ	- <i>Threshold</i> de ajuste da escala

Independentemente do tipo da aplicação, é claro que aumentar o número de eventos de *sensing* resultará numa maior precisão nas medições dos parâmetros a observar. Este aumento na precisão tem, no entanto, uma grande desvantagem que é a carga nos subsistemas de *sensing*, de processamento e de comunicação, no que diz respeito ao consumo energético. De maneira a encontrar o melhor equilíbrio entre a precisão e o consumo energético, o e-LiteSense aproveita as variações nos parâmetros em observação e o nível atual de energia dos sensores para regular, em concordância, a frequência de *sensing*. Desta forma, se a deteção dos dados não exibir variabilidade significativa, a frequência de *sensing* é reduzida, ou seja, o período temporal até ao próximo evento de *sensing* ΔT_{next} é aumentado de forma a economizar energia e recursos de processamento. Este ajuste deve ser definido e ajustado cuidadosamente de forma a que a precisão nas medições se mantenha em valores aceitáveis. Todavia, se os dados detetados pelos sensores apresentarem uma variação considerável, ΔT_{next} é diminuído de forma a aumentar a reatividade do e-LiteSense a variações escalares dos parâmetros em observação. Desta forma, a definição de ΔT_{next} tem sempre em conta o nível atual da bateria do sensor.

Em continuidade da secção anterior, o e-LiteSense recorre a processos aritméticos simples, procurando a simplicidade de implementação e consumo de recursos reduzido como princípios de conceção. Usando pseudocódigo, os Algoritmos 1 e 2 ilustram o comportamento do e-LiteSense. A Tabela 1 apresenta as principais variáveis e constantes usadas no algoritmo. Como ilustrado no Algoritmo 1, quando um evento de *sensing* acontece (linha 4 e 5), o valor médio da variável em observação \bar{X}_i é calculado usando

a média móvel entre a média obtida no evento de *sensing* anterior e a nova amostra S , isto é,

$$\bar{X}_i = (1 - \alpha)\bar{X}_{i-1} + \alpha S \quad (1)$$

(ver Linha 6). A constante α expressa o peso do novo valor observado S em \bar{X}_i . Neste caso, de forma a simplificar, a variação escalar do parâmetro em observação é definida como a média móvel desse parâmetro \bar{X}_i e o seu desvio médio \bar{V}_i . Esta é uma abordagem simples e aceitável para calcular uma aproximação desvio padrão, como apresentado em [21]. Assim, quando o evento de *sensing* S ocorre, o algoritmo compara como a nova leitura evoluiu

$$\bar{X}_{i-1} \xleftarrow{\bar{V}_{i-1}} \bar{X}_i \xleftarrow{\bar{V}_i} S$$

de forma a conseguir determinar o valor de ΔT_{next} , ou seja, o período de tempo até ao próximo evento de *sensing*. Se o desvio entre a variação atual e a anterior (\bar{V}_i e \bar{V}_{i-1}), em valor absoluto, exceder um threshold γ , o novo valor lido S sofreu uma variação que não pode ser desprezada, portanto, ΔT_{next} é diminuído de forma a capturar a nova tendência de variação. Caso contrário, se esta variação for menor, significa que o valor lido S está num ponto estável, o que permite definir um ΔT_{next} maior de forma a preservar a energia do sensor.

A função *getScale()* permite ajustar a escala de reatividade entre as leituras do sensor (Ver Algoritmo 2). Como apresentado, os ajustes do ΔT têm em conta o nível atual de energia na bateria do sensor. Na presença de níveis de bateria altos, o ΔT tende a diminuir rapidamente ou aumentar lentamente, enquanto que para níveis de bateria baixos, o comportamento é o oposto, ou seja, ΔT diminui lentamente ou aumenta rapidamente. As constantes ϵ_{min} e ϵ_{max} definem o limite máximo e mínimo da escala de reatividade. Estes valores são ajustados dependendo do tipo da aplicação e os correspondentes requisitos de medição dos fenómenos físicos.

O algoritmo define também limites superiores e inferiores para o ΔT , nomeadamente ΔT_{min} e ΔT_{max} (Ver linha 9). Enquanto que ΔT_{max} previne que o valor de ΔT cresça indefinidamente, garantindo assim um número mínimo de leituras por unidade de tempo, ΔT_{min} limita a frequência de *sensing* máxima de forma a que ΔT não tenda para zero, resultando num consumo de recursos extremo. Mais uma vez, definir estes limites depende também do cenário da aplicação.

Algorithm 1 Determinação do ΔT_{next}

```

main()
1:  $\bar{X}_{i-1} \leftarrow 0$ 
2:  $\bar{V}_{i-1} \leftarrow 0$ 
3: repeat
4:    $S \leftarrow \text{getSense}()$ 
5:    $\Delta T_i \leftarrow \text{timestamp de } S$ 
6:    $\bar{X}_i \leftarrow (1 - \alpha)\bar{X}_{i-1} + \alpha S$ 
7:    $\bar{V}_i \leftarrow (1 - \beta)\bar{V}_{i-1} + \beta|\bar{X}_i - S|$ 
8:    $\Delta T_{next} \leftarrow \Delta T_i + (\Delta T_i \times \text{getScale}(\bar{V}_i, \bar{V}_{i-1}))$ 
9:    $\Delta T_{next} \leftarrow \max(\min(\Delta T_{max}, \Delta T_{next}), \Delta T_{min})$ 
10:   $\bar{X}_{i-1} \leftarrow \bar{X}_i$ 
11:   $\bar{V}_{i-1} \leftarrow \bar{V}_i$ 
12:  wait( $\Delta T_{next}$ )
13: until end execution;

```

Algorithm 2 Ajuste da reatividade do algoritmo e-LiteSense

```

main()
1: if ( $|\bar{V}_i - V_{i-1}| \geq \gamma$ ) then
2:   return  $-((\text{getBattery}()) \times (\epsilon_{max} - \epsilon_{min}))/100 + \epsilon_{min}$ 
3: else
4:   return  $((100 - \text{getBattery}()) \times (\epsilon_{max} - \epsilon_{min}))/100 + \epsilon_{min}$ 

```

Figura 11: Arquitetura do e-LiteSense

3.4 RESUMO

Neste capítulo, inicialmente foram apresentados os princípios que estão na base da conceção do algoritmo e-LiteSense, referindo que o algoritmo deve ser uma solução *versátil*, capaz de funcionar corretamente independentemente do cenário da RSSF em questão e deve ser uma solução com *consciência energética* de forma a aumentar o tempo de vida da RSSF. Com estes dois princípios, surge um grande argumento do e-LiteSense que é o *sensing auto-adaptativo*, sendo que o algoritmo deve ser capaz de adaptar a frequência dos eventos de *sensing* de acordo com o nível de energia dos sensores e a variação dos parâmetros em observação. Para que isto seja possível é necessário encontrar um equilíbrio entre a *precisão* dos eventos de *sensing*, a *sobrecarga* do impacto dos eventos de *sensing* e a *baixa complexidade* dos algoritmos na base do comportamento auto-adaptativo do e-LiteSense.

Seguidamente, foi apresentada uma arquitetura tri-dimensional de forma a entender o papel e posicionamento do algoritmo e-LiteSense no desenvolvimento de um cenário de RSSF. Na dimensão aplicacional são definidos os requisitos de medição dos diferentes

parâmetros em observação. Na dimensão de controlo é onde o algoritmo e-LiteSense é implementado e também é feito o processamento do resultado das medições recebidas da dimensão de dados e na dimensão de dados é onde os dados de *sensing* são recolhidos.

Para finalizar, foram apresentados os algoritmos de *sensing* que controlam o comportamento do e-LiteSense, mostrando como é que o e-LiteSense aproveita as variações nos parâmetros em observação e o nível atual de energia dos sensores para regular a frequência dos eventos de *sensing*.

No próximo capítulo, serão apresentados e discutidos os resultados obtidos nos testes de ajuste do algoritmo e-LiteSense.

AJUSTE DO ALGORITMO E-LITESENSE

Tal como vem sendo mencionado, esta dissertação é continuação do trabalho previamente realizado sobre o algoritmo e-LiteSense [11], e será, por isso, dado seguimento ao mesmo, no sentido em que continuará a ser feito um estudo à eficiência do esquema adaptativo proposto. Serão realizados dois tipos de testes. Em primeiro lugar, será feito um estudo à alteração do *threshold* de ajuste da escala de reatividade (γ) em situações de alteração repentina dos parâmetros em observação (referidas como *stress*) de forma a entender o comportamento do algoritmo ao alterar esse parâmetro. Em segundo lugar, será feita uma análise à versatilidade do algoritmo, de forma a tentar entender até que ponto o algoritmo é capaz de funcionar corretamente, independentemente do cenário de aplicação em questão.

4.1 CENÁRIO DE TESTE

Os testes de avaliação de desempenho no contexto desta dissertação são realizados usando a ferramenta CupCarbon uma vez que esta permite o desenvolvimento de diferentes cenários, o uso de diferentes esquemas de *sensing* e ainda fornece funcionalidades que nos permitem criar um perfil detalhado dos eventos de *sensing* e do consumo energético dos diferentes sensores.

A análise de desempenho do algoritmo passa pela comparação do consumo energético e na precisão das medições dos fenómenos físicos usando dois sensores: um com uma abordagem sistemática (abordagem mais usada em cenários de RSSF) e outro usando a abordagem adaptativa do algoritmo e-LiteSense, tal como mostra a Figura 11. Estes sensores foram configurados com o protocolo de comunicação ZigBee pois este protocolo permite obter uma melhor eficiência energética. Para além disso, como modelo de consumo energético foi usado o consumo energético Clássico disponibilizado pelo CupCarbon.



Figura 12: Cenário de simulação dos testes

Apresenta-se de seguida os datasets públicos usados como *input* para os diferentes sensores, tanto para o caso do estudo do γ como para o caso do estudo da versatilidade do algoritmo, com o objetivo de tornar os resultados obtidos nos testes realizados o mais realistas possível.

- **Estudo da reatividade (parâmetro γ)** - neste caso, são usados dois datasets obtidos por sensores num ambiente interior. Estes sensores faziam a medição da temperatura e humidade ambiente, em intervalos de 5 segundos, durante 6 horas [24]. De forma a conseguir analisar a precisão do e-LiteSense a identificar situações de *stress*, ou seja, mudanças significativas nos parâmetros em observação, este sensor foi exposto a vapor de água de forma a simular uma situação de *stress* no dataset.
- **Estudo da versatilidade** - de forma a avaliar a capacidade do algoritmo em funcionar corretamente em cenários heterogéneos serão usados, neste estudo, 6 datasets diferentes. Estes datasets foram selecionados tendo em conta o seguinte critério: os datasets devem incluir parâmetros numéricos, pertencentes a áreas aplicacionais distintas, terem uma frequência de sensing relativamente reduzida e serem captados durante períodos de tempo consecutivos. Sendo assim, os datasets selecionados são os seguintes :

- **Latitude/Longitude** - Este conjunto de dados de trajetórias GPS foi recolhido no projeto Geolife (Microsoft Research Asia) por 182 utilizadores num período de mais de três anos (de Abril de 2007 a Agosto de 2012). As trajetórias GPS deste conjunto de dados são representadas por uma sequência de pontos com registo de data e hora, cada um contendo as informações de latitude e longitude. Para o estudo realizado neste trabalho é usado apenas parte deste dataset, no qual a localização do utilizador é recolhida a cada 3 segundos durante um período de 5 horas. [27]
- **Velocidade do Vento** - Este dataset foi criado em 2017 no âmbito do projeto "Estudo de Qualidade do Ar em Escala Local"(KC-TRAQS), para aprender mais sobre a qualidade do ar da comunidade local de três bairros em Kansas City. Neste dataset a velocidade do vento é recolhida a cada 30 segundos durante 8 meses. Para este estudo apenas é usado um excerto de 7 horas. [25]
- **Voltagem** - Este dataset regista as medições de consumo de energia elétrica numa casa localizada em Sceaux (7km of Paris, França) entre Dezembro de 2006 a Dezembro de 2010. A voltagem média é recolhida a cada 60 segundos sendo que para este estudo é usado apenas um período de 8 horas. [26]
- **Monóxido de Carbono e Ozono** - Ambos os datasets [29] [28] foram recolhidos no âmbito do projeto "CityPulse EU FP7", que tem como objetivo facilitar a obtenção de dados para utilização em projetos de Cidades Inteligentes. Sendo assim, estes dois datasets foram recolhidos na cidade de Aarhus na Dinamarca. As medições de Monóxido de Carbono e de Ozono são recolhidas com uma frequência de 5 minutos, sendo que para este estudo é usado um excerto do dataset de 10 horas.

A avaliação do desempenho do algoritmo nos diferentes testes consiste em avaliar: (1) o número de eventos de *sensing* durante o período de monitorização; (2) a capacidade e precisão do algoritmo em identificar as variações dos parâmetros em observação; e (3) o consumo energético dos diferentes sensores da rede.

Para estimar a precisão das medições dos parâmetros obtidas pelo e-LiteSense é usado o *Coefficiente de Correlação* e o *Erro Quadrático Médio* (EQM) entre os valores obtidos e a dinâmica real dos fenómenos físicos, uma metodologia usada frequentemente para avaliar estimadores baseados em amostragem. Para o cálculo do EQM, é necessário determinar a média dos erros ao quadrado, ou seja, a diferença média ao quadrado entre cada um dos valores obtidos pelo e-LiteSense e o valor atual do dataset de input do sensor. Como o e-LiteSense reduz o número de eventos de *sensing*, o número de

entradas no dataset será maior que o número de valores obtidos pelo algoritmo. Deste modo, é necessário preencher os valores em "falta" para conseguir calcular corretamente o EQM. Para isso, é considerado o último valor lido até que ocorra um novo valor de amostragem para preencher os valores em "falta". Para além disso, será também comparado o consumo energético entre o sensor com o algoritmo e-LiteSense e o sensor sem o algoritmo.

Importante acrescentar ainda que nos casos de nível de *stress* médio e alto serão também utilizadas algumas métricas que são normalmente usadas para representar as características de *burstiness* de Tráfego de Redes [31] [32]. O principal objetivo destas métricas é o de avaliar a variabilidade dos dados obtidos. Sendo assim, estas métricas são utilizadas neste estudo no intuito de ajudar a entender se o algoritmo se comporta corretamente em casos de *bursts* nos parâmetros em observação, indicando a variabilidade dos dados obtidos nos momentos de *bursts*. Posto isto, as métricas utilizadas são as seguintes:

- **Coefficiente de Variação (CV)** - pode ser definido como o rácio entre o desvio padrão(σ) e a média(μ) :

$$\sigma / \mu. \quad (2)$$

- **Índice de Dispersão (IDC)** - pode ser definido como o rácio entre a variância (σ^2) e a média (μ) :

$$\sigma^2 / \mu \quad (3)$$

- **Rácio Peak to Mean** - pode ser definido como o rácio entre o maior valor obtido (*peak*) e a média (μ) :

$$peak / \mu \quad (4)$$

Depois de apresentadas as métricas a serem usadas para avaliar o desempenho do algoritmo, é necessário indicar a configuração do algoritmo para os diferentes testes que serão discutidos na Secção 4.2. Esta configuração foi selecionada com base experimental (estudo prévio ao algoritmo) e considera um $\Delta T_{min} = 5$ e $\Delta T_{max} = 45$, significando que inicialmente a frequência de *sensing* será de 5 segundos e que pelo menos um evento de *sensing* ocorre a cada 45 segundos. Os limites ao fator da escala reatividade é definido como $\epsilon_{min} = 0.05$ e $\epsilon_{max} = 0.30$. Considera-se também que o peso das observações anteriores \bar{X}_{i-1} e \bar{V}_{i-1} em \bar{X}_i e \bar{V}_i (Ver Algoritmo 1) são $\alpha = 0.7$ e $\beta = 0.7$. Para terminar,

a variável γ irá ser alterada a cada execução da simulação, pois é nesta variável que incide o estudo da Secção 4.2, no qual se pretende entender o comportamento do algoritmo em termos de reatividade, alterando o valor do γ .

4.2 ESTUDO DA VARIAÇÃO DA ESCALA DE REATIVIDADE EM SITUAÇÃO DE *stress*

O objetivo de fazer um estudo do parâmetro γ é o de perceber o impacto da alteração do mesmo no comportamento do algoritmo em termos de reatividade, tal como já vem sendo referido. Os testes realizados e reportados nesta secção consideram a utilização de dois datasets. Especificamente, um dataset de um sensor de temperatura e um de humidade, ambos colocados num ambiente interior. Estes sensores foram expostos a um vapor de água, gerando assim um pico de temperatura e humidade, pico este que pode ajudar a entender a capacidade do algoritmo em captar variações repentinas nos parâmetros em observação e ainda tentar definir qual seria o valor ótimo para atribuir à variável γ em diferentes cenários de *stress*.

Por conseguinte, decidiu-se dividir estes testes em três partes. A primeira, onde é usado um dataset de temperatura para simular uma situação de baixo nível de *stress*. A segunda, onde é usado o mesmo dataset, mas o pico de temperatura é repetido 2 vezes por hora, de forma a ser criada uma situação de *stress* de nível médio. Em último lugar, é novamente criado um dataset, mas desta vez com um dataset de humidade. O pico de *stress* é repetido 4 vezes por hora, de forma a replicar um período de *stress* alto.

4.2.1 Nível de *stress* baixo ou pontual

Um dos parâmetros de desempenho em avaliação é capacidade do algoritmo e-LiteSense em reduzir o número de eventos de *sensing*. De acordo com o que já foi exposto, a frequência de *sensing* é de 5s e a simulação foi executada durante 5 horas.

Como se pode verificar pela Figura 13, o algoritmo tem a capacidade de reduzir significativamente o número de eventos de *sensing*, sendo que para γ igual a 0.005 a redução é de 16%; para γ igual a 0.01 a redução é de 42%; para $0.1 \leq \gamma \leq 0.4$ mantém-se na ordem dos 52% e para o caso do γ igual a 0.5 a redução é de cerca de 60%. Pode-se também reparar que, com o aumento do γ aumenta também a redução do número de eventos de *sensing*. Isto deve-se ao facto de que, com o aumento do γ aumenta também a variação entre o desvio atual (\bar{V}_i) e o desvio anterior (\bar{V}_{i-1}) necessária para que o algoritmo diminua a frequência de *sensing* (ΔT_{next}).

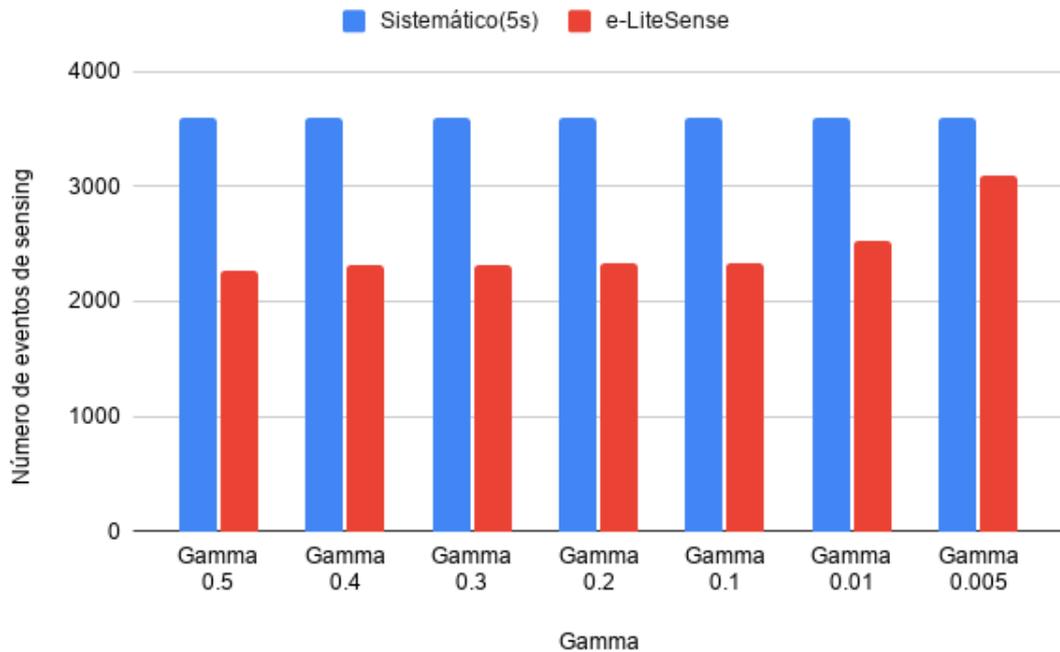


Figura 13: Número de eventos de *sensing* variando o gamma

No entanto, de forma a atingir a eficiência esperada, esta redução significativa no número de eventos de *sensing* deve ser acompanhada por medições precisas. Medições precisas consistem, mais concretamente, em produzir medidas estatisticamente representativas quando comparadas com a distribuição real dos parâmetros.

Consequentemente, a Tabela 2 destaca essa relação, apresentando os valores do EQM e da correlação para os diferentes valores atribuídos ao threshold γ . Pode-se visualizar que os valores apresentados na tabela mostram que para qualquer valor atribuído ao gamma, o algoritmo e-LiteSense foi capaz de obter uma precisão elevada.

Tabela 2: Precisão das medições - nível de *stress* baixo ou pontual

Gamma	EQM	Correlação
0.5	0.0944	0.9667
0.4	0.0745	0.9742
0.3	0.0347	0.9915
0.2	0.0196	0.9952
0.1	0.0139	0.9962
0.01	0.0013	0.9990
0.005	0.0001	0.9999

Para se obter uma percepção visual da distribuição dos valores obtidos pelo e-LiteSense, comparando diretamente com o dataset original, as diferentes figuras presentes no apêndice A.1 mostram que para $0.005 \leq \gamma \leq 0.4$, apesar de haver um aumento repentino da temperatura (situação de *stress*), o algoritmo e-LiteSense foi capaz de produzir uma distribuição que coincide na sua maioria com o dataset original (*sensing* sistemático a cada 5s), mas baseando-se num número significativamente menor de eventos de *sensing*. No entanto, para $\gamma \geq 0.5$ verifica-se que, apesar do baixo valor de EQM e de correlação (ver Tabela 2), no momento do pico de *stress* o algoritmo e-LiteSense não é capaz de captar todos os valores corretamente, como se pode confirmar na Figura 14.

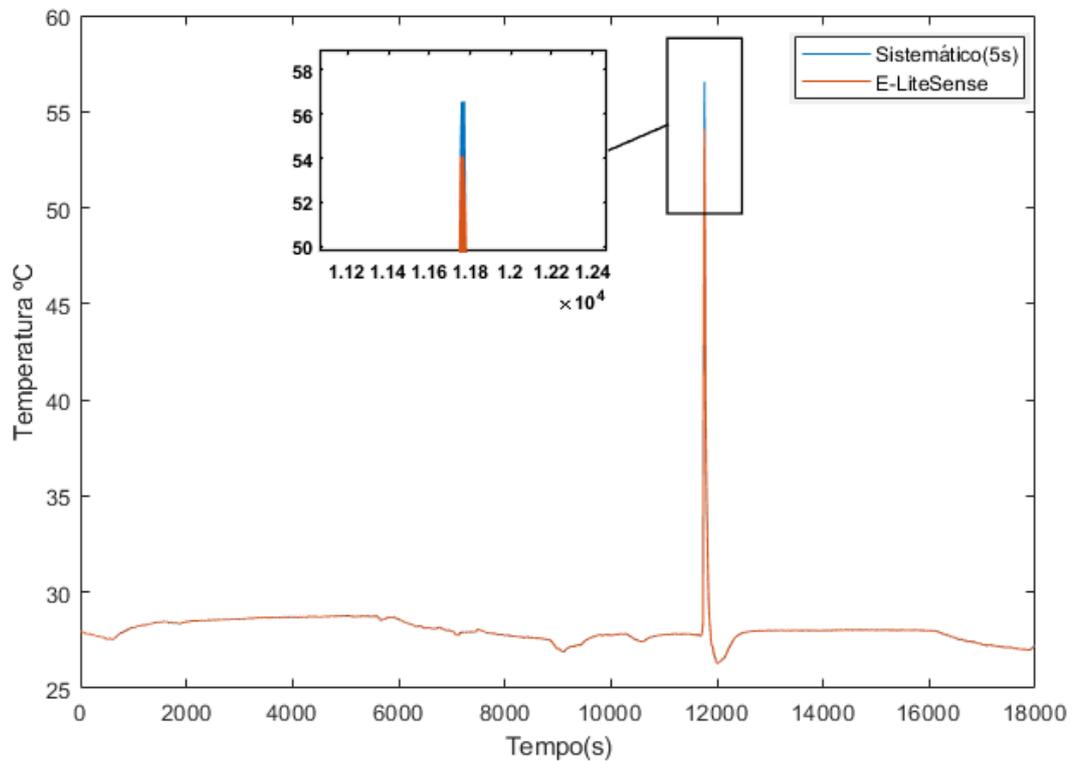


Figura 14: Comparação das leituras para gamma igual a 0.5

Como consequência da redução do número de eventos de *sensing* e, consequentemente, menos dados a serem processados e transmitidos, o algoritmo e-LiteSense promove também melhorias no consumo energético dos sensores. Como apresenta a Figura 15, o sensor no qual estava implementado o algoritmo e-LiteSense apresenta melhorias significativas no consumo energético quando comparado com o sensor com frequência de *sensing* sistemática (5s). Podemos também verificar que para $\gamma \geq 0.1$ não existe grande alteração no consumo energético do sensor com o algoritmo, devendo-se

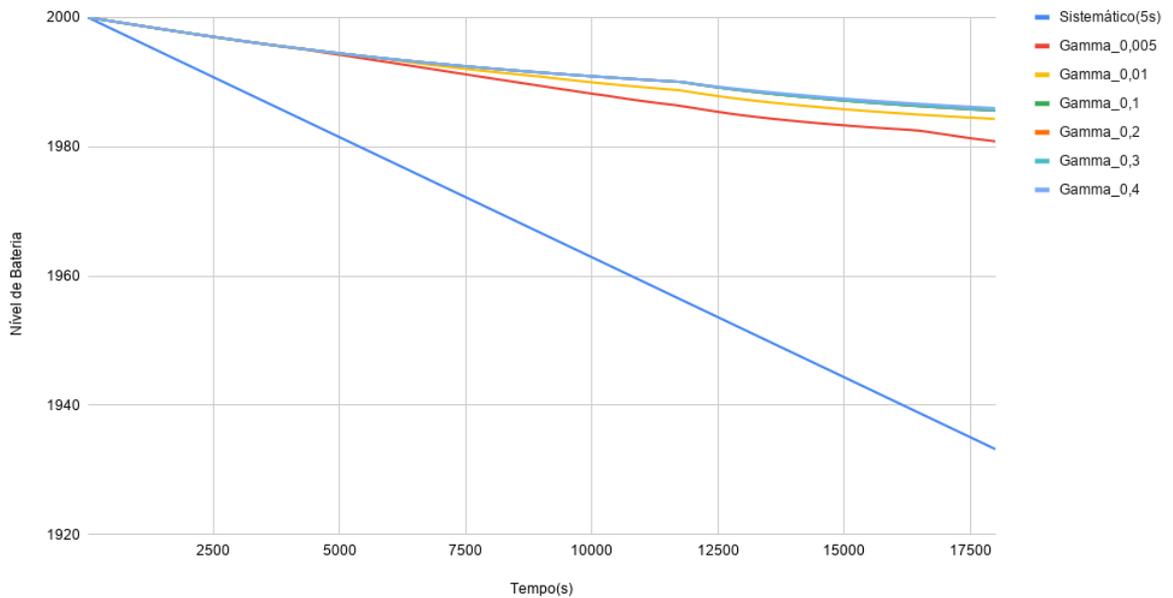


Figura 15: Consumo energético para os diferentes valores de gamma

isto ao facto de não existir grande diferença entre o número de eventos de *sensing* para esses valores de γ (ver Figura 13).

4.2.2 *Nível de stress médio*

Para o caso do estudo da variável γ em caso de *stress* médio, foi necessário criar um dataset que fosse capaz de simular essa situação de *stress*. Para isso, e na tentativa de criar um dataset o mais realista possível, a abordagem usada foi a de utilizar o dataset real usado na secção anterior (Sensor de Temperatura com frequência de 5s e um pico de *stress*) e replicar o pico de temperatura presente no mesmo. Esse pico de temperatura foi replicado duas vezes por hora.

Posto isto, podemos começar por avaliar o algoritmo a nível do *stress* médio no que diz respeito à sua capacidade de reduzir o *número de eventos de sensing*.

Como podemos verificar pela Figura 16, o algoritmo mantém a capacidade de reduzir significativamente o número de eventos de *sensing*, sendo que para $0.005 \leq \gamma \leq 1$ a redução mantém-se na ordem dos 15%; para o caso do γ igual a 1.5 a redução é de cerca de 22%; para γ igual a 2 é de 29%; para γ igual a 2.5 é de 31%; para γ igual a 3.5 é de 43% e para γ igual a 5 é de 45%. Podemos também reparar que com o aumento do γ aumenta também a redução do número de eventos de *sensing*, algo que também se verificou no caso da secção anterior.

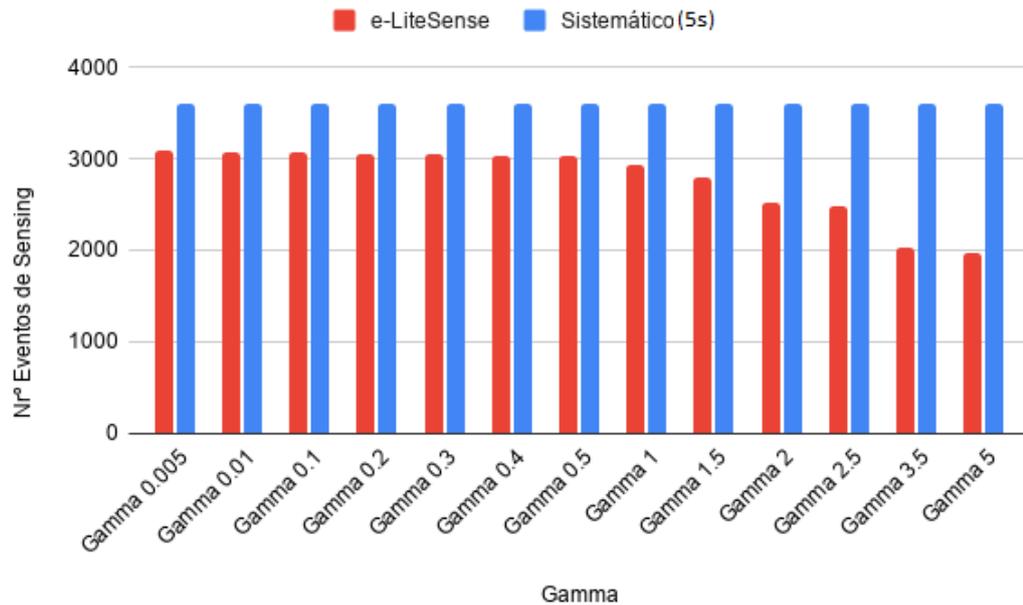


Figura 16: Número de eventos de *sensing* em função do parâmetro gamma

Mais uma vez, de forma a atingir o desempenho esperado, esta redução do número de eventos de *sensing* não deve afetar a precisão das medições. Assim, a Tabela 3 apresenta os valores obtidos nos cálculos do EQM e da correlação. A tabela destaca, novamente, o facto de, apesar da redução do número de eventos de *sensing*, o algoritmo continuar a conseguir manter a precisão das suas medições a um nível elevado para a grande maioria dos valores de γ .

Tabela 3: Precisão medições - nível de *stress* médio

Gamma	EQM	Correlação
0.005	0.00344	0.9998
0.01	0.00398	0.9998
0.1	0.0372	0.9983
0.2	0.0629	0.9973
0.3	0.1052	0.9954
0.4	0.1301	0.9930
0.5	0.1651	0.9915
1	0.1917	0.9913
1.5	0.2333	0.9911
2	0.2574	0.9909
2.5	0.4340	0.9879
3.5	0.4694	0.9803
5	0.5974	0.9743

Depois de analisada a capacidade do algoritmo em obter leituras precisas, mesmo com a redução do número de eventos de *sensing*, avaliou-se o impacto da alteração do γ no comportamento do algoritmo nos momentos de *bursts* de temperatura. Para tal ser possível, a Tabela 4 foi criada de forma a apresentar os resultados obtidos ao calcular o CV e o IDC, métricas apresentadas na secção anterior. Esta tabela mostra que, para $0.005 \leq \gamma \leq 1$ as leituras efetuadas pelo e-LiteSense têm uma variabilidade semelhante, mostrando por isso que o algoritmo se comporta de forma semelhante para esses valores, e indo também ao encontro dos valores obtidos na Figura 16. Para $\gamma \geq 1.5$ verificamos que existe uma diminuição nos valores do CV e do IDC, mostrando que para estes valores de γ e indo ao encontro dos valores da precisão das medições na Tabela 3, o algoritmo começa a mostrar incapacidade em conseguir captar de forma correta os parâmetros em observação.

Tabela 4: Métricas de *burstiness* - nível de *stress* médio

Gamma	CV	IDC
0.005	0.1237	0.4327
0.01	0.1250	0.4498
0.1	0.1262	0.4558
0.2	0.1258	0.4503
0.3	0.1254	0.4503
0.4	0.1244	0.4459
0.5	0.1242	0.4441
1	0.1245	0.4464
1.5	0.1208	0.4291
2	0.1188	0.4159
2.5	0.1173	0.4023
3.5	0.1167	0.3920
5	0.1162	0.3888

De forma a tentar entender melhor os resultados obtidos na tabela anterior, o rácio *Peak to Mean* é calculado e apresentado no Apêndice A.4. Para o cálculo desta métrica, o dataset foi dividido em séries temporais (30 min cada série) de forma a ajudar a avaliar de que maneira esta métrica evolui ao longo da simulação. Neste caso, a tabela mostra que para $0.005 \leq \gamma \leq 1$ o valor do *peak to mean* apresenta um valor estável, não mostrando grande variação ao longo da simulação. Para $\gamma \geq 1.5$ podemos visualizar que com o decorrer da simulação, em alguns dos períodos criados (períodos de 30 min), o valor do rácio *Peak to Mean* varia consideravelmente, dando a entender que o algoritmo começa a perder a capacidade de recolher as leituras do parâmetro em

observação corretamente, resultados estes que vão ao encontro dos resultados obtidos na Tabela 3.

Numa tentativa de obter confirmação dos resultados obtidos anteriormente e uma percepção visual da distribuição das leituras efetuadas pelo algoritmo e-LiteSense, são incluídos resultados complementares no Apêndice A.2 que comparam diretamente as leituras obtidas pelo e-LiteSense com o dataset original. Para $0.005 \leq \gamma \leq 1$ mostram que, apesar das diversas situações de *stress* presentes ao longo da simulação, o algoritmo é capaz de produzir leituras que coincidem na maioria das ocasiões, com o dataset original, embora essas leituras sejam baseadas num menor número de eventos de *sensing*. Porém, como apresenta a Figura 17, e as incluídas no apêndice A.2, pode-se verificar que para $\gamma \geq 1.5$, em alguns dos momentos do pico de *stress* do parâmetro em observação, a partir desse valor o algoritmo mostra sinais de incapacidade em obter os valores corretamente, indo, por isso, ao encontro dos valores obtidos no cálculo das métricas de burstiness.

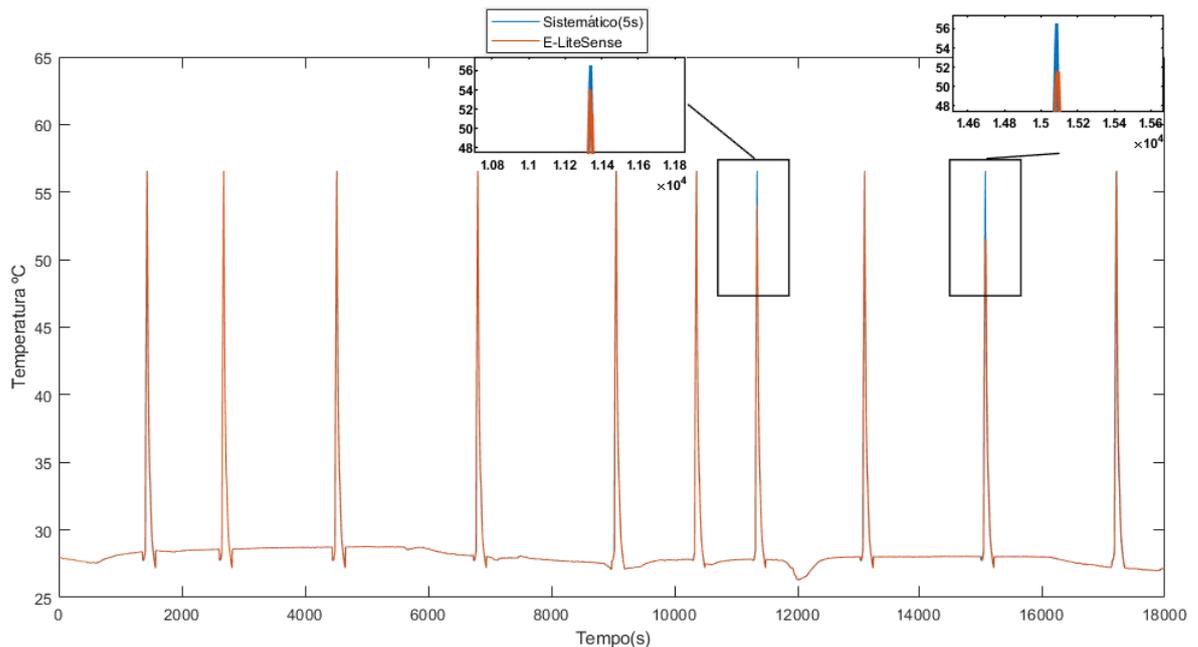


Figura 17: Comparação das leituras para gamma igual a 1.5

Reforçando o que já vem sendo visto, devido à redução do número de eventos de *sensing* e, como resultado, à existência de um menor volume de dados a serem processados e transmitidos, o e-LiteSense promove também reduções no consumo energético dos diferentes sensores. A Figura 18 representa o consumo energético comparando diretamente o consumo do sensor onde é implementado o algoritmo

e-LiteSense com o sensor normal (frequência de 5s). Podemos verificar que, para todos os valores atribuídos ao parâmetro γ , o sensor com o algoritmo apresenta melhorias no consumo energético. Reparamos também que, para $0.1 \leq \gamma \leq 1.5$ não existe grande variação no consumo energético, devendo-se isto ao facto de não existir grande diferença entre o número de eventos de *sensing* para esses valores de γ .

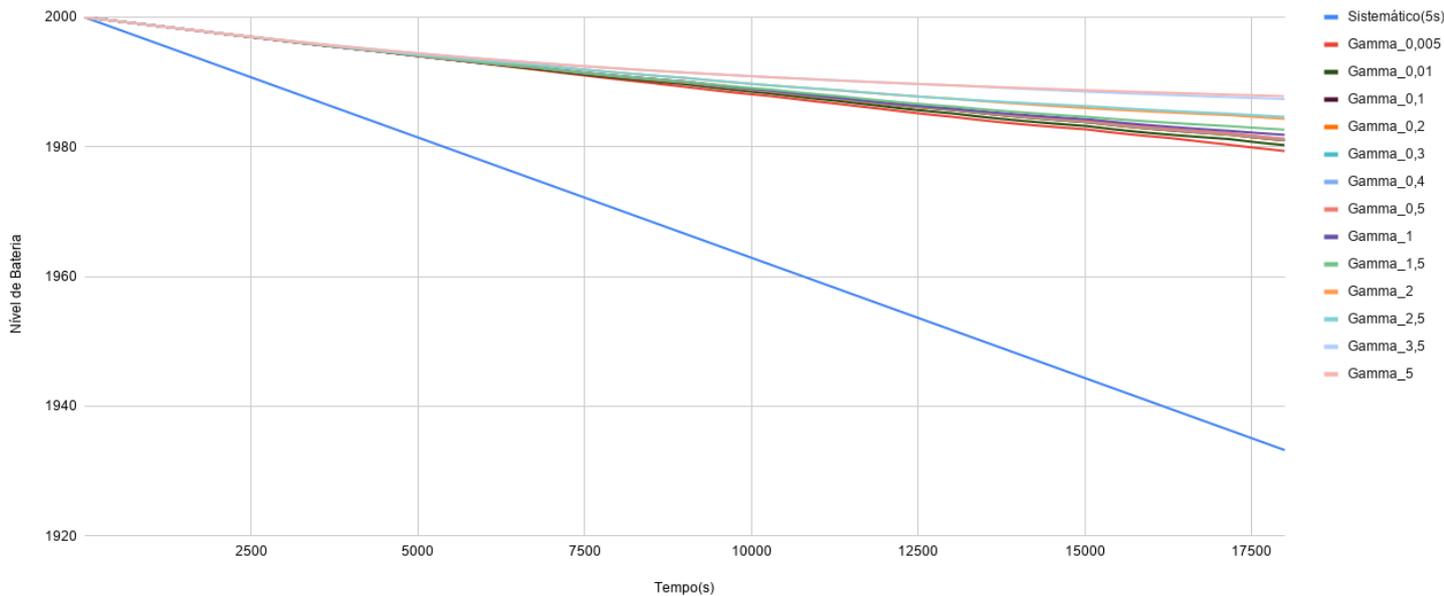


Figura 18: Consumo energético para os diferentes valores de gamma

4.2.3 *Nível de stress alto*

Para o caso de estudo da variável γ em caso de *stress* alto foi, uma vez mais, necessário construir o dataset para usar na simulação. Para tal ser possível, e com o objetivo de tentar replicar uma situação o mais real possível, utilizamos um dataset recolhido previamente (sensor de humidade com frequência de 5s), dataset este já com um burst de humidade, e replicamos esse pico de humidade 4 vezes por hora, tentando assim simular uma situação de *stress* elevado.

Neste contexto, a Figura 19 apresenta a capacidade do algoritmo em reduzir o número de eventos de *sensing*. Como podemos confirmar, o algoritmo e-LiteSense mantém a capacidade de reduzir o número de eventos de *sensing* para qualquer dos valores atribuídos ao parâmetro γ . Mais concretamente, 2% para γ igual a 0.005, 5% para γ igual a 0.01; na ordem dos 8% para $0.1 \leq \gamma \leq 1.5$; 15% para $2 \leq \gamma \leq 2.5$; 38% para γ igual a 3.5 e 44% para γ igual a 5. Podemos, novamente, verificar que com o aumento do

γ aumenta também a redução do número de eventos de *sensing*, bem como, a diferença entre o valor obtido na leitura atual e o valor obtido na leitura anterior que é necessária para reduzir o valor do ΔT_{next} .

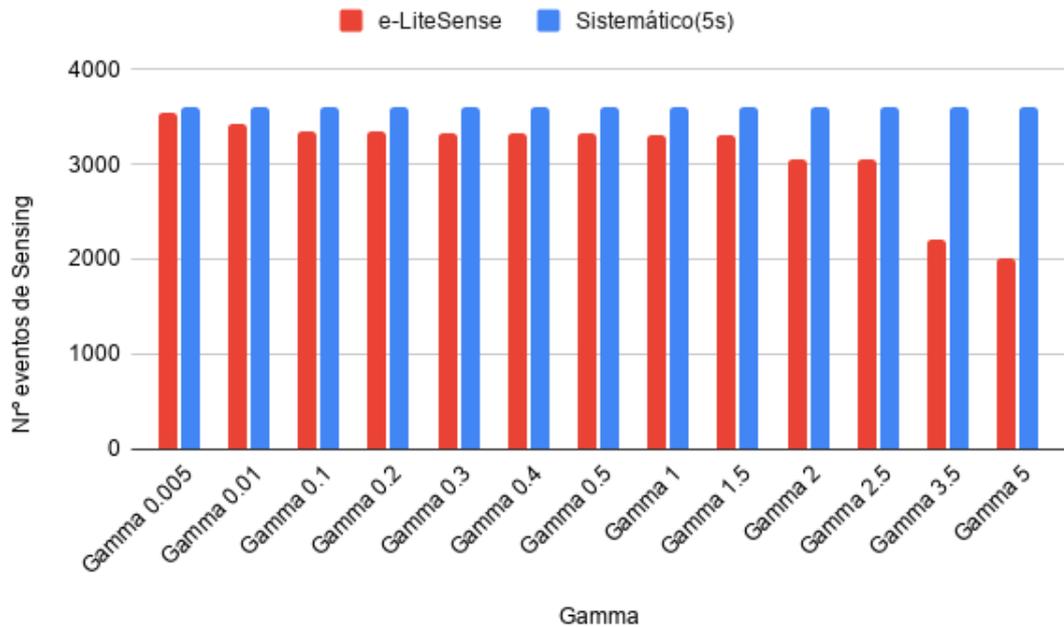


Figura 19: Número de eventos de *sensing* variando o gamma

Esta redução no número de eventos de *sensing* deve ser, assim como vem sendo referido, acompanhada por leituras com alta precisão, de forma a manter o comportamento que é esperado pelo algoritmo. Desta maneira, a Tabela 5 apresenta os valores obtidos para o EQM e a correlação. Podemos verificar que para $\gamma \leq 5$ o valor do MSE obtido é significativamente reduzido, conseguindo assim o algoritmo manter medições precisas apesar da redução do número de eventos de *sensing*.

Concluiu-se que seria interessante analisar de que forma o algoritmo se comporta nos momentos de *bursts* de temperatura depois de analisada, novamente, a precisão das leituras obtidas pelo algoritmo e-LiteSense, apesar do facto de existir uma redução do número de eventos de *sensing*. Para isso, foi criada a Tabela 6, contendo os valores obtidos para as métricas de *burstiness* (CV e IDC). Como se observa, para $0.005 \leq \gamma \leq 2.5$, o valor das métricas tem uma variabilidade reduzida, mantendo um valor semelhante para os diferentes valores de γ , mostrando assim que, para esses valores de γ , o algoritmo se comporta de forma semelhante.

Tabela 5: Precisão das medições - nível de *stress* alto

Gamma	EQM	Correlação
0.005	0.0057	0.9999
0.01	0.0042	0.9999
0.1	0.0065	0.9998
0.2	0.0186	0.9998
0.3	0.0215	0.9998
0.4	0.0223	0.9998
0.5	0.0249	0.9997
1	0.0834	0.9996
1.5	0.2849	0.9989
2	0.2922	0.9988
2.5	0.3018	0.9988
3.5	0.7928	0.9969
5	1.1971	0.9954

Tabela 6: Métricas de burstiness - nível de *stress* alto

Gamma	C	IDC
0.005	0.2395	2.8266
0.01	0.2400	2.8409
0.1	0.2427	2.8558
0.2	0.24305	2.8632
0.3	0.2432	2.8690
0.4	0.2429	2.8608
0.5	0.2434	2.8738
1	0.2435	2.8733
1.5	0.2406	2.8579
2	0.2406	2.8387
2.5	0.2407	2.8297
3.5	0.2064	2.7036
5	0.1912	2.6544

No entanto, para $\gamma \geq 3.5$, reparamos pela redução dos valores das métricas que o algoritmo apresenta um comportamento diferente, dando assim a entender que o algoritmo começa a perder a capacidade de obter leituras precisas e corretas.

Por forma a tentar auxiliar a análise dos dados obtidos na tabela anterior, foi calculado o rácio Peak to Mean e apresentado na Tabela presente no Apêndice A.5. Reitera-se que para o cálculo desta métrica o dataset foi dividido em períodos de 15 minutos de forma a perceber a evolução das métricas com o decorrer da simulação. Consequentemente, a tabela indica que, para $0.005 \leq \gamma \leq 2.5$, mais uma vez, não existe

grande variação do valor de Peak to Mean, indicando que ao longo da simulação o algoritmo apresenta um comportamento semelhante e aparentemente correto para esses valores de γ . Para $\gamma \geq 3.5$ podemos reparar que em alguns dos períodos de 15 minutos criados o valor do Peak to Mean varia de forma considerável, dando assim indicadores de que o algoritmo, com o decorrer da simulação, começa a perder a capacidade de obter todos os valores do parâmetro em observação corretamente.

Numa tentativa de confirmar visualmente os resultados das métricas de precisão e de burstiness das medições, as figuras presentes no apêndice A.3 comparam diretamente as leituras obtidas pelo algoritmo com o dataset original. Para $0.005 \leq \gamma \leq 2.5$, verificamos que o algoritmo, na maioria das ocasiões, consegue captar o parâmetro em observação corretamente, produzindo, deste modo, leituras que coincidem com o dataset original.

Contudo, como mostram algumas das imagens presentes no apêndice A.3 e como representa a Figura 20, verificamos que para $\gamma \geq 3.5$, em diversos momentos de picos de humidade, o algoritmo não consegue acompanhar corretamente as variações sofridas pelo parâmetro em observação. Esta confirmação visual vai ao encontro dos resultados obtidos anteriormente, resultados estes que, a partir desse valor de γ , mostram que o algoritmo apresenta uma redução na precisão e alguma incapacidade de em certos momentos capturar corretamente o parâmetro em observação.

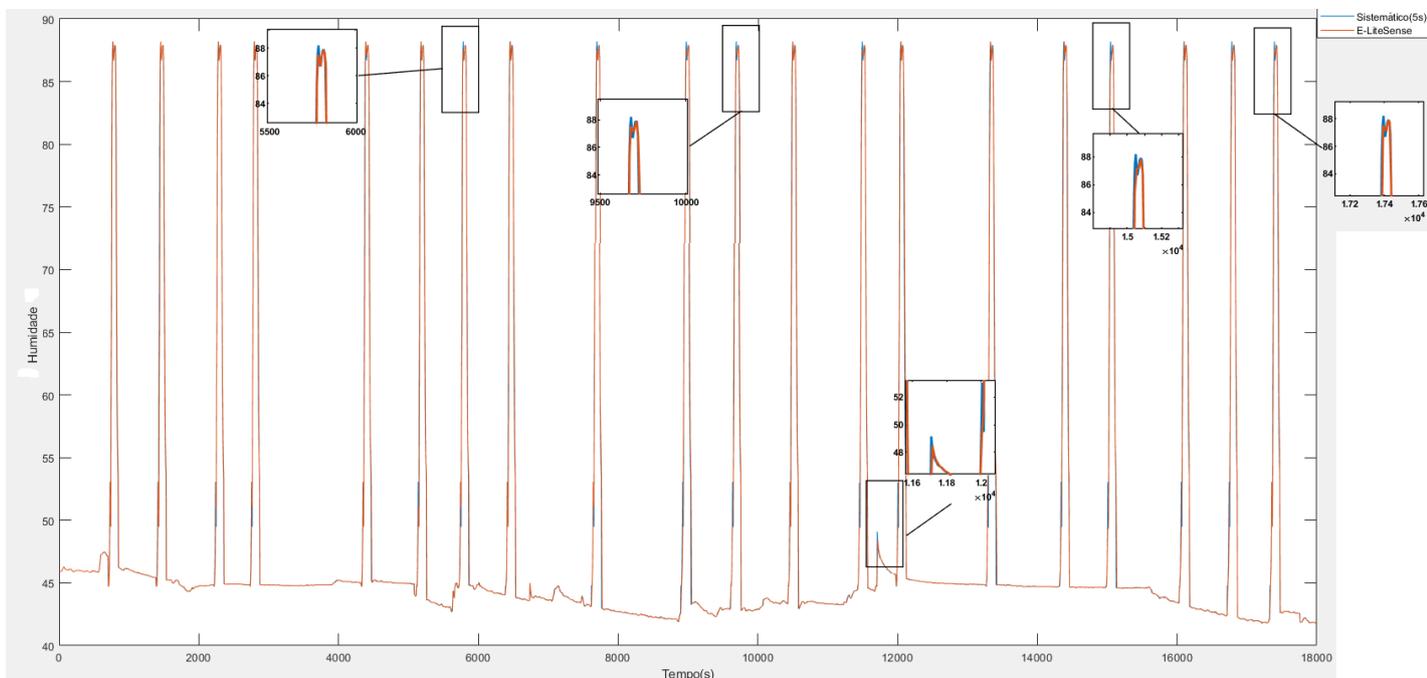


Figura 20: Comparação das leituras para gamma igual a 3.5

Reforçadamente, devido à capacidade do algoritmo em reduzir o número de eventos de *sensing*, existe também uma redução no número de dados a precisarem de transmissão e processamento. Posto isto, é necessário perceber se essa redução é sinónimo de melhoria de consumo energético dos diferentes sensores. A Figura 21 apresenta a comparação entre o consumo energético do sensor com frequência de 5 segundos e o sensor com o algoritmo e-LiteSense implementado. Verificamos novamente que para todos os valores atribuídos à variável γ o algoritmo apresenta melhoria no consumo energético para o caso de *stress* alto. Podemos também reparar que, para $0.1 \leq \gamma \leq 1.5$ não existe praticamente nenhuma variação no consumo energético, sendo este comportamento esperado pois, como mostra a Figura 19, não existe grande variação do número de eventos de *sensing* para estes valores de γ .

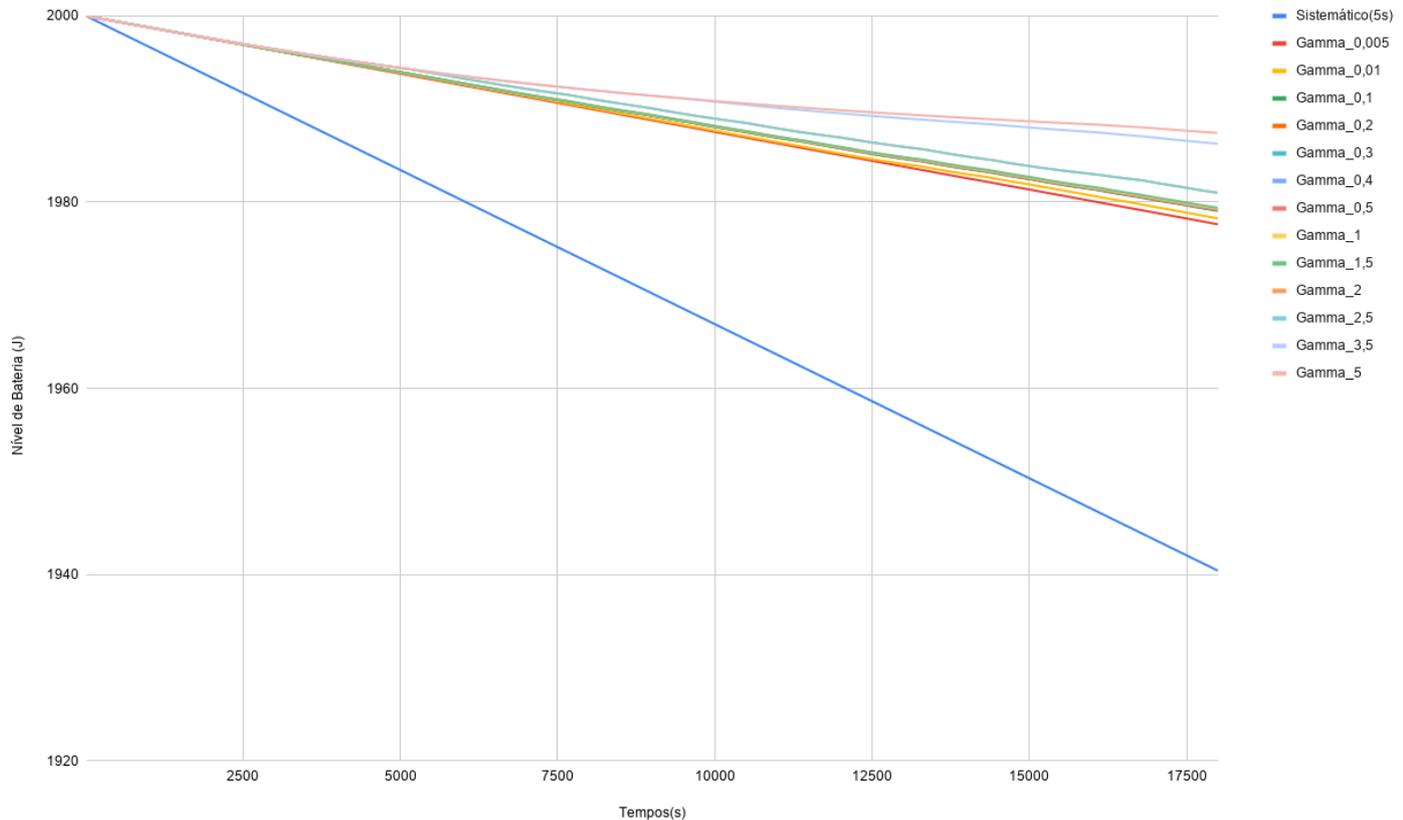


Figura 21: Consumo energético para os diferentes valores de gamma

4.3 ESTUDO DA VERSATILIDADE DO ALGORITMO

De forma a continuar a realizar o estudo ao algoritmo e-LiteSense, esta secção apresenta um estudo à versatilidade do algoritmo. Este estudo é importante pois poderá ajudar a entender se o algoritmo é capaz de funcionar corretamente numa diversidade de cenários. De forma a avaliar o desempenho do e-LiteSense em diferentes cenários, ambos os esquemas de *sensing* (sistemático / auto-adaptativo) são aplicados a datasets públicos de diferentes áreas de aplicação e com características distintas, como descrito e sumarizado na Tabela 7.

Tabela 7: Identificação e descrição dos datasets usados

Dataset	Descrição	Simulação/Freq. Sensing
Longitude e Latitude	Trajetórias GPS, contendo informação da latitude e longitude	5Horas / 3 segundos
Velocidade do vento(m/s)	Velocidade do vento em Kansas City	7 Horas / 30 s
Voltagem	Consumo energético de uma residência	8Horas / 60 s
MonCarbon	Medições da poluição do ar	10 horas / 5min
Ozone	Medições da poluição do ar	15 horas / 5min

Em relação à avaliação do desempenho, esta passa por determinar: (i) o número de eventos de *sensing* ao longo da simulação; (ii) a precisão em identificar as variações temporais dos parâmetros em observação e (iii) o consumo energético dos diferentes sensores. Para estimarmos a precisão das medições serão usadas as mesmas métricas da secção anterior (EQM e Correlação). Para o cálculo do EQM foi uma vez mais necessário preencher os valores em "falta", sendo que, para isso foi usada a abordagem apresentada na Secção 4.1. Será também apresentado o consumo energético dos diferentes sensores (sensor com *sensing* sistemático vs sensor com algoritmo e-LiteSense).

Depois de apresentar os datasets que irão ser usados nas diferentes simulações e as métricas de avaliação do desempenho, é necessário apresentar a configuração do algoritmo para os diferentes testes. Mais uma vez, esta configuração foi selecionada com base experimental (estudo prévio ao algoritmo), considerando, para o caso da **Latitude/Longitude**, um $\Delta T_{min} = 3$ e $\Delta T_{max} = 30$, que significa que a frequência de *sensing* mínima será de 3 segundos e pelo menos um evento de *sensing* ocorre a cada 30 segundos. Para o da **Velocidade do Vento** um $\Delta T_{min} = 30$ e $\Delta T_{max} = 270$, significa que a frequência de *sensing* mínima será de 30 segundos e pelo menos um evento de *sensing* ocorre a cada 4 minutos e 30 segundos. Para o caso da **Voltagem**, um $\Delta T_{min} = 60$ e $\Delta T_{max} = 540$, significa que a frequência de *sensing* mínima será de 60 segundos e

pelo menos um evento de *sensing* ocorre a cada 9 minutos. Para o caso do **Monóxido de Carbono/Ozono**, um $\Delta T_{min} = 300$ e $\Delta T_{max} = 2700$, significa que a frequência de *sensing* mínima será de 60 segundos e pelo menos um evento de *sensing* ocorre a cada 45 minutos. Para além desta configuração inicial, os limites ao fator da escala de reatividade são definidos como $\epsilon_{min} = 0.05$ e $\epsilon_{max} = 0.30$. Para além disso, o peso das observações anteriores \bar{X}_{i-1} e \bar{V}_{i-1} em \bar{X}_i e \bar{V}_i (ver algoritmo 1) são $\alpha = 0.7$ e $\beta = 0.7$. Para finalizar, a variável gamma γ irá ser diferente para cada cenário em questão, sendo que, os valores para esta variável foram definidos de forma a abranger o acordo com o estudo experimental realizado na secção 4.2. Neste estudo cheguei à conclusão de que para casos em o nível de stress é baixo (como é o caso dos datasets usados), para $\gamma \leq 0.5$ o algoritmo consegue obter os resultados de forma precisa. Sendo assim, e de forma a testar com diferentes valores, para o dataset de **Longitude** o γ é de 0.005, para o de **Latitude** o γ é de 0.01, para o dataset da **Velocidade do Vento** o γ é de 0.1, para o dataset da **Voltagem** o γ é de 0.2 e para o dataset da **Mon.Carbono/Ozono** o γ é de 0.4.

4.4 ANÁLISE DOS RESULTADOS

Como supramencionado, o primeiro parâmetro de desempenho em avaliação é a capacidade do algoritmo em reduzir o número de eventos de *sensing* necessários para recolher corretamente os parâmetros em observação. Tendo isso em consideração, a Figura 22 apresenta os resultados dessa avaliação de desempenho. Podemos perceber que, para todos os cenários em questão, o algoritmo consegue reduzir o número de eventos de *sensing* significativamente. Mais concretamente, na ordem dos 60% tanto para o cenário da Latitude como da Longitude, 23% para o caso da Velocidade do Vento, 20% para o caso da Voltagem, e na casa dos 5% para os casos do Monóxido de Carbono e Ozono.

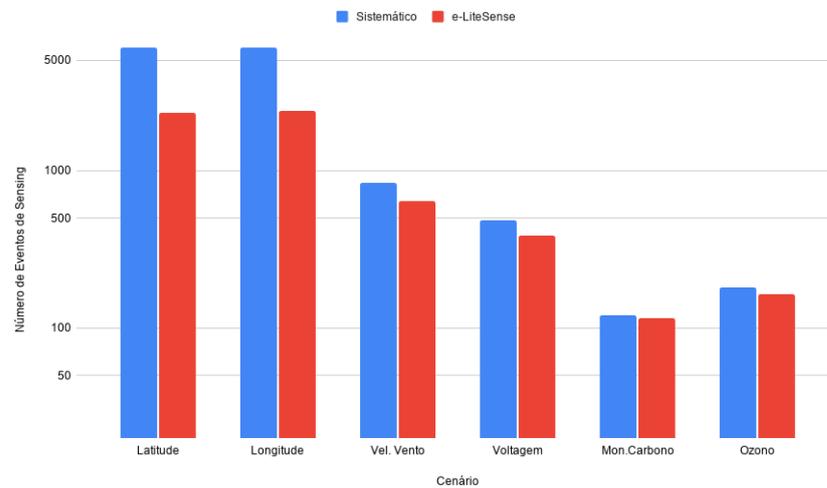


Figura 22: Redução do número de eventos de *sensing* para os diferentes cenários

Verifica-se também que o algoritmo e-LiteSense consegue obter um maior impacto na redução do número de eventos de *sensing* quanto menor for a frequência de *sensing*. Este facto era algo espectável, pois quanto maior for o número de eventos de *sensing* maior será o número de vezes que o algoritmo e-LiteSense é executado e, conseqüentemente, maior será a possibilidade de o e-LiteSense ter um impacto mais elevado.

Apesar de existir esta redução significativa do número de eventos de *sensing*, e de forma a conseguirmos atingir o desempenho esperado, esta redução deve ser acompanhada por leituras precisas dos parâmetros em observação. Logo, de forma a comparar diretamente o dataset original com as leituras obtidas pelo e-LiteSense, as figuras presentes no Apêndice A.6, 23 e 24 mostram precisamente essa relação, onde podemos verificar que para todos os cenários em questão, o algoritmo é capaz de obter medições que coincidem com o dataset original, mesmo estas tendo sido obtidas usando um número menor de eventos de *sensing*.

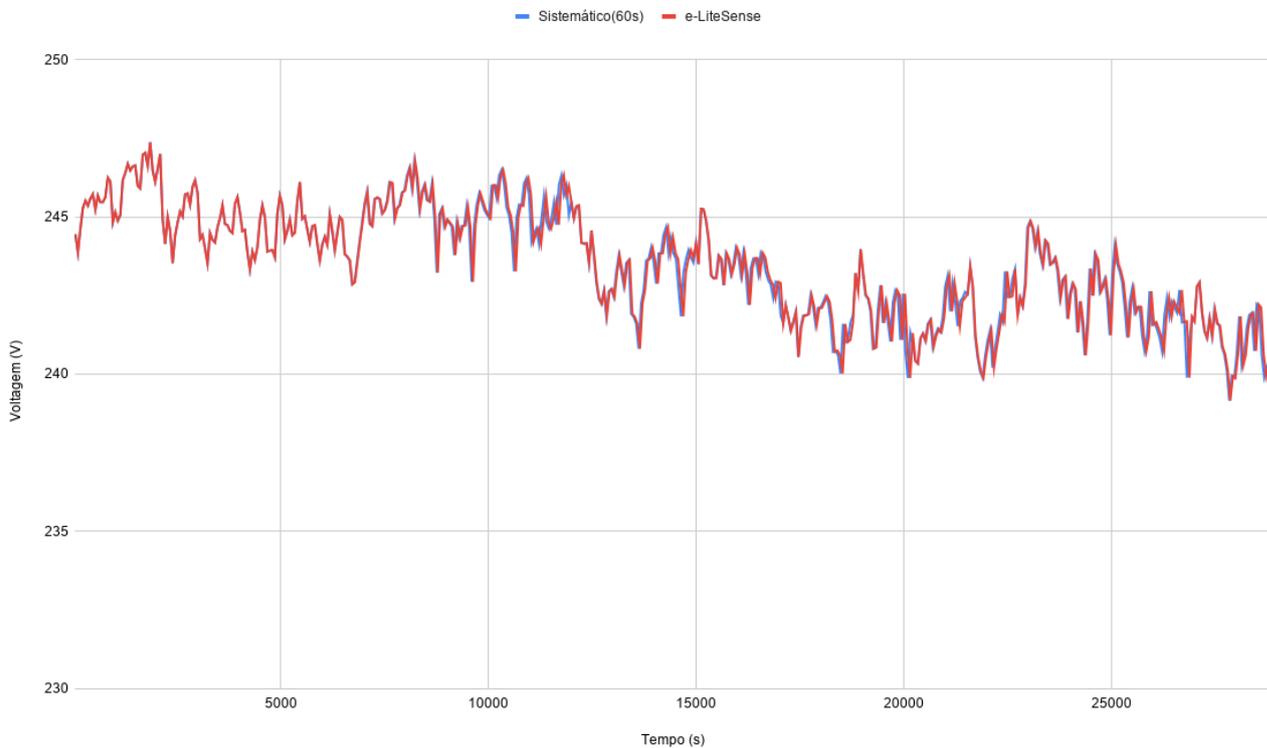


Figura 23: Comparação das leituras para o dataset da Voltagem

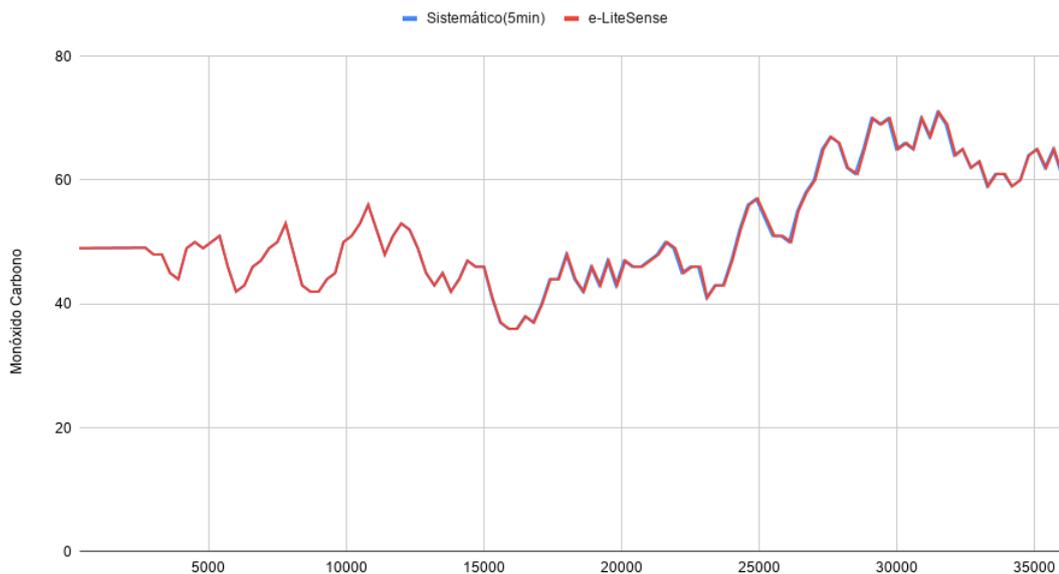


Figura 24: Comparação das leituras para o dataset de Monóxido de Carbono

De forma a confirmar estatisticamente os resultados apresentados anteriormente, na Tabela 8 incluem-se os resultados obtidos para o EQM e do Coeficiente de Correlação. A tabela mostra que para todos os cenários avaliados o algoritmo é capaz de obter medições relativamente precisas. Assim, valida-se a percepção visual dos resultados obtidos anteriormente, demonstrando a eficiência do comportamento auto-adaptativo do e-LiteSense, de acordo com as variações dos fenómenos em avaliação.

Tabela 8: Precisão das medições - versatilidade

	EQM	Correlação
Latitude	0.0281	0.9999
Longitude	0.0381	0.9999
Vel. Vento	0.1806	0.9388
Voltagem	0.1091	0.9817
Mon.Carbono	0.216	0.9986
Ozono	0.3277	0.9992

Com a redução do número de eventos de *sensing*, e consequentemente a necessidade de processamento e transmissão de dados ser menor, o algoritmo e-LiteSense promove também melhorias no consumo energético dos sensores. Como apresentado nas Figura 25, 26 e nas restantes figuras presentes no Apêndice A.7, quando comparado diretamente com os cenários operacionais dos diferentes datasets, o algoritmo e-LiteSense consegue

obter uma melhoria no consumo energético para os diferentes datasets, confirmando, portanto, a eficiência do mesmo. Podemos também verificar que com o aumento da frequência de *sensing* o impacto do e-LiteSense é reduzido, indo assim ao encontro dos resultados obtidos anteriormente. Isto deve-se ao facto de quanto maior for a frequência de *sensing*, menor serão as vezes que o algoritmo e-LiteSense é executado e, consequentemente, o impacto da execução do mesmo é reduzido.

4.5 RESUMO

Inicialmente, neste capítulo foi feita uma apresentação dos diferentes testes realizados, começando por enumerar os datasets usados tanto no estudo do parâmetro γ como no estudo da versatilidade do algoritmo. Seguidamente, para além de serem enumeradas as métricas usadas para avaliação do algoritmo nos diferentes testes, é também apresentada a configuração do algoritmo na execução das diferentes simulações.

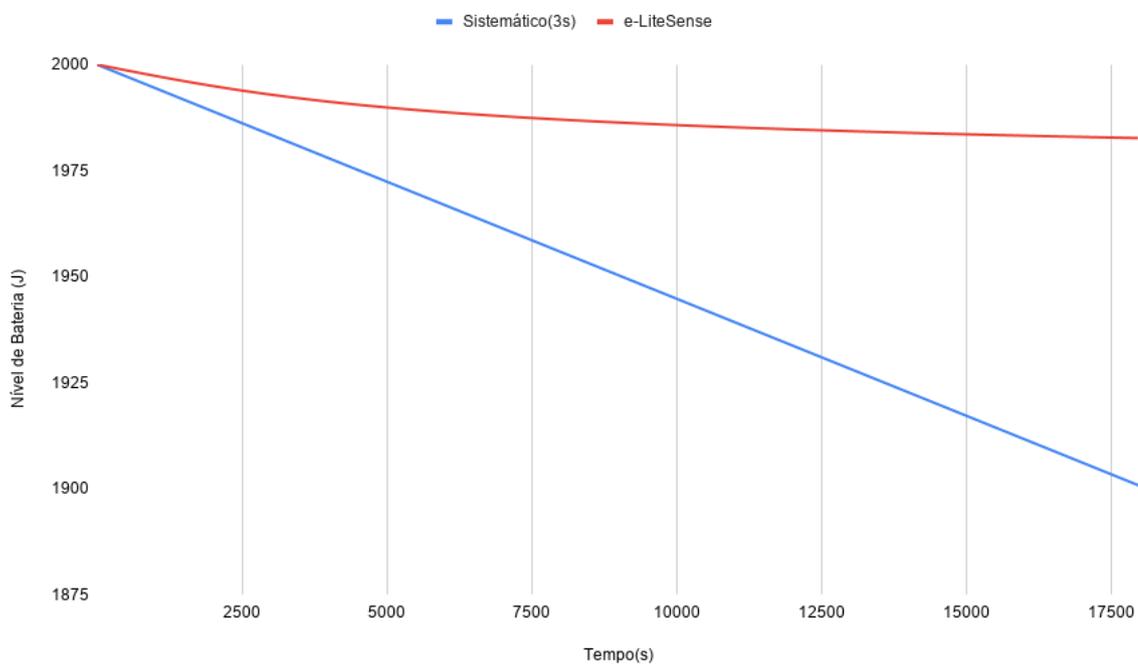


Figura 25: Comparação do consumo energético para o dataset de Latitude

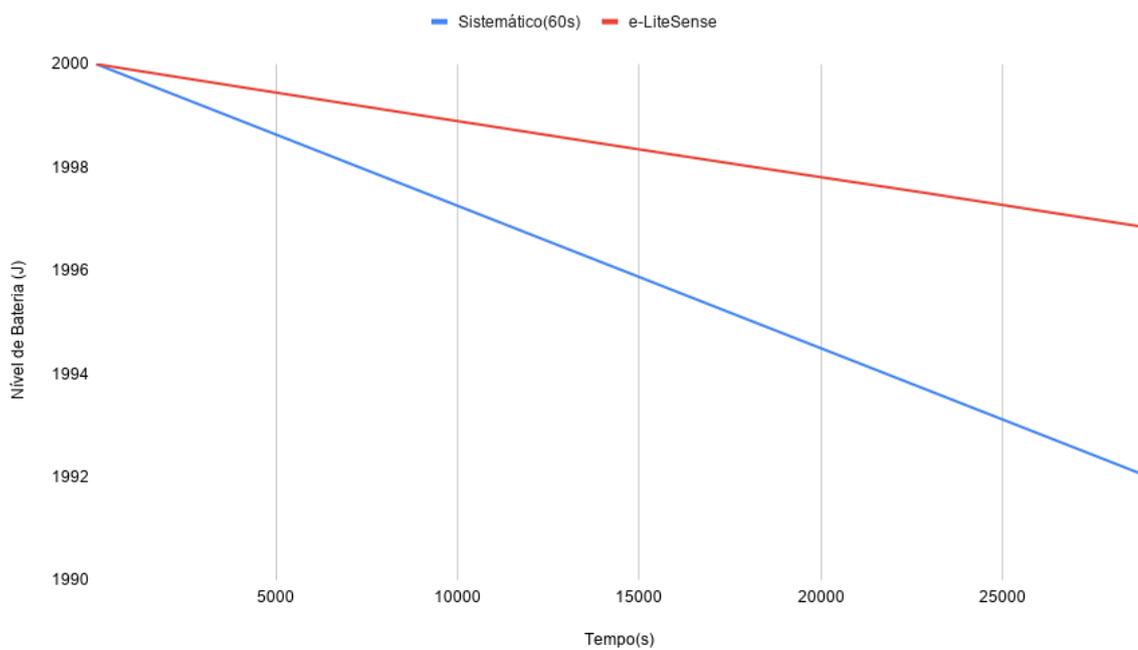


Figura 26: Comparação do consumo energético para o dataset da Voltagem

Depois de explanada a planificação dos testes, foram apresentados os resultados obtidos nos testes de *stress* baixo ou pontual, médio e alto. Para a avaliação de desempenho do algoritmo nestes testes foram usadas as seguintes métricas : EQM, Coeficiente de Correlação, CV, IDC e também foi feita a comparação do consumo energético entre os diferentes sensores. Para o caso do nível de *stress* baixo ou pontual, médio e alto, os resultados mostram que o algoritmo funciona corretamente, na maioria dos casos. Porém, os resultados mostram também que nos casos de nível baixo, médio e alto o algoritmo começa a perder a capacidade de captar corretamente os parâmetros em observação, para valores de $\gamma \geq 0.5$, $\gamma \geq 1.5$ e $\gamma \geq 3.5$, respetivamente.

Por último, foram apresentados os testes realizados no estudo da versatilidade do algoritmo. Os resultados obtidos mostram que para os diferentes datasets(Latitude, Longitude, Vel. Vento, Voltagem, Mon.Carbono, Ozono) usados nas simulações, o algoritmo e-LiteSense é capaz de reduzir o número de eventos de *sensing*, mantendo a precisão das medições obtidas e reduzindo o consumo energético da rede comparando diretamente com o consumo do sensor com uma abordagem sistemática.

PROVA DE CONCEITO

O objetivo principal do algoritmo e-LiteSense é o de reduzir o consumo energético da RSSF, mantendo a precisão nas medições dos parâmetros em observação. Assim, o objetivo da prova de conceito é o de analisar o desempenho do algoritmo e-LiteSense num cenário de simulação que se enquadre mais com a realidade. Para isso, é usado um Sistema de Rega Inteligente, previamente implementado no CupCarbon. Posto isto, inicialmente será feita uma introdução ao caso de estudo, seguindo-se então a análise à eficiência do esquema adaptativo proposto.

5.1 SISTEMA DE REGA INTELIGENTE

O sistema de rega utilizado é constituído por diversos nós Sink, sendo que estes comunicam com um nó Sink principal através do método de comunicação de grande distância LoRa. Este protocolo é utilizado porque é ideal para fazer a transferência de dados com o nó Sink principal a distâncias maiores.

Adicionalmente foram inseridos 5 sensores nas proximidades do seu respetivo nó Sink, usando o protocolo de comunicação ZigBee [33]. Este protocolo é usado porque a distância entre os sensores e o gateway é reduzida, permitindo assim uma melhor eficiência energética na transferência de dados. Consequentemente, é adicionado outro método de comunicação para o nó Sink para que este possa comunicar tanto com os sensores como com o nó principal.

No simulador CupCarbon, quando se pretende que os sensores enviem dados relativos, por exemplo, ao solo, torna-se necessário conetar cada um deles a um gás, ou seja, a um gerador de eventos analógicos. Este gerador tanto pode ser associado a um *script* que simula valores reais criados por um gerador de eventos naturais, como pode ser associado a um dataset real para uso na simulação. Nesta simulação os valores simulados correspondem aos parâmetros de temperatura, humidade, luminosidade,

pH e fluxo de água. Para o caso da temperatura e da humidade os datasets são reais, enquanto que para os restantes parâmetros os datasets foram obtidos usando o simulador de valores reais do CupCarbon.

A frequência de *sensing* dos diferentes sensores é de 5 segundos, excepto para o caso do fluxo de água que é de 4 segundos.

Foi seguidamente implementado um sistema do controlo da irrigação, decidindo quando deve ou não ser ligado/desligado o sistema de rega, como mostra a imagem seguinte:

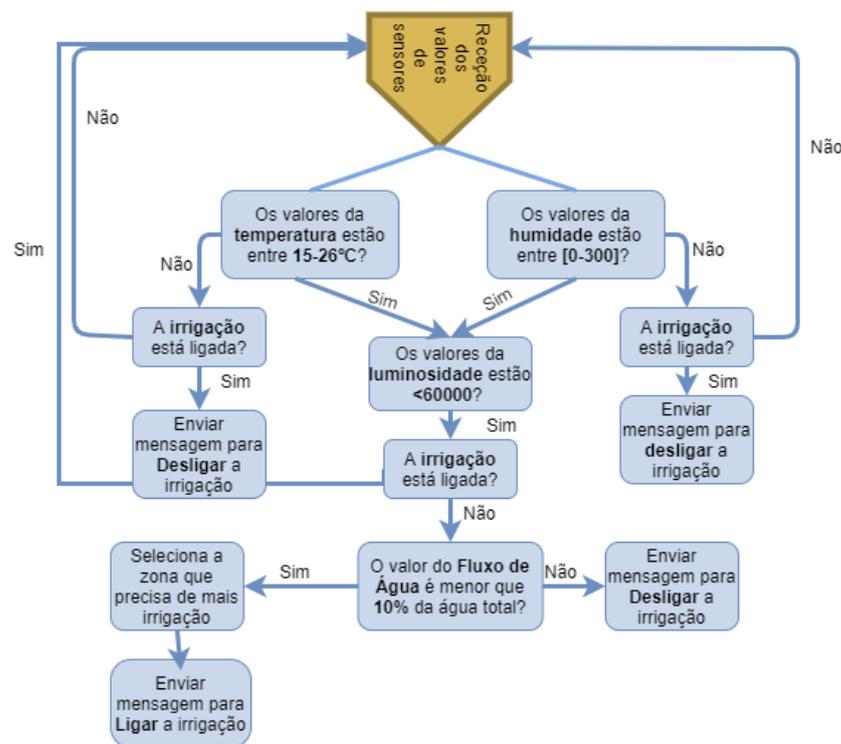


Figura 27: Algoritmo de controlo da irrigação [10]

Para a implementação deste controlo de irrigação, foi necessário ter em conta algumas condições no que diz respeito aos parâmetros em observação, de forma a ativar/desativar o sistema de rega [36] [37]:

- **Temperatura:** Foi considerada que a melhor temperatura para irrigar o solo era entre os 15°C e os 26°C . Se a temperatura estiver abaixo desse valor o solo poderá estar demasiado frio dificultando a infiltração da água. Acima desse valor a água poderá ser evaporada, mantendo-se assim o solo seco.
- **Humidade:** Entre [0%,42%] o solo está seco; solo húmido está entre os [43%,95%]; solo é considerado saturado em água quando os valores estão entre [95%, 100%].

- **Luminosidade:** Leituras acima dos 60000 lux são obtidas quando o sol está a brilhar ao máximo. Assim sendo, a altura ideal considerada para ligar o sistema de rega é quando a luminosidade está abaixo do valor referido.
- **pH:** Foi considerado que o solo possui os nutrientes principais quando o seu pH está perto de neutro, entre os 6.6 - 7.3.

Posteriormente, os sensores que representam os aspersores são adicionados para garantir que o algoritmo decide, com base nos valores que lê, se os aspersores são ativados ou desativados.

Numa última fase, foram consideradas diferentes zonas de intervenção, sendo que os sensores e aspersores são duplicados, com exceção do sensor do fluxo de água, de forma a identificar qual a zona que precisa de ser regada, sendo apenas esses aspersores ativados.

Posto isto, o cenário de simulação final é apresentado na Figura 28:

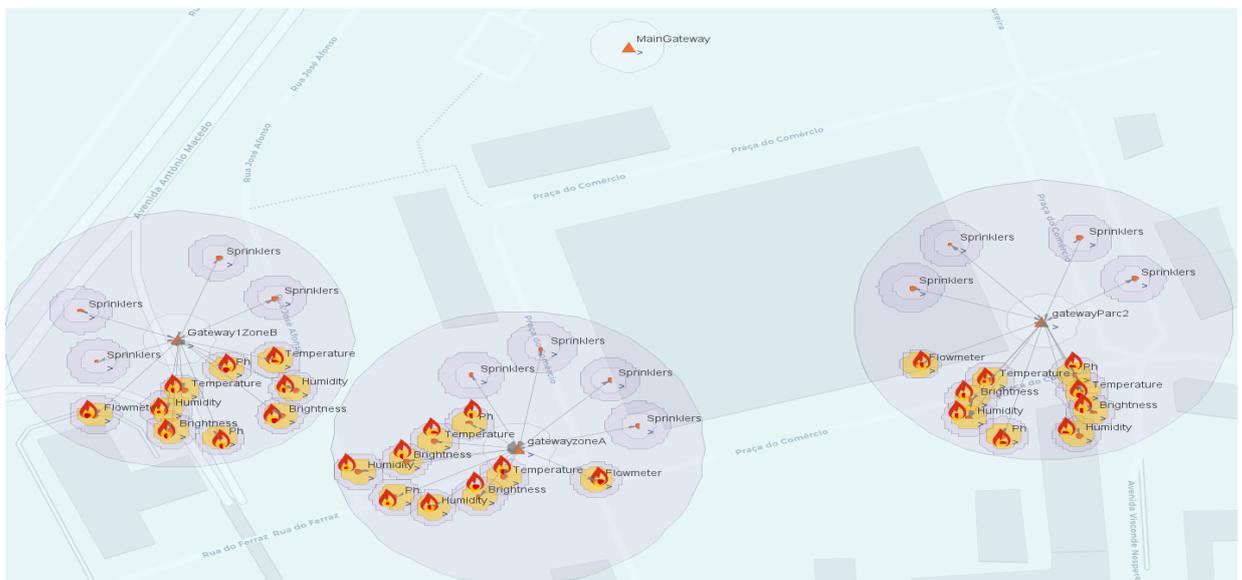


Figura 28: Cenário de Simulação

5.2 CENÁRIO DE TESTE

Como apresentado na secção acima, o cenário para a prova de conceito é um Sistema de Rega Inteligente implementado no CupCarbon. De forma a avaliarmos o desempenho do algoritmo, será efetuada a comparação das precisões das medições obtidas e do consumo energético entre o sistema de rega original (com uma frequência

constante de sensoriamento) e o sistema rega com o algoritmo adaptativo e-LiteSense implementado.

Posto isto, na Tabela 9 são apresentados os datasets usados para os diferentes tipos de sensores usados nesta simulação.

Tabela 9: Descrição dos datasets usados na prova de conceito

Sensor	Dataset	Freq. Sensing
Temperatura	Sensor de Temperatura (°C) em ambiente exterior [24]	5s
Humidade	Sensor de Humidade (%) em ambiente exterior [24]	5s
Ph	Gerador de Eventos Naturais do CupCarbon	5s
Luminosidade	Gerador de Eventos Naturais do CupCarbon	5s
Fluxo Água	Gerador de Eventos Naturais do CupCarbon	4s

Como apresenta a Figura 28, o cenário do sistema de rega é composto por 21 sensores divididos por 3 zonas. Por cada zona existem dois sensores de cada tipo, excepto para o caso do sensor de fluxo de água (apenas um sensor). Sendo assim, foi considerado o mesmo dataset para todos os sensores de cada tipo, dado que os sensores do mesmo tipo têm as mesmas características e as diferentes zonas estão a uma proximidade reduzida.

Em relação aos datasets usados para os sensores de Ph, Luminosidade e Fluxo de água, não foi possível encontrar qualquer dataset público que se enquadrasse com a simulação do sistema de rega. A solução encontrada para esse problema passou por usar uma das funcionalidades do CupCarbon, o gerador de eventos naturais, que permite gerar parâmetros ambientais.

É importante ainda referir que, como os datasets da Temperatura e Humidade têm apenas uma duração de 6 horas, e a simulação foi executada durante 24 horas, o CupCarbon repete o dataset para o tempo restante de simulação.

No que diz respeito à avaliação da desempenho do algoritmo e-LiteSense no Sistema de Rega, serão realizados os seguintes passos:

1. comparação do número de eventos de *sensing* entre a simulação original e a simulação com o algoritmo;
2. cálculo da precisão das medições obtidas pelo algoritmo e-LiteSense;
3. comparação do consumo energético entre a simulação original e simulação com o algoritmo.

Para o cálculo da precisão das medições, serão usadas as mesmas métricas usadas nas secções anteriores, o EQM e a Correlação. Reitera-se que, como o algoritmo e-LiteSense promove uma redução do número de eventos de *sensing*, é necessário igualar o número de leituras do algoritmo ao número de entradas no dataset original. Para isso, foi usada a mesma abordagem apresentada na Secção 4.1.

A configuração do algoritmo para as simulações realizadas neste capítulo, foi selecionada com base experimental (estudo prévio ao algoritmo) e considera: $\Delta T_{min} = 5$ e $\Delta T_{max} = 45$, significando que inicialmente a frequência de *sensing* será de 5 segundos e que pelo menos um evento de *sensing* ocorre a cada 45 segundos. Para o caso do sensor de fluxo de água, a configuração é de $\Delta T_{min} = 4$ e $\Delta T_{max} = 40$. Os limites ao factor da escala de reatividade são definidos como $\epsilon_{min} = 0.05$ e $\epsilon_{max} = 0.30$. Considera-se também que o peso das observações anteriores \tilde{X}_{i-1} e \tilde{V}_{i-1} em \tilde{X}_i e \tilde{V}_i (ver algoritmo 1) são $\alpha = 0.7$ e $\beta = 0.7$. Para terminar, os valores para a variável gamma γ foram definidos de acordo com o estudo experimental realizado na secção 4.2. Neste estudo cheguei à conclusão de que para casos em o nível de stress é baixo (como é o caso dos datasets usados), para $\gamma \leq 0.5$ o algoritmo consegue obter os resultados de forma precisa. Sendo assim, e de forma a testar com diferentes valores, para o dataset de **Temperatura** o γ é de 0.005; para o de **Humidade** o γ é de 0.01; para o dataset da **Luminosidade** o γ é de 0.1; para o dataset da **pH** o γ é de 0.2 e para o dataset da **Fluxo de água** o γ é de 0.3.

5.3 AVALIAÇÃO DE DESEMPENHO

Inicialmente, de forma a avaliar a capacidade do algoritmo em reduzir o número de eventos de *sensing*, foi feita a comparação desse número entre as duas simulações. É importante relembrar que, por defeito, a frequência de *sensing* é de 5 segundos para todos os sensores, (excepto o do fluxo de água que é de 4 segundos) e que a simulação foi executada durante 24 horas.

Podemos verificar pela Figura 29, que o algoritmo consegue reduzir de forma considerável o número de eventos de *sensing*, mais concretamente 65% para os sensores de Temperatura; 59% para os sensores de Humidade; 66% para os sensores de Luminosidade; 61% para os sensores de Ph e 49% para os sensores de fluxo de água. Estes valores comprovam assim a capacidade do algoritmo e-LiteSense em reduzir o *número de eventos de sensing*.

Depois de comprovada a capacidade de redução do número de eventos de *sensing*, é necessário determinar se esta redução significativa dos eventos de *sensing* não causa

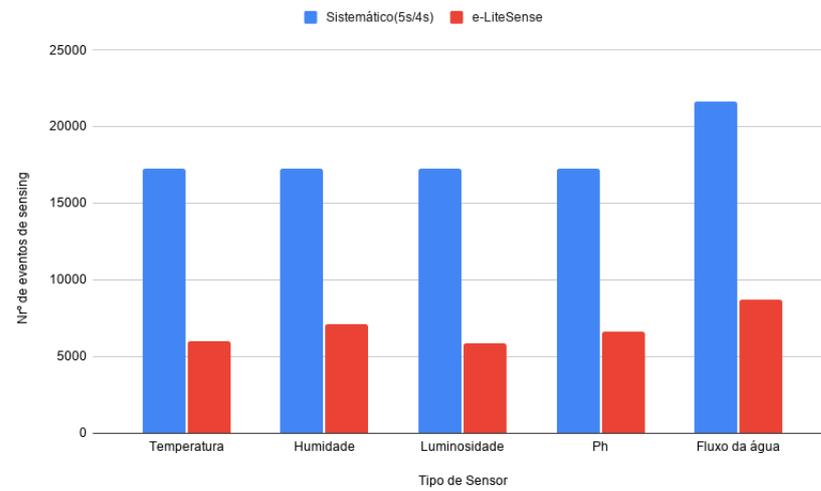


Figura 29: Redução do número de eventos de *sensing* para os diferentes tipos de sensores

impacto significativo na precisão das medições. De acordo com o referido na secção anterior, medições precisas significam medições que tenham uma distribuição semelhante á distribuição original dos parâmetros em observação.

Com o intuito de avaliar a precisão das medições, a Figura 30 mostra que o algoritmo e-LiteSense consegue obter leituras com uma distribuição que coincide, na maioria dos casos, com a distribuição do dataset original (frequência de *sensing* de 5 segundos), mesmo considerando que esta distribuição é baseada em apenas 35% dos eventos de *sensing*.

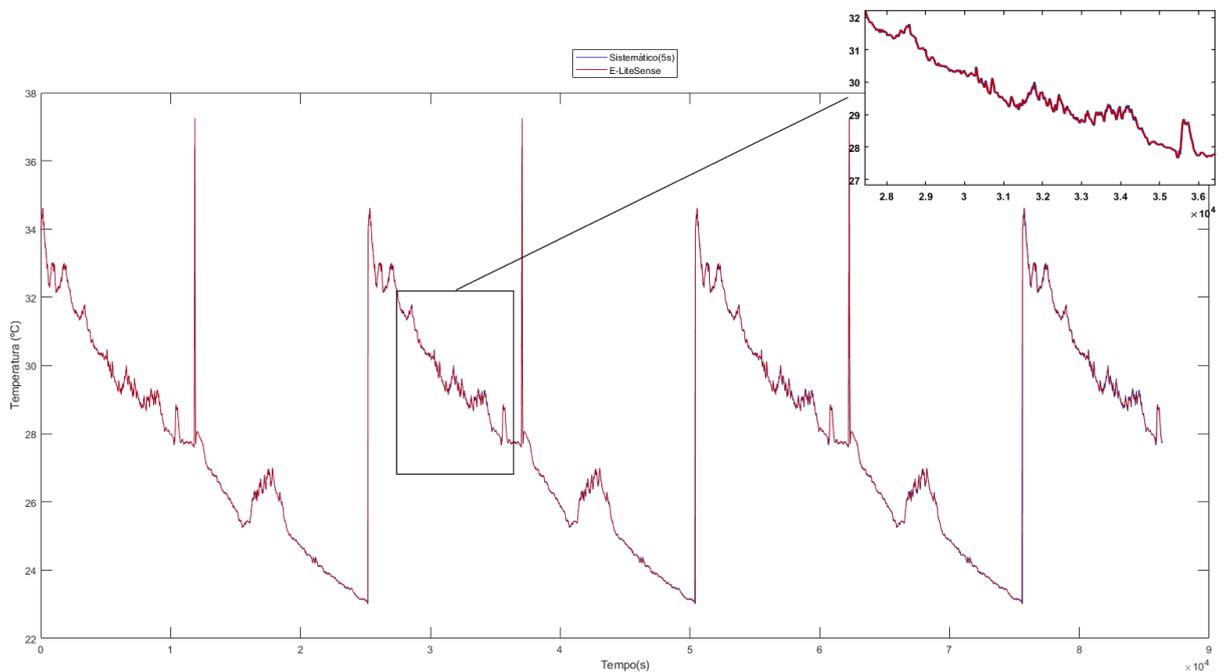


Figura 30: Comparação das leituras para o sensor da Temperatura

Com o objetivo de complementar e validar a análise visual feita anteriormente, apresentam-se na Tabela 10 os resultados obtidos para o cálculo do EQM e do Coeficiente de Correlação. Os resultados mostram que para todos os tipos de sensores o algoritmo e-LiteSense obtém medições com baixo EQM e elevado Coeficiente de Correlação, validando assim estatisticamente a percepção visual obtida na Figura 30. Esta elevada precisão obtida para todos os sensores demonstra repetidamente a eficiência do comportamento auto-adaptativo do e-LiteSense tendo em conta as variações dos parâmetros físicos em observação.

Tabela 10: Precisão das medições - sistema de rega

	EQM	Correlação
Temperatura	0.0438	0.9974
Humidade	0.0908	0.9982
Luminosidade	0.2906	0.9830
Ph	0.4189	0.9781
Fluxo de Água	0.2123	0.9696

Depois de efetuada a análise da precisão das medições, é necessário verificar se o algoritmo e-LiteSense promove melhorias no consumo energético dos diferentes sensores e, conseqüentemente, da RSSF. Como o algoritmo é capaz de reduzir substancialmente

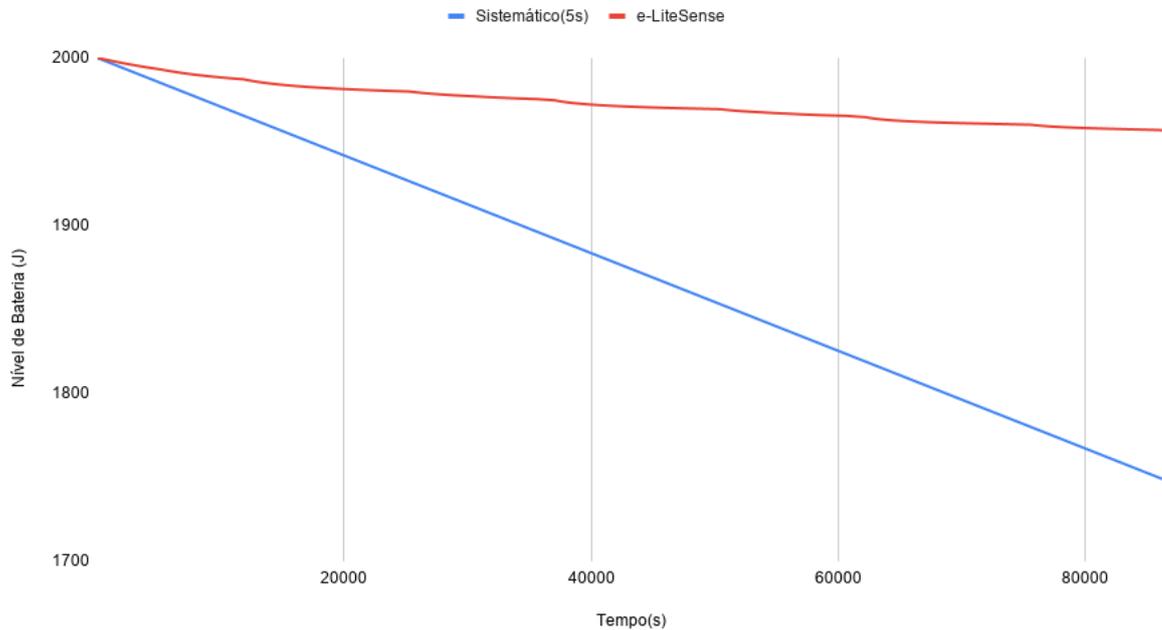


Figura 31: Comparação do consumo energético para o sensor da Temperatura

o número de eventos de *sensing* existe uma menor necessidade de processamento e transmissão de dados, sendo expectável que o algoritmo seja capaz de reduzir o consumo energético dos sensores.

De forma a demonstrar essa intuição, a Figura 31 apresenta a comparação do consumo energético entre um dos sensores de Temperatura, ao executar a simulação com o sistema de rega original e ao executar a simulação com o e-LiteSense implementado nos sensores. Para simplificar, é apenas comparado o consumo energético de uma das três zonas que compõem o sistema de rega (zona situada mais á esquerda). A figura mostra que o algoritmo foi capaz de reduzir substancialmente o consumo energético, mais concretamente uma redução de cerca de 9% em 24 horas de simulação.

A Tabela 11 apresenta a poupança de energia para todos os sensores obtida pela implementação do e-LiteSense, quando comparadas diretamente com o consumo energético do sistema de rega original. Os resultados mostram uma melhoria entre 6.61% (sensores de Fluxo de Água) e 9.44%(sensores de Temperatura), confirmando a eficiência do algoritmo e-LiteSense, principalmente tendo em conta o boa precisão obtida pelas medições produzidas. (ver Tabela 10)

Tabela 11: Poupança de energia para os diferentes sensores de cada zona

Zona 1								
S13_Hum	S17_Hum	S12_Temp	S16_Temp	S15_Lum	S18_Lum	S19_pH	S14_pH	S20_Fluxo
9.46%	9.46%	8.97%	8.97%	9.45%	9.45%	7.89%	7.89%	6.61%
Zona 2								
S4_Hum	S25_Hum	S3_Temp	S23_Temp	S24_Lum	S6_Lum	S26_pH	S5_pH	S21_Flux
9.46%	9.46%	8.97%	8.97%	9.45%	9.45%	7.89%	7.89%	6.61%
Zona 3								
S35_Hum	S31_Hum	S30_Temp	S34_Temp	S37_Lum	S33_Lum	S32_pH	S36_pH	S22_Fluxo
9.46%	9.46%	8.97%	8.97%	9.45%	9.45%	7.89%	7.89%	6.61%

5.4 RESUMO

Neste capítulo, foi apresentado o Sistema de Rega Inteligente usado como caso de estudo, descrevendo-se, primeiramente, o sistema de rega, as características dos diferentes sensores, o sistema de controlo de irrigação e o design final do sistema de rega.

Ato contínuo, foram apresentados os datasets usados nos diferentes sensores e a frequência de *sensing* dos mesmos, bem como as métricas de avaliação de desempenho usadas para avaliar os resultados obtidos pelo algoritmo e-LiteSense durante a simulação. É ainda descrita a parametrização do algoritmo e-LiteSense para os diferentes sensores do sistema de rega.

Depois de apresentada de que forma a simulação foi executada, foram descritos os resultados obtidos pelo algoritmo. A avaliação de desempenho passou por: comparar o número de eventos de *sensing* entre o sistema de rega original e o sistema de rega com o algoritmo implementado nos sensores; calcular o EQM e o Coeficiente de Correlação; Comparar o consumo energético entre as duas simulações. Os resultados mostram que o algoritmo tem a capacidade de reduzir o número de eventos de *sensing* para todos os sensores. Ademais, mostram que esta redução não apresenta impacto significativo na precisão das medições, obtendo-se assim um baixo EQM e elevado Coeficiente de Correlação. Ao ser capaz de reduzir o número de eventos de *sensing*, os resultados mostram também que o algoritmo é capaz de reduzir o consumo energético de todos os sensores e, conseqüentemente, aumentar o tempo de vida da RSSF.

CONCLUSÕES

As RSSF são, atualmente, aplicadas a uma grande diversidade de cenários envolvendo *sensing* de fenómenos ambientais ou parâmetros físicos. Um dos principais desafios, independentemente da heterogeneidade desta área, é o facto de os recursos disponibilizados aos sensores da rede serem reduzidos. Com o intuito de mitigar este problema, o algoritmo e-LiteSense foi desenvolvido, recorrendo a características tais como a capacidade de auto-adaptação consoante as variações dos parâmetros em observação, a baixa complexidade e a consciência energética.

O presente estudo teve como objetivo realizar uma avaliação detalhada do algoritmo e-LiteSense, no sentido de avaliar o comportamento do algoritmo em casos de *stress* (variação repentina dos parâmetros em observação) ao alterarmos o valor do γ , avaliar se o algoritmo consegue manter o desempenho esperado em cenários diversificados e, por fim, testar o algoritmo usando um cenário da vida real. Com este estudo pretende-se contribuir para a melhoria de um algoritmo que poderá ser útil em diversas situações envolvendo RSSF.

6.1 CONCLUSÕES

No que diz respeito ao estudo da variação da escala de reatividade (γ) em situação de *stress*, os testes foram divididos em três partes: nível de *stress* baixo ou pontual, médio e alto. Para o nível de *stress* baixo ou pontual, os resultados mostram que para $0.005 \leq \gamma \leq 0.4$ o algoritmo consegue captar corretamente os parâmetros em observação mantendo a precisão das medições e reduzindo o consumo energético do sensor. Para valores de $\gamma \geq 0.5$ é notório que o algoritmo começa a perder a capacidade de captar corretamente os parâmetros em observação. Para os casos de nível de *stress* médio e alto, os resultados obtidos mostram que para $0.005 \leq \gamma < 1.5$ e $0.005 \leq \gamma < 3.5$, respetivamente, o algoritmo apresenta o comportamento correto,

mantendo a capacidade de reduzir o número de eventos de *sensing*, mantendo a precisão das medições e reduzindo o consumo energético do sensor. Para valores de γ superiores, o algoritmo demonstra uma perda na capacidade de recolher corretamente os valores dos parâmetros em observação. Tendo tudo isto em consideração, é possível concluir que na utilização do algoritmo e-LiteSense, o valor de γ deverá ser configurado da seguinte forma:

- nível de *stress* baixo ou pontual : $0.005 \leq \gamma < 0.5$
- nível de *stress* médio : $0.005 \leq \gamma < 1.5$
- nível de *stress* alto : $0.005 \leq \gamma < 3.5$

Relativamente aos resultados obtidos no estudo da versatilidade do algoritmo, foram usados datasets de áreas de intervenção bastante distintas de forma a inferir se o algoritmo consegue funcionar de forma correta independentemente do cenário em questão e dos parâmetros em observação. Os resultados mostram que, para os diferentes datasets usados na simulação, o algoritmo é capaz de captar corretamente os parâmetros em observação e ainda reduzir o consumo energético dos sensores em comparação com a abordagem sistemática. Os resultados demonstram também que o algoritmo é mais eficiente se o cenário em questão tiver uma frequência de *sensing* reduzida.

A prova de conceito final foi realizada usando uma simulação de um Sistema de Rega, de forma a avaliar o algoritmo quando aplicado a um cenário mais complexo e realista. Este sistema de rega é composto por 21 sensores divididos por três zonas, sendo que para dois dos sensores são usados datasets reais. Os resultados mostram que, tanto para os sensores com os datasets de reais (Temperatura e Humidade) como para os sensores nos quais são usados datasets obtidos pelo CupCarbon, o algoritmo e-LiteSense apresenta, repetidamente, o comportamento esperado. O algoritmo é capaz de reduzir os eventos de *sensing*, manter a precisão das medições dos diferentes parâmetros de interesse e reduzir o consumo energético de cada um dos sensores e, conseqüentemente, da RSSF.

Os resultados obtidos nesta dissertação demonstraram assim os níveis de precisão elevada do e-LiteSense na captura do comportamento dos parâmetros em observação, com uma redução significativa dos eventos de detecção e aumento da vida útil dos sensores e da RSSF. Demonstraram ainda a sua versatilidade e capacidade de auto-adaptação a diferentes cenários no âmbito das CI.

6.2 TRABALHO FUTURO

Face aos positivos resultados observados e explanados ao longo desta dissertação, não descurando os trabalhos anteriormente realizados relativamente ao algoritmo, concluí-se que ulteriormente, como trabalho futuro, seria interessante avaliar a possibilidade de implementar diretamente o algoritmo e-LiteSense como funcionalidade do CupCarbon. Caso isso venha a ser possível, seria relevante fazer um estudo de comparação entre o algoritmo e-LiteSense e outros algoritmos que estejam disponíveis no CupCarbon. Seria também pertinente procurar algoritmos com abordagens semelhantes ao e-LiteSense e fazer a comparação do desempenho dos mesmos.

Todos os testes realizados até ao momento ao algoritmo foram baseados em simulações pelo que, de forma a validar os resultados obtidos, seria relevante implementar o algoritmo e-LiteSense em sensores reais, no sentido de avaliar todo o real potencial do desempenho do algoritmo e-LiteSense.

GRÁFICOS E TABELAS EXTRA

A.1 NÍVEL DE *stress* BAIXO OU PONTUAL

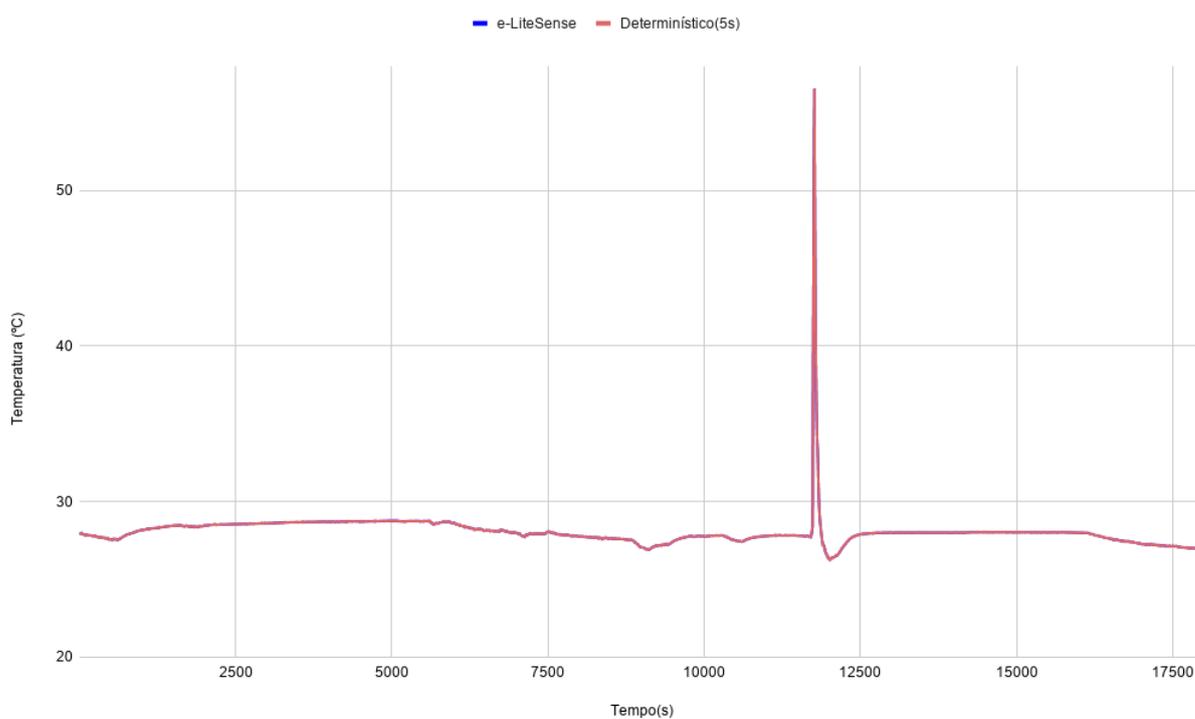


Figura 32: Medições para gamma igual a 0.005

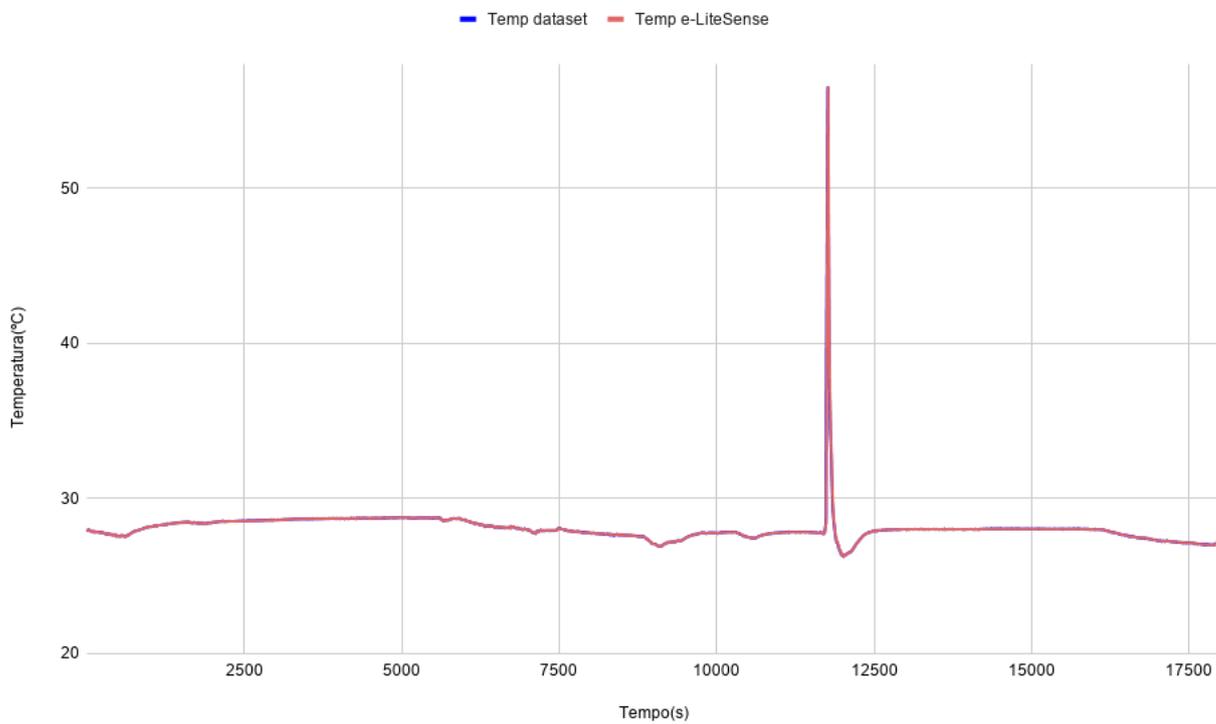


Figura 33: Medições para gamma igual a 0.01

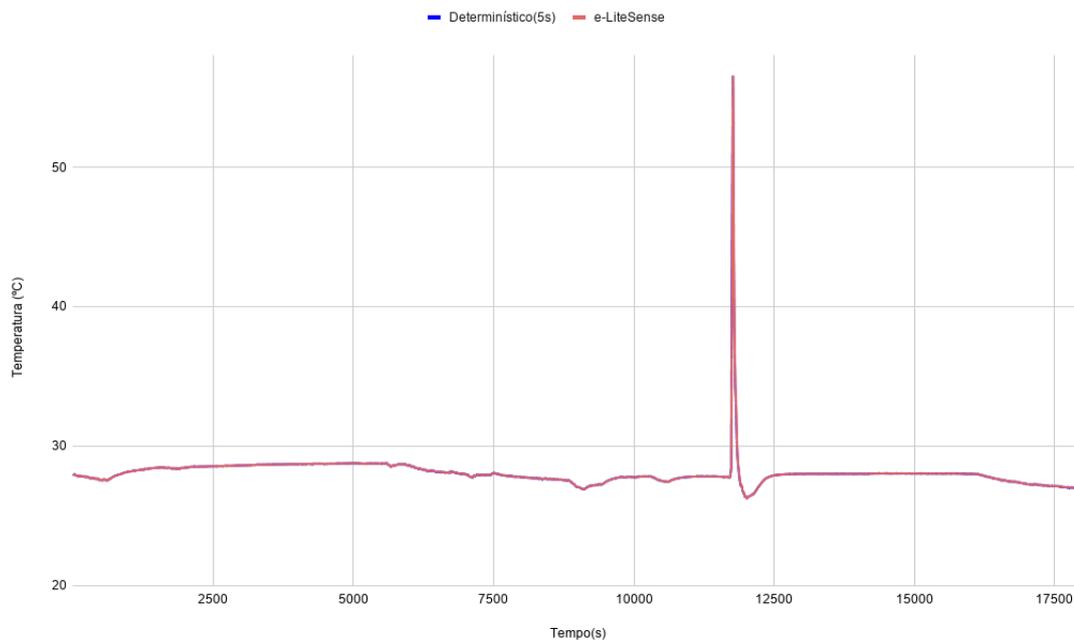


Figura 34: Medições para gamma igual a 0.1

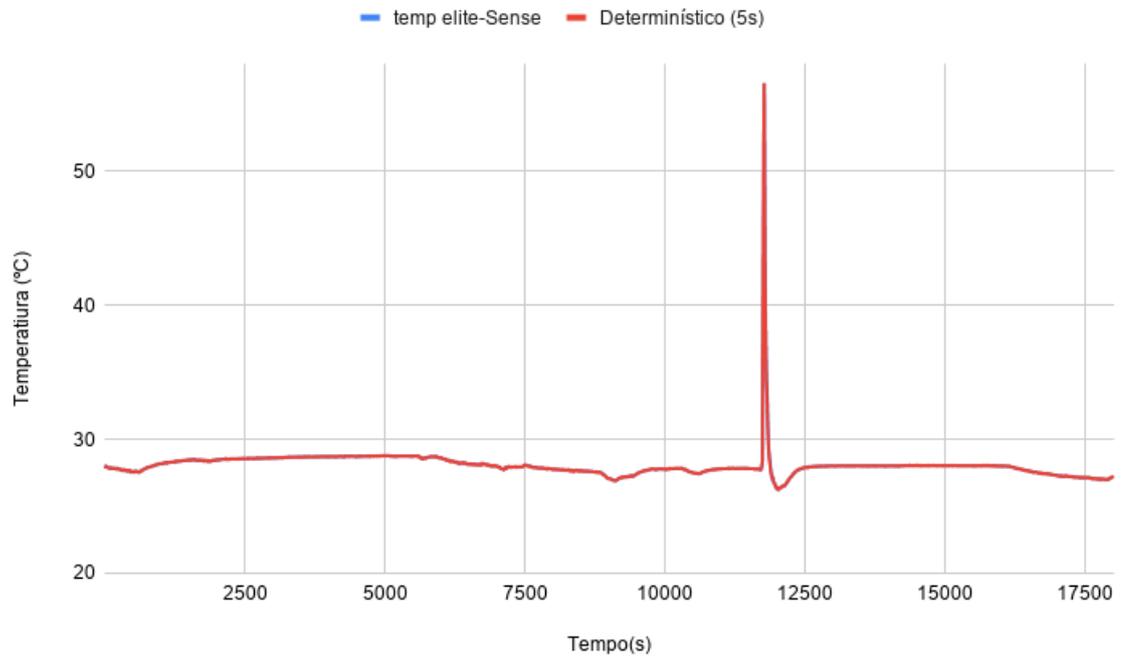


Figura 35: Medições para gamma igual a 0.2

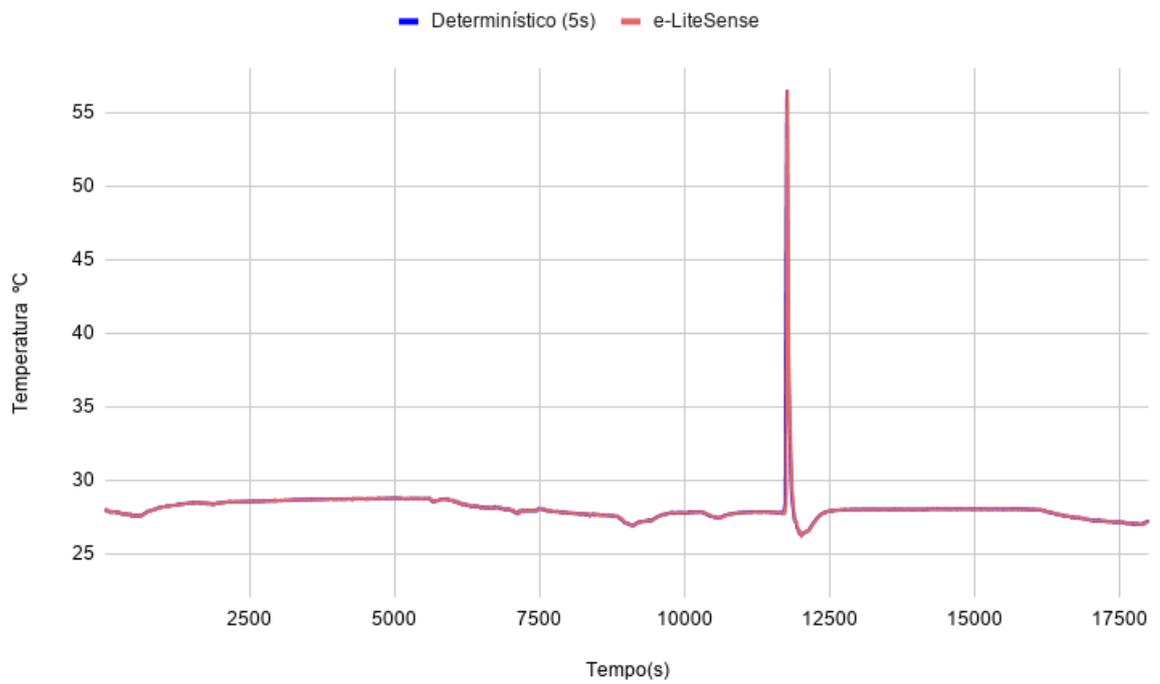


Figura 36: Medições para gamma igual a 0.3

A.2 NÍVEL DE *stress* MÉDIO

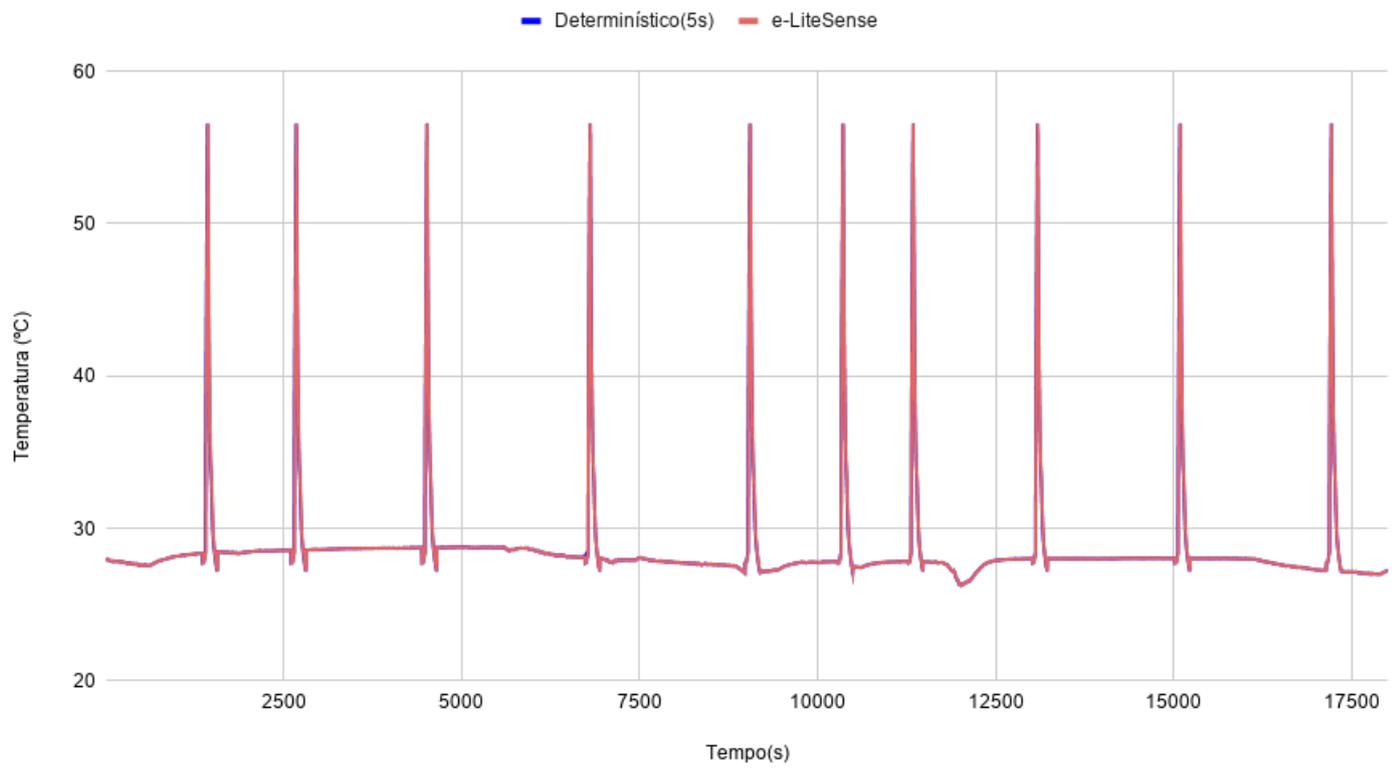


Figura 37: Medições para gamma igual a 0.005

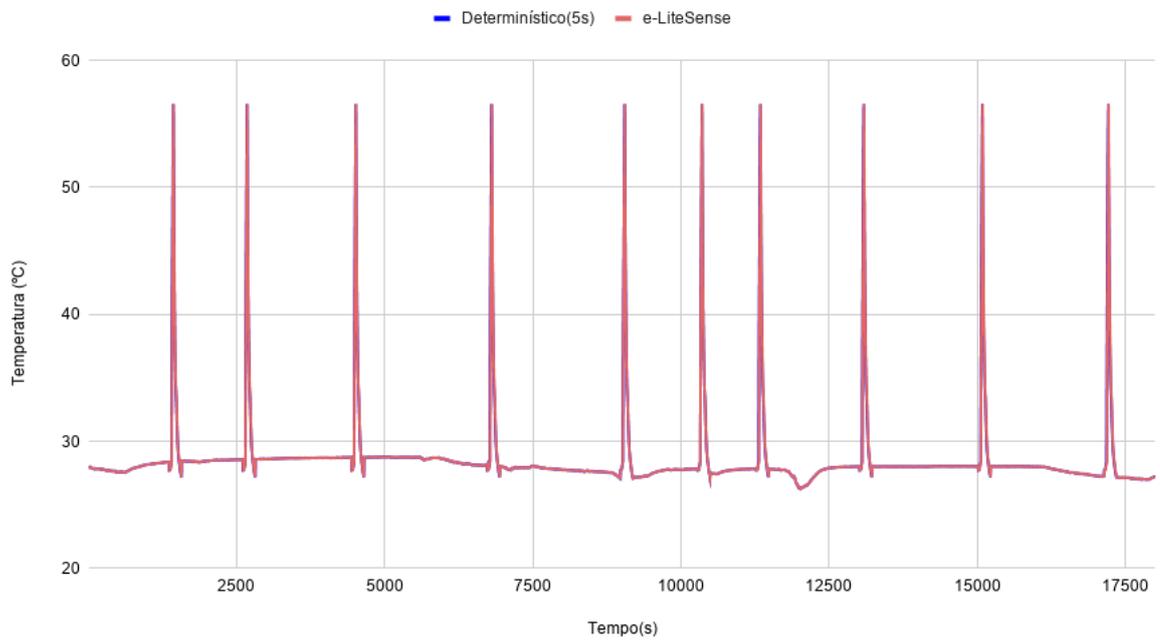


Figura 38: Medições para gamma igual a 0.01

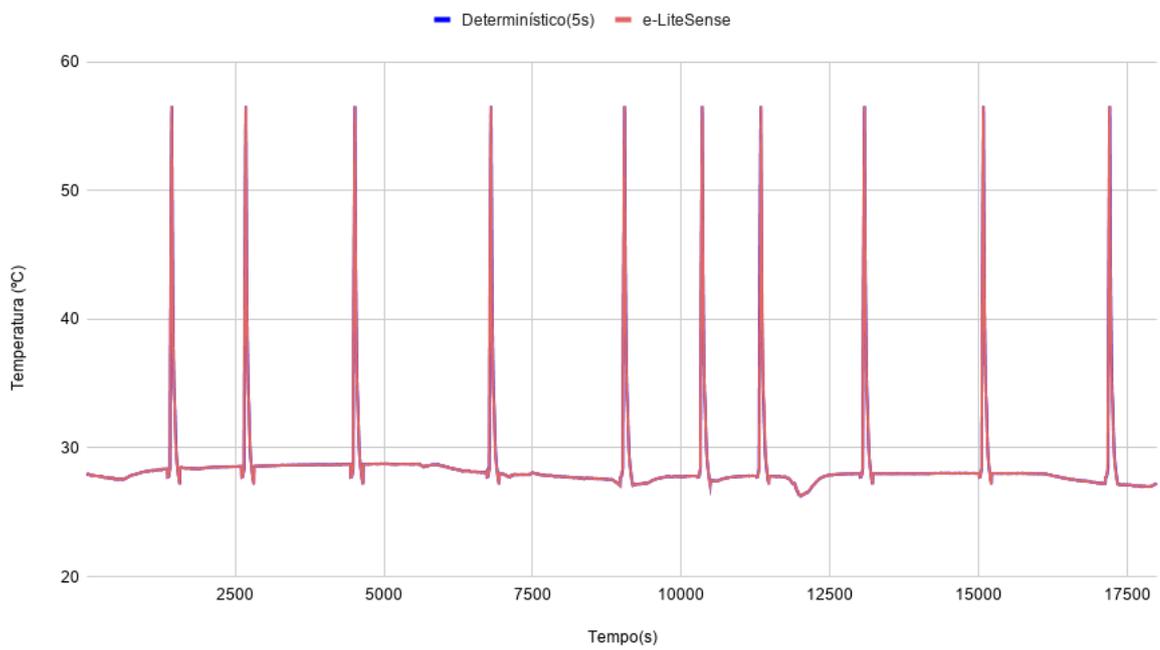


Figura 39: Medições para gamma igual a 0.1

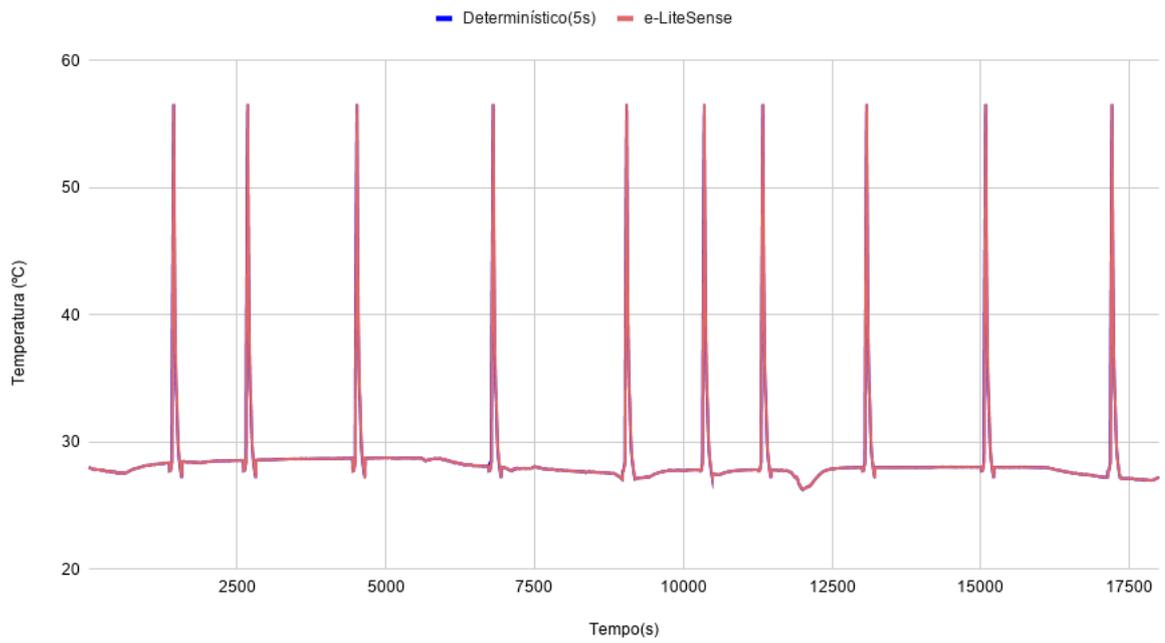


Figura 40: Medições para gamma igual a 0.2

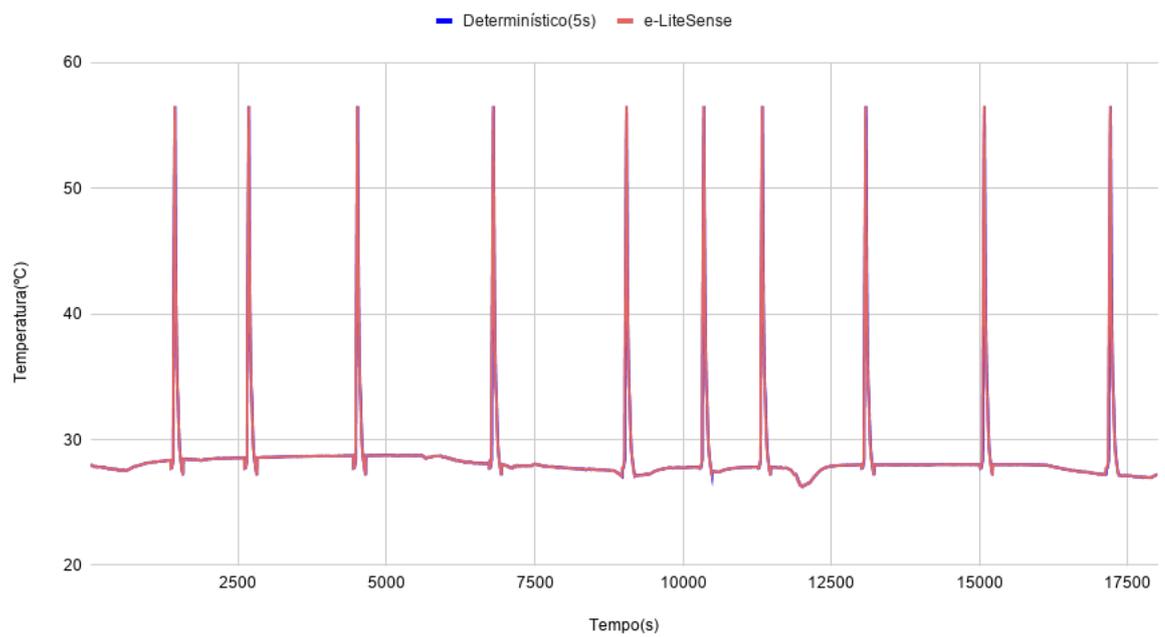


Figura 41: Medições para gamma igual a 0.3

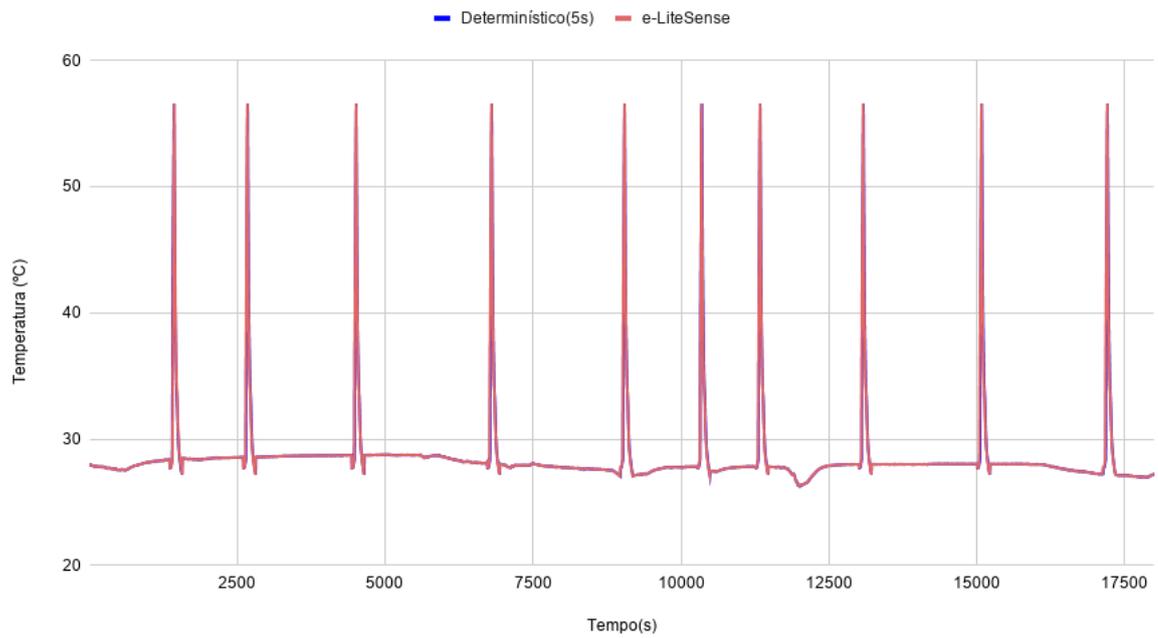


Figura 42: Medições para gamma igual a 0.4

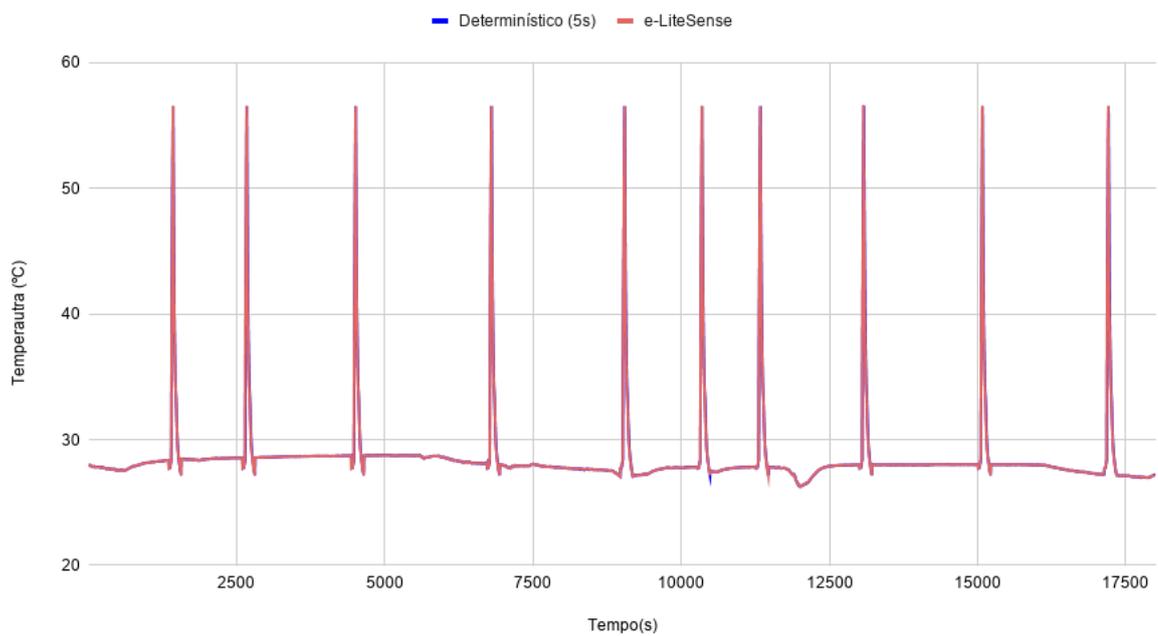


Figura 43: Medições para gamma igual a 0.5

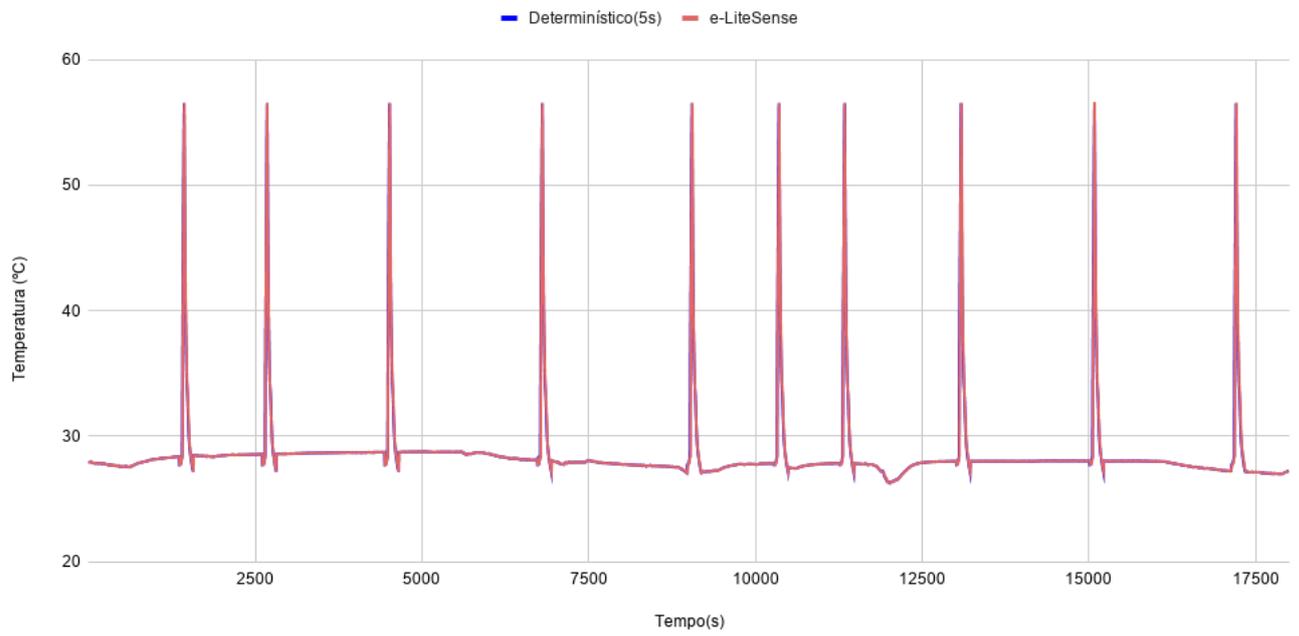


Figura 44: Medições para gamma igual a 1

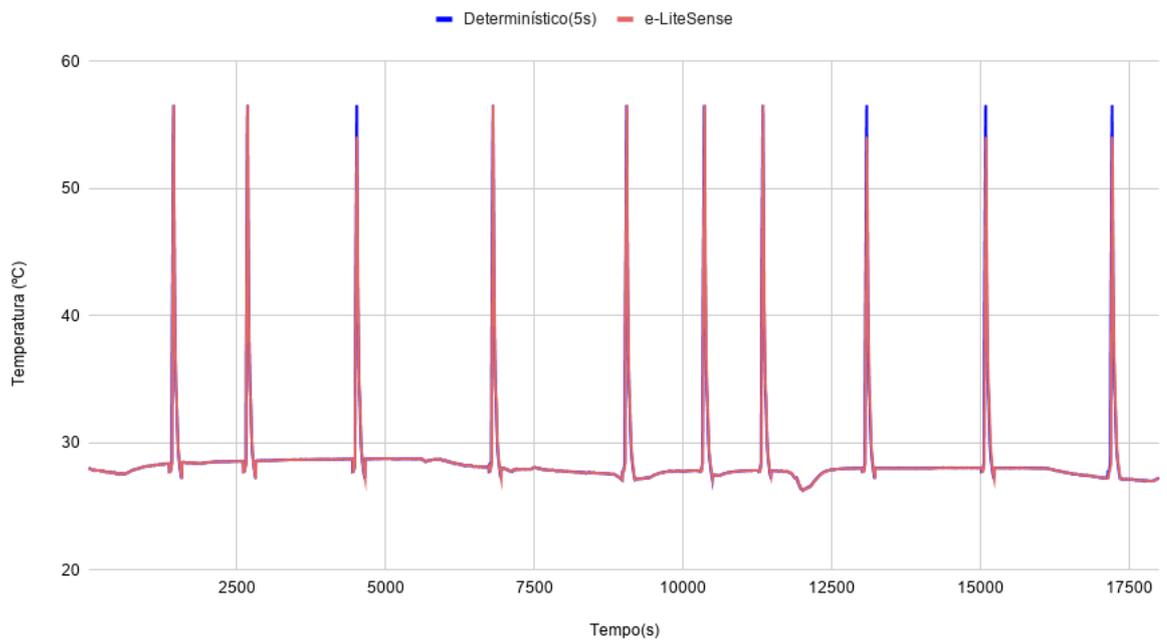


Figura 45: Medições para gamma igual a 2

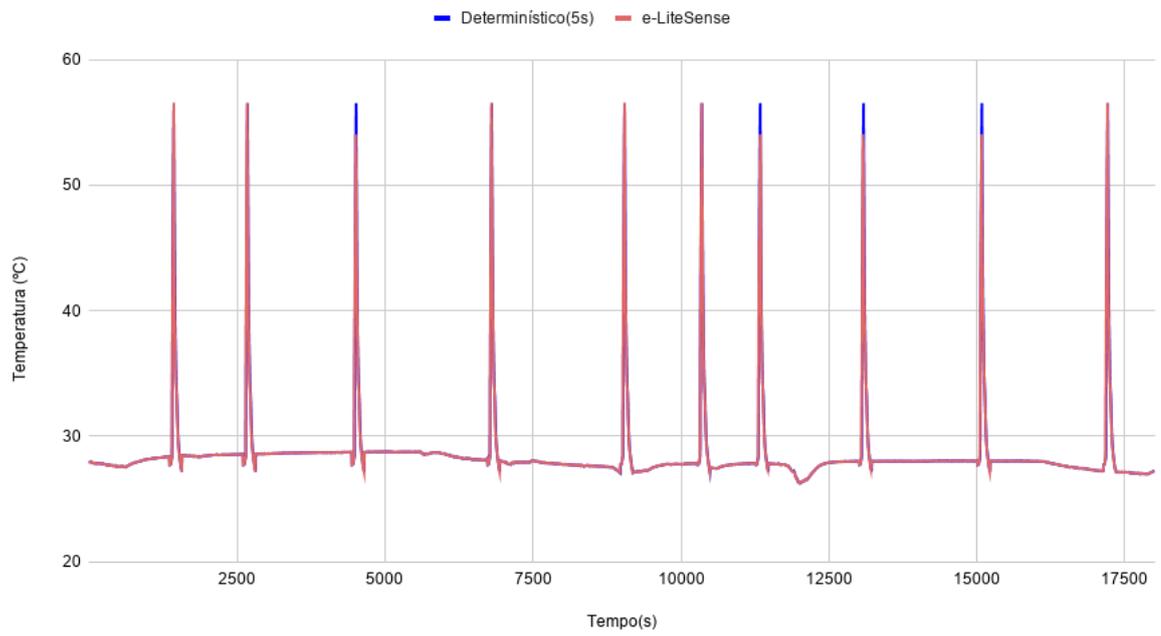


Figura 46: Medições para gamma igual a 2.5

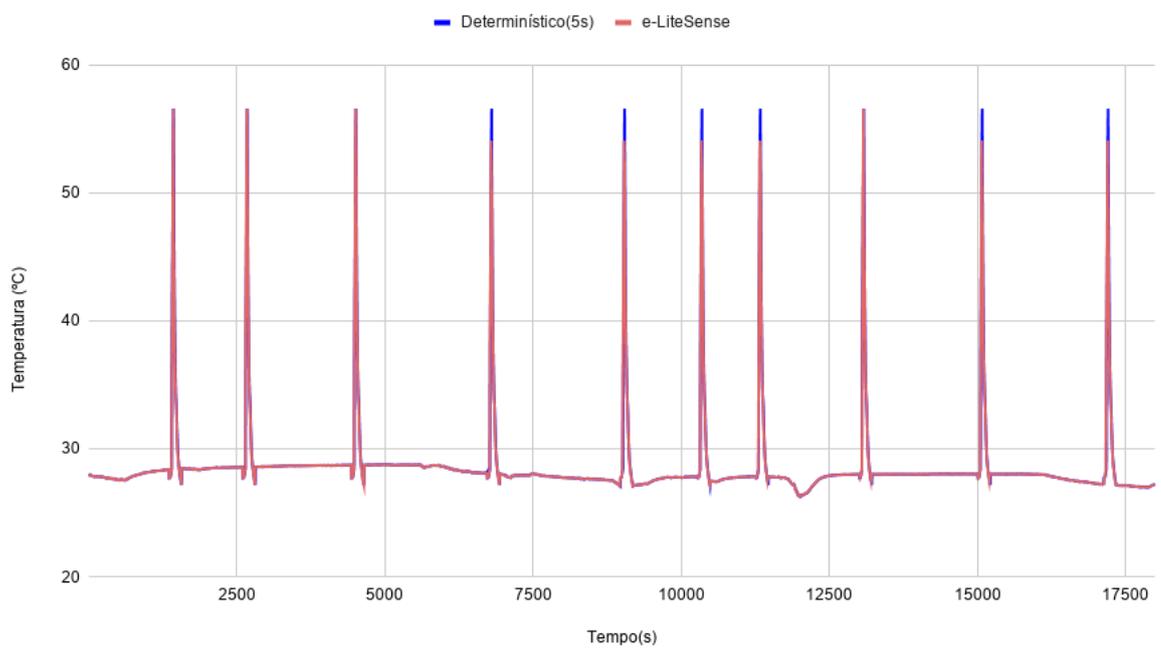


Figura 47: Medições para gamma igual a 3.5

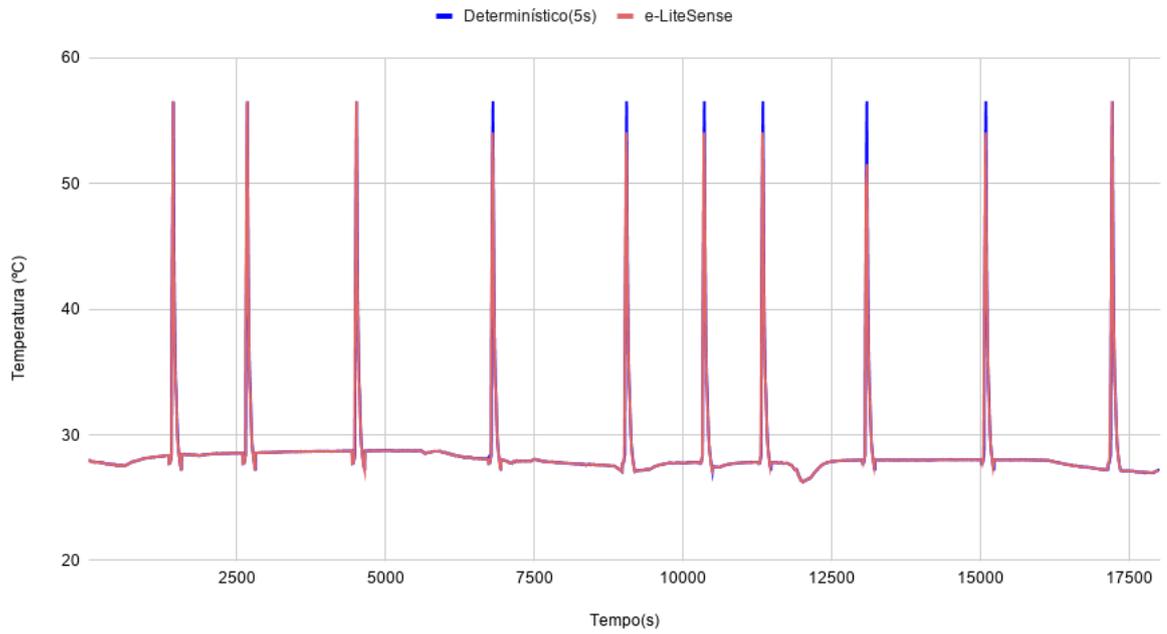


Figura 48: Medições para gamma igual a 5

A.3 NÍVEL DE *stress* ALTO

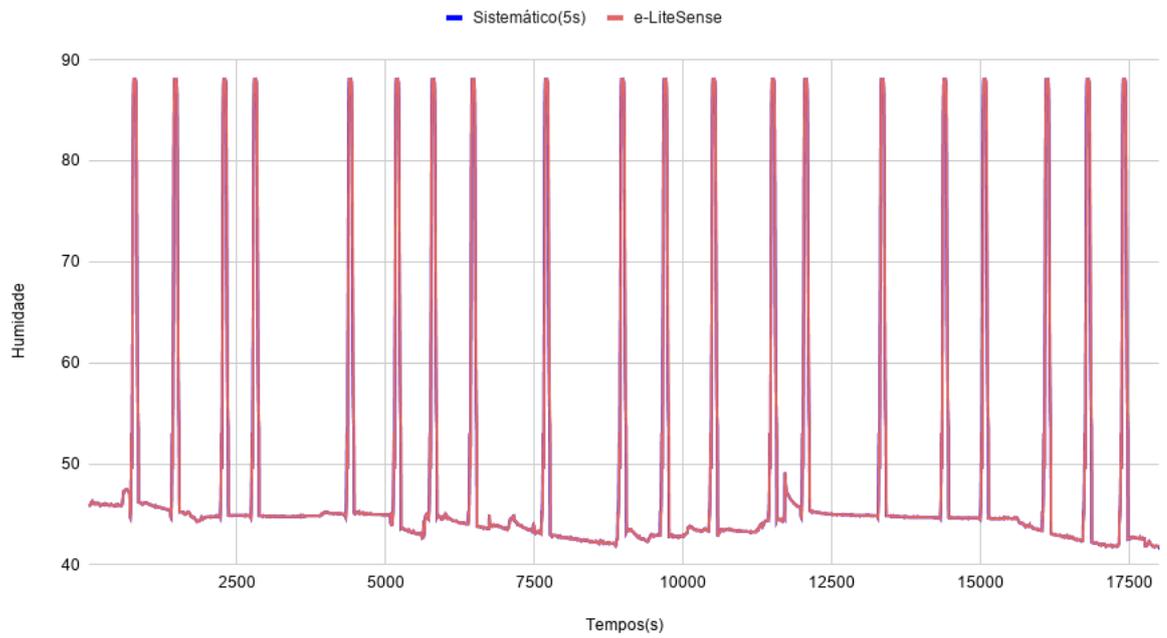


Figura 49: Medições para gamma igual a 0.005

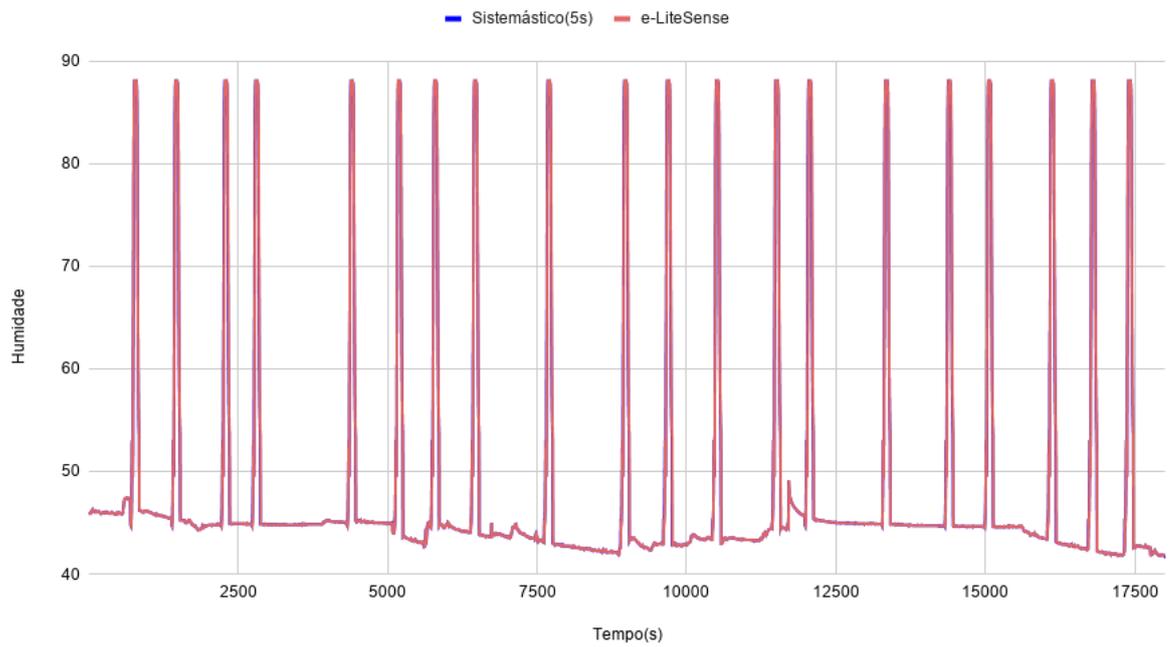


Figura 50: Medições para gamma igual a 0.01

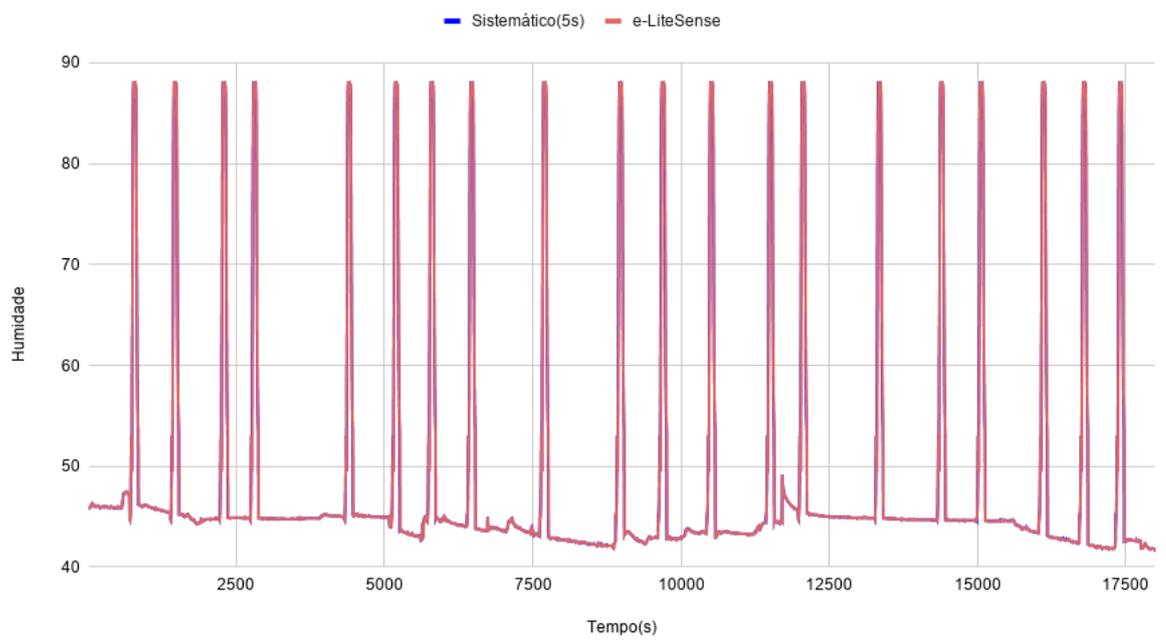


Figura 51: Medições para gamma igual a 0.1

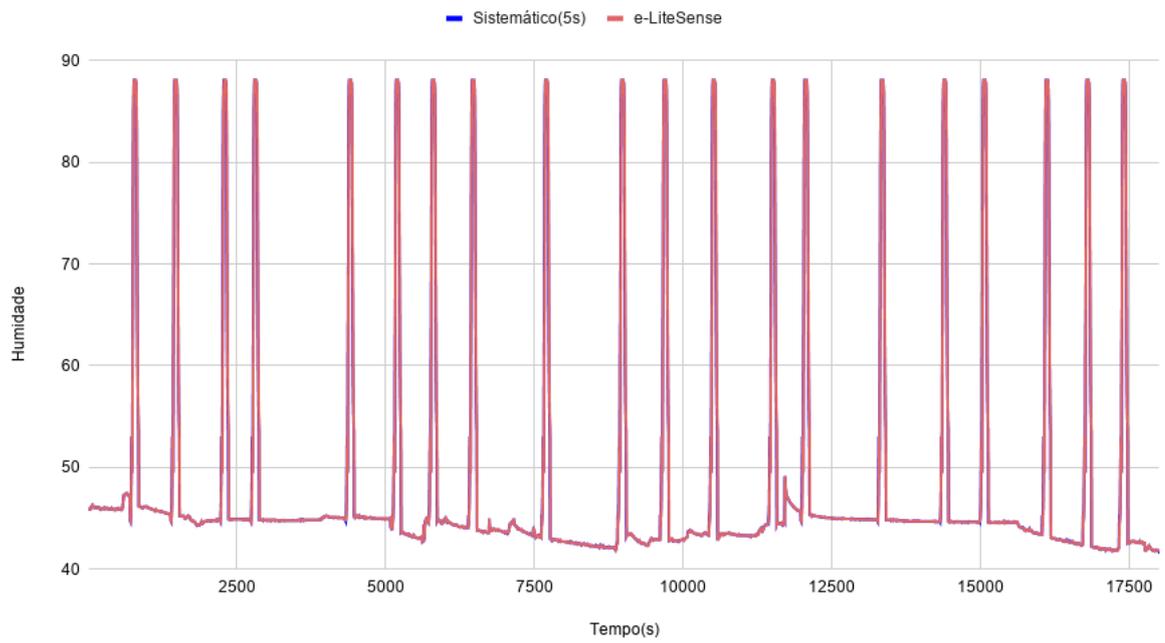


Figura 52: Medições para gamma igual a 0.2

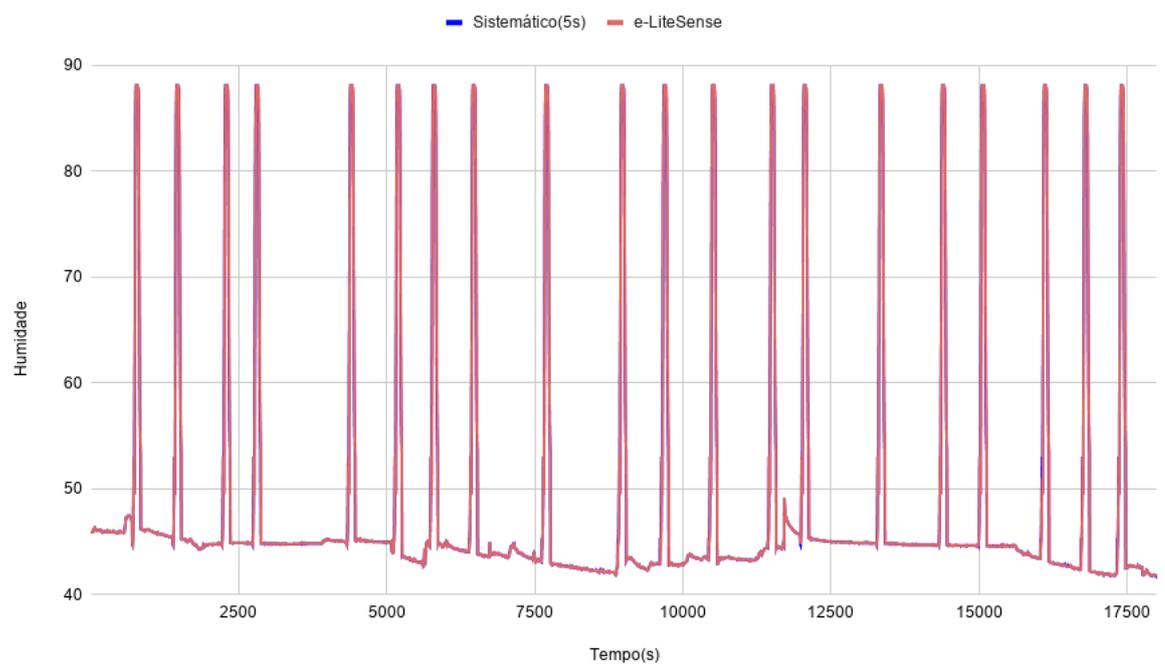


Figura 53: Medições para gamma igual a 0.3

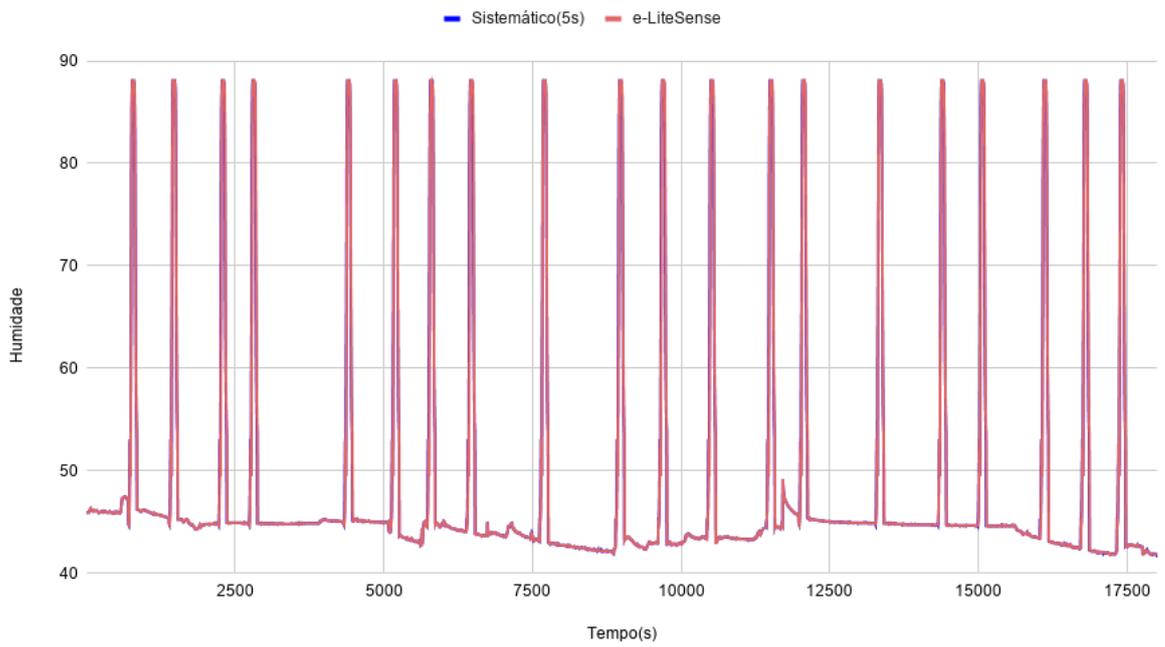


Figura 54: Medições para gamma igual a 0.4

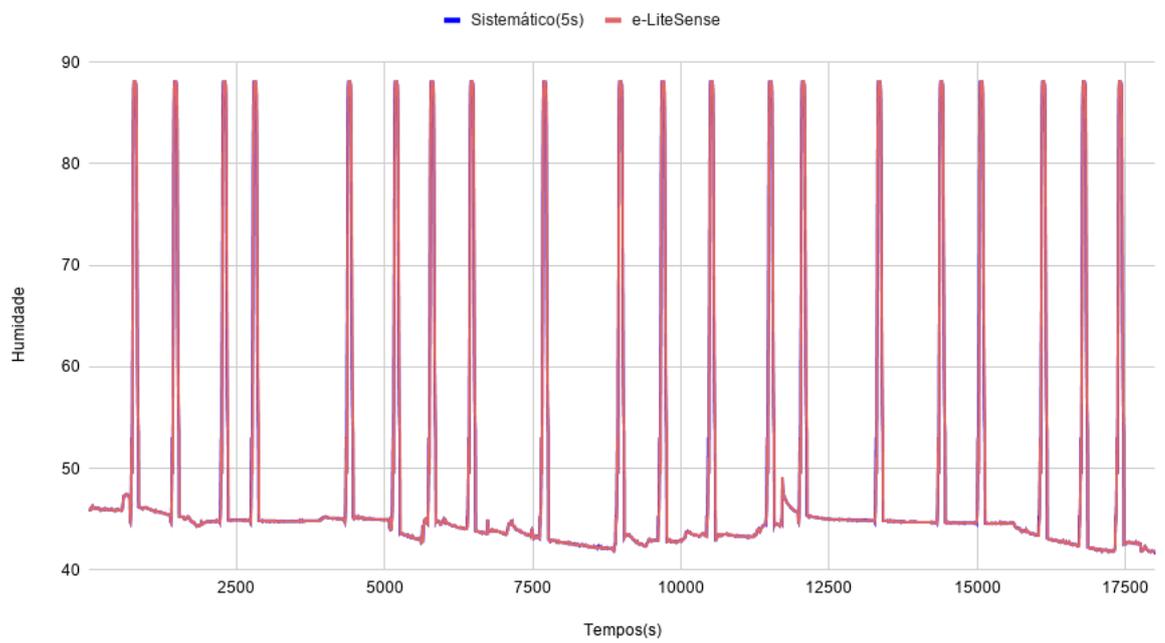


Figura 55: Medições para gamma igual a 0.5

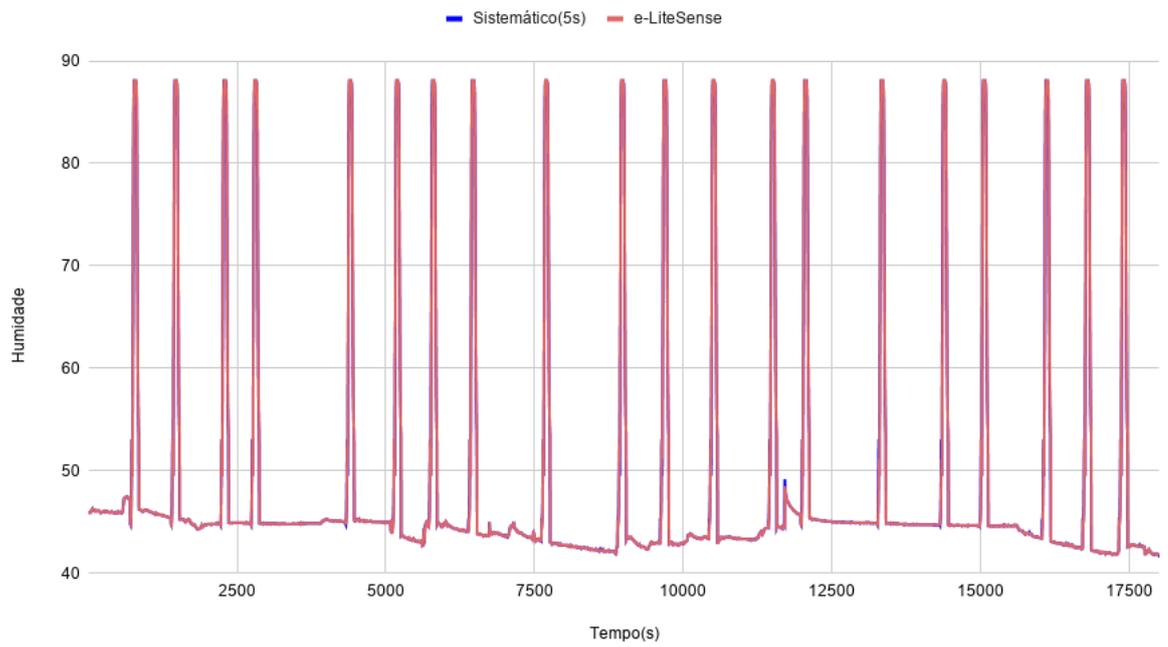


Figura 56: Medições para gamma igual a 1

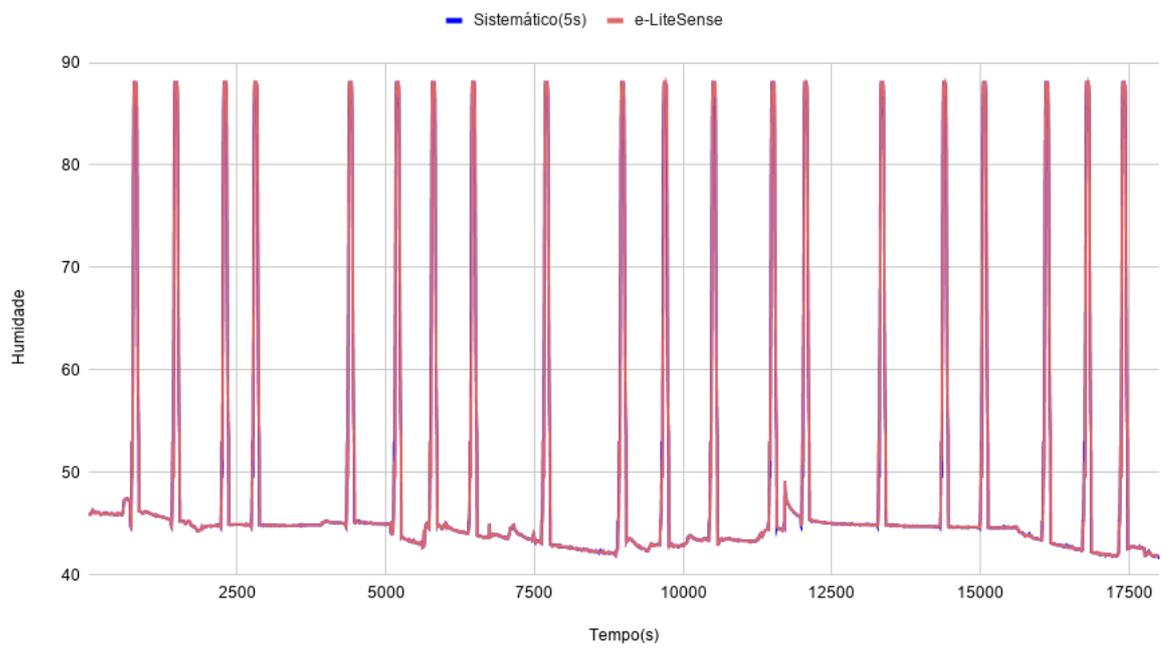


Figura 57: Medições para gamma igual a 1,5

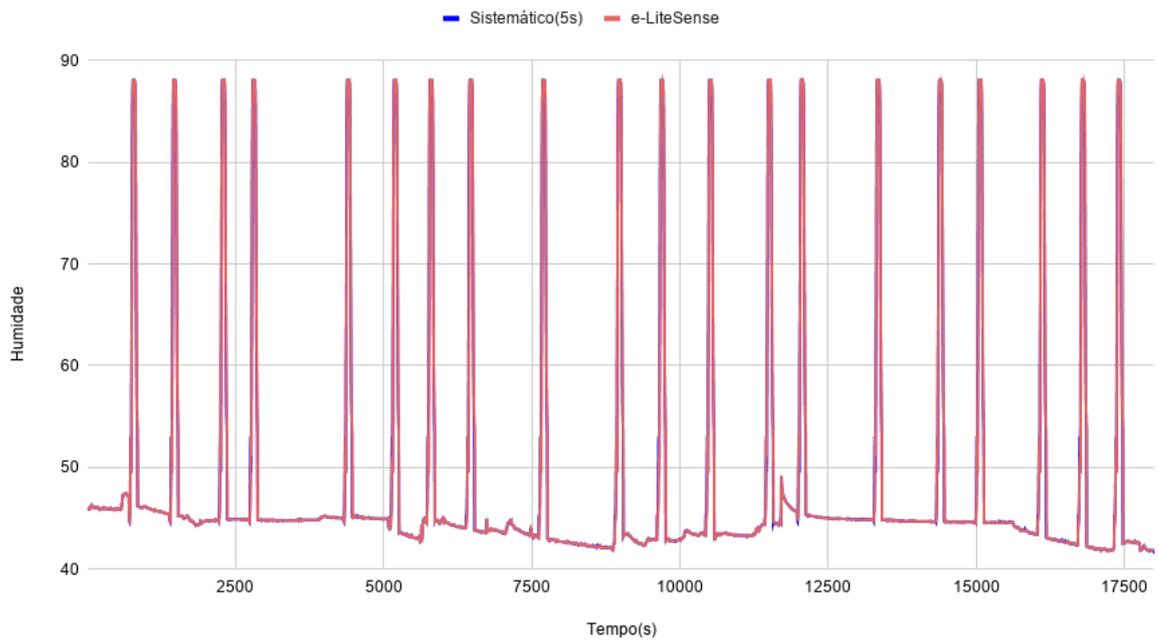


Figura 58: Medições para gamma igual a 2

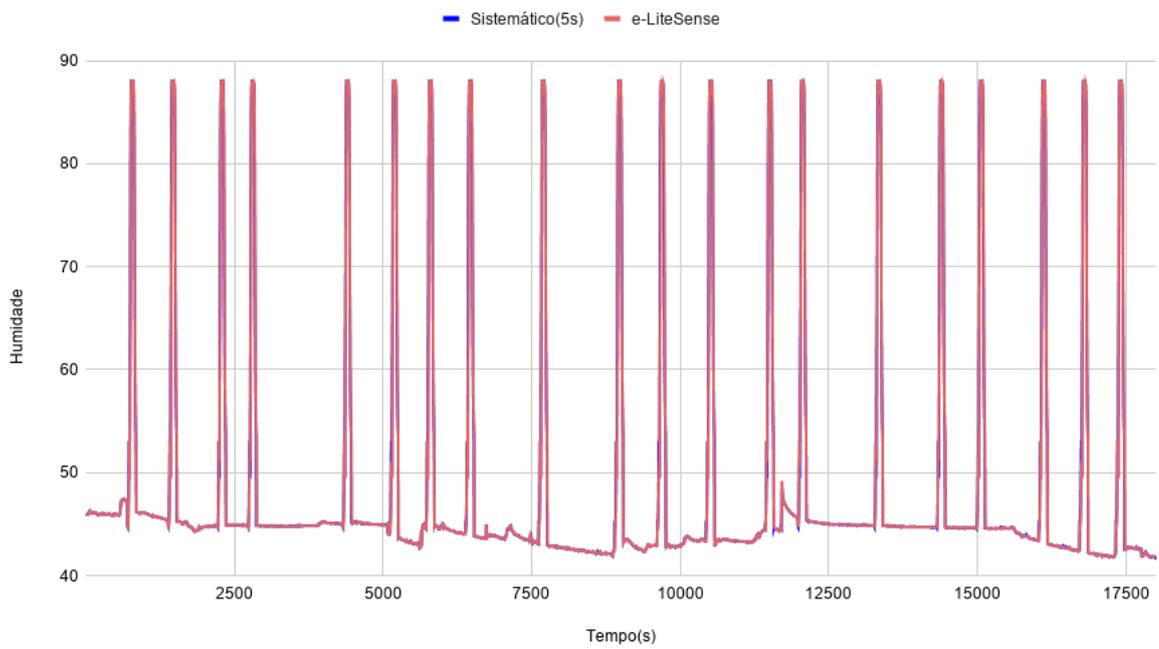


Figura 59: Medições para gamma igual a 2.5

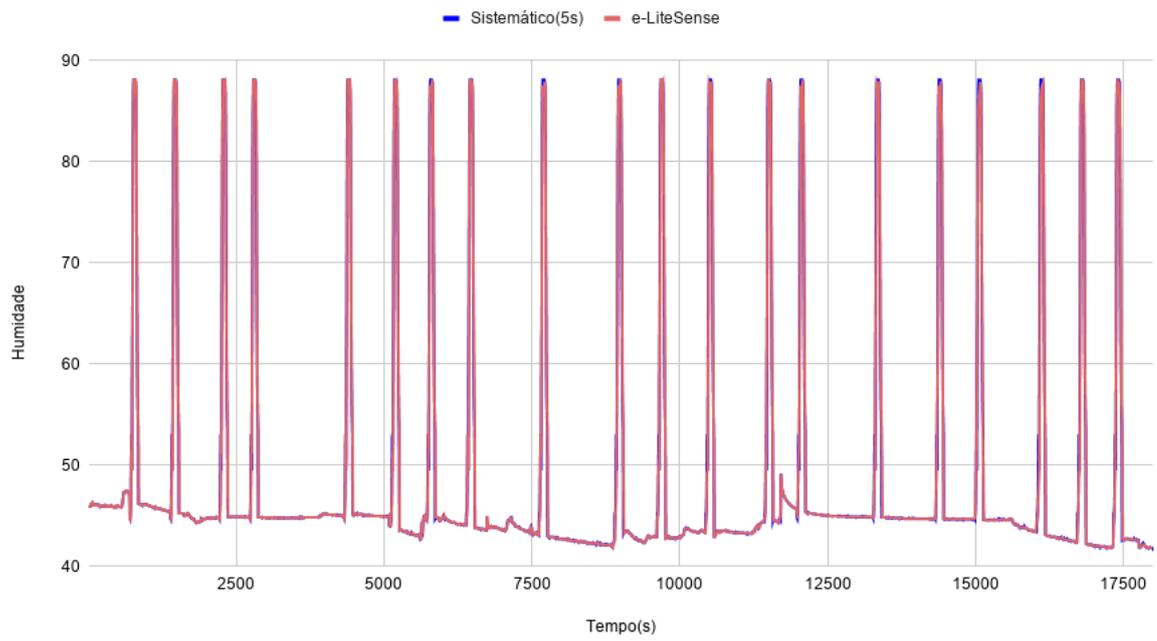


Figura 60: Medições para gamma igual a 5

A.4 RÁCIO PEAK TO MEAN - NÍVEL DE *stress* MÉDIO

Gamma	1-1800	1800-3600	3600-5400	5400-7200	7200-9000	9000-10800	10800-12600	12600-14400	14400-16200	16200-18000
0.005	1.9717	1.9371	1.9257	1.9491	1.9894	1.9880	2.0011	1.9634	1.9578	2.0089
0.01	1.9716	1.9369	1.9257	1.9470	1.9807	1.9881	1.9992	1.9625	1.9582	1.9967
0.1	1.9715	1.9369	1.9256	1.9448	1.9805	1.9867	1.9928	1.9620	1.9593	1.9999
0.2	1.9715	1.9368	1.9253	1.9423	1.9816	1.9861	1.9929	1.9609	1.9578	2.0014
0.3	1.9714	1.9365	1.9246	1.9422	1.9791	1.9863	1.9944	1.9617	1.9585	1.9989
0.4	1.9715	1.9364	1.9244	1.9418	1.9799	1.9916	1.9926	1.9667	1.9607	2.0025
0.5	1.9714	1.9364	1.9247	1.9438	1.9841	1.9882	1.9921	1.9672	1.9627	2.002
1	1.9714	1.9364	1.9247	1.9438	1.9839	1.9866	1.9920	1.9623	1.9637	2.0056
1.5	1.9714	1.9364	1.9247	1.9435	1.9855	1.9864	1.9554	1.9344	1.9109	1.9540
2	1.9759	1.9372	1.9246	1.9453	1.9839	1.9963	1.9323	1.8841	1.8855	1.9200
2.5	1.9759	1.9372	1.9046	1.9453	1.9839	1.9563	1.9123	1.8828	1.8783	1.9182
3.5	1.9759	1.9369	1.9286	1.8663	1.90609	1.9083	1.9119	1.8674	1.8837	1.9040
5	1.9759	1.9369	1.9286	1.8663	1.906	1.9085	1.9119	1.7987	1.8712	1.8840

A.5 RÁCIO PEAK TO MEAN - NÍVEL DE *stress* ALTO

Gamma	1-900	900-1800	1800-2700	2700-3600	3600-4500	4500-5400	5400-6300	6300-7200	7200-8100	8100-9000	9000-9900	9900-10800	10800-11700	11700-12600	12600-13500	13500-14400	14400-15300	15300-16200	16200-17100	17100-18000
0.005	1.7895	1.7998	1.8121	1.8107	1.8060	1.8141	1.8346	1.8597	1.82256	1.8408	1.8339	1.8275	1.8361	1.8177	1.8010	1.8008	1.8095	1.8278	1.8538	1.8359
0.01	1.7895	1.7989	1.8121	1.8074	1.8010	1.8126	1.8337	1.8587	1.8297	1.8495	1.8319	1.8166	1.8332	1.8114	1.78944	1.8034	1.7966	1.8294	1.8391	1.8339
0.1	1.7888	1.7988	1.8105	1.8074	1.7933	1.8107	1.8312	1.8536	1.8213	1.8477	1.8351	1.8173	1.8296	1.8181	1.7981	1.8028	1.7949	1.8061	1.8206	1.8343
0.2	1.7888	1.7988	1.8105	1.8074	1.7932	1.8101	1.8292	1.8536	1.8248	1.8477	1.8352	1.8472	1.8277	1.8199	1.8094	1.8019	1.7949	1.8063	1.8206	1.8376
0.3	1.7888	1.7988	1.8112	1.8065	1.7933	1.8101	1.8289	1.8509	1.8243	1.8482	1.8351	1.8476	1.8298	1.8199	1.7892	1.8019	1.7949	1.8062	1.8206	1.8376
0.4	1.7883	1.7988	1.8104	1.8064	1.7932	1.8100	1.8179	1.8345	1.8280	1.8470	1.8308	1.8463	1.8288	1.8199	1.7892	1.8019	1.7949	1.8047	1.8227	1.8270
0.5	1.7888	1.7980	1.8104	1.8064	1.7932	1.8100	1.8164	1.8311	1.8288	1.8463	1.8345	1.8485	1.8278	1.8059	1.7801	1.8019	1.7949	1.8047	1.8331	1.8370
1	1.7888	1.7988	1.8105	1.8065	1.7933	1.8111	1.8168	1.8321	1.8287	1.8481	1.8260	1.8470	1.8294	1.8183	1.7829	1.8094	1.7949	1.8032	1.8220	1.8276
1.5	1.7883	1.7939	1.8123	1.8043	1.8098	1.8022	1.8316	1.8428	1.8252	1.8487	1.8134	1.8445	1.8263	1.8160	1.7936	1.8163	1.8170	1.8102	1.8310	1.8370
2	1.7888	1.7939	1.8123	1.8043	1.8098	1.8022	1.8316	1.8350	1.8115	1.8387	1.8234	1.8445	1.8132	1.8160	1.7936	1.8163	1.8170	1.8102	1.8372	1.8370
2.5	1.7888	1.7939	1.8123	1.8043	1.8098	1.8022	1.8316	1.8350	1.8215	1.8276	1.8244	1.8445	1.8163	1.8160	1.7936	1.8163	1.8170	1.8102	1.8210	1.8570
3.5	1.7888	1.7939	1.8123	1.8043	1.8098	1.8169	1.8255	1.8357	1.8869	1.8951	1.9233	1.8924	1.8563	1.8441	1.8070	1.9003	1.8044	1.8102	1.8058	1.9278
5	1.7888	1.7939	1.8125	1.8043	1.8098	1.8187	1.8255	1.8357	1.9169	1.9172	1.9451	1.9461	1.8590	1.8500	1.8688	1.9252	1.7996	1.8048	1.8048	1.9169

A.6 VERSATILIDADE - MEDIÇÕES

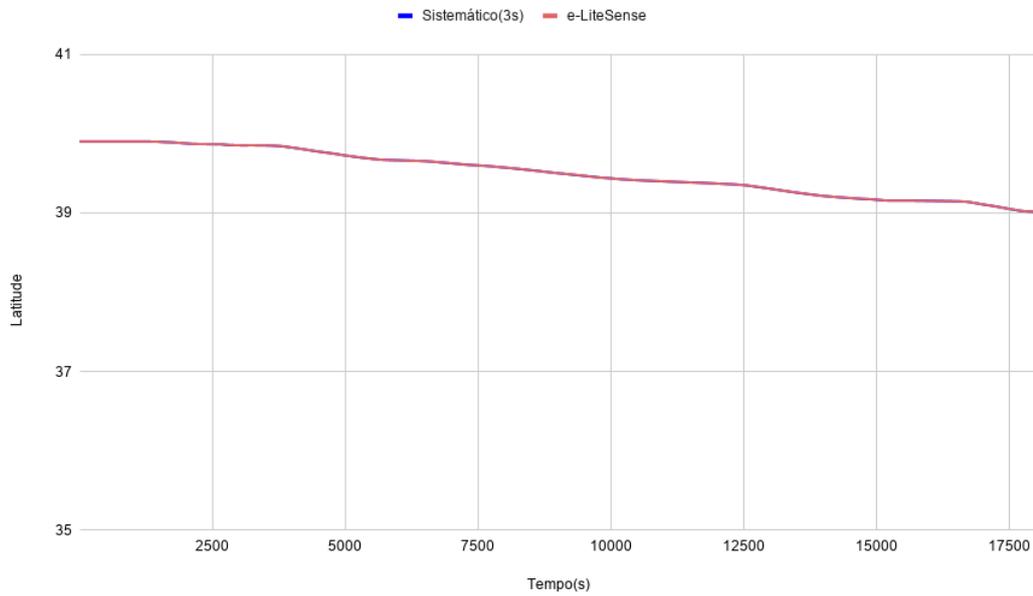


Figura 61: Comparação das leituras para o dataset da Latitude

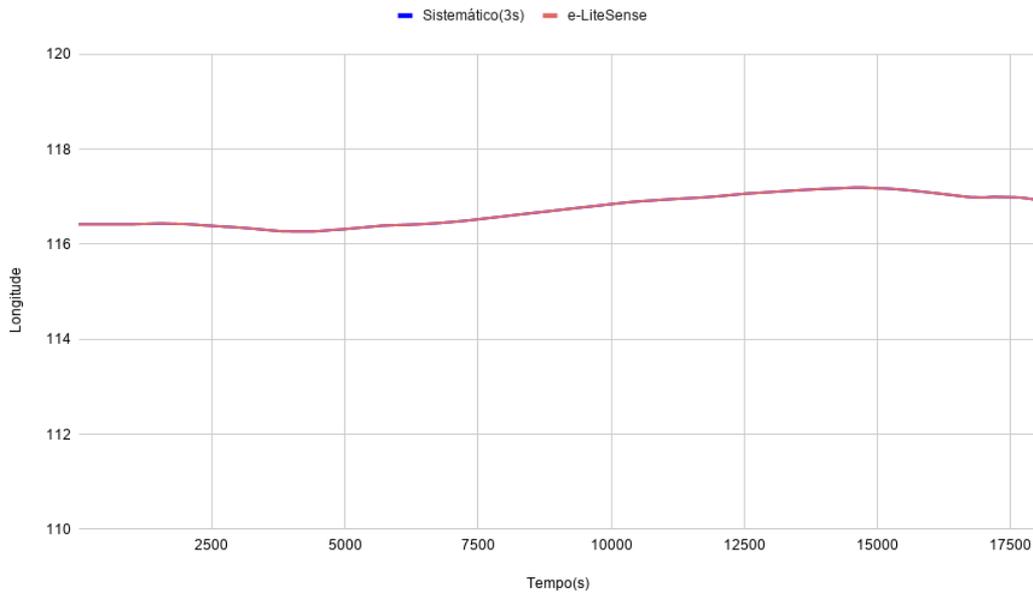


Figura 62: Comparação das leituras para o dataset da Longitude

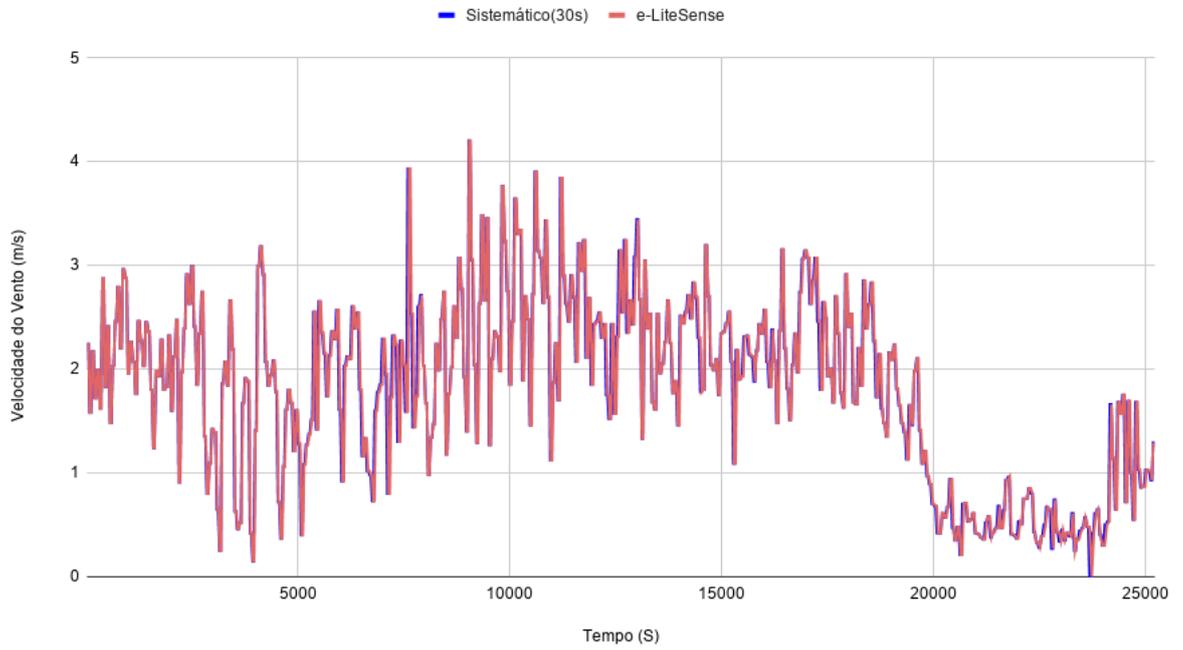


Figura 63: Comparação das leituras para o dataset da Velocidade do Vento

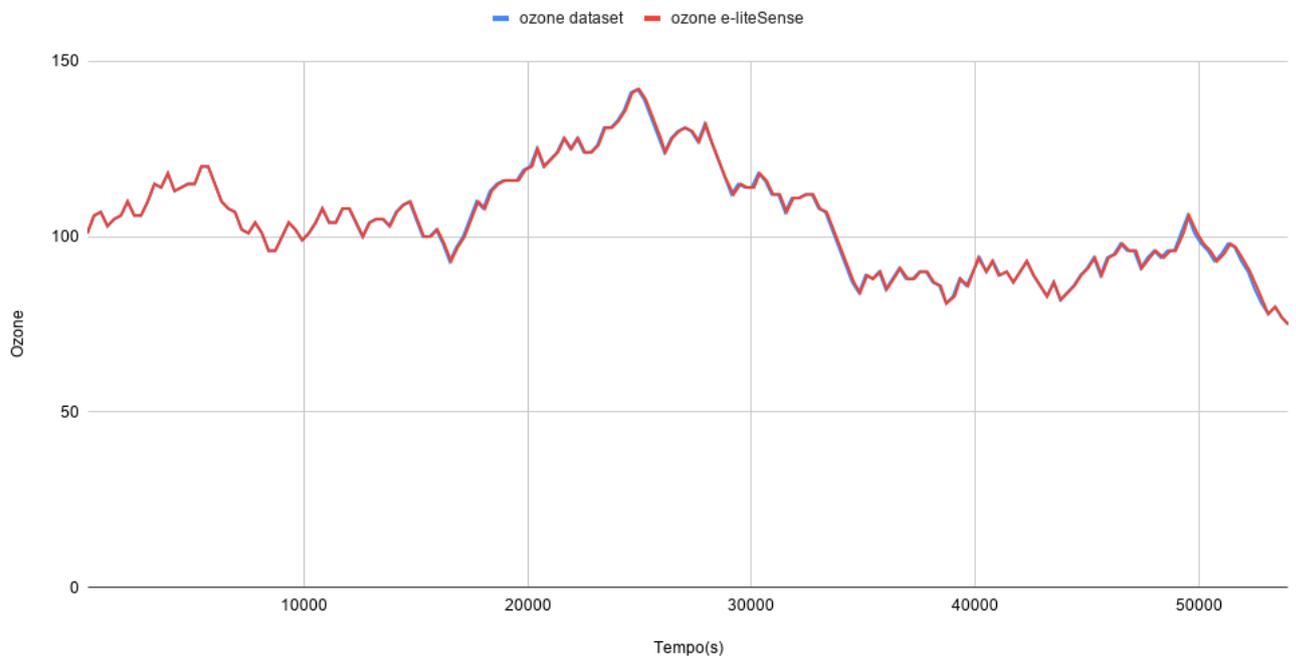


Figura 64: Comparação das leituras para o dataset da Ozono

A.7 VERSATILIDADE - CONSUMO ENERGÉTICO

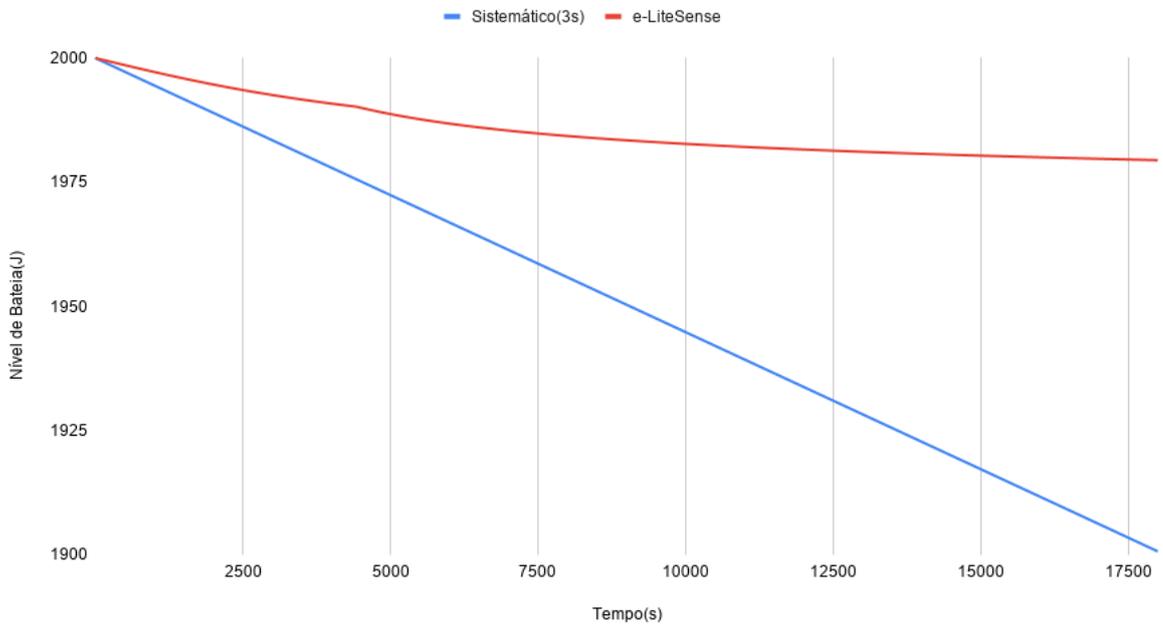


Figura 65: Comparação do consumo energético para o dataset da Longitude

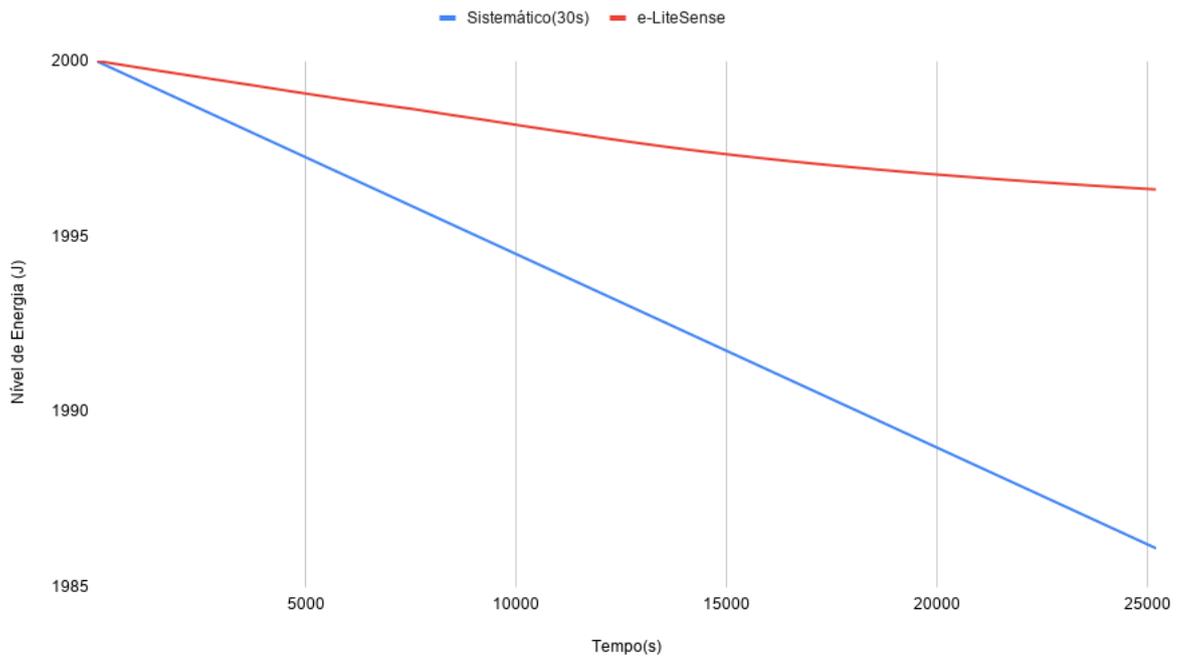


Figura 66: Comparação do consumo energético para o dataset da Vel. Vento

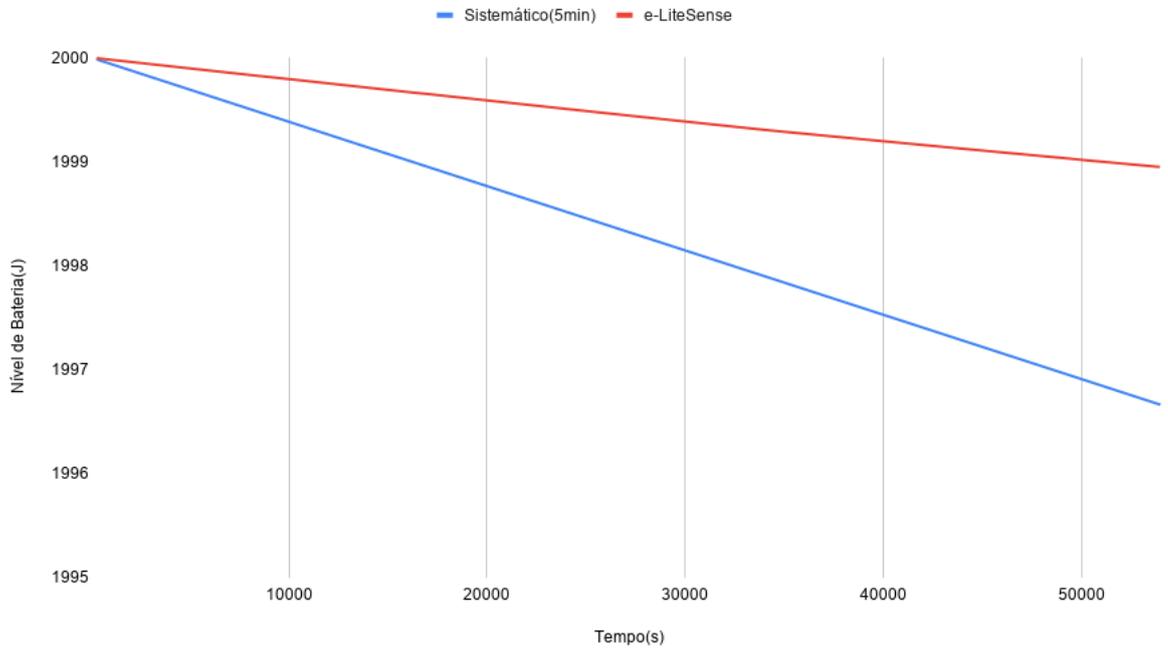


Figura 67: Comparação do consumo energético para o dataset de Ozono

BIBLIOGRAFIA

- [1] JAtzori L, Iera A, Morabito G., The Internet of Things: A survey. *Computer Networks* oct 2010, URL <http://linkinghub.elsevier.com/retrieve/pii/S1389128610001568> .
- [2] Kamal Mehdi, Massinissa Lounis, AhcÃne Bounceur and Tahar Kechadi. CupCarbon: A multi-agent and discrete event wireless sensor network design and simulation toll. *In IEEE 7th Internation Conference on Simulation Tools and Techniques (SIMUTools'14, Lisbon, Portugal, March 17-19 2014.*
- [3] M. Lounis K. Mehdi, and A. Bounceur. A cupCarbon tool for simulating destructive insect movements. *1st IEEE 7th Internation Conference on Information and Communication Technologies for Disaster Management, Algiers, Algeria, March 24-25 2014*
- [4] ANR PROJECT PERSETPEUR. Cupcarbon simulator, 2015.
- [5] Alexandru Lavric, Popa Valentin, Stefan Sfichi, Street lighting control system based on large-scale WSN: A step towards a smart city. *Conference: 2014 International Conference and Exposition on Electrical and Power Engineering (EPE) October 2014*
- [6] Vijender Kumar Solanki, M.Venkatesan, Somesh Katiyar, Conceptual Model for Smart Cities: Irrigation and Highway Lamps using IoT. *International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 4, N^o3*
- [7] Luis Canom Claudio Ortega, Alvaro Talavera, JuanG Lazo. *Smart City Park Irrigation System: A Case Study of San Isidro*. Academic Department of Engineering, Universidad del Pacífico, Lima Perú 25 Outubro 2018
- [8] Shweta B. Saraf , Dhanashri H. Gawali, IoT based smart irrigation monitoring and controlling system. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) 19-20 May 2017*
- [9] "Rega Inteligente em Castelo Branco", URL http://www.allbesmart.pt/reports/Smart_irrigation_use_case.pdf
- [10] S. Silva, T. Cardoso, P. Barros, H. Ribeiro, P. Carvalho and S. Rito Lima, "A Flexible System for Optimising Green Spaces Irrigation," 2020 5th International Conference

- on Smart and Sustainable Technologies (SpliTech), Split, Croatia, 2020, pp. 1-6, doi: 10.23919/SpliTech49282.2020.9243756.
- [11] João Marco C. Silva, Kalil Araujo Bispo, Paulo Carvalho, Solange Rito Lima, e-LiteSense: Self-adaptive Energy-aware Data Sensing in WSN Environments. The 22nd IEEE Symposium on Computers and Communications (IEEE ISCC 2017), Heraklion, Crete, Greece.
- [12] Yetgin H, Cheung KTK, El-Hajjar M, Hanzo LH. A survey of network lifetime maximization techniques in wireless sensor networks. IEEE Communications Surveys Tutorials Secondquarter 2017;
- [13] Application of Compressive Sensing Techniques in Distributed Sensor Networks: A Survey. arXiv e-prints Sep 2017;
- [14] Mehrjoo S, Khunjush F. Distributed semi-adaptive compressive sensing data collection in wireless sensor networks. International Journal of Communication Systems 2018; 31(9):e3546, doi:10.1002/dac.3546. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/dac.3546>
- [15] Wu X, Wu J, Yang J. Adaptive sensing scheduling for energy harvesting sensors with finite battery. 2015 IEEE International Conference on Communications (ICC), 2015;
- [16] Chou CT, Rana R, Hu W. Energy efficient information collection in wireless sensor networks using adaptive compressive sensing. 2009 IEEE 34th Conference on Local Computer Networks, 2009
- [17] Quer G, Masiero R, Pillonetto G, Rossi M, Zorzi M. Sensing, Compression, and Recovery for WSNs: Sparse Signal Modeling and Monitoring Framework. IEEE Transactions on Wireless Communications oct 2012;
- [18] Gupta V, De S. Sbl-based adaptive sensing framework for wsn-assisted iot applications. IEEE Internet of Things Journal Dec 2018;
- [19] Kerasiotis F, Prayati A, Antonopoulos C, Koulamas C, Papadopoulos G. Battery Lifetime Prediction Model for a WSN Platform. 2010 Fourth International Conference on Sensor Technologies and Applications, IEEE, 2010; 525– 530, doi:10.1109/SENSORCOMM.2010.85.URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5558125>

- [20] Antonopoulos C, Prayati A, Stoyanova T, Koulamas C, Papadopoulos G. Experimental evaluation of a WSN platform power consumption. 2009 IEEE International Symposium on Parallel & Distributed Processing, IEEE, 2009; 1–8, doi:10.1109/IPDPS.2009.5161185. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5161185>
- [21] Jacobson V. Congestion Avoidance and Control. Symposium Proceedings on Communications Architectures and Protocols, SIGCOMM '88, ACM: New York, NY, USA, 1988; 314–329, doi:10.1145/52324.52356. URL <http://doi.acm.org/10.1145/52324.52356>.
- [22] Kobo HI, Abu-Mahfouz AM, Hancke GP. A survey on software-defined wireless sensor networks: Challenges and design requirements. IEEE Access 2017; 5:1872–1899, doi:10.1109/ACCESS.2017.2666200.
- [23] Sikora-Fernandez, Dorota and Stawasz, Danuta. THE CONCEPT OF SMART CITY IN THE THEORY AND PRACTICE OF URBAN DEVELOPMENT MANAGEMENT. Romanian Journal of Regional Science, 10:86–99,06 2016
- [24] Suthaharan S, Alzahrani M, Rajasegarar S, Leckie C, Palaniswami M. Labelled. Data collection for anomaly detection in wireless sensor networks. Intelligent sensors, sensor networks and information processing (ISSNIP), 2010 sixth international conference on, IEEE, 2010; 269–274
- [25] S.Kimbrough, S. Krabbe, R. Baldauf, T. Barzyk, M. Brown, S. Brown, C. Croghan, M. Davis, P. Deshmukh, R. Duvall, S. Feinberg, V. Isakov, R. Logan, T. McArthur, and A. Shields. Characterizing Community Air Pollution Impacts Near Complex Sources: The Kansas City Transportation and Local-Scale Air Quality Study (KC-TRAQS). URL <https://data.world/us-epa-gov/94da11c8-9c5e-4993-bda5-cd91e4dbae7f>.
- [26] Georges Hebrai, Alice Berard. UCI Machine Learning Repository. URL <https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption>
- [27] Yu Zheng, Lizhu Zhang, Xing Xie, Wei-Ying Ma. Mining interesting locations and travel sequences from GPS trajectories. In Proceedings of International conference on World Wild Web (WWW 2009), Madrid Spain. ACM Press: 791-800
- [28] Muhammad Intizar Ali, Feng Gao and Alessandra Mileo, "CityBench: A Configurable Benchmark to Evaluate RSP Engines Using Smart City Datasets", The

Semantic Web - ISWC 2015 - 14th International Semantic Web Conference, October 11-15, 2015, Bethlehem, PA, USA

- [29] Stefan Bischof, Athanasios Karapantelakis, Cosmin-Septimiu Nechifor, Amit Sheth, Alessandra Mileo and Payam Barnaghi, "Semantic Modeling of Smart City Data", Position Paper in W3C Workshop on the Web of Things: Enablers and services for an open Web of Devices, 25-26 June 2014, Berlin, Germany
- [30] W. Shi and S. Dustdar, "The Promise of Edge Computing," in *Computer*, vol. 49, no. 5, pp. 78-81, May 2016, doi: 10.1109/MC.2016.145.
- [31] Radu Dobrescu and Florin Ionescu. "Topological models of complex networks", in *Large Scale Networks: Modeling and Simulation au.*(Boca Raton: CRC Press, 10 out 2016)
- [32] Boudewijn R. Haverkort, Henrik C. Bohnenkamp, Connie U. Smith. *Computer Performance Evaluation: Modelling Techniques and Tools*, 11th International Conference, TOOLS 2000, Schaumburg, IL, USA, March 27-31, 2000, Proceedings.
- [33] "Zigbee alliance," <https://zigbeealliance.org/solution/zigbee/>.
- [34] "Wifi alliance" <https://www.wi-fi.org/>
- [35] A. Zourmand, A. L. Kun Hing, C. Wai Hung and M. AbdulRehman, "Internet of Things (IoT) using LoRa technology," 2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Selangor, Malaysia, 2019, pp. 324-330, doi: 10.1109/I2CACIS.2019.8825008.
- [36] "IPMA", https://www.ipma.pt/pt/agrometeorologia/mapas/diario/index.jsp?page=dts5_co.xml , aceso a 12 Março.
- [37] D. de la Rosa and R. Sobral, *Soil Quality and Methods for its Assessment*. Dordrecht: Springer Netherlands, 2008, pp. 167-200. [Online]. Available: https://doi.org/10.1007/978-1-4020-6778-5_9