UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

Juan David Arias Correa

Compressão de dados com perda em dispositivos da Internet das Coisas

Florianópolis
2020

Juan David Arias Correa

Compressão de dados com perda em dispositivos da Internet das Coisas

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas para a obtenção do título de Mestre em Engenharia de Automação e Sistemas.
Orientador: Prof. Alex S. Roschildt Pinto, Dr.
Coorientador: Prof. Carlos Barros Montez, Dr.

Florianópolis
2020

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

Arias Correa, Juan David
Compressão de dados com perda em dispositivos da Internet das Coisas / Juan David Arias Correa ; orientador, Alex Sandro Roschildt Pinto, coorientador, Carlos Barros Montez, 2020.
105 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Florianópolis, 2020.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Internet das coisas. 3. Compressão de dados. 4. Compressão com perda. I. Roschildt Pinto, Alex Sandro . II. Barros Montez, Carlos . III. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia de Automação e Sistemas. IV. Título.

Juan David Arias Correa

Compressão de dados com perda em dispositivos da Internet das Coisas

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

Prof[a]. Kalinka Regina Lucas Jaquie Castelo Branco, Dra.
Universidade de São Paulo

Prof[a]. Luciana de Oliveira Rech, Dra.
Universidade Federal de Santa Catarina

Prof. Marcos Fagundes Caetano, Dr.
Universidade de Brasilia

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de Mestre em Engenharia de Automação e Sistemas.

Documento assinado digitalmente
Werner Kraus Junior
Data: 30/04/2020 08:34:38-0300
CPF: 531.085.239-53

Prof. Werner Kraus Junior, Dr
Coordenador do Programa

Documento assinado digitalmente
Alex Sandro Roschildt Pinto
Data: 23/04/2020 09:48:02-0300
CPF: 802.787.440-87

Prof. Alex S. Roschildt Pinto, Dr.
Orientador

Florianópolis, 2020.

dedicado à minha família, amigos e a todos aqueles que
acompanharam e contribuíram para que este processo resultasse
em grande aprendizado e crescimento pessoal.

AGRADECIMENTOS

Agradeço a meus orientadores pelo excelente trabalho alcançado e por todo o apoio a mim dispendido. À banca, pelo esforço e pelo asserto das considerações, meu muito obrigado. Agradeço também aos professores do PPGEAS pelos ensinamentos ao longo desses dois anos. Aos estudantes do LAPESD por todo o apoio, comentários e sugestões neste processo, bem como a todos os mestrandos anônimos que estudaram comigo no PPGEAS, obrigado pelo conhecimento, comentários e experiencias compartilhadas na minha trajetória na UFSC.

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

Por la ignorancia nos han dominado más que por la fuerza.
(Simon Bolivar)

RESUMO

A Internet das Coisas ou *Internet of Things* (IoT) apresenta a visão de conectar *coisas* do mundo real com o mundo digital através da internet. As coisas são embarcadas com tecnologias que as transformam em dispositivos inteligentes, sensores e atuadores que lhes permitem interagir com o ambiente, e estes contam com um sistema de comunicação para conectá-los com outros dispositivos, serviços ou servidores. A transmissão do estado atual e das amostras coletadas dos sensores em um dispositivo da IoT é uma tarefa fundamental na maioria das aplicações, porém é um dos maiores desafios para a IoT atualmente. O uso do sistema de comunicação é uma das principais fontes de consumo energético do dispositivo. Considerando que os dispositivos tendem a ter limitações energéticas, soluções que permitam reduzir a utilização do sistema de comunicação são essenciais para melhorar o desempenho e tempo de vida do sistema. A coleta de amostras por parte do dispositivo e seu envio, além dos custos causados pelo transporte, gerenciamento, processamento e armazenamento destes, podem afetar a viabilidade de projetos de IoT, assim, soluções de Compressão de Dados (CD) se fazem necessárias. Os mecanismos de CD representam os dados de uma forma comprimida e, posteriormente, com um processo de descompressão, se pode recuperar os dados originais. Existem algoritmos sem perda nos quais os dados originais podem ser completamente recuperados e, com perda, nos quais só é possível recuperar uma aproximação destes. Os algoritmos com perda têm uma complexidade computacional menor, além de conseguir representar os dados com uma menor quantidade de dados. A presente dissertação tem como objetivo projetar mecanismos de compressão de dados com perdas capazes de serem executados em ambientes com capacidades de processamento e memória principal extremamente limitados, como são os dispositivos utilizados na IoT. Considerando as propostas verificadas nos trabalhos relacionados , o método *Swinging Door Trending* (SDT) é o estudado. O SDT é um método de interpolação linear que tem uma complexidade computacional igual a O(1). O SDT demanda a definição de um parâmetro de tolerância que é um mecanismo comum nos algoritmos de CD com perda, quando estes parâmetros são escolhidos incorretamente podem reduzir o nível de compressão ou incrementar o erro o suficiente para que os dados originais fiquem irreconhecíveis. Desta forma, a proposta da dissertação é adicionar uma etapa de autoajuste e autodefinição do parâmetro de tolerância utilizando como estudo de caso o SDT. O foco da dissertação são os dados dos sensores coletados pelo dispositivo, o comportamento do sinal coletado em cada dispositivo sera utilizado como base para a etapa de autoajuste e autodefinição. Os algoritmos de CD com perdas são avaliados considerando a Taxa de erro e de compressão que permitem avaliar o desempenho dos algoritmos. Uma métrica de avaliação chamada de Critério de Compressão (CC) baseada no *F-Score* é proposta como métrica de qualidade e de seleção dos parâmetros em algoritmos com perdas. Os resultados mostram que os mecanismos propostos funcionam adequadamente com o SDT, conseguindo bons resultados em termos de compressão e erro, sem aumentar a complexidade computacional do algoritmo a ser executado no dispositivo.

Palavras-chave: Internet das Coisas, Compressão de dados. Compressão com perdas. Sensores.

ABSTRACT

Internet of Things (IoT) is the vision to connect things that are part of the real world with the digital world through the internet. These things are embedded with technologies to turn them into smart device. They must have communication systems for connecting with services, servers and other devices. Devices interact with the environment through sensors and actuators. Things have as an important task the transmission of information such as their current state or sensed data. The use of the communication system by devices consumes a significant portion of their energy supply, making energy efficiency an important challenge for IoT due to energy limitations of devices. The collection of large volumes of data from IoT devices implies a significant storage and transmission cost. Data Compression (DC) is a possible solution for the economic and energetic viability of some IoT projects in face of IoT popularization. DC is the process of reducing the quantity of data required to represent the same volume of information.A decompression process can restore the original data from the compressed one. There are lossless mechanisms that can recover the original data and lossy that only restore an approximation of them. The lossy methods can compress some information with less quantity of data and has less computational complexity than the lossless one. This dissertation focuses in a lossy DC mechanism to compress the sensor measurements inside IoT devices. The *Swinging Door Trending* (SDT) is chosen considering the proposals found in the literature. The SDT is a linear interpolation method with a computational complexity equal to O(1). The SDT requires the selection of the Deviation Compression (DC) parameter that is fundamental for compression. When the DC is incorrectly selected, it can reduce the Compression Rate or increase the Error Rate (some cases so much that the data is unrecognizable). This dissertation proposes to modify the SDT, adding a DC auto-definition process and an auto-adjustment proces, considering the behavior of the collected signal from the sensors. lossy CD algorithms can be evaluated by equations that represent the Compression Error (CE) and the Compression rate (CR). This dissertation proposes a new performance metric called Compression Criterion (CC), this is based on the *F-Score* used to evaluate the performance of Artificial intelligence (AI) algorithms. The *F-Score* provides a single measure combining two other desirables metrics such as precision and call in AI, the proposes uses the CE and CR as metrics. The CC is additionally used as a selective criterion of the DC value. The results conclude that the proposed mechanisms work properly with the SDT in IoT devices and CC proved to be a good evaluation metric.

Keywords: Internet of Things, Data compression, Lossy compression, sensors.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE LISTAGENS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

ADCS	*Adaptive Data Compression Scheme*. 50, 51
AE	*neural AutoEncoder*. 54
AEE	*Adaptive Entropy Encoder*. 52
AM	Aprendizado de Máquina. 27, 48, 50, 53
AR-MWCEB	*self-Adaptive Regression-based Multivariate Data Wavelet Compression Scheme with Error Bound*. 50, 57
ASDT	*Adaptive Swinging Door Trending*. 58, 61, 64, 78, 79, 80, 81
AVSC	*Approximate Vector Stream Compression*. 50, 51
BEDCA	*Bounded-Error Data Compression and Aggregation*. 51
CC	Critério de Compressão. 15, 17, 30, 31, 62, 64, 66, 68, 75, 76, 77, 78, 81, 82, 84, 86, 90
CD	Compressão de Dados. . 15, 29, 30, 31, 33, 34, 35, 36, 37, 38, 41, 42, 43, 44, 46, 48, 49, 50, 53, 54, 55, 56, 59, 61, 62, 64, 89, 90, 91
CE	Critério de Exclusão. 17, 44, 45, 47
CI	Critério de Inclusão. 17, 44, 45, 47
CNN	*Compressing Neural Network*. 54
D-BEDCA	*Dynamic Bounded-Error Data Compression and Aggregation*. 51
DC	Desvio da Compressão. 15, 17, 38, 40, 59, 61, 63, 64, 65, 66, 75, 77, 78, 79, 80, 81, 82, 83, 84, 86, 90, 99
DCA	*Data Compression Algorithm*. 54
DCT	*Discrete Cosine Transform*. 58
DHC	Diferença Horária de Chegada. 53, 55, 58
DPCM	*Differential Pulse Code Modulation*. 50, 52
DPNR	Diferença percentual Normalizada da RMQ. 36
DPR	Diferença percentual da RMQ. 36, 37, 62
DSDT	*Distributed Swinging Door Trending*. 50, 55, 59
DWT	*Discrete Wavelet transform*. 58
EAR	Erro Absoluto Relativo. 36, 62
EE	Espaço Economizado. 37, 62, 75
EHCC	*Embedded Harmonic Components Coding*. 50, 52, 53
EQM	Erro Quadrático Médio. 36
EQR	Erro Quadrático Relativo. 36, 62
ETEO	*Extension Temporal Edge-operator*. 57

FFNN *Feed-Forward Neural Network*. 57
FQ Fator de Qualidade. 37
FTC *Fuzzy Transform Compression*. 58
FuzzyCat *Fuzzy Compression Adaptative Transform*. 50, 58

GC Ganho da Compressão. 37
GPC *Generalized Predictive Coding*. 52

HASDC *Hierarchical Adaptive Spatio-Temporal Data Compression*. 50, 58

IA Inteligência artificial. 17, 27, 48, 49, 53, 54, 89
IE Inferência estatística. 27, 48, 50, 54
IL Interpolação linear. 28, 48, 50, 55, 56
INL Interpolação Não Linear. 48, 50
IoT *Internet of Things*. . . 15, 29, 30, 31, 33, 35, 41, 42, 43, 44, 45, 48, 60, 61, 64, 69, 72, 89, 90
IQ Índice de qualidade. 37, 75
ISDT *Improved Swinging Door Trending*. 59, 64

JSON *JavaScript Object Notation*. 70, 72

K-RLE *K-precision Run Length Encoding*. 50, 53

LaPeSD Laboratório de Pesquisa de Sistemas Distribuidos. 72, 78
LCF *Linear-Curve Fitting*. 57
LSWT *Lifting Scheme Wavelet Transform*. 53
LTC *Lightweight temporal compression*. 50, 56

MDCT *Modified Discrete Cosine Transform*. 50, 52, 53
MEGC *Modified Exponential Golomb Code*. 52
MME Média Móvel exponencial. 64, 65, 78, 79
MQTT *Message Queuing Telemetry Transport*. 70

NADPCMC *Non-linear Adaptive Pulse Coded Modulation-Based Compression*. . . 50, 57

PBSA *Precision Based Sampling and Transmit Algorithm* 50, 54
PC Proporção de Compressão. 36, 37, 75
PLAMLiS *top-down Piecewise Linear Approximation with Minimum number of Line Segments*. 50, 55, 56, 63

PNF *Preceding Neighbor Fitting*. 57
PP Pergunta de Pesquisa. 42, 44, 45, 46

QCF *Quadratic-Curve Fitting*. 57

RBM *standard Restricted Boltzmann Machine*. 53
RC Razão de Compressão. 36, 37
RLE *Run Length Encoding*. 53
RMQ Raiz da Média Quadrática. 36, 37
RSL Revisão Sistemática da Literatura. 30, 31, 32, 41, 43, 44, 46, 47, 48
RSR Relação Sinal-Ruído. 36
RSSF Rede de Sensores Sem Fio. 35, 42, 44, 53, 58, 59

S-LEC *Sequential Lossless Entropy Compression*. 52
SQS *Summarizing event seQuenceS*. 51
S-RBM-AE *Stacked RBM Autoencoder*. 50, 53
SCF Sistema Ciberfísicos. 29
SDT *Swinging Door Trending*. . . 15, 30, 31, 35, 38, 39, 40, 55, 58, 59, 61, 63, 64, 65, 66, 68, 72, 73, 78, 80, 81, 83, 90, 99, 100, 101, 102
SME *Sensor Manufacture Error*. 17, 51, 52, 81, 83, 84
SSDT *Self-definition Swinging Door Trending*. 28, 64, 65, 68, 73, 78, 80, 90
STI Sistemas de Transporte Inteligente. 53, 55, 58
SZ *Squeeze*. 57

TC Taxa de Compressão. . . 15, 30, 32, 36, 37, 50, 52, 53, 54, 56, 57, 62, 63, 64, 68, 75, 76, 77, 78, 79, 80, 83, 84, 90
TE Taxa de Erro.30, 32, 35, 36, 37, 50, 53, 55, 56, 57, 62, 63, 64, 68, 75, 77, 79, 80, 83, 84, 90
TI Tecnologias da Informação. 29, 35
TS Taxa de Similaridade. 15, 62, 63, 76, 77, 78
TSDT *Training Swinging Door Trending*. . 17, 66, 68, 73, 75, 76, 81, 82, 83, 84, 86, 90

SUMÁRIO

1 INTRODUÇÃO

As Tecnologias da Informação (TI) têm adquirido uma maior relevância nos âmbitos industrial e social, com a popularização e distribuição de novas tecnologias, visões e serviços como as redes sociais, Internet das Coisas ou *Internet of Things* (IoT), Sistema Ciberfísicos (SCF), Indústria 4.0 entre outros. Atualmente, a taxa de crescimento da geração de dados por parte de TI é maior que a das tecnologias de armazenamento e transmissão (UTHAYAKUMAR; VENGATTARAMAN; DHAVACHELVAN, 2018; HOSSEINI; MOHAMMAD, 2012). Este fenômeno é causado pela diversificação de usos da TI que abarca desde o entretenimento, cuidado da saúde, segurança, accessórios para animais de estimação, cidades inteligentes, entre muitas outras aplicações.

A IoT é uma das áreas de TI que demonstra expressiva evolução nas últimas duas décadas. Ela representa uma visão na qual as *coisas* do mundo real conectam-se com o mundo digital através da internet (COETZEE; EKSTEEN, 2011). A IoT torna transparente a separação entre o mundo físico e digital, o que permite a criação e melhoramento de produtos e serviços (DORSEMAINE et al., 2016). Em 2011, a quantidade de objetos conectados excedia a população mundial (GUBBI et al., 2013). Estima-se que, até o ano de 2021, aproximadamente 27 bilhões de nodos IoT estarão conectados (Ud Din et al., 2019).

As *coisas*, sob a visão de IoT, são elementos embarcados com tecnologias que as transformam em dispositivos inteligentes. Sensores e atuadores permitem a interação destas com o entorno e, nesse cenário, as tecnologias de comunicação são fundamentais. A transmissão das amostras coletadas de sensores pelos dispositivos é uma tarefa comum.

As plataformas baseadas em computação na nuvem e névoa (*Cloud and Fog computing*) provêem um sistema de armazenamento, gerenciamento e processamento das *coisas*, o que cria um ecossistema dinâmico da IoT (PAUL; SARASWATHI, 2017). Atualmente, a grande quantidade de dispositivos conectados gera um alto volume de dados e, como em qualquer sistema, estes precisam ser transmitidos, processados e armazenados. A IoT depende do sistema de comunicação usado pelo dispositivo, e esse sistema é responsável por uma parte relevante do consumo energético destes (AZAR et al., 2019a; Stojkoska; Nikolovski, 2017; Harb; Makhoul; Abou Jaoude, 2018). Este consumo pode ser problemático, uma vez que os dispositivos tendem a ter limitações energ éticas. Além disso, o processamento e armazenamento da informação transmitida pelos dispositivos têm consequências econômicas devido ao uso dessas plataformas (BISWAS; GIAFFREDA, 2014; PAUL; SARASWATHI, 2017).

A diminuição da quantidade de dados que precisa ser transmitida pelos dispositivos pode reduzir o consumo energético causado pelo sistema de comunicação (Shravana; Veena, 2017; HOSSEINI; MOHAMMAD, 2012), além dos custos advindos do gerenciamento dos dados. Os mecanismos de Compressão de Dados (CD) são uma das soluções apresentadas na literatura. A CD é a utilização de um processo ou método de compressão em que um sinal passa a ser representado de uma forma reduzida, com o objetivo de poupar espaço de armazenamento e tempo de transmissão (MAHDI; MOHAMMED; MOHAMED, 2013). Os mecanismos de

compressão podem ser *sem perda*, nos quais o resultado do processo de descompressão gera dados exatamente iguais aos dados originais. Outros mecanismos de CD são denominados *com perda* ou *irreversíveis*, já que, por meio destes, só é possível recuperar uma aproximação dos dados originais.

Considerando esse contexto, esta dissertação de mestrado busca responder a seguinte pergunta de pesquisa: **"– É possível contribuir com o aprimoramento dos algoritmos de compressão de dados com perda voltados para serem usados nos dispositivos da IoT? Contribuir, principalmente, com os algoritmos de compressão de dados amostrados por dispositivos sensores?"**.

Mais especificamente, a presente dissertação tem como objetivo projetar mecanismos de compressão de dados com perdas capazes de serem executados em ambientes com capacidades de processamento e memoria limitados, como são os dispositivos utilizados na IoT. Os algoritmos de CD reduzem a utilização do sistema de comunicação e armazenamento e, assim, diminuem o consumo energético. Os mecanismos de CD com perda, apesar do prejuízo na precisão dos dados, contam com benefícios em nível de compressão e utilização de recursos computacionais interessantes. Outro benefício da utilização da CD é diminuir o consumo dos sistemas de armazenamento e gerenciamento, o que significaria uma despesa menor pelo uso das plataformas baseadas em computação em nuvem ou de névoa.

Considerando os objetivos desta dissertação, uma Revisão Sistemática da Literatura (RSL) foi feita para conhecer o estado atual da literatura. De acordo com os resultados obtidos nessa revisão, o *Swinging Door Trending* (SDT) (BRISTOL, 1990) foi escolhido como base para uma nova proposta de compressão de dados em dispositivos da IoT. O SDT é um método de interpolação linear que tem uma complexidade computacional igual a O(1).

O SDT demanda a definição de um parâmetro de tolerância, prática comum nos algoritmos de CD com perda. Quando estes parâmetros são escolhidos incorretamente, eles podem reduzir o nível de compressão ou aumentar o erro de forma que os dados originais fiquem irreconhecíveis. O desempenho dos algoritmos de CD com perdas são usualmente avaliados considerando duas métricas denominadas Taxa de Erro (TE) e a Taxa de Compressão (TC). Nesta dissertação, uma nova métrica de avaliação chamada de Critério de Compressão (CC) baseada no *F-Score* (ou F1-Score) é proposta como métrica de qualidade da compressão. A métrica *F-Score* é empregada na área de inteligência artificial, e avalia o rendimento dos algoritmos de classificação, combinando duas outras métricas denominadas "precisão" e "*recall*". Baseada nessa métrica, O CC proposta nesta dissertação é criada através da combinação das métricas TE e TC.

As propostas encontradas na RSL, em sua maioria, focam em respeitar um nível de erro ou cumprir um objetivo de compressão. Estes enfoques podem limitar consideravelmente a compressão alcançada, ou gerar um erro alto por cumprir uma meta de compressão não adequada para o comportamento dos dados. O CC é utilizado também como critério de seleção dos parâmetros, esta métrica foi desenhada utilizando a média harmônica que tem maior sensibilidade a grandes diferenças entre métricas, o que a torna adequada para equilibrar a TE e TC, e

assim escolher os parâmetros considerando estas duas métricas.

Os resultados mostram que os mecanismos propostos funcionam adequadamente com o SDT conseguindo bons resultados em termos de compressão e erro, sem aumentar a complexidade computacional do algoritmo a ser executado no dispositivo. O CC demostrou ser uma boa métrica de qualidade e avaliação, além de conseguir bons resultados como métrica de seleção de parâmetros, o que abre a porta para ser utilizada com outros algoritmos com perdas.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Este trabalho tem como objetivo construir soluções de compressão de dados com perda que serão executadas dentro dos dispositivos da IoT em amostras de sensores.

1.1.2 Objetivos Específicos

- Propor um mecanismo de compressão de dados com perda para ser usado embarcado em dispositivos da IoT.
- Implementar uma aplicação da IoT para testar o algoritmo de compressão de dados proposto;
- Desenvolver processos para automatizar a seleção e ajuste de parâmetros dos algoritmos de compressão de dados.
- Desenvolver uma métrica de avaliação adequada para avaliar algoritmos com perda.

1.2 METODOLOGIA

A metodologia utilizada no desenvolvimento desta pesquisa é composta pelas seguintes etapas:

1. Conduzir uma RSL focada nos algoritmos, métodos e esquemas de CD com perda propostos para serem executados dentro de dispositivos da IoT. Adicionalmente, apresentar uma taxonomia para classificar as propostas.
2. Escolher um algoritmo de CD com perda para ser usado dentro dos dispositivos da IoT.
3. Realizar uma pesquisa sobre o algoritmo de CD escolhido.
4. Implementar um ambiente de teste que permita medir o rendimento da proposta avaliada.

5. Usando o algoritmo escolhido, propor um novo esquema de compressão de dados que considere as características técnicas dos sensores e o comportamento do sinal a comprimir.

6. Implementar o esquema proposto no ambiente de testes.

7. Avaliar o desempenho do esquema proposto, considerando a TE e a TC, e compará-lo com outras propostas encontradas na RSL.

1.3 ORGANIZAÇÃO DO TEXTO

Este documento está organizado em 6 capítulos. No segundo capítulo são descritos os principais conceitos sobre compressão de dados voltados para compressão com perda de dados de sensores. No terceiro capítulo são apresentados os resultados de uma revisão sistemática da literatura sobre os trabalhos relacionados. Nesse capítulo, uma taxonomia proposta para classificar esses trabalhos também é apresentada. No quarto capítulo é apresentada a proposta de auto-definição de parâmetros em um algoritmo de compressão baseado na técnica *Swinging Door Trending*. No quinto capítulo é introduzido um ambiente de testes e os resultados obtidos pela proposta. Finalmente, no sexto capítulo são apresentados os principais resultados e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capitulo serão brevemente apresentados os principais conceitos da pesquisa. A Seção 2.1 traz uma pequena definição de IoT, a Seção 2.2 aborda a CD e na Subseção 2.2.1 apresentam-se algumas métricas básicas de avaliação dos algoritmos de compressão de dados.

2.1 A INTERNET DAS COISAS

Os avanços nas tecnologias de comunicação sem fio, eletrônica e sensores têm fornecido suportes à evolução do paradigma da IoT. Nesse paradigma, os objetos do mundo físico são embarcados com essas tecnologias e transformados em dispositivos com a capacidade de comunicar-se com servidores ou outros objetos, de processar informação, pedidos e eventos, além de interagir com o ambiente por meio de sensores e atuadores. Os objetos têm a capacidade de coletar informação e comunicar-se sem a necessidade de intervenção humana. Cada objeto possui identificação, e sua posição e estado são conhecidos. Dessa forma, o paradigma da IoT representa uma visão onde é possível conectar as "coisas" do mundo real com o mundo digital por meio da internet, e permite a criação de novos serviços ou o melhoramento dos existentes, adicionando-os a uma versão expandida da internet (COETZEE; EKSTEEN, 2011; Perera et al., 2014; GUBBI et al., 2013).

As funcionalidades dos objetos da IoT podem ser consideradas como serviços de software com a capacidade de otimizar atividades humanas ou incrementar a qualidade de vida da população (Pattar et al., 2018). Atualmente muitos problemas podem ser abordados combinando serviços, servidores e dispositivos, criando combinações que permitam a união transparente entre os mundos digital e físico. IoT é uma abordagem para muitos tipos de aplicações como: cuidado da saúde, agricultura, aplicativos para celulares inteligentes, soluções para cidades, transportes, edifícios e casas inteligentes, entre outros (Pattar et al., 2018).

A IoT é geralmente caracterizada por dispositivos amplamente distribuídos com limitações de processamento, energia e capacidades de armazenamento, que envolvem preocupações relacionadas à confiabilidade, desempenho, segurança e privacidade (BOTTA et al., 2016). Algumas tecnologias que permitiram uma maior evolução da IoT são a computação na nuvem, na névoa e na borda da rede pois essas tecnologias facilitam a criação de um ecossistema heterogêneo de "coisas" e serviços (BOTTA et al., 2016; BELLAVISTA et al., 2019; Yu et al., 2018; PAUL; SARASWATHI, 2017)

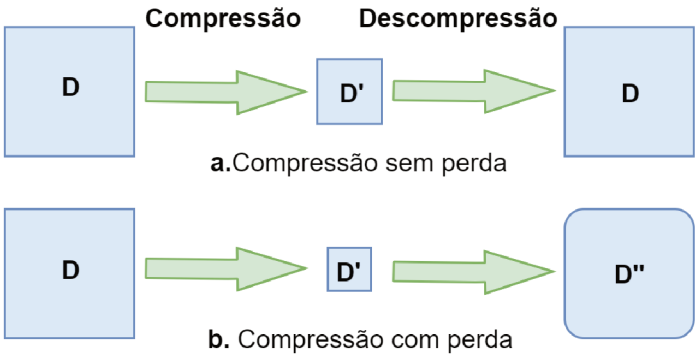
A IoT tem obtido grande relevância nos últimos anos. Em 2011, a quantidade de objetos conectados excedia a população mundial (GUBBI et al., 2013), e estima-se que, até 2021, pelo menos 27 bilhões de nodos ou objetos da IoT estarão conectados (Ud Din et al., 2019).

2.2 COMPRESSÃO DE DADOS

A Compressão de Dados (CD) é o processo no qual uma dada informação é representada de maneira comprimida.

O uso dos mecanismos de CD diminui o tamanho da informação que deve ser transmitida e armazenada, o que é importante em aplicações que requerem um menor consumo de seus recursos, tais como o sistema de armazenamento, a largura de banda e o tempo de ocupação de um sistema de comunicação, a energia usada na transmissão e transporte dos dados, entre outros (Shravana; Veena, 2017; HOSSEINI; MOHAMMAD, 2012; KOTHA; TUMMANAPALLY; UPADHYAY, 2019; DOLFUS; BRAUN, 2010). A CD conta, principalmente, com dois processos básicos, o de compressão que transforma ou representa os dados originais em uma forma reduzida, e o de descompressão, que permite recuperar os dados originais, ou recuperar uma aproximação dos mesmos. De uma forma geral, os métodos podem ser classificados como: com perda ou sem perda (Figura 1).

Figura 1 – Compressão sem perda (a) e com perda (b). D: Dados originais. D': Dados comprimidos. D": aproximação dos dados originais.



Fonte – O autor.

As técnicas de CD **sem perda** permitem realizar uma compressão de dados completamente reversível. Esses algoritmos permitem uma reconstrução exata dos dados na etapa de descompressão e são normalmente usados em aplicações e situações que requerem uma integridade absoluta da informação, como em documentos de texto, planilhas, registros de bancos de dados, entre outros. A codificação de Huffman adaptativa, codificação aritmética, e LZS são exemplos deste enfoque (KOTHA; TUMMANAPALLY; UPADHYAY, 2019; DOLFUS; BRAUN, 2010).

Os mecanismos **com perda** conseguem reconstruir uma aproximação dos dados originais nas etapas de descompressão. Essas técnicas são algumas vezes chamadas como compressão irreversível já que é impossível recuperar os dados anteriores ao processo de compressão. Eles são usados em aplicações que aceitam um grau de erro ou que, pela natureza dos dados, contam com informação e amostras redundantes que, ao serem eliminadas ou representadas de uma maneira comprimida, não afetam o funcionamento. Exemplos de aplicações para esse tipo

de técnica são a codificação de áudio e vídeo, a compressão de imagens, a coleta de dados provenientes de sensores, o SDT, a aproximação linear, MP3, JPEG são exemplos deste enfoque (HOSSEINI; MOHAMMAD, 2012). Ainda, os algoritmos com perda tendem a ter uma menor complexidade computacional e conseguir uma compressão maior que os sem perda, mas adicionam erro. Portanto, é preciso usá-los com precaução já que podem alcançar níveis de erro indesejáveis, podendo tornar irreconhecíveis os dados e afetar o processo de análise e a confiabilidade dos resultados obtidos.

A necessidade de comprimir os dados não é algo recente, a evolução da CD começou no século XlX com o código Morse. As técnicas de CD, atualmente, são fundamentais em aplicações de imagens de satélite, sistemas de informação geográfica, gráficos, Rede de Sensores Sem Fio (RSSF), IoT entre outras (UTHAYAKUMAR; VENGATTARAMAN; DHAVACHELVAN, 2018). Elas são usadas em compressão de áudio e vídeo, processamento de imagens, transferência de dados, redes de telecomunicações digitais entre outros (KOTHA; TUMMANAPALLY; UPADHYAY, 2019).

As Tecnologias da Informação (TI) estão crescendo rapidamente, o que causa um aumento na quantidade de dados gerados e das necessidades de armazenamento e transmissão de informação. Atualmente, a taxa de crescimento de dados gerados é maior que a de tecnologias de transmissão e armazenamento (UTHAYAKUMAR; VENGATTARAMAN; DHAVACHELVAN, 2018), o que é um desafio para a sustentabilidade econômica e energética do modelo atual de sociedade na que os indivíduos usam ativamente serviços e plataformas de TI. Portanto, as técnicas de CD estão se posicionando como uma possível solução para reduzir esta tendência, já que permitem um maior armazenamento de informação na mesma quantidade de espaço e a redução do tempo de uso do canal de comunicação, bem como do consumo energético.

2.2.1 Métricas de desempenho e avaliação

Atualmente existem muitas abordagens de CD que usam os mais variados mecanismos para comprimir dados. As métricas de desempenho permitem avaliar e analisar os algoritmos de compressão, além de permitir a comparação entre os métodos. As métricas tradicionais tendem a medir: **(i)** o nível de compressão obtido, considerando a redução em bytes do espaço ocupado ou de amostras necessárias para representar um sinal; e **(ii)** o nível de erro de perda de informação (nos métodos com perda) que representa o grau de diferença entre os dados originais e resultantes da compressão. Ocasionalmente alguns trabalhos avaliam também o nível de complexidade computacional ou o tempo de execução do algoritmo, já que são usados em aplicações com restrições de tempo real.

2.2.1.1 Métricas de Erro

As métricas de erro representam o erro observado ou a Taxa de Erro (TE) depois de um processo de compressão. Nestas, compara-se o sinal reconstruído depois do processo de

descompressão (S^r) e o sinal original (S^o). Algumas das equações usam um valor de amostra normalizado S^n como referência. Entre as equações mais usadas encontram-se o Erro Quadrático Médio (EQM) (Equação 2.1), o Erro Absoluto Relativo (EAR) (Equação 2.2), o Erro Quadrático Relativo (EQR) (Equação 2.3), a Diferença percentual da RMQ (DPR) (Equação 2.4) e a Diferença percentual Normalizada da RMQ (DPNR) (Equação 2.5) baseadas na Raiz da Média Quadrática (RMQ) (Equação 2.6), a Relação Sinal-Ruído (RSR) (Equação 2.7), entre outras métricas que têm uma logica semelhante (YILDIRIM; TAN; ACHARYA, 2018; UTHAYAKUMAR; VENGATTARAMAN; DHAVACHELVAN, 2018; LIU; CHEN; WANG, 2018). As métricas RSR e TE não contam com uma fórmula única Zhang e Li (2006). Elas usam uma fórmula da RSR diferente da Equação 2.7 por exemplo. Algumas propostas e esquemas de CD usam métricas não tradicionais para avaliar o funcionamento de seus algoritmos com relação ao erro apresentado. Por exemplo, Zhang et al. (2013) usa o coeficiente de Pearson que avalia o grau de correlação entre grupo de amostras; Harb, Makhoul e Abou Jaoude (2018) usa o erro de regressão e Liu e Yu (2009) usa o erro da aplicação.

$$EQM = \frac{1}{N} \cdot \sum_{i=1}^{N} (S_i^r - S_i^o)^2 \tag{2.1}$$

$$EAR(\%) = 100 \cdot \frac{\sum_{i=1}^{N} |S_i^r - S_i^o|}{\sum_{i=1}^{N} |S_i^o|} \tag{2.2}$$

$$EQR(\%) = 100 \cdot \frac{\sum_{i=1}^{N} (S_i^r - S_i^o)^2}{\sum_{i=1}^{N} (S_i^o)^2} \tag{2.3}$$

$$DPR(\%) = 100 \cdot \sqrt{\frac{\sum_{i=1}^{N} (S_i^o - S_i^r)^2}{\sum_{i=1}^{N} (S_i^o)^2}} \tag{2.4}$$

$$DPNR(\%) = 100 \cdot \sqrt{\frac{\sum_{i=1}^{N} (S_i^o - S_i^r)^2}{\sum_{i=1}^{N} (S_i^o - S^n)^2}} \tag{2.5}$$

$$RMQ = \sqrt{\frac{\sum_{i=1}^{N} (S_i^o - S_i^r)^2}{N}} \tag{2.6}$$

$$RSR(\%) = 10 \cdot \log \frac{\sum_{i=1}^{N} (S_i^o - S^n)^2}{\sum_{i=1}^{N} (S_i^o - S_i^r)^2} \tag{2.7}$$

2.2.1.2 Métricas de compressão

As métricas de compressão permitem avaliar e analisar os resultados da compressão ou a TC considerando a redução de dados obtida com o método. Nestas, compara-se a quantidade de amostras ou de bytes resultantes depois da compressão (A^c, B^c) com o número de bytes iniciais (A^o, B^o). Entre as equações comumente usadas para este propósito, encontram-se a Razão de Compressão (RC) (Equação 2.8), a Proporção de Compressão (PC) (Equação 2.9)

, o Espaço Economizado (EE) (Equação 2.10) e o Ganho da Compressão (GC) (2.11) (YILDIRIM; TAN; ACHARYA, 2018; UTHAYAKUMAR; VENGATTARAMAN; DHAVACHELVAN, 2018; LIU; CHEN; WANG, 2018).

$$RC = \frac{A^o}{A^c * 2} = \frac{B^o}{B^c * 2} \tag{2.8}$$

$$PC = \frac{A^o}{A^c} = \frac{B^o}{B^c} \tag{2.9}$$

$$EE(\%) = 100 \cdot (1 - \frac{A^c}{A^o}) = 100 \cdot (1 - \frac{B^c}{B^o}) \tag{2.10}$$

$$GC = 100 \cdot \log e(\frac{A^o}{A^c}) \tag{2.11}$$

2.2.1.3 Métricas de qualidade da compressão

As métricas de qualidade da compressão medem a efetividade dos algoritmos de CD, tendo em conta a TE e a TC. Yildirim, Tan e Acharya (2018) usam o Índice de qualidade (IQ) (Equação 2.12) que considera a PC (Equação 2.9) e a DPR (Equação 2.4), V, M e O (2015) avaliam usando o Fator de Qualidade (FQ) (Equação 2.13).

$$IQ = \frac{PC}{DPR} \tag{2.12}$$

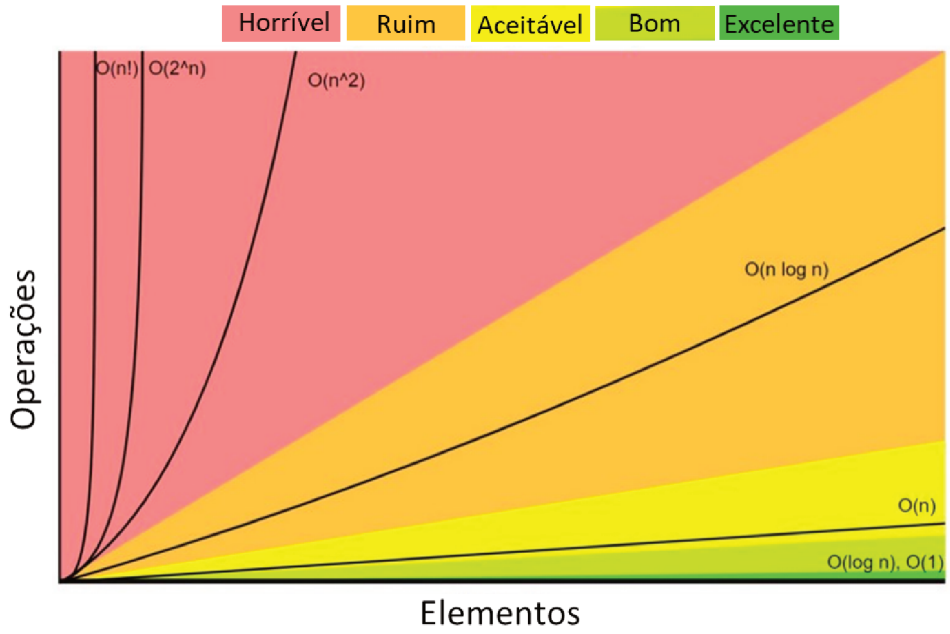
$$FQ = \frac{PC^2}{RMQ} \tag{2.13}$$

2.2.1.4 Notação Big-O

A notação ***Big-O***, representada por $O(f(n))$, é comumente usada na área da ciência da computação para comparar custos computacionais dos algoritmos. Essa notação classifica o efeito que o tamanho da entrada tem no tempo de execução (tempo de processamento ou espaço de trabalho requerido). Ela dá um limite superior a uma função entre um fator constante (2.14), onde $f(n) = O(g(n))$ indica que a função $f(n)$ é membro do conjunto $O(g(n))$. Essa notação permite descrever o tempo de execução de um algoritmo por meio da inspeção da estrutura geral do mesmo (CORMEN et al., 2009). Na Figura 2 apresentam-se as notações *Big-O* mais comuns e o efeito do número de elementos na quantidade de operações necessárias.

$$\begin{aligned} O(g(n)) = \{f(n) : &\text{Existem umas constantes positivas c e } n_o \text{ tal que} \\ &0 \leq f(n) \leq c \cdot g(n) \text{para todo } n \geq n_o\} \end{aligned} \tag{2.14}$$

Figura 2 – Operações versus número de elementos para diferentes notações *Big-O*



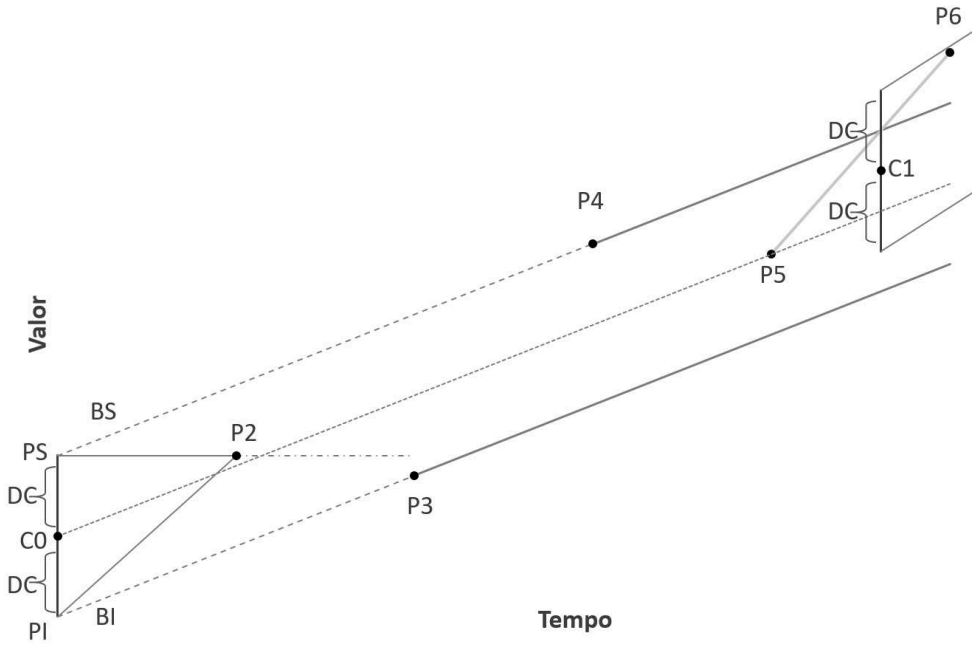
Fonte – Adaptado do (ROWELL, 2019)

2.2.2 Swinging Door Trending

O SDT é um método de CD que usa uma linha de tendência para representar certa quantidade de amostras. Esse algoritmo de compressão com perda tem como parâmetro mais importante o DC, que representa a diferença máxima que um certo ponto pode ter com relação a linha atual para ser representado na mesma (BRISTOL, 1990; NETO LUIZ AUGUSTO, 2014; BRISTOL, 1987).

A Figura 3 ilustra um exemplo do SDT usando seis amostras. O método cria uma área de cobertura com forma de paralelogramo. Essa área cresce com relação ao primeiro ponto da linha de tendência atual ou ponto Computado (c), o qual é uma dupla $\langle x,t \rangle$, sendo x o valor atual e t o tempo. A área tem uma dimensão longitudinal de duas vezes o DC tendo no centro inicial o (c). O paralelogramo tem um pivô superior (s) e outro inferior (i) com uma distância de $+DC$ e $-DC$ em relação ao c. Este conta com uma borda superior (BS) e uma inferior (BI) que começa no s E i, respectivamente. O BS só pode movimentar-se no sentido anti-horário e o BI no sentido horário. Quando uma nova amostra chega, estes limites podem movimentar-se para inseri-la na área, o que somente é possível quando o coeficiente angular da reta ou inclinação s do BS for menor ou igual ao do BI. Quando esta condição não é cumprida, um novo ponto é criado dentro da área para representar o último ponto da linha de tendência. Esse mesmo ponto é usado como o novo c, e o processo começa novamente (NETO LUIZ AUGUSTO, 2014).

Figura 3 – Exemplo gráfico do SDT



Fonte – O autor.

De acordo com a patente do SDT (BRISTOL, 1987), este conta com oito passos apresentados no Algoritmo 1:

1. Receber o primeiro ponto.
2. O pivô superior e inferior são estabelecidos.
3. Receber o ponto seguinte.
4. Calcular as inclinações atuais em relação ao pivô superior e inferior.
5. Comparar os declives atuais com os casos de declives extremos (S^s_{max}, S^i_{min}). Se S^s_{max} é maior que S^i_{min}, o algoritmo continua no sexto passo; caso contrário, regressa ao terceiro.
6. Localizar o ponto final dentro da área do paralelogramo. A inclinação entre este e o ponto atual é calculada, e a borda cruzada é ajustada para ser paralela à outra. Uma interseção entre a borda cruzada e a inclinação do ponto final e atual é calculada e definida como o novo valor de c.
7. Emitir o ponto final c.
8. O novo ponto inicial c é usado como inicio do paralelogramo, as inclinações extremas são restauradas, os pivôs são novamente definidos e o ponto atual é avaliado. Finalmente, o algoritmo volta ao terceiro passo.

Algoritmo 1 SDT

Initialization
$CD = selected\ error$ ▷ DC - *Compression Deviation*
$c = d = \langle Time, Value \rangle$ ▷ Recebe o primeiro ponto
$u = c + \langle 0, CD \rangle, \quad l = c + \langle 0, -CD \rangle$ ▷ Estabelecer pivôs
$s^u_{Max} = -\infty, \quad s^l_{Min} = \infty$
DELIVER(c) ▷ enviar primeiro ponto

function NEW_PARALLELOGRAM()
    $u = c + \langle 0, CD \rangle$ ▷ Atualizar pivôs
    $l = c + \langle 0, -CD \rangle$
    $s^u = (d_y - u_y)/(d_x - u_x)$
    $s^l = (d_y - l_y)/(d_x - l_x)$
    $s^u_{max} = s^u$
    $s^l_{min} = s^l$

function SDT($Time, Value$)
    $p = d$ ▷ Ponto anterior
    $d = \langle Time, Value \rangle$ ▷ Ponto novo
    $s^u = (d_y - u_y)/(d_x - u_x)$ ▷ Inclinação respeito pivô superior
    $s^l = (d_y - l_y)/(d_x - l_x)$ ▷ Inclinação respeito pivô inferior
    if $s^u > s^u_{max}$ **then** ▷ Comparar inclinação superior atual com S^s_{max}
        $s^u_{max} = s^u$
        if $s^u_{max} > s^l_{min}$ **then** ▷ Verificar restrição
            $s^o = (d_y - p_y)/(d_x - p_x)$ ▷ Calcular a inclinação do punto externo respeito o anterior
            $c^u = (u_y - p_y + s^o p_x - s^l_{min} u_x)/(s^o - s^l_{min})$
            $c = \langle c^u\ ,\ u_y + s^l_{min}(c^u - u_x) - CD/2 \rangle$ ▷ Ponto final
            NEW_PARALLELOGRAM()
            return c ▷ retornar ponto final
    if $s^l < s^l_{min}$ **then** ▷ Comparar inclinação inferior atual com S^i_{min}
        $s^l_{min} = s^l$
        if $s^u_{max} > s^l_{min}$ **then** ▷ Verificar restrição
            $s^o = (d_y - p_y)/(d_x - p_x)$ ▷ Calcular a inclinação do punto externo respeito o anterior
            $c^l = (l_y - p_y + s^o)p_x - s^u_{max} l_x)/(s^o - s^u_{max})$
            $c = \langle c^l\ ,\ l_y + s^u_{max}(c^l - l_x) + CD/2 \rangle$ ▷ Ponto final
            NEW_PARALLELOGRAM()
            return c ▷ Retornar ponto final
    return $\emptyset$ ▷ Não tem ponto final

Note: d é uma amostra coletada, p é a amostra anterior e c é um valor calculado pelo algoritmo. Todos os pontos são representados como uma dupla da forma $\langle Tempo, Valor \rangle$. . $S^u_{max} = S^l_{min}$ significa que a borda superior é inferior são paralelas

2.3 CONCLUSÕES DO CAPÍTULO

Este capítulo apresentou, de forma sucinta, conceitos básicos sobre o paradigma IoT. Os principais conceitos sobre compressão de dados e suas principais métricas também foram apresentados e discutidos. Finalmente, o método SDT, que serve de base para a proposta deste trabalho de mestrado, é descrito passo a passo.

3 TRABALHOS RELACIONADOS

Buscando responder a pergunta de pesquisa apresentada nesta dissertação, uma RSL foi executada. Foi usado como método, o guia para o planejamento e execução de uma RSL proposto por Kitchenham (2004).

Os trabalhos relacionados discutidos neste capítulo são divididos em dois grupos. O primeiro grupo é apresentado na Seção 3.1 e tem por objetivo apresentar o resultado da pesquisa sobre as propostas atuais de CD com perda em dispositivos da IoT para compressão em amostras de sensores. O segundo, na Seção 3.2, apresenta os trabalhos relacionados com o SDT, tendo em conta as modificações propostas e as áreas em que essa técnica é aplicada.

3.1 RSL SOBRE COMPRESSÃO DE DADOS COM PERDA NA IoT

As técnicas de CD são comumente usadas em diferentes áreas de pesquisa, como no processamento multimídia de imagem ou vídeo, ou na comunicação via satélite. Na IoT, a CD pode ser executada de maneira local dentro do próprio dispositivo, centralizada, usando um servidor ou um nó coletor, ou distribuída. Em razão do foco deste trabalho ser a redução na transmissão de dados coletados pelos sensores e do consumo energético desses dispositivos, os mecanismos distribuídos ou centralizados não são estudados, sendo o foco desta RSL apenas em técnicas de compressão de dados de sensores na IoT. Além disso, ao considerar a natureza da maioria das aplicações de IoT, apenas os métodos com perda de CD foram escolhidos na revisão da literatura por obterem uma maior taxa de compressão.

A Subseção 3.1.1 apresenta outras revisões encontradas atualmente na literatura, com um grau de relação com os objetivos desta dissertação. O protocolo usado na revisão é apresentado na Subseção 3.1.2. A Subseção 3.1.3 mostra a taxonomia criada para a classificação das propostas. Finalmente, a Subseção 3.1.4 apresenta e discute os resultados.

3.1.1 Outras Revisões da Literatura Relacionadas com este Trabalho

Esta subseção foca em apresentar outras revisões encontradas na literatura que abordam propostas de CD usadas para a compressão de amostras de sensores. Essas revisões são discutidas de forma breve.

Bose et al. (2016) foca na classificação de algoritmos de compressão com perda baseada nas características dos dados dos sensores. As propostas são categorizadas entre aquelas que trabalham no domínio do tempo ou nas que trabalham no domínio da transformação do dado. As técnicas que trabalham no domínio do tempo envolvem uma análise do comportamento dos dados ao longo do tempo, sem envolver qualquer transformação no dado. Por outro lado, as técnicas que trabalham no domínio da transformação dos dados, transformam os dados antes de tratá-los. Alguns exemplos dessas técnicas são *Discrete Fourier Transform* (DFT),

Fast Fourier transform (FFT), *Discrete Cosine Transform* (DCT) e Transformada Discreta de Wavelet (DCT).

Srisooksai et al. (2012) realiza uma revisão dos algoritmos de compressão de dados para Rede de Sensores Sem Fio (RSSF). As propostas são classificadas como locais ou distribuídas. A revisão apresenta informação de grande relevância, e explica apropriadamente o protocolo implementado. No entanto, pode-se considerar esse estudo desatualizado, pois foi realizado em 2012.

Uma revisão semelhante foi realizada em (TUAMA et al., 2018). Nela, as propostas são classificadas entre aquelas com perda ou sem perda. Uma desvantagem desse trabalho é que apenas duas propostas com perda são discutidas.

Além dessas, Uthayakumar, Vengattaraman e Dhavachelvan (2018) trata sobre as técnicas de compressão sob uma perspectiva de qualidade de dados, esquema de codificação, tipo de dados e aplicação. No entanto, somente 6 propostas sobre CD para amostras de sensores são estudadas, das quais apenas uma é com perda.

Propostas de CD para RSSF são tratadas em (KOTHA; TUMMANAPALLY; UPADHYAY, 2019) e em (DOLFUS; BRAUN, 2010). No entanto, esses trabalhos tratam apenas com técnicas de compressão sem perdas, estando fora do escopo desta dissertação.

Analisando as revisões encontradas na literatura, conclui-se que há necessidade de um levantamento atualizado da literatura sobre técnicas de compressão de dados com perdas para dispositivos IoT.

3.1.2 Protocolo da Revisão Sistemática da Literatura

Kitchenham (2004) propõe um protocolo de 10 passos, a disseminação e a tabela de tempo do projeto não estão nos interesses desta dissertação, pelo qual só são realizados os outros 8 passos.

3.1.2.1 Pergunta de pesquisa

A Pergunta de Pesquisa (PP) usada para a revisão foi: “quais são os algoritmos de compressão de dados com perda que existem para ser usados em dispositivos IoT para a compressão de dados de sensores?”. Essa PP pode ser estruturada usando o PICOC (População, Intervenção, Comparação, Resultado, Contexto) sugerido em (KITCHENHAM, 2004):

- A **população** alvo são os dados de sensores provenientes de sensores em dispositivos da IoT;
- a **intervenção** é usar algoritmos de CD com perda no dispositivo nos dados provenientes de sensores;
- o componente de **comparação** não é considerado;
- o **resultado** esperado é uma redução nos custos econômicos ou no consumo energético do dispositivo;

- o **contexto** é uma aplicação ou ambiente da IoT no qual os dispositivos estejam conectados com um servidor (local, na internet, ou provedores de serviços de computação na nuvem ou na névoa).

O PICOC é apresentado na Tabela 1

Tabela 1 – PICOC

PICOC	Resultado
População	Dados de sensores coletados
Intervenção	CD com perda para dispositivos da IoT
Resultados	Redução de custos e melhorar o uso dos recursos energéticos e de armazenamento
Contexto	Dispositivos da IoT conectados com um servidor

Fonte – O autor.

3.1.2.2 Estratégia de busca

A busca foi executada usando SCOPUS como base de dados. Essa base indexa artigos de revistas cientificas e de conferências provenientes de vários publicadores voltados para áreas de pesquisa relevantes para a RSL, como o IEEE Xplore, IEEE, ScienceDirect, ACM, Elsevier, Emerald. IOS Press, Springer, entre outros. SCOPUS foi escolhido, também, pela sua interface de consulta, que permite consultas mais sofisticadas e expressivas. O operador W/n por exemplo, permite a definição de uma quantidade máxima de palavras (representado por n) entre duas palavras-chave, o que permite aumentar a faixa de busca de documentos, ao poder incluir mais trabalhos relacionados com a área.

A Tabela 2 apresenta as palavras-chave em Português provenientes dos elementos do PICOC apresentados na Tabela 1. A Listagem 1 apresenta a consulta realizada em SCOPUS, onde usou-se o operador W/n para incluir outros trabalhos relacionados. As palavras-chave da população consistem em adjetivos da palavra ***data*** em inglês. Neste caso usou-se o operador W/n com o valor de $n = 2$ para incluir outros trabalhos relacionados. Por outro lado, a intervenção combina uma ação (comprimir – do Inglês "compression" ou "compressing" ou "compress") com o objeto (dados – do Inglês "data"). Com relação às palavras em Inglês relacionadas ao contexto da aplicação do método de compressão ("iot", "internet of thing", "ioe", "internet of everything", "scada" etc), foi usado o valor de $n = 5$ para capturar as relações de dependência entre palavras na mesma frase (BANSAL; GIMPEL; LIVESCU, 2014). O resultado final dessa consulta coletou 314 documentos relacionados ao tema da pesquisa desta dissertação.

Tabela 2 – Palavras-chave

Componentes	Palavras-chave
População	Dados coletados, dados de sensores, dados de dispositivos IoT
Intervenção	Compressão de dados, dados comprimidos, comprimir
Resultado	Redução, economia, custos, optimização,
Contexto	IoT, SCADA, indústria 4.0, sistemas ciberfísicos, RSSF, Indústria inteligente

Fonte – O autor.

Listagem 1 – Consulta usada em SCOPUS

```
TITLE-ABS-KEY
(
 (
  (
    data W/2 (sensed OR sensing OR sensor OR IoT )
  ) W/5
  (compression OR compressing OR compress)
 )AND
 (
   iot OR "internet of thing" OR ioe
   OR "internet of everything"
   OR scada OR "industry 4.0" OR cps
   OR "cyber physical system"
   OR "smart industry" OR wsn
   OR "Wireless sensor network"
  )
) AND
DOCTYPE(ar OR cp OR ip) AND LANGUAGE(english) AND
SUBJAREA(engi OR comp)
```

Fonte – O autor.

3.1.2.3 Critérios de seleção

Os artigos escolhidos precisam cumprir com todos os Critérios de Inclusão (CIs) e nenhum dos Critérios de Exclusão (CEs), sendo que três CI e dez CE foram definidos para essa RSL. Os CIs foram executados nas etapas iniciais da revisão. Eles usaram a PP e consideraram somente artigos relacionados com CD realizados em dados provenientes de sensores.

Os critérios de inclusão foram os seguintes:

Critérios de Inclusão

1. O documento propõe algoritmos ou métodos de CD.
2. O método compressão deve ser aplicado em um fluxo de dados do sensor.
3. O método de compressão deve ser aplicado em dados de sensores.

Os critérios de exclusão, elencados abaixo, foram executados tendo em conta a PP e as tabelas 1-2. Os critérios 1-2 limitam os documentos apenas aos que contam com uma proposta de algoritmo ou método com perda que foram usados em dispositivos da IoT para comprimir dados de sensores. O terceiro exclui as propostas executadas em sinais analógicos. Alguns artigos de outras áreas de pesquisa contam com palavras-chave semelhantes. Os artigos que não apresentam suficiente informação são descartados pelo quarto critério. O quinto critério é usado quando dois documentos apresentam a mesma proposta, sendo que a proposta com maior detalhe é a única considerada. O sexto é usado no caso do artigo não ser encontrado. Nesse caso, foram enviadas mensagens aos autores via Researchgate ou via e-mail, mas não houve resposta. Os critérios sétimo ao décimo foram executados na consulta feita em SCOPUS (Listagem 1). O sétimo critério exclui documentos que não têm relação com as áreas de engenharia e ciências da computação, o oitavo só permite artigos escritos em inglês e o nono e décimo limitam a leitura a artigos aceitos e publicados. Os critérios 1-2 e 7-10 foram criados para serem usados nas etapas iniciais da revisão, critérios do 3 ao 6 demandam uma revisão mais exaustiva do artigo.

Critérios de Exclusão

1. A compressão não é feita dentro do dispositivo que coleta diretamente os dados do sensor.
2. O algoritmo de compressão é sem perda.
3. A proposta é executada em sinais analógicos.
4. O documento tem informação insuficiente sobre o método.
5. Existe outra versão do artigo com mais detalhes e que apresenta a mesma intervenção.
6. Não foi possível obter o texto completo do documento usando todos os meios disponíveis, incluindo mensagens enviadas aos autores.
7. O documento não é relacionado com as áreas de engenharia e ciências da computação.
8. O documento não está escrito em inglês.
9. O não foi sujeito a uma revisão científica por pares, por exemplo: monografia, relatório técnico, normatização de padrões, editoriais, artigo convidado, entre outros.
10. O documento não é um artigo publicado em uma conferência ou jornal.

3.1.2.4 Procedimento de seleção

A seleção dos documentos pode ser resumida em três passos. O primeiro é realizado automaticamente e usa a consulta realizada em SCOPUS que filtra artigos que não cumprem certas condições básicas ou que não têm relação com o objetivo da PP. O segundo e terceiro passos são executados manualmente. O segundo consiste na aplicação dos CI e CE no título, resumo, palavras-chave de todos os documentos que resultaram do uso da consulta em SCOPUS. O terceiro passo busca executar os mesmos critérios, porém, no texto completo. O segundo e terceiro passos introduzem uma possibilidade de erro humano. A possibilidade de inclusão ou exclusão incorreta de documentos tem que ser considerada. A exclusão incorreta precisa de uma segunda execução dos passos 2 e 3. Os documentos finalmente selecionados são formados

pela união entre as duas execuções. A inclusão incorreta é controlada na extração dos dados explicada na Subsubseção 3.1.2.6.

3.1.2.5 Avaliação da qualidade

Esta RSL não exclui documentos em base à sua qualidade ou baixo rigor científico. Dessa forma, há três situações que podem afetá-la: uma exposição pobre da pesquisa; informação incompleta; conclusões sem suporte teórico ou empírico.

3.1.2.6 Extração de dados

A extração de dados consiste em escrever um resumo de cada artigo lido e responder um grupo de perguntas. As questões para esta RSL foram as seguintes:

1. Qual é a entrada do algoritmo?
2. Qual é a saída do algoritmo?
3. Há parâmetros de configuração? Em caso positivo, quais?
4. A proposta fornece alguma característica ou mecanismo adicional? Se fornece, quais?
5. Quais são as limitações do algoritmo?
6. Qual a classificação do algoritmo baseado na taxonomia proposta na Subseção 3.1.3?

Caso um documento não tenha informação suficiente para responder alguma das perguntas apresentadas na extração de dados, a pergunta não respondida será marcada como “não definido”.

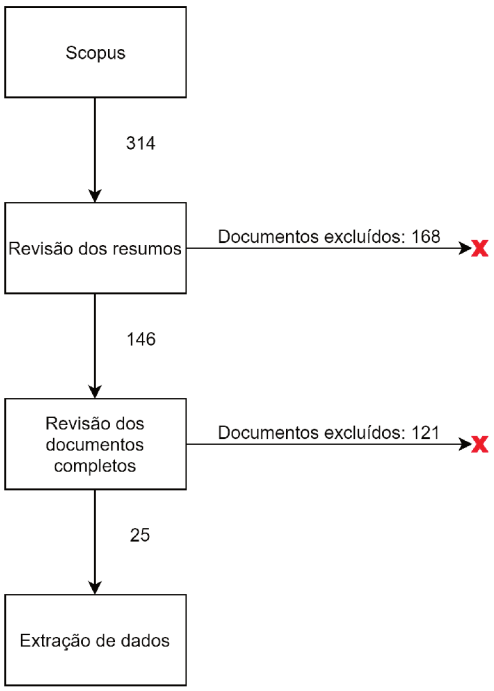
3.1.2.7 Síntese dos dados

A PP desta RSL requer uma síntese descritiva dos dados (KITCHENHAM, 2004). Inicialmente, é feita uma análise dos resultados conseguidos na extração de dados (Subsubseção 3.1.2.6). Um dos resultados mais importantes foi identificar os algoritmos de compressão que foram propostos para serem usados em dados de sensores, e realizar uma avaliação de todos os documentos e propostas, além de caracterizá-los com relação a seus métodos de CD para dados de sensores. As propostas também foram classificadas para entender os diferentes tipos de métodos de CD encontrados na literatura. Eles foram classificados com base na taxonomia proposta na Subseção 3.1.3.

3.1.2.8 Execução

A execução desde protocolo de RSL foi realizada em duas partes. Na primeira parte, o protocolo foi executado, resultando em 21 artigos selecionados. Na segunda parte, usou-se o protocolo novamente, adicionando 4 documentos, dando, como resultado, 25 documentos.

Figura 4 – Visão geral do processo de seleção dos documentos.



Fonte – O autor.

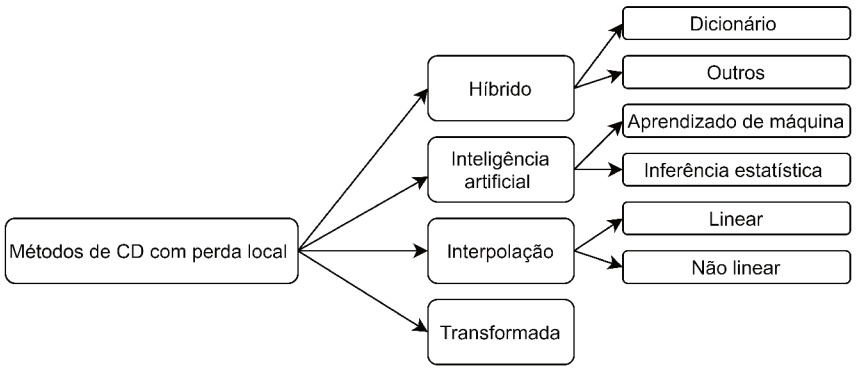
A primeira execução foi realizada entre fevereiro e abril, e a segunda entre abril e agosto. A Figura 4 apresenta uma visão global do resultado da RSL. A Tabela 3 apresenta o número de artigos descartados por não satisfazer um CI ou por satisfazer um CE, dos quais não são representados os descartados entre o critério 7 e 10.

Tabela 3 – Documentos excluídos por etapa e por critério (não respeita um CI ou cumpre um CE)

Critério	**Revisão do resumo**	**Revisão documento completo**
CI1	77	20
CI2	29	9
CI3	19	16
CE1	14	23
CE 2	29	8
CE 3	0	19
CE 4	0	6
CE 5	0	12
CE 6	0	8
Excluídos	**168**	**121**

Fonte – O autor.

Figura 5 – Taxonomia dos algoritmos de CD com perda em dispositivos IoT



Fonte – O autor.

3.1.3 Taxonomia

As propostas de CD usam diferentes tipos de mecanismos para realizar o processo de compressão. As etapas básicas nos algoritmos de CD são o processo de compressão e descompressão. Bose et al. (2016), por exemplo, classificou os algoritmos no domínio do tempo ou da transformação. Srisooksai et al. (2012), Tuama et al. (2018) os separa entre propostas distribuídas ou locais. A taxonomia proposta foca na RSL realizada. Neste caso, o objetivo é exclusivamente as propostas locais, que são algoritmos com perda e os mecanismos que usados para a compressão de dados coletados de sensores. Na Figura 5 é apresentada a taxonomia proposta.

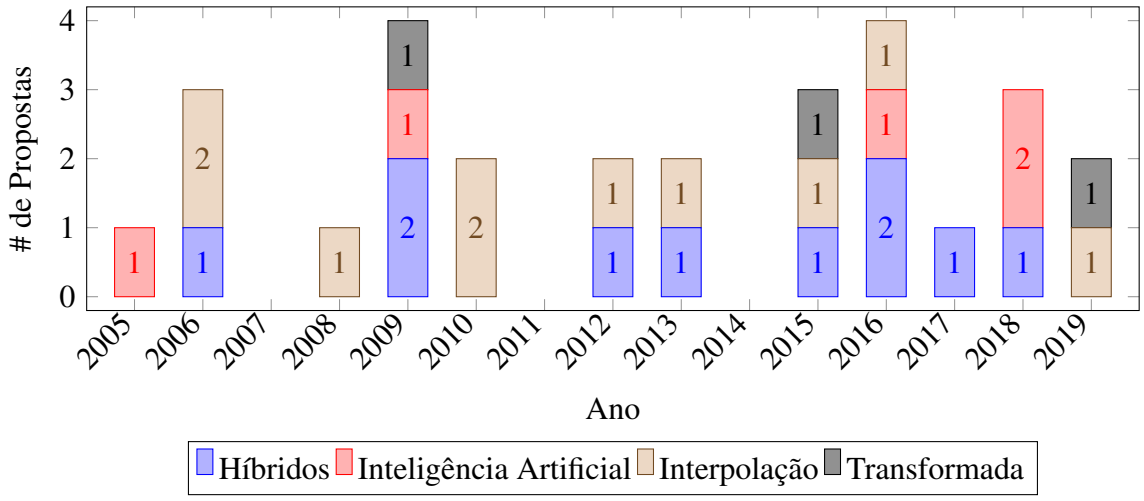
Os métodos híbridos usam um algoritmo de compressão de dados com perda e outro sem perda trabalhando juntos. Eles são separados entre as propostas que usam dicionário (Dic) que focam em métodos de dicionário sem perda, que usam um mecanismo ou algoritmo com perda para melhorar sua taxa de compressão e outros (Otr). Os IA propõem métodos de Aprendizado de Máquina (AM) ou Inferência estatística (IE) para melhorar ou realizar o processo de compressão ou descompressão. Os métodos de Interpolação usam mecanismos para criar um novo dado usando um conjunto de dados conhecidos. O sinal do sensor é representado como o valor de uma função. Os métodos de Interpolação são separados entre duas subcategorias: Interpolação linear (IL) e Interpolação Não Linear (INL). As abordagens de transformada (Tran) comprimem os dados mudando o domínio da sinal como por exemplo, mudando o sinal do domínio do tempo ao domínio da frequência.

3.1.4 Resultados

Os dados extraídos dos documentos selecionados são sintetizados nesta subseção. A Figura 6 apresenta as propostas em base ao seu ano de publicação e classificação na taxonomia proposta na Figura 5. Na revisão, 25 documentos foram selecionados. Porém, alguns dos documentos contam com mais de uma proposta, contabilizando, assim, 28 abordagens. O do-

cumento mais antigo selecionado foi publicado no 2005. Metade dos artigos foram realizados entre o 2015 a 2019. O processo de classificação mostrou que 10 mecanismos de CD são híbridos, 5 usam IA, 10 são considerados como de interpolação e 3 apresentam um processo de mudança do domínio (transformada). A Tabela 4 divide os 25 documentos pelo edição e a quantidade deles que foram publicados em revista científica. As Subsubseção 3.1.4.1 à Subsubseção 3.1.4.4 mostram um resumo das propostas. A Tabela 5 mostra a proposta e a classificação da mesma.

Figura 6 – Mecanismos propostos por ano de publicação



Fonte – O autor.

Tabela 4 – Número de documentos classificados conforme o editor

Editor	**Documentos**	**Periódico**
IEEE	13	2
SAGE	2	2
Springer	2	1
ACM	2	1
MDPI	1	1
Inderscience	1	1
ISA	1	1
Elsevier	1	1
ASCE	1	1
Trans Tech Publications	1	0

Fonte – O autor.

Tabela 5 – Informação geral das propostas selecionadas

Documento	**técnica proposta**	**Classificação**
Azar et al. (2019b)	Compressor com perda com limite de erro rápido	INL
Chen et al. (2019)	HASDC	Tran
Liu, Chen e Wang (2018)	Stacked S-RBM-AE	AM
Ramijak, Pal e Kant (2018)	AVSC	Dic
Harb, Makhoul e Abou Jaoude (2018)	Redução de dados de dois níveis usando coeficientes de Pearsone Ekmeans ou TopK	IE
Giorgi (2017)	Filtro preditivo de zero latência baseado no DPCM	Otr
Li e Liang (2016)	Codificação preditiva generalizada bidimensional	Dic
Wang et al. (2016)	Esquema de aproximação linear com estrutura de árvore	IL
Abu Alsheikh et al. (2016)	Rede Neural de Compressão usando limite de erro	AM
Kolo et al. (2016)	ADCS	Dic
Sharma (2015)	LTC	IL
Alsalaet e Ali (2015)	MDCT-EHCC	Otr
V, M e O (2015)	FuzzyCat	Tran
Mohamed, Wu e Moniri (2013)	CD adaptativa baseada nas técnicas de predição	Otr
Zhang et al. (2013)	AR-MWCEB	IL
Zhao et al. (2012)	DSDT	IL
Chen et al. (2012)	Agregação e Compressão de Dados com limite de erro dinâmico	Dic
Chen et al. (2010)	Algoritmo no nível do nó não baseado nos limites de erro	IL
Kasirajan, Larsen e Jagannathan (2010)	NADPCMC	INL
Liu e Yu (2009)	Transformada wavelet com codificação aritmética	Otr
Liu e Yu (2009)	Análise de pacotes Wavelet com informação de Fisher	Tran
Liu e Yu (2009)	Compressão de aumento monótono	IE
Capo-Chichi, Guyennet e Friedt (2009)	K-RLE	Otr
Pham, Le e Choo (2008)	PLAMLiS	IL
Li, Loke e Ramakrishna (2006)	Aproximação linear	IL
Li, Loke e Ramakrishna (2006)	Aproximação usando a Curva de Bézier	INL
Zhang e Li (2006)	Transformada Wavelet de esquema de elevação	Otr
Li e Li (2005)	PBSA	IE

Fonte – O autor.

3.1.4.1 Híbridos

Os mecanismos de CD Híbridos são algoritmos com perda que aumentam sua capacidade de compressão com ajuda de propostas sem perda, estas propostas conseguem melhorar a TC ao aumentar o nível de compressão sem um aumento na TE, elas tendem a considerar o estado do sistema ou as necessidades para melhorar os resultados do processo de compressão. Um ponto fraco a se considerar neste enfoque é a tendência a demandar mais passos e processos que os outros tipos, aumentando assim o tempo de execução ou a complexidade computacional para alcançar um resultado maior em termos da TC. A Tabela 6 mostra uma revisão das características das propostas considerando a Extração de dados apresentada na Subsubseção 3.1.2.6

3.1.4.1.1 Dicionário

Uma abordagem comum para CD são os algoritmos de codificação que trabalham com dicionários para representar os dados. Eles funcionam como algoritmos sem perda e usam

Tabela 6 – Informação geral dos mecanismos de compressão híbridos

Documento	Entrada	Saída	Parâmetros	Características adicionais
Ramijak, Pal e Kant (2018)	Séries de tempo	Código	B: # de bits para Discretização τ: Distância máxima euclidiana ι: Longitude de sequência predefinida	Limite de erro usando τ SQS
Chen et al. (2012)	Amostra	Código	τ_{max}:Erro Máximo	Limite de erro dinâmico usando correlação temporal, espacial ou dos dados
Kolo et al. (2016)	Vetor	ID da tabela+ código	SME Tabela estática	Tabela estática de huffman Compressão adaptativa
Mohamed, Wu e Moniri (2013)	Amostra	Código	ER: Faixa de Erro	Limite de erro adaptativo Detecção de eventos Considera qualidade dos dados Considera a energia economizada
Giorgi (2017)	Amostra Uints	Código	δ: Limite de tolerância	
Alsalaet e Ali (2015)	Vetor	Código	N: Tamanho do vetor	Esquema de codificação embarcado
Capo-Chichi, Guyennet e Friedt (2009)	Amostra	NV V: Valor N: contagem	k: Limite de tolerância	
Li e Liang (2016)	Amostra	Código	R: Radio de compressão ε: limite de erro de resíduos	
Zhang e Li (2006)	Vetor	código	Tamanho de quantização	
Liu e Yu (2009)	Vetor	código	L: Nível de decomposição N: # de amostras	

Fonte – O autor.

diferentes mecanismos para criar lista de códigos e representar os dados usando uma menor quantidade de bits. Adicionar um mecanismo ou algoritmo com perda em etapas preliminares gera um esquema com perda que consegue melhores resultados de compressão e, em alguns casos, reduz o tamanho do dicionário ao diminuir a faixa de valores que pode receber o algoritmo. A necessidade de armazenar a tabela de códigos do dicionário tem que ser analisada, já que pode ser necessário considerar as capacidades de armazenamento do dispositivo.

Ramijak, Pal e Kant (2018) apresentam o *Approximate Vector Stream Compression* (AVSC) que tem uma versão correlacionada e outra não correlacionada. Essa abordagem usa *Summarizing event seQuenceS* (SQS) antes da execução do dicionário, e assegura-se de que a sequência de eventos possa ser derivada ou inferida desde os dados comprimidos. A parametrização do algoritmo requer atenção, já que são fundamentais para o funcionamento adequado do esquema.

Chen et al. (2012) propõem o *Dynamic Bounded-Error Data Compression and Aggregation* (D-BEDCA), o qual é uma melhoria do *Bounded-Error Data Compression and Aggregation* (BEDCA) proposto em (Li et al., 2010). Ele conta com duas fases. Na primeira fase, o coeficiente de variação para cada sensor é calculado e usado para a seleção do limite de erro. Na segunda, a compressão é realizada considerando a correlação temporal (amostras passadas), espacial (amostras de vizinhos) ou de codificação (com as codificações passadas) e a tolerância definida na primeira fase. O dicionário proposto em (Marcelloni; Vecchio, 2008) é usado. No dispositivo é realizada somente a compressão baseada na correlação temporal. O ponto fraco da proposta é o uso de um mecanismo de erro que não considera a tendência dos dados, só compara o dado atual com relação ao último enviado.

O *Adaptive Data Compression Scheme* (ADCS) é proposto em (KOLO et al., 2016).

Dependendo da aplicação e necessidades, o esquema muda entre uma proposta sem perda e outra com perda. O componente principal da proposta é o *Adaptive Entropy Encoder* (AEE), um codificador sem perda que usa duas tabelas estáticas de Huffman que foram desenhadas para lidar com diferentes níveis de correlação nos dados de entrada. Essas tabelas são criadas fora do dispositivo para reduzir a complexidade computacional da proposta. Os métodos de força bruta e região de decisão são usados para codificar eficientemente os blocos de dados enquanto usam as tabelas adaptativamente. Quando for necessário realizar uma compressão com perda, o SME é usado como limite do erro para filtrar o sinal. O objetivo é somente filtrar o erro do sinal. A proposta incrementa a complexidade computacional no processo de decisão da tabela a usar. Adicionalmente, surge a necessidade de adicionar um bit ou vários para identificar a tabela usada. Outro ponto fraco é a necessidade de criar tabelas estáticas para cada aplicação, o que limita a capacidade de adaptação do esquema.

Li e Liang (2016) propõe usar um filtro com tolerância de erro antes do *Generalized Predictive Coding* (GPC) e um algoritmo sem perda chamado *Sequential Lossless Entropy Compression* (S-LEC). Ambos formam uma solução de compressão unificada. O dicionário é realizado baseado em grupos de resíduos. O limite de erro trabalha com base ao resíduo gerado por cada amostra referentemente à última enviada, quando ele é menor que o limite é considerado igual a zero. O resultado da compressão é limitada.

3.1.4.1.2 Outras propostas

Mohamed, Wu e Moniri (2013) propõem uma estrutura que decide entre usar um método de compressão com perda ou um sem perda dependendo dos recursos energéticos disponíveis e a relevância dos dados. Um processo de decisão de Markov é usado para otimizar o nível de qualidade dos dados e a economia energética. Adicionalmente, um detector de eventos avalia o nível de importância dos dados. Dados importantes são enviados sem perdas, enquanto os outros dados são fortemente comprimidos. Uma codificação de entropia sem prefixo é usada na compressão. A proposta usa um raio de erro para definir quando o erro é aceitável, funcionando como um limite de erro que decide quando o dado precisa ser codificado e enviado.

Giorgi (2017) apresenta um algoritmo de compressão de zero latência de dois blocos. No primeiro bloco é executada uma técnica com perda, que é um filtro preditivo de zero latência baseado no *Differential Pulse Code Modulation* (DPCM). Este usa um limite de tolerância para decidir quando é necessário transmitir uma amostra para o seguinte bloco. No segundo bloco, usa-se um componente sem perda chamado *Modified Exponential Golomb Code* (MEGC), que é uma técnica de codificação de prefixo de tamanho fixo.

Alsalaet e Ali (2015) um esquema de compressão com dois componentes. O primeiro é o *Modified Discrete Cosine Transform* (MDCT), que é um mecanismo com perda. Este representa um vetor de amostras com tamanho definido (N) em coeficientes. No segundo componente é usado o *Embedded Harmonic Components Coding* (EHCC) que permite melhorar a TC, além

de prover codificação embarcada, que permite transmitir os coeficientes progressivamente de acordo a sua relevância.

EHCC foi proposto em (ALSALAET; ALI, 2015). O objetivo com esta proposta é aproveitar a redundância harmônica do sinal. O MDCT requer $O(N^2)$ operações, e o EHCC tem quatro passos: o primeiro de complexidade computacional $O(N^2)$, o segundo e quarto com complexidade de $O(1)$, e o terceiro de $O(N)$. Um valor maior de N gera melhores resultados de TC, porém aumenta o tempo de execução o que tem que ser considerado dependendo da aplicação e o dispositivo a usar.

Uma modificação do codificador sem perda *Run Length Encoding* (RLE) é proposto em (CAPO-CHICHI; GUYENNET; FRIEDT, 2009), chamada *K-precision Run Length Encoding* (K-RLE). A precisão K adiciona um sistema de limite de erro aceitável com tolerância K. A proposta é leve e funciona adequadamente em dispositivos com baixa capacidades computacionais, porém a TC é muito limitada e depende altamente da estabilidade do sinal.

Zhang e Li (2006) apresenta uma técnica de CD para sensores de vibração baseada no *Lifting Scheme Wavelet Transform* (LSWT), que consiste em realizar a descorrelação, a quantização e posteriormente a codificação dos dados. A descompressão representa um conjunto de valores em um set menor. A proposta foca exclusivamente em sensores de vibração.

A transformada wavelet com codificação aritmética é proposta em (LIU; YU, 2009). O objetivo é calcular a Diferença Horária de Chegada (DHC) em Sistemas de Transporte Inteligente (STI), sendo esta analisada como uma RSSF com comunicação constante entre sensores. Esse trabalho está voltado para os resultados da aplicação e não para a qualidade dos dados. A proposta tem uma alta TE, porém, consegue-se calcular a DHC apropriadamente.

3.1.4.2 Inteligência artificial (IA)

A IA permite abordar os desafios de diferentes áreas das ciências da computação e engenharia. No caso da CD, essa técnica pode ser usada como mecanismo de compressão ou como uma ferramenta para a melhoria das métricas TE e TC. A Tabela 7 contém um resumo das características básicas das abordagens de IA.

3.1.4.2.1 Aprendizado de Máquina (AM)

O AM é uma ferramenta importante que pode ser usada para direcionar muitos problemas e desafios. No caso de CD, essa técnica permite criar novos métodos, ou melhorar os atuais.

Li e Li (2005) propõem um novo método de compressão usando quatro *standard Restricted Boltzmann Machine* (RBM) combinados com um método não linear de aprendizado profundo chamado *Stacked RBM Autoencoder* (S-RBM-AE). Esse método comprime usando as características matemáticas dos dados coletados. Adicionalmente, um método de optimização energética é usado para reduzir o consumo energético da proposta e assim fazer ela viável

Tabela 7 – Informação geral dos mecanismos baseados em IA

Documento	Entrada	Saída	Parâmetros	Características adicionais
Liu, Chen e Wang (2018)	Vetor	Vetor comprimido	S: Dados de treinamento I: # de iterações do treinamento A: taxa de aprendizado	Optimização energética Processo de pre-treino
Abu Alsheikh et al. (2016)	Vetor	Vetor comprimido	Dados de treinamento Tolerância de erro	Detector básico de outliers Aprendizado off-line Dicionário de compressão
Li e Li (2005)	Amostra Serie de tempo	Amostra Grupo+intervalo	δ:Toleranca de erro T: Período de amostragem	Frequência de amostragem
Harb, Makhoul e Abou Jaoude (2018)	Vetor	Vetor representativo	Tamanho do periodo t_p: Limite de pearson	Eliminação de dados redundantes gerados por nós vizinhos
Liu e Yu (2009)	Vetor	Conjunto de componentes	Taxa de Compressão (TC)	

Fonte – O autor.

para os dispositivos. A proposta é recomendada para dispositivos sem muitas limitações em seus recursos computacionais, além de ser recomendada exclusivamente para aplicações com uma baixa tolerância a erros.

Abu Alsheikh et al. (2016) aborda o uso de *Compressing Neural Networks* (CNNs) com um limite de erro. O algoritmo foi criado para ter uma baixa complexidade computacional no processo de compressão e descompressão. O processo de compressão usa operações lineais e sigmoides. O método usa compressão temporal ou espacial usando a correlação nestes domínios. Ele comprime usando um dicionário de descompressão aprendido e necessário para recuperar os dados. O *neural AutoEncoders* (AEs) é usado nesta proposta com três camadas, a complexidade computacional do algoritmo é $O(LXK)$, onde L é o tamanho dos dados de entrada e K é o tamanho dos dados comprimidos.

3.1.4.2.2 Inferência estatística (IE)

A Inferência estatística pode deduzir o comportamento de determinada população com relação à informação fornecida por certas amostras, usando análise dos dados pode-se inferir as propriedades atuais do sistema.

Li e Li (2005) combinam o controle da frequência de amostragem e um algoritmo de compressão. A técnica *Precision Based Sampling and Transmit Algorithm* (PBSA) ajusta a frequência de amostragem dos dados coletados dinamicamente usando modelos de regressão linear para predizer o intervalo de tempo a ser usado. Quando a frequência não pode ser controlada é usado o mecanismo de compressão proposto chamado *Data Compression Algorithm* (DCA). Ele divide as amostras de um intervalo de tempo em grupos que respeitam um critério de erro. A proposta foca na economia energética e em reduzir a frequência de amostragem mais do que na CD. Além de DCA apresentar resultados pobres em termos de qualidade de dados, o TC é baixo quando o critério de erro não é grande.

Uma proposta de redução de dados de dois níveis é apresentada em (Harb; Makhoul; Abou Jaoude, 2018). O primeiro nível comprime no dispositivo. A compressão é baseada nos

coeficientes de Pearson que representam o grau de correlação entre dois conjuntos de dados. Um critério de Pearson é implementado como critério de compressão (limite de erro). Um conjunto de amostras é separado em subconjuntos que são analisados pelo algoritmo. Um conjunto representativo é criado para representar os dados originais, O segundo nível é usado no nó de coleta ou servidor, o qual busca-se eliminar os dados redundantes criados por nós vizinhos usando algoritmos de agrupamento (EKmeans or TopK). A seleção do tamanho do conjunto ou vetor e do critério de Pearson é crítico, já que o tempo de execução e a qualidade dos dados depende muito destes.

Liu e Yu (2009) apresenta o método de compressão de aumento monótono para calcular o DHC em STI. O objetivo do método é a estimativa de parâmetros por cima da capacidade de reconstruir os dados originais. A proposta foca exclusivamente nesta aplicação. Dados comprimidos com este método podem ficar irreconhecíveis, porém atingem resultados adequados para calcular o DHC.

3.1.4.3 Interpolação

A criação de novos pontos usando um conjunto de valores discretos é definido como interpolação. Os sinais de sensores podem ser vistos como os valores de uma função. Os valores dentro de uma linha de tempo podem ser usados como variáveis independentes em um sinal no domínio do tempo. Quando um sinal é interpolado, os valores da função são estimados para um valor intermediário.

A interpolação é uma ferramenta útil para a CD. Os sinais dos sensores podem ser representados com valores representativos ou aproximados que permitem estimar os dados originais com uma menor quantidade de amostras. A Tabela 8 apresenta a informação básica das propostas que usam mecanismos de interpolação.

3.1.4.3.1 Interpolação linear (IL)

Os métodos de interpolação mais simples são os que representam um sinal usando uma função linear. Esses métodos não são muito precisos, porém podem ser combinados com outras técnicas para controlar a TE.

Zhao et al. (2012) apresentam o *Distributed Swinging Door Trending* (DSDT) que é uma modificação do SDT (BRISTOL, 1990). Eles usam uma linha de tendência para representar uma quantidade de amostras considerando um critério de limite de erro. Essa proposta permite comprimir os dados de um sensor no nó ou usando um centro de compressão. A complexidade computacional é $O(1)$. A correta seleção do critério de erro afeta consideravelmente os resultados da compressão, o sistema considera a tendência dos dados.

O *top-down Piecewise Linear Approximation with Minimum number of Line Segments* (PLAMLiS) é proposto em (Pham; Le; Choo, 2008). Ele usa um sistema de controle de erro focado na TE.

Tabela 8 – Informação geral das propostas de Interpolação

Documento	Entrada	Saída	Parâmetros	Características adicionais
Zhao et al. (2012)	Serie de tempo	Valores representativos	ε: Limite de erro	Compressão distribuída
Pham, Le e Choo (2008)	serie de tempo	Serie de tempo representativa	Limite de erro	Redução da complexidade computacional
Wang et al. (2016)	Vetor	Vetor representativo	W: Tamanho da janela D: Distorção objetivo	Limite de erro baseado na distorção Detector de out-liner Estimação de erro Melhor partição por partes
Li, Loke e Ramakrishna (2006)	Serie de tempo	Ponto no tempo representativo	Limite de erro	
Sharma (2015)	Serie de tempo	Punto no tempo representativo	E: Limite de erro	
Chen et al. (2010)	Conjunto de serie de tempo	Serie de tempo representativa	W: Tamanho da janela η : TC	Operador de borda temporal
Zhang et al. (2013)	Vetor 2-D	Dados não tratados Coeficientes de regresao	EPS: Limite de erro	Limite de erro da regressão CD multivariável
Azar et al. (2019b)	Vetor flutuante 2-D	Vetor de bytes	E: Limite de erro P: Período	Compressão de M sensores Encaixe de curva linear
Li, Loke e Ramakrishna (2006)	Serie de tempo	Coeficientes da curva: Três (Curva completa) Quatro (Curva incompleta)	Limite de erro N: Tamanho da janela	
Kasirajan, Larsen e Jagannathan (2010)	Serie de tempo	Valor quantizado	Limite de erro	Limite de erro de reconstrução

Fonte – O autor.

PLAMLiS analisa um conjunto de amostras que são separadas em subconjuntos. Cada um deles tem uma representação linear que cumpre o critério de seleção definido. A complexidade computacional é $O(nlogn)$.

Um esquema de aproximação linear com estrutura da árvore espaço-temporal é proposto em (Wang et al., 2016). Esse trabalho conta com duas contribuições importantes, um procedimento para explorar a divisão por partes mais adequada ao conjunto de amostras e um esquema adaptável para sensores heterogêneos, várias taxas de amostragem e dados *outlier*. A proposta usa poda ótima estruturada em árvore e regressão linear para extrair a inclinação e a intercepção de uma linha possível que pode ser usada para aproximar adequadamente o conjunto de dados. No pior caso, o algoritmo tem uma complexidade computacional de $O(Wlog2W)$.

As propostas apresentadas em (ZHAO et al., 2012; Wang et al., 2016) têm maiores complexidades computacionais que outras propostas de IL, porém conseguem melhores resultados de TE. Nelas, a TC depende da quantidade de dados usados na execução. Usar mais dados em cada rodada, pode melhorar os resultados da compressão, porém aumentam o tempo de execução.

Li, Loke e Ramakrishna (2006) usa aproximação linear para representar os dados coletados. Ele usa um sistema de limite de erro simples. Os dados são representados com um segmento de linha com dois pontos. A proposta tem uma complexidade computacional de $O(1)$, porém não considera a tendência do sinal, somente a diferença com o ultimo valor transmitido ou liberado.

Sharma (2015) apresenta o *Lightweight temporal compression* (LTC) que é baseado na aproximação linear por partes. Essa técnica usa um limite superior e outro inferior de tamanho. Para avaliar quando é necessário enviar o ponto final de uma linha de tendência, a proposta apresenta um sistema de controle de erro melhor que (Li; Loke; Ramakrishna, 2006) já que

considera a tendência dos valores.

Um algoritmo não baseado no limite de erro é proposto em (CHEN et al., 2010). A proposta não requer conhecimento prévio nem a definição de um limite. A ideia básica de operadores de borda são usados para desenhar o *Extension Temporal Edge-operator* (ETEO). A convolução entre o ETEO e a série de tempo é usada para gerar uma nova sequência. O algoritmo foca em alcançar um valor de TC, o que gera uma TE não controlada.

Focando exclusivamente na compressão e deixando de lado a qualidade dos dados, um esquema chamado *self-Adaptive Regression-based Multivariate Data Wavelet Compression Scheme with Error Bound* (AR-MWCEB) é proposto em (ZHANG et al., 2013). Essa proposta é projetada para ser usada em dispositivos que coletam dados de diferentes sensores. Ela é baseada em três premissas: a correlação multivariável existe; a correlação espacial não existe; e o erro é limitado. o AR determina o número de dados envolvidos no cálculo da regressão. O algoritmo decide entre enviar coeficientes de regressão o dados não tratados considerando o erro e as necessidades da aplicação.

3.1.4.3.2 Interpolação não linear

Azar et al. (2019b) apresentam um compressor com perda com limite de erro rápido que usa uma modificação ligeira do *Squeeze* (SZ). Ele foca em dispositivos que coletam M dados de N fontes (sensores, coordenadas de um giroscópio, entre outros). Inicialmente, o vetor de 2-D é transformado em um de 1-D. A técnica SZ usa um modelo de ajuste da curva (DI; CAPPELLO, 2016) que analisa cada ponto no vetor de 1-D para verificar se ele pode ser predito usando um critério de erro definido pelo usuário. Esse modelo consegue o melhor ajuste considerando três modelos de predição; o *Preceding Neighbor Fitting* (PNF); o *Linear-Curve Fitting* (LCF); e o *Quadratic-Curve Fitting* (QCF). Os dados comprimidos são reconstruídos em um nó de borda, este processo descomprime dados usados em *Feed-Forward Neural Network* (FFNN). O objetivo é analisar o efeito da compressão com perda no desempenho de modelos de aprendizado de máquina e aprendizado profundo. A proposta apresenta bons resultados no caso estudado, dois de seus processos têm complexidade computacional de $O(M)$.

Uma aproximação de curva de Bezier cúbica é proposta em (Li; Loke; Ramakrishna, 2006). Bezier usa três coeficientes para descrever um sinal. O algoritmo considera um limite de erro, quando uma quantidade N de dados não pode ser representada respeitando este erro. Os dados são cortados e a primeira curva é considerada incompleta e necessita quatro coeficientes para ser representada (três coeficientes e o ponto final). Esta proposta pode ter uma alta TC, porém depende de um valor alto do erro aceitável o que gera uma alta TE. O tamanho da janela N influencia na compressão, porém um valor de N muito elevado aumenta o tempo de execução.

Kasirajan, Larsen e Jagannathan (2010) propõem o *Non-linear Adaptive Pulse Coded Modulation-Based Compression* (NADPCMC). A ideia é representar os dados com uma relação não linear e usar técnicas desde a teoria de estimação adaptativa para obter uma estimativa precisa. A proposta tem dois passos: um processo de estimação no qual é realizada uma estimação

Tabela 9 – Informação geral das propostas de transformada

Document	Entrada	Saída	Parametros	Características adicionais
V, M e O (2015)	Vetor	Vetor Fuzzy	W: Quantidade de amostras N: # de funções de associação	Esquema de predição entre sensores
Chen et al. (2019)	Vetor	Vetor comprimido	N: Longitude δ: Limite	Passos de compressão adicionais
Liu e Yu (2009)	Vetor	Coeficientes	L: Nível de decomposição N: # de amostras	flexibilidade melhorada

Fonte – O autor.

adaptativa não linear; e um processo de quantização que quantifica a diferença entre o estado atual e o estado estimado.

3.1.4.4 Transformada

As transformadas são processos em que há a conversão de dados de um domínio a outro. Este novo domínio pode representar o sinal com menos quantidade de amostras ou informação conseguindo um efeito de compressão, as transformadas tendem a representar os dados dentro de um pequeno número de coeficientes.

V, M e O (2015) apresentam o *Fuzzy Compression Adaptative Transform* (FuzzyCat) que é baseado no *Fuzzy Transform Compression* (FTC). Essa técnica adapta os parâmetros de transformada para inferir a curvatura do sinal desde derivadas de tempo e incrementar a resolução da transformada de Fuzzy sempre que o sinal exibir alta curvatura. A detecção de curvatura é realizada monitorando a segunda derivada do sinal.

Chen et al. (2019) apresenta o *Hierarchical Adaptive Spatio-Temporal Data Compression* (HASDC) que usa um algoritmo de compressão de limite adaptativo. O *Discrete Cosine Transform* (DCT) é usado para a compressão temporal no dispositivo, e a *Discrete Wavelet transform* (DWT) para a compressão espacial em um centro de compressão, servidor ou nó externo.

Liu e Yu (2009) usa uma análise de pacotes Wavelet com informação de Fisher voltadas para RSSF com uma comunicação constante entre sensores, especificamente em STI para calcular o DHC. A presente proposta apresenta melhor análise que as outras duas apresentadas neste artigo, porém possui o defeito de focar exclusivamente nos resultados da aplicação, tornando até mesmo possível que os dados fiquem irreconhecíveis.

3.2 SWINGING DOOR TRENDING: TRABALHOS RELACIONADOS

Neto Luiz Augusto (2014) apresentam o *Adaptive Swinging Door Trending* (ASDT). Essa proposta conta com as mesmas características básicas que o SDT, além de realizar uma análise de tendência em tempo real como mecanismo para incorporar variações nos parâmetros principais. Uma média móvel exponencial foi usada nesta análise como um filtro passa baixas.

Feng et al. (2002) apresenta o *Improved Swinging Door Trending* (ISDT). Essa proposta conta com um DC que varia em relação aos resultados das compressões anteriores. O ISDT modifica os parâmetros de compressão dinamicamente quando um dado é adquirido. Ele incorpora o parâmetro $F_{adj}(0 < F_{adj} < 1)$. F_{adj} é usado para ajustar o valor do CD, o qual tem uma faixa de valores possíveis.

Zhao et al. (2012) apresenta o *Distributed Swinging Door Trending* (DSDT), com objetivo de usar os recursos computacionais dos dispositivos para comprimir os dados coletados. Essa proposta conta com um centro de compressão com mudança auto-adaptativa, considerando as condições de uso da largura de banda e dos recursos computacionais dos dispositivos. DSDT conta com dois algoritmos, um que comprime no dispositivo outro no centro de compressão.

Liu et al. (2014) propõem o KSDT que é uma melhoria do SDT para ser usado em RSSF para aquisição de sinais de falha mecânica. Essa técnica ajusta o valor do CD dinamicamente baseada nos parâmetros específicos usados no monitoramento da condição da máquina. Esta proposta foi criada para um caso especifico e sua aplicação em outros cenários de uso não pode ser considerada.

Silva, Guedes e Vasques (2008) usa o SDT como algoritmo de compressão em RSSF. O objetivo desta pesquisa foi investigar o rendimento de um algoritmo de CD em uma aplicação de monitoramento para um ambiente de automação industrial. O algoritmo foi implementado diretamente nos dispositivos. O resultado mostra uma redução do 85% nos dados transmitidos e a tipificação da vida do tempo de vida da rede.

Um acionador de eventos assíncrono para uma arquitetura de rede de sensores inteligentes foi apresentado em (LEãO; GUEDES; VASQUES, 2007a) para ser usado em padronização OMG'S, ele integra um algoritmo de CD dentro dos sensores inteligentes, o resultado mostra uma redução considerável da troca de mensagens na rede de comunicação, a proposta foi modificada para ser usada na padronização IEEE 802.15.4 (LEãO; GUEDES; VASQUES, 2007b)

3.3 CONCLUSÕES DO CAPÍTULO

Analisando as propostas apresentadas neste capítulo, pode-se fazer um sumário das principais conclusões:

- Os esquemas de CD analisados normalmente desconsideram o efeito dos valores anômalos ou *outliers*, os quais podem afetar o desempenho do algoritmo.
- A utilização de algoritmos de codificação ou de CD sem perda nos dados comprimidos por um algoritmo com perda consegue melhorar os resultados de compressão sem aumentar o erro, porém aumenta o custo computacional ao executar dois algoritmos sequencialmente.
- Os mecanismos de controle de erro e de compressão mais comuns são baseados em limites de erro. Estes podem ser feitos usando métricas de desempenho, a diferença entre o

valor atual e o anterior, a tendência do sinal, entre outros.

- As propostas analisadas que usam interpolação linear tendem a ter menor complexidade computacional.
- Nem sempre a qualidade e capacidade de recuperar os dados é relevante. Aplicações específicas podem funcionar adequadamente com dados irreconhecíveis.

Os estudos efetuados neste capítulo, por meio de uma revisão da literatura, permitiram uma análise das principais técnicas de CD com perda que podem ser aplicadas em sensores da IoT. No próximo capítulo é apresentada a proposta de CD deste trabalho de mestrado, a qual é baseada na técnica SDT.

4 PROPOSTA: AUTO-DEFINIÇÃO DE PARÂMETROS DE SDT

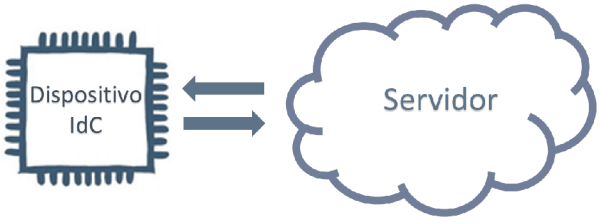
As análises, considerações, modificações e propostas desta dissertação consideram um sistema como o apresentado na Figura 7, no qual existem dispositivos IoT que podem ter várias atividades e tarefas, ou a necessidade de conectar-se com outros dispositivos, porém precisam transmitir dados coletados a partir de sensores para um servidor, o qual poderá ser local ou fornecido por um prestador de serviços baseado na computação em nuvem, névoa ou na borda. A coleta e transmissão das amostras de sensores não requerem a interação, intermediação ou colaboração entre os dispositivos.

A proposta é definir um processo que permita a auto-definição do parâmetro Desvio da Compressão (DC) do SDT, utilizado como estudo de caso. Adicionalmente, uma nova métrica de qualidade da compressão é proposta e utilizada como critério de seleção do DC (Seção 4.1). O objetivo é avaliar o erro e a compressão conseguida em uma compressão, utilizando uma única métrica que represente a qualidade da compressão. Na Seção 4.2 mostram-se duas modificações do SDT. Estas adicionam um processo de auto-definição do parâmetro DC. A primeira modificação (Subseção 4.2.1) apresenta um mecanismo que auto-define e ajusta o valor do DC utilizando fórmulas simples e o sinal coletado pelo sensor. Esse mecanismo foca em tomar a decisão no dispositivo da IoT sem intervenção do servidor, além de explorar a utilização do sinal como métrica de avaliação sem definir um limite como no ASDT.

Na segunda modificação (Subseção 4.2.2), explora-se a métrica proposta como critério de seleção. Essa abordagem requer mais recursos computacionais pelo que é implementado no servidor. Nessa implementação, simulam-se vários cenários de compressão utilizando N amostras de treinamento. O DC do cenário com melhores resultados na métrica utilizada é definido no SDT do dispositivo.

Essas propostas de modificação podem ser utilizadas em qualquer algoritmo de CD de interpolação linear. Os mecanismos dessa classificação conseguem bons resultados de compressão sem precisar de recursos computacionais altos, o que é adequado para ser executado em dispositivos da IoT.

Figura 7 – Dispositivo IoT transmitindo dados para um servidor



Fonte – O autor.

4.1 CRITÉRIO DE COMPRESSÃO

Os algoritmos de CD com perda devem ser avaliados considerando a compressão obtida e o erro resultante. Na Subseção 2.2.1 foram apresentadas algumas das métricas mais comuns para representar os níveis de erro e de compressão, além de fórmulas para determinar a qualidade da compressão (2.2.1.3) considerando estes níveis. Esta dissertação propõe uma nova métrica de avaliação da qualidade da compressão que possa ser utilizada como métrica de decisão dos parâmetros dos algoritmos com perda. Este critério é chamado de Critério de Compressão (CC) e foi fundamentado na média harmônica entre dois números.

A Equação 4.2 apresenta a fórmula resultante. Pode-se perceber que Taxa de Erro (TE) foi trocado pela Taxa de Similaridade (TS) (Equação 4.1), a qual representa o nível de similaridade entre os sinais original e comprimida. Esta mudança é realizada para que as métricas tenham um comportamento semelhante. A média harmônica foi escolhida por apresentar uma maior sensibilidade quando um dos valores é pequeno respeito ao outro, sendo adequado porque maximiza as compressões com valores altos de Taxa de Compressão (TC) e TS, e reduz quando um valor fica para trás, garantindo um equilíbrio entre uma compressão adequada e um erro aceitável.

O Critério de Compressão (CC) segue uma lógica semelhante do *F-Score* utilizado nos algoritmos de Aprendizado de Máquina (*Machine Learning*) e algoritmos de busca e recuperação de informação como critério de avaliação. O *F-Score* representa em um único valor ponderado entre a precisão e o *recall*, a confiabilidade (*dependability*) de um método na etapa de testes. Este tem um intervalo de valores entre 0% que significa o pior caso e 100% no caso de ter precisão e *recall* perfeitos (RIJSBERGEN, 1979; SASAKI; YUTAKA et al., 2007).

O CC precisa seguir a mesma lógica do *F-Score*, mas utilizando o Taxa de Similaridade (TS) e o TC em vez da precisão e *recall*. Das métricas apresentadas na Subsubseção 2.2.1.2 é escolhida como métrica da TC o Espaço Economizado (EE) (Equação 2.10), esta tem o intervalo de valores de [0%,100%) adequado para o CC.

No caso do Taxa de Similaridade (TS) (4.1) requere-se utilizar métricas que representem o erro da compressão, considerando as métricas apresentadas na Subsubseção 2.2.1.1, o Erro Absoluto Relativo (EAR) (2.2) é utilizado, porém métricas como o Erro Quadrático Relativo (EQR) (Equação 2.3) e o Diferença percentual da RMQ (DPR) (Equação 2.4) poderiam também ser utilizadas. Estas métricas contam com um intervalo de valores possíveis de $[0, \infty)$, inadequado para ser utilizado no Critério de Compressão (CC), para tornar elas fatíveis no TS (Equação 4.1) limita-se o intervalo de possíveis valores a $[0\%, 100\%]$, retornando valores iguais a 0 quando o erro é igual ou maior ao 100%. Este limite é adequado considerando que um erro de 100% representa que a somatória de todas as magnitudes das diferenças entre os sinais original e comprimido é equivalente à somatória de todas as magnitudes das amostras do sinal original.

A Equação 4.2 apresenta uma constante β utilizada no *F-Score* quando é necessário dar um maior peso ou relevância a precisão ou ao *recall* (RIJSBERGEN, 1979; SASAKI; YU-

TAKA et al., 2007). β permite modificar o nível de relevância da TS e da TS considerando as necessidades dos sistemas. Um β igual a 1 significa que a TC e a TE contam com a mesma ponderação (Equação 4.3). Um $\beta > 1$ da maior importância ao TS, e quando $0 \leq \beta < 1$ o TC tem maior relevância.

$$TS(\%) = \begin{cases} 100 - TE, & \text{SE } TE < 100 \\ 0, & \text{SE } TE \geq 100 \end{cases} \tag{4.1}$$

$$CC = \frac{(1+\beta^2) \cdot TC \cdot TS}{(\beta^2 \cdot TC) + TS} \tag{4.2}$$

$$CC = \frac{2 \cdot TC \cdot TS}{TC + TS}, \qquad \beta = 1 \tag{4.3}$$

4.2 SWINGING DOOR TRENDING PARA DISPOSITIVOS IOT

A seleção do algoritmo de compressão desta dissertação foi realizada através dos seguintes critérios:

1. O algoritmo precisa ter uma complexidade computacional constante O(1).
2. O algoritmo deve poder seguir a tendencia dos dados no tempo
3. O algoritmo executa-se em cada rodada de amostragem
4. O algoritmo tem poucos parâmetros e variáveis (além de ser constantes)
5. Fundamentação teoria e informação.

Os algoritmos que cumpriram melhor estes critérios foram os de interpolação linear, já que requerem menos recursos computacionais.Duas propostas cumpriram os requisitos, O *Swinging Door Trending* (SDT) (BRISTOL, 1990; BRISTOL, 1987) e o *top-down Piecewise Linear Approximation with Minimum number of Line Segments* (PLAMLiS) (Pham; Le; Choo, 2008).

O *Swinging Door Trending* (SDT) foi escolhido como algoritmo de compressão por ter uma melhor sustentação teórica e um maior impacto na literatura, além de conseguir melhores resultados de compressão no tempo. o SDT também possui um único parâmetro de configuração e um sistema de tolerância de erro simples, dinâmico e que segue a tendência dos dados analisados. O *Swinging Door Trending* (SDT) tem como seu parâmetro principal o Desvio da Compressão (DC).

Este define a largura do paralelogramo criado e a diferença que um ponto pode ter com relação à linha de tendência atual, o que significa que funciona como um limite de tolerância ao erro. A magnitude deste parâmetro afeta diretamente os resultados da compressão e o erro final do sinal comprimido. Um valor muito pequeno do Desvio da Compressão (DC) pode implicar um resultado em termos de TC pobre, porém um sinal mais semelhante ao original

e um DC alto incrementa a TC além de um aumento no nível de erro no sinal comprimido. Propostas como o *Adaptive Swinging Door Trending* (ASDT) (NETO LUIZ AUGUSTO, 2014) e o *Improved Swinging Door Trending* (ISDT) (FENG et al., 2002) implementam mecanismos de ajuste automático do DC, porém o projetista decide o intervalo de valores que estas propostas podem usar.

Uma seleção apropriada do valor da DC pode melhorar o resultado da compressão em termos de TC e TE. A seguir, na Subseção 4.2.1 modifica-se o SDT para adicionar uma etapa de treinamento que permita a seleção do Desvio da Compressão (DC) desde o próprio dispositivo, e na Subseção 4.2.2 é proposto um esquema que se utiliza de uma quantidade de dados para definir o valor do DC desde o servidor usando o CC (Equação 4.3) como critério de seleção. Estes mecanismos de auto definição podem ser usados com outros algoritmos de interpolação linear que usam sistemas de tolerância ao erro.

4.2.1 *Self-definition Swinging Door Trending* (SSDT)

O *Self-definition Swinging Door Trending* (SSDT) define o valor da DC automaticamente (CORREA et al., 2019). A proposta é projetada para ser executada nos dispositivos da IoT, sem aumentar a complexidade computacional do SDT. A seleção da DC tem repercussões no resultado final da compressão. Um valor muito alto por exemplo, pode implicar uma TC alta, porém pode incrementar a TE também. Nesta situação o conhecimento do comportamento do sinal e as características do sensor são relevantes. Tendo isto em conta, procurou-se usar uma métrica simples que permita decidir um valor do CD adequado.

O SDT é um algoritmo de interpolação linear que usa a inclinação entre amostras em um plano bidirecional (valor-tempo) e certos critérios como apresenta-se na Subseção 2.2.2. O valor absoluto da diferença entre o ponto atual e o anterior é proposto como métrica (Equação 4.4), supondo intervalos de tempo constantes e analisando-o em tempo discreto. A diferença entre dois pontos contínuos é igual à inclinação da reta. Esta métrica oferece informação sobre a mudança do sinal no tempo, relativo ao seu valor anterior.

$$s^i(d,p) = |d_v - p_v| \tag{4.4}$$

A métrica escolhida é utilizada em quatro possíveis equações matemáticas as quais foram escolhidas pela sua simplicidade, permitindo seu uso em dispositivos IoT sem aumentar consideravelmente o tempo de execução do SDT e sem aumentar sua complexidade computacional. A média aritmética (Equação 4.5), a Média Móvel exponencial (MME) (Equação 4.6), a média sem as inclinações nulas (*Zmean*) (Equação 4.7) e o *range* (Equação 4.8) são propostos. A média aritmética soma as diferenças e as divide pela quantidade destas. O Média Móvel exponencial (MME) é um filtro da resposta infinita ao impulso de primeira ordem que aplica fatores de ponderação que diminuem exponencialmente. Esse filtro conta com o parâmetro α que tem valores entre $[0,1]$. A média sem diferenças nulas é uma modificação da média arit-

mética. A ideia é somente considerar quando o sinal sofre alterações. O *range* corresponde à media entre as diferenças do maior e do menor valor.

$$f_m = \frac{1}{n}\sum_{i=1}^{n} S^i \tag{4.5}$$

$$f_{mme}(d,p) = (1-\alpha)CD + \alpha \cdot S^i(d,p) \tag{4.6}$$

$$f_{Zmean}(d,p) = \begin{cases} CD + \dfrac{s^i(d,p)}{N^z} & \text{if } s^i(d,p) \neq 0 \\ \dfrac{CD \cdot (N^z+1)}{N^z} & \text{if } S^i(d,p) = 0 \end{cases} \tag{4.7}$$

$$f_r = \frac{MAX + MIN}{2} \tag{4.8}$$

O Algoritmo 2 apresenta o código do *Self-definition Swinging Door Trending* (SSDT). São utilizadas inicialmente como dados, $N+1$ amostras (N diferenças), com objetivo de decidir o valor de DC. Essas amostras são enviadas diretamente ao servidor sem comprimi-las. Somente quando essa etapa de treinamento finaliza, o dispositivo começa o processo de compressão dos dados.

No processo de compressão, ao finalizar um paralelogramo, o SDT utilizará os resultados das métricas para realizar um ajuste do DC. Isso é feito com o propósito de obter um sistema de auto-ajuste do valor do DC, considerando o valor do sinal. O valor α do Média Móvel exponencial (MME) utilizado é igual ao proposto em (NETO LUIZ AUGUSTO, 2014).

Algoritmo 2 SSDT

```
 1: Initialization
 2: d = ⟨Time, Value⟩
 3: CD = Count = 0
 4: Training_min = TRAINING
 5: Trained = False
 6: α = 2/(TRAINING + 1)                                   ▷ EMA
 7: N^z = TRAINING                                         ▷ Zmean
 8: Max = −∞                                               ▷ Range
 9: Min = ∞                                                ▷ Range
10: BROADCAST(d)

11: function SSDT(Time, Value)
12:     if Trained == False then
13:         TRAIN(Time, Value)
14:     else
15:         SDT(Time, Value)

16: function TRAIN(Mode, Time, Value)
17:     p = d
18:     d = ⟨Time, Value⟩
19:     DELIVER(d)
20:     if Mode == MEAN then       CD = MEAN(d, p)
21:     else if Mode == ZMEAN then CD = ZMEAN(d, p)
22:     else if Mode == RANGE then CD = RANGE(d, p)
23:     else if Mode == EMA then   CD = EMA(d, p)
24:     if ++Count == Training_min then
25:         c = d
26:         s^u_Max = −∞
27:         s^l_Min = ∞
28:         Trained = True
```

4.2.2 Treinamento do Swinging Door Trending

No TSDT, o algoritmo de compressão é executado pelo próprio dispositivo, porém um processo de decisão da DC é executado no servidor. O critério de decisão Critério de Compressão (CC), apresentado em Seção 4.1, é usado como critério de decisão. O Algoritmo 3 apresenta o código usado no dispositivo. Nele, o dispositivo envia todas as amostras sem comprimi-las até receber do servidor, o valor do DC. Quando esse valor é recebido, o dispositivo começa a executar o SDT e a enviar os dados comprimidos.

O Algoritmo 4 mostra o código no servidor. Quando um novo dispositivo conecta-se ao servidor, este coleta N amostras que serão usadas no processo de treinamento. Nesse treinamento, o servidor faz um mapeamento dos possíveis valores do DC. Um critério de precisão (α) e um valor máximo aceitável da DC (β_{max}) são necessários. Esses parâmetros permitem determinar o número de execuções do algoritmo. Usando o CC como critério de seleção, o valor de β com melhor desempenho é usado como DC. Esse processo basicamente executa o SDT usando valores diferentes de β, permitindo explorar diferentes possíveis cenários de compressão.

Algoritmo 3 TSDT no dispositivo

```
1:  Initialization
2:  d = ⟨Time, Value⟩                                   ▷ Primeiro ponto
3:  c, u, l = ⟨0, 0⟩                                    ▷ Variaveis do SDT
4:  s^u_Max = −∞,      s^l_Min = ∞
5:  CD, Round = 0
6:  Window_period = WINDOWS
7:  Trained = False
8:  DELIVER(d)                                          ▷ Primeira amostra enviada

9:  function TSDT(Time, Value)
10:     if Trained == False then
11:         TRAIN(Time, Value)
12:     else
13:         if SDT(Time, Value) then
14:             DELIVER(c)                              ▷ amostra enviada
15: function TRAIN(Time, Value)
16:     d = ⟨Time, Value⟩
17:     DELIVER(d)                                      ▷ amostra enviada
18:     if RECEIVE() then
19:         CD = MESSAGE()
20:         c = d
21:         NEW_PARALLELOGRAM()
22:         Trained = True
```

Algoritmo 4 TSDT processo de treinamento no servidor

```
1:  function TSDT(δ_max, α, V = ⟨⟨t_0, v_0⟩, ..., ⟨t_n, v_n⟩⟩)
2:      cc = 0
3:      cd = 0
4:      δ = 0
5:      while δ < δ_max do
6:          δ += α
7:          R = ⟨SDT(t, v) | ⟨t, v⟩ ∈ V⟩
8:          cr = CR(V.lenght(), R.lenght())
9:          ce = CE(V, R)
10:         cs = CS(ce)
11:         if CC(cr, cs) < cc then
12:             cc = CC(ce, cr)
13:             cd = δ
14:     DELIVER(cd)
```

4.3 CONCLUSÕES DO CAPÍTULO

Este capítulo descreveu o *Training Swinging Door Trending* (TSDT) – a principal proposta deste trabalho–, o qual é fundamentado na necessidade de configurar os parâmetros do algoritmo de compressão considerando o comportamento do sinal, da Taxa de Compressão (TC) e da Taxa de Erro (TE). O *Self-definition Swinging Door Trending* (SSDT) também foi descrito, o qual foi proposto nas etapas iniciais da pesquisa como um mecanismo de auto-definição e auto-ajuste dos parâmetros do *Swinging Door Trending* (SDT). O *Self-definition Swinging Door Trending* (SSDT) considera o comportamento do sinal, focando na diferença entre o valor anterior e atual. O Critério de Compressão (CC) foi proposto como uma nova métrica para avaliar a qualidade de uma compressão em algoritmos com perda. No próximo capítulo é descrito um ambiente de testes construído para a avaliação da proposta. Os principais resultados obtidos também são apresentados e discutidos.

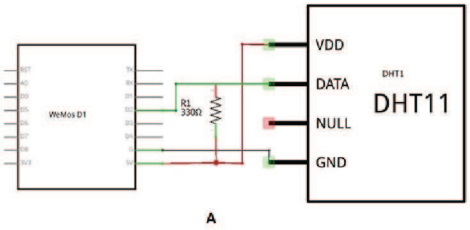
5 RESULTADOS

5.1 AMBIENTE DE TESTES

Para avaliar o funcionamento do SDT em dispositivos da IoT, foi implementado um ambiente de coleta de dados, com o objetivo de reduzir a quantidade de dados transmitida pelo dispositivo e armazenada pelo servidor. Apesar de não ter sido assumido qualquer tipo de restrição temporal a ser cumprida na tarefa de compressão, avaliou-se, também, o tempo de execução do algoritmo por parte do dispositivo.

A placa de desenvolvimento **WeMos D1 Wi-Fi UNO ESP-12E** (AI-THINKER, 2015) foi usada como dispositivo da IoT. Esta placa é baseada no **ESP8266** (ESPRESSIF, 2018) e suporta os padrões de Wi-Fi 802.11 b/g/n na banda dos 2.4-2.5 GHz (AI-THINKER, 2015). Já o sensor de temperatura e umidade relativa **DHT11** (ADAFRUIT, 2019) é usado para coletar dados. A Figura 8 (A) apresenta o circuito implementado, enquanto a Figura 8 (B) mostra o protocolo usado em cada camada no modelo TCP/IP.

Figura 8 – (A) circuito do sistema, (B) modelo TCP/IP



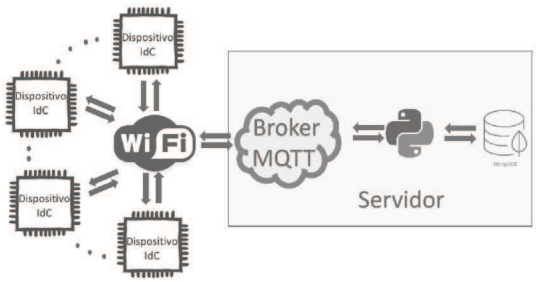
Modelo TCP/IP	
Camada	**Protocolo**
Aplicação	MQTT
Transporte	TCP
Rede	IPv4
Ligação de dados	802.11 b/g/n

B

Fonte – O autor.

A Figura 9 apresenta a implementação utilizada. Essa implementação usa Mosquitto (LIGHT, 2017) como *broker* MQTT. Um *script* em Python foi implementado para receber e gerenciar as informações dos sensores, além estabelecer uma comunicação com o MongoDB, que é o gerenciador de base de dados não relacional usado.

Figura 9 – Arquitetura implementada

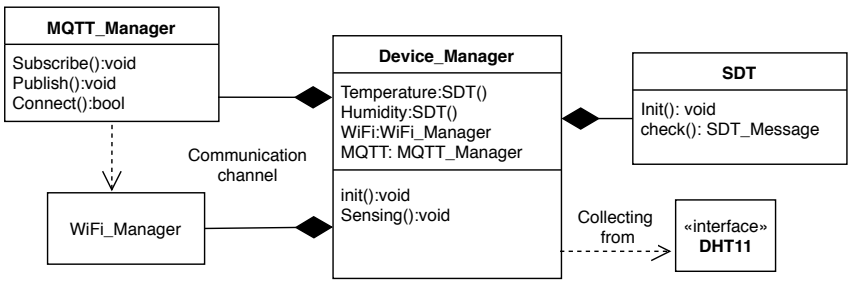


Fonte – O autor.

5.1.1 Dispositivo

Os dispositivos foram programados usando o Arduíno IDE, através do projeto apresentado em (GROKHOTKOV, 2017), o que permite programar as placas Wi-Fi ESP8266 usando uma lógica semelhante das placas Arduíno convencionais. A Figura 10 apresenta o diagrama de classes do código dos dispositivos. Na imagem observa-se que uma classe *Device_Manager* coleta os dados do sensor DHT11 e os comprime através de um objeto da classe SDT. O dispositivo usa como canal de comunicação uma rede WiFi, fornecida pela classe *WiFi_Manager*, e para a comunicação, o protocolo *Message Queuing Telemetry Transport* (MQTT) pela classe *MQTT_Manager*.

Figura 10 – UML Classes no Dispositivo

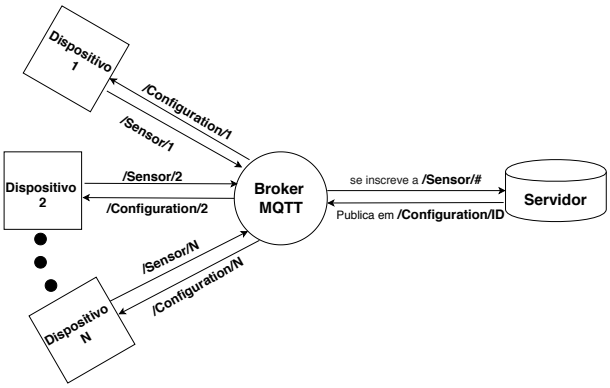


Fonte – O autor.

5.1.2 Comunicação

A comunicação foi realizada utilizando o protocolo *Message Queuing Telemetry Transport* (MQTT) e o JSON como formato para o intercâmbio de dados. A Figura 11 mostra os tópicos para a publicação e a inscrição de mensagens usadas no sistema. A Listagem 2 mostra a estrutura do JSON usada pelo dispositivo para a comunicação com o servidor, e a Listagem 3 apresenta a mensagem enviada ao dispositivo desde o servidor. A coleta e transmissão de dados no dispositivo é realizada cada 5 segundos.

Figura 11 – MQTT publicação e inscrição



Fonte – O autor.

Listagem 2 – Estrutura JSON mensagens a partir dos dispositivos

```
{
    "id":int, //ID do dispositivo
    "Humidity":
    {
        "Value":int, //Valor atual
        "Compressed":float, //Valor comprimido (se houver)
        "CD":float, //\gls{DC}, envia-se com o valor comprimido
        "Time":double //Tempo de execução em microssegundos
    },
    "Temperature":
    {
        "Value":int,
        "Compressed":float,
        "CD":float,
        "Time":double
    },
    "Time":double, //Tempo de execução total com comunicação
    "Round":int //Rodada de compressão
}
```

Fonte – O autor.

5.1.3 Servidor

No servidor, usou-se um código em Python, que administra a comunicação MQTT com os dispositivos e a base de dados MongoDB. A Figura 12 apresenta a estrutura geral do servidor. Este não foi estruturado usando lógica de classes. O servidor primordialmente coleta a informação enviada pelos dispositivos e a separa em cinco diferentes documentos na base de dados: Informação do dispositivo, Temperatura, Temperatura comprimida, Umidade e Umidade comprimida. A data de quando foi recebida a informação do dispositivo, um contador de reinícios e uma lógica para evidenciar comportamentos estranhos foram programados para evi-

Listagem 3 – Estrutura JSON mensagens para o dispositivos

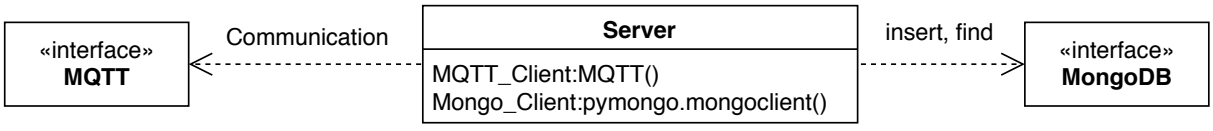
```
{
    "Humidity":
    {
        "CD":float, //Valor do \gls{DC}
        "Round":int, //restaurar ultimo estado (se for necessário)
        "Value":int //Ultimo valor recebido (se for necessário)
    },
    "Temperature":
    {
        "CD":float,
        "Round":int,
        "Value":int
    }

}
```

Fonte – O autor.

tar amostras repetidas, amostras sem sentido, e também para permitir a restauração do último estado conhecido. Os dados são armazenados usando o formato JSON.

Figura 12 – Estrutura geral do servidor



Fonte – o autor

5.1.4 Resultados

No Laboratório de Pesquisa de Sistemas Distribuidos (LaPeSD) da Universidade federal de Santa Catarina, dois dispositivos da IoT foram implementados seguindo o exposto na Figura 8. Os dispositivos foram programados para enviar os dados originais e comprimidos com o SDT. Assim, os dados eram coletados a cada 5 segundos, o que resultou em 192610 amostras de temperatura e umidade relativa de cada dispositivo. Como estes dispositivos têm limitações computacionais (SCHRICKTE et al., 2016), analisou-se se eles contavam com recursos suficientes para realizar uma compressão de dados online.

A principio, o SDT tem uma complexidade computacional na notação Big-O igual a $O(1)$. No experimento, calculou-se o tempo de execução do SDT e suas modificações no WeMos D1 com uma velocidade de relógio de 160 MHz. O pior caso de tempo de execução obtido nos dispositivos foi de 162 microssegundos. Esse tipo de informação pode ser importante na etapa de projeto de sistemas com restrições de tempo real. No trabalho desenvolvido

nesta dissertação, não se assume quaisquer restrições temporais para a tarefa de compressão. Contudo, observando que o tempo máximo de execução medido é menor que 2 milissegundos, considera-se que esse tipo de algoritmo pode ser adequado para aplicações que precisem de uma compressão funcionando em tempo de execução (*online*), tais como, monitoramento energético (60Hz), monitoramento climático, aplicações médicas (*healthcare*), entre outras.

Todas as amostras foram enviadas sem compressão e comprimidas pelo SDT. O tempo de execução total do sistema não foi representado, uma vez que o código conta com funcionalidades exclusivas da simulação e para a interpretação e estudo do pesquisador.

5.2 BASE DE DADOS DE AMOSTRAS DE SENSORES

Uma base de dados de sensores de Intel Lab Data (MADDEN et al., 2004) foi utilizada para a simulação de diferentes cenários de compressão usando o SDT, TSDT e SSDT. Esta base de dados conta com amostras de 54 dispositivos que foram posicionados no Laboratório de pesquisa Intel Berkeley conforme apresenta-se na Figura 13. As amostras foram coletadas a cada 31 segundos e o dispositivo coletou amostras de umidade relativa, temperatura, iluminação e voltagem entre 28 de fevereiro e 5 de abril de 2004. A placa meteorológica Mica2DoT WB foi usada como dispositivo. Essa placa conta com um sensor de umidade relativa SHT11 (SENSIRION, 2008) e um sensor de luz ambiental TSL2250. A base de dados conta com 2.3 milhões de leituras coletadas desde os dispositivos.

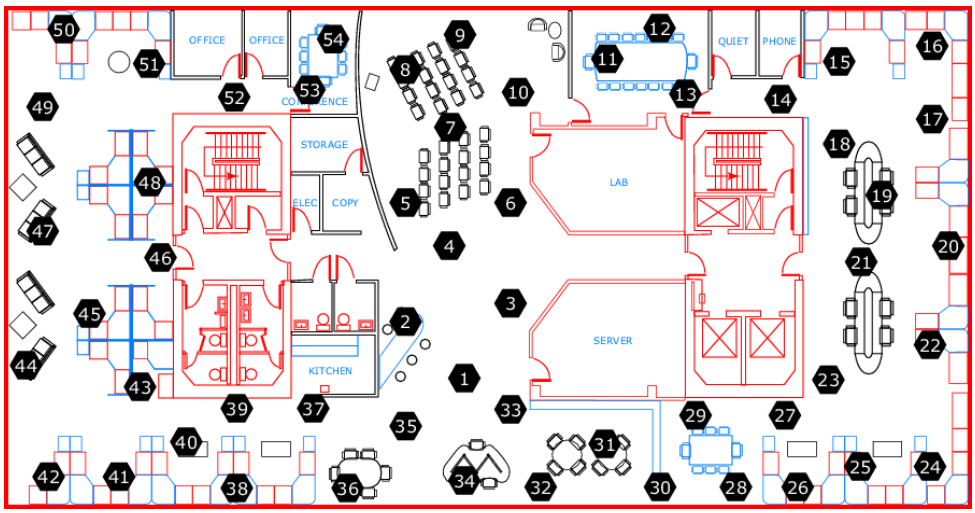
A base de dados do Intel Lab Data e os dados coletados no experimento apresentado na Seção 5.1 são utilizadas como estudos de caso para avaliar diferentes cenários de compressão de dados utilizando diferentes configurações nos algoritmos baseados no SDT, somente dez dispositivos da Intel são utilizados, devido à dificuldade de apresentar os resultados de 54 dispositivos. A Tabela 10 apresenta as informações gerais dos sensores DHT11 e o SHT11 utilizados no ambiente de testes e no Intel Lab Data, respectivamente.

Tabela 10 – Informação básica do SHT11 e o DHT11 (UR=Umidade Relativa,°C=Graus Celsius)

Umidade Relativa				
Característica	**SHT11**		**DHT11**	
	Valores típicos	**Unidade**	**Valores típicos**	**Unidade**
Intervalo	0.0-100.0	%UR	20-90	%UR
Resolução	0.05	%UR	1	%UR
Precisão	±3.0	%UR	±5	%UR
Temperatura				
Característica	**SHT11**		**DHT11**	
	Valores típicos	**Unidade**	**Valores típicos**	**Unidade**
Intervalo	−40.0 - 123.8	°C	0-50	°C
Resolução	0.01	°C	1	°C
Precisão	±0.4	°C	2	°C

Fonte – (ADAFRUIT, 2019; SENSIRION, 2008)

Figura 13 – Distribuição das placas meteorológicas Mica2DoT WB no Laboratório de pesquisa Intel Berkeley



Fonte – (MADDEN et al., 2004)

5.3 CRITÉRIO DE COMPRESSÃO

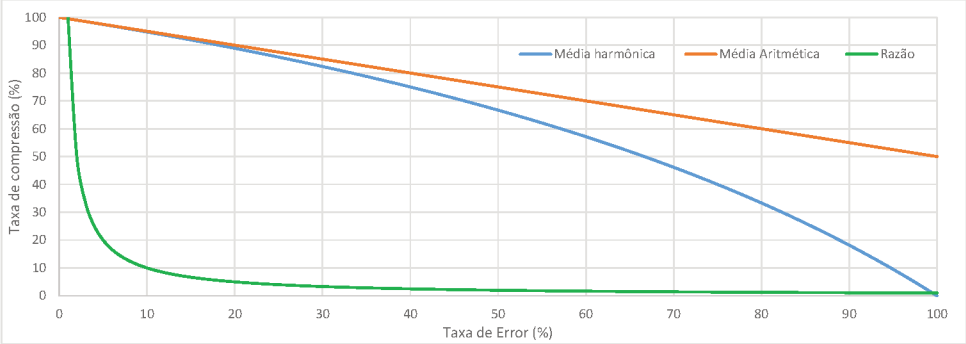
A média harmônica entre dois números foi proposta como métrica de qualidade da compressão e critério de decisão. Esse tipo de média apresenta uma maior sensibilidade quando um dos valores é pequeno com relação ao outro.

A Figura 14 mostra o comportamento da razão, da média aritmética e média harmônica usando um valor hipotético de 100% na Taxa de Compressão (TC). A média aritmética tem um comportamento linear, enquanto a razão tem um comportamento assintótico com uma queda alta de seu valor com valores pequenos de erro. Já a média harmônica tem um comportamento semelhante à média aritmética, onde a diferença vai aumentando quando um valor é menor com relação ao outro.

Na Tabela 11 apresenta-se o Desvio da Compressão (DC) com melhor resultado em 10 dispositivos da Intel (MADDEN et al., 2004). Todas as amostras de umidade e temperatura foram consideradas. O TSDT utiliza quatro métricas para definir o melhor DC. O Critério de Compressão (CC) e a média apresentam resultados iguais nos dados de umidade. Nesse caso, o DC com melhores resultados tem valores baixos em TE e altos em Taxa de Compressão (TC), dando assim resultados iguais. No caso da temperatura, o sinal é um pouco instável, resultando em uma queda nos valores do TC, o que faz que o Critério de Compressão (CC) escolha um valor com um erro maior para ganhar um pouco mais de TC.

Os dados do laboratório da Intel têm alguns erros nas amostras. Por isso, essas amostras foram filtradas, considerando o intervalo de valores dos sensores apresentados na Tabela 10. No caso da temperatura, o intervalo é maior, permitindo mais amostras atípicas, o que gera um comportamento instável no sinal, requerendo uma maior quantidade de amostras para representá-lo. A razão usa a mesma fórmula que o Índice de qualidade (IQ) (Equação 2.12), porém usa o Espaço Economizado (EE) e não a Proporção de Compressão (PC). Os valores da razão são maiores quando o TE é pequeno, escolhendo valores de DC pequenos com resultados pobres de compressão. No caso do IQ, o PC (Equação 2.9) tem um intervalo de valores maior, dando melhores resultados no TC. Porém, ao comprimir um sinal mais instável (temperatura), seleciona DC pequenos, sabendo que o crescimento do PC pelo crescimento da tolerância é pequeno.

Figura 14 – Valores do CC para um TC de 100% e diferentes níveis de erro (As médias usam o TS)



Fonte – O autor.

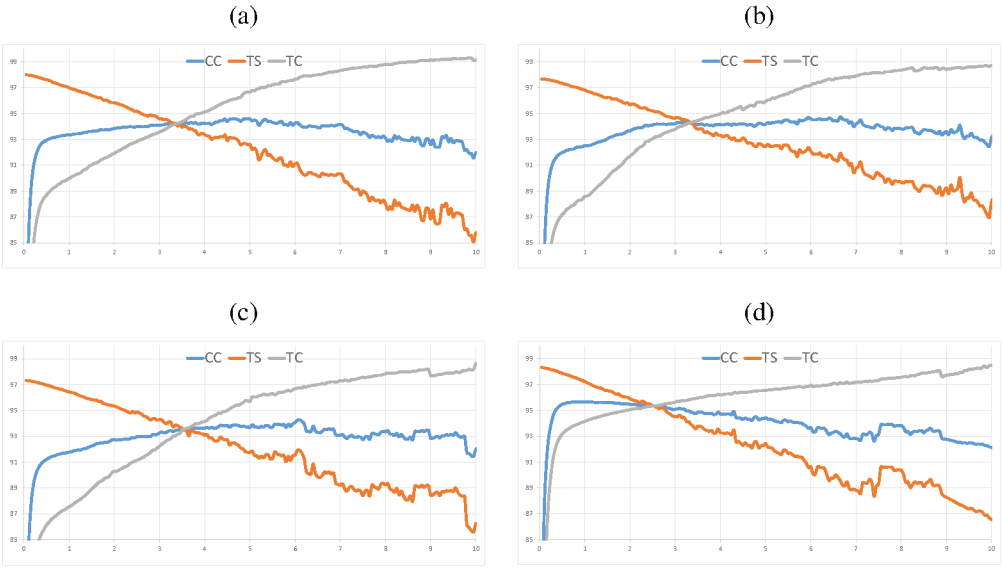
Tabela 11 – Melhor compressão respeito a 4 métricas diferentes no TSDT nos primeiros 10 dispositivos da Intel (MADDEN et al., 2004). Umidade: $\delta max = 15.0, \alpha = 0.05$, Temperatura: $\delta max = 10.0, \alpha = 0.01$

Umidade	**CC**			**Média**			**Razão**			**IQ**		
Dispositivo	**DC**	**TE**	**TC**	**DC**	**TE**	**TC**	**DC**	**TE**	**TC**	**DC**	**TE**	**TC**
Intel 1	1.7	1.9	97.94	1.7	1.9	97.94	0.05	0.08	67.78	0.15	0.21	92.13
Intel 2	0.55	0.62	96.81	0.55	0.62	96.81	0.05	0.08	56.57	0.2	0.25	93.85
Intel 3	1.75	2.14	97.26	1.75	2.14	97.26	0.05	0.08	77.01	0.2	0.25	95.35
Intel 4	0.85	0.77	97.78	0.85	0.77	97.78	0.05	0.09	73.47	0.85	0.77	97.78
Intel 5	2.5	2.5	97.98	2.5	2.5	97.98	0.05	0.11	64.65	0.35	0.46	93.1
Intel 6	1.2	1.14	97.94	1.2	1.14	97.94	0.05	0.1	78.72	0.1	0.15	86.21
Intel 7	1.35	1.14	97.53	1.35	1.14	97.53	0.05	0.09	50.0	0.65	0.63	96.7
Intel 8	0.8	1.14	95.92	0.8	1.14	95.92	0.05	0.07	78.02	0.1	0.11	87.64
Intel 9	2.8	3.08	96.97	2.8	3.08	96.97	0.05	0.1	61.11	0.15	0.19	83.52
Intel 10	1.65	1.4	97.92	1.65	1.4	97.92	0.1	0.15	63.16	0.5	0.53	95.29
Temperatura	**CC**			**Média**			**Razão**			**IQ**		
Dispositivo	**DC**	**TE**	**TC**	**DC**	**TE**	**TC**	**DC**	**TE**	**TC**	**DC**	**TE**	**TC**
Intel 1	1.7	3.05	81.34	1.55	2.8	81.11	0.01	0.02	50.36	0.01	0.02	50.36
Intel 2	2.11	3.42	73.94	2.04	3.35	73.88	0.01	0.02	47.49	0.01	0.02	47.49
Intel 3	1.95	3.71	79.68	0.65	1.21	77.49	0.01	0.03	50.5	0.01	0.03	50.5
Intel 4	0.77	1.12	71.2	0.43	0.62	70.82	0.01	0.02	49.87	0.01	0.02	49.87
Intel 5	0.59	1.5	68.97	0.59	1.5	68.97	0.01	0.03	48.22	0.01	0.03	48.22
Intel 6	0.24	0.68	94.86	0.24	0.68	94.86	0.01	0.04	69.34	0.03	0.08	87.96
Intel 7	0.47	0.67	68.47	0.35	0.5	68.35	0.01	0.02	51.38	0.01	0.02	51.38
Intel 8	0.95	1.58	76.47	0.49	0.82	75.93	0.01	0.02	53.72	0.01	0.02	53.72
Intel 9	1.25	2.48	80.18	1.03	1.97	79.78	0.01	0.03	46.11	0.01	0.03	46.11
Intel 10	0.71	1.15	73.83	0.51	0.84	73.57	0.01	0.02	44.07	0.01	0.02	44.07

Fonte – O autor

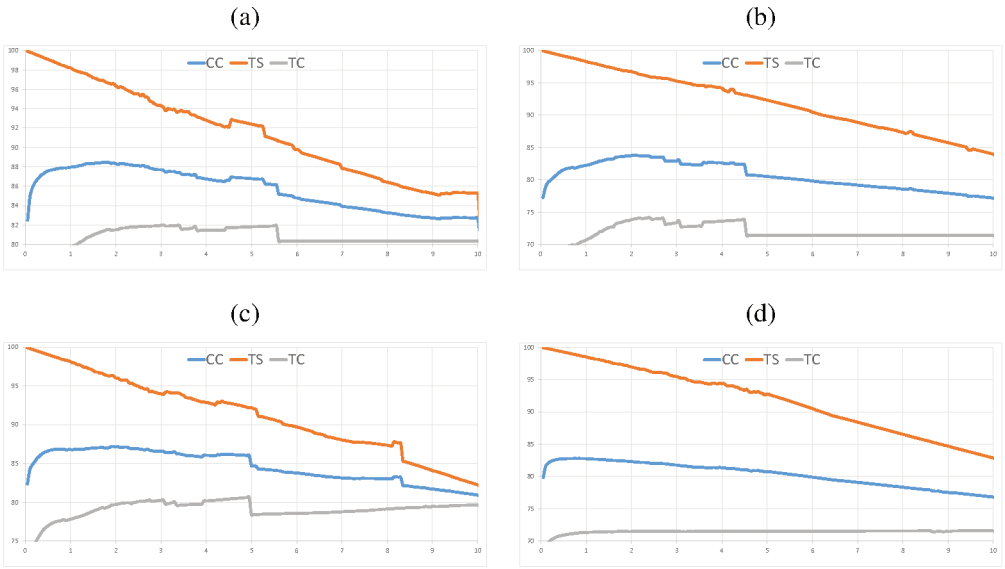
A média harmônica pode ser utilizada como critério de seleção em um processo de treinamento. Essa média considera a Taxa de Compressão (TC) e a Taxa de Erro (TE), conseguindo um equilibro entre estas. Nas Figuras 15 e 16 apresentam-se os valores de CC, Taxa de Similaridade (TS), TC com respeito aos valores do DC nos sinais de umidade e temperatura de 4 dispositivos. Estas mostram os resultados no mesmo sinal de diferentes configurações, e sua representação em termos do erro e compressão, além de mostrar os resultados do CC nos diferentes cenários.

Figura 15 – Valores em porcentagem do CC, TS e TC respeito ao valor do DC nas amostras de umidade de quatro dispositivos do (MADDEN et al., 2004)



Fonte – O autor

Figura 16 – Valores em porcentagem do CC, TS e TC respeito ao valor do DC nas amostras de temperatura de quatro dispositivos do (MADDEN et al., 2004)



Fonte – O autor

5.4 SWINGING DOOR TRENDING

Nesta seção apresenta-se os resultados das modificações propostas do SDT. Dois estudos de caso são utilizados. O LAPESD, do ambiente de testes realizado no Laboratório de Pesquisa de Sistemas Distribuidos (LaPeSD) (Seção 5.1), e o Intel Lab Data (MADDEN et al., 2004), utilizando os dados coletados por 10 dispositivos que foram posicionados no Laboratório de pesquisa Intel Berkeley (Seção 5.2).

5.4.1 Self-Definition Swinging Door Trending

O *Self-definition Swinging Door Trending* (SSDT) propõe utilizar métricas simples para decidir o valor inicial e o ajuste do Desvio da Compressão (DC). Nesta dissertação, essa modificação é comparada com o *Adaptive Swinging Door Trending* (ASDT). Essa proposta tem o mesmo objetivo: utilizar o Média Móvel exponencial (MME) (Equações 5.1 e 5.2) para escolher o valor do DC, utilizando as amostras coletadas. O valor do DC é decidido utilizando a Equação 5.3. O γ assume valores entre [0,1], o que representa o percentual de influência de S_n no DC.

$$S_n = (X + S_{n-1})\alpha + S_{n-1} \tag{5.1}$$

$$\alpha = \frac{2}{N+1} \tag{5.2}$$

$$DC = \gamma \cdot S_n \tag{5.3}$$

5.4.1.1 *Estudo de caso: LAPESD*

As Tabelas 12 e 13 mostram os resultados das compressões nos sinais de umidade e temperatura, respectivamente. O primeiro valor do DC é escolhido utilizando 100 amostras. O Média Móvel exponencial (MME) apresenta o pior resultado ao utilizar a diferença entre o ponto atual e o anterior. A média, *range* e Zmean apresentaram melhores resultados que o *Adaptive Swinging Door Trending* (ASDT), exceto o Zmean no Lapesd 2. Nas amostras de temperatura, o *range* e Zmean conseguem resultados menores de TE com valores maiores no TC. No caso dos dados de temperatura no lapesd 1, conseguem-se altos níveis de TC, porém duplicando o TE na média, e o *range* e incrementando-o 40% no Zmean. Ainda, os resultados não representam uma melhora considerável comparando com o ASDT. Conseguem-se bons resultados considerando que não se utiliza um limitante de influencia da métrica como no ASDT.

Tabela 12 – Resultados das compressões nos sinais de umidade utilizando o SSDT e o ASDT com 100 amostras de treinamento inicial

	Média		***Range***		**Zmean**		**MME**		**ASDT** $\gamma = 0.05$		**ASDT** $\gamma = 0.1$	
Dispositivo	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**
Lapesd 1	5.92	99.83	4.84	99.64	3.8	98.3	1.49	66.43	2.52	86.17	4.92	95.15
Lapesd 2	4.79	99.85	5.09	99.89	3.24	99.5	0.62	72.91	2.39	97.01	4.05	99.36

Fonte – O autor.

Tabela 13 – Resultados das compressões nos sinais de temperatura utilizando o SSDT e o ASDT com 100 amostras de treinamento inicial

	Média		***Range***		**Zmean**		**MME**		**ASDT** $\gamma = 0.05$		**ASDT** $\gamma = 0.1$	
Dispositivo	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**
Lapesd 1	13.69	99.82	15.23	99.78	9.64	96.52	2.5	76.77	3	79.36	6.78	85.85
Lapesd 2	5.47	99.84	5.39	99.89	4.27	99.41	0.73	80.32	2.6	97.4	4.44	99.51

Fonte – O autor.

5.4.1.2 *Caso de estudo: Intel Lab Data*

As tabelas 14 e 15 mostram os resultados das compressões nos sinais de umidade e temperatura, respectivamente. O primeiro valor do DC é escolhido utilizando 100 amostras. O MME apresenta o pior resultado ao utilizar a diferença entre o ponto atual e anterior. Nas amostras de umidade a média e o *range* conseguiram melhores resultados de TC, porém em alguns casos o TE consegue valores maiores ao 10%. No caso do Zmean os resultados do TC são ligeiramente menores ao *Adaptive Swinging Door Trending* (ASDT), embora tenha melhores resultados de erro que a média e o *range*. Nas amostras de temperatura os resultados

da média, *range* e zmean em termos de TC são na maioria dos casos melhores, porém o nível de erro é muito alto chegando perto de 90%, em alguns casos.

Tabela 14 – Resultados dos sinais de umidade utilizando o SSDT e o ASDT com 100 amostras de treinamento inicial

	Média		***Range***		**Zmean**		**MME**		**ASDT** $\gamma = 0.05$		**ASDT** $\gamma = 0.1$	
Dispositivo	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**
Intel 1	6.74	98.62	26.6	99.53	5.24	93.69	2.58	65.97	2.9	91.88	5.7	94.92
Intel 2	7.49	98.48	18.02	99.05	5.11	91.36	3.83	63.87	2.94	91.66	5.43	94.84
Intel 3	9.9	98.56	6.35	98.47	4.38	89.08	5.15	69.53	3.06	90.11	5.98	94.04
Intel 4	10.53	98.39	9.09	98.17	4.09	90.25	3.1	67.53	2.88	94.71	5.76	95.78
Intel 5	1.42	98.49	6.68	99.5	0.29	94.41	0.09	62.48	2.78	99.29	5.28	99.49
Intel 6	6.26	98.18	14.02	99.03	3.78	95.8	0.51	67.07	2.96	96.76	6.21	98.96
Intel 7	13.87	98.37	10.96	98.98	2.6	93.02	2.36	71.67	2.78	96.85	5.85	97.79
Intel 8	10.74	98.72	14.74	99	7.47	94.58	2.75	66.02	2.88	93.66	5.16	95.73
Intel 9	6.36	97.41	10.54	97.79	4.08	87.34	3.62	64.1	3.05	87.86	6.19	91.42
Intel 10	5.34	97.32	9.97	98.66	3.43	89.35	2.5	62.44	2.63	94.36	5.37	95.67

Fonte – O autor.

Tabela 15 – Resultados dos sinais de temperatura utilizando o SSDT e o ASDT com 100 amostras de treinamento inicial

	Média		***Range***		**Zmean**		**MME**		**ASDT** $\gamma = 0.05$		**ASDT** $\gamma = 0.1$	
Dispositivo	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**
Intel 1	42.49	99.11	82.97	97.11	44.53	98.14	46.77	84.27	3.5	79.68	7.15	79.71
Intel 2	60.48	99.28	90.68	95.93	13.56	95.1	53.38	75.93	3.38	70.71	6.74	71
Intel 3	55.1	99.4	89.61	99.56	50.62	97.37	41.72	68.67	3.71	77.76	7.14	77.07
Intel 4	55.22	98.78	54	96.44	48.88	93.87	44.83	63.33	3.39	71.21	7.34	71.26
Intel 5	72.35	98.79	80	96.37	79.55	92.89	52.32	73.97	3.29	68.22	7.29	68.35
Intel 6	27.18	93.69	69.77	97.29	9.5	92.15	3.8	73.09	3.26	95.47	7.68	98.25
Intel 7	18.09	96.77	48.12	95.68	44.04	85.49	44.14	62.72	3.48	68.66	7.34	68.57
Intel 8	54.42	99.12	86.42	98	25.36	92.46	42.39	82.76	3.11	74.29	6.55	75.04
Intel 9	18.91	96.26	40.78	94.63	31.75	80.13	38.19	63.76	3.7	79.64	7.22	78.05
Intel 10	79.66	99.32	74.09	97.27	57.12	97.92	48.28	82.98	3.17	73.86	6.31	74.01

Fonte – O autor.

5.4.1.3 Analises de resultados

Utilizar a diferença entre duas amostras é uma métrica aceitável. Porém, quando o sinal tem mudanças abruptas, o resultado do DC é altamente afetado, o que pode gerar uma compressão com TE alto. Nos dados do Lapesd, os resultados obtidos pelo *Self-definition Swinging Door Trending* (SSDT) são melhores, na maioria dos casos. Porém, nos dados da Intel, o TE final é muito elevado. Em alguns casos, a diferença entre o valor atual e o anterior é maior que 100 °C na temperatura e 50% em umidade. O ASDT consegue controlar melhor o efeito destes dados atípicos pelo γ que limita o efeito que a métrica tem na tolerância do SDT.

Existem duas potenciais soluções para evitar este problema: (i) primeiro utilizar uma lógica semelhante ao ASDT, e utilizar um parâmetro γ para evitar um efeito incontrolável, quando o sinal for instável, ou (ii) adicionar algum mecanismo de detecção de *outliers*.

5.4.2 Training Swinging Door Trending

O critério de seleção CC utilizado no TSDT é comparado com o *Sensor Manufacture Error* (SME) (KOLO et al., 2016). Este utiliza a precisão do sensor definida pelo fabricante na folha de dados do sensor para definir a tolerância de erro. A base de dados apresentada na Seção 5.2 é utilizada como estudo de caso. A Tabela 10 apresenta a informação necessária dos sensores.

5.4.2.1 Caso de Estudo: LAPESD

Na Tabela 16 encontram-se os resultados do SDT utilizando o critério de decisão do *Sensor Manufacture Error* (SME), e na Tabela 17 encontram-se os resultados utilizando o CC como critério de seleção. Os resultados com o critério SME são adequados e conseguem uma boa compressão, além de um nível de erro aceitável, exceto na temperatura do Lapesd 1. O TSDT escolhe valores adequados, além de conseguir melhores resultados de compressão no caso da temperatura do Lapesd 1. O nível de erro neste caso é alto pela instabilidade do sinal, porém o valor do CC é mais alto. Caso o CC fosse usado como critério de seleção, o aumento no erro é compensando pelo aumento considerável no nível de compressão. A Tabela 18 apresenta o valor do DC escolhido e o CC resultante. As figuras 17 e 18 apresentam uma parte do sinal comprimida dos dados de temperatura e umidade do Lapesd 1. Nestas figuras apresentam-se os resultados utilizando diferentes valores do DC. Os resultados utilizando o CC como métrica e 1000 amostras de treinamento apresentam resultados que segue com melhor detalhe o comportamento do sinal.

Tabela 16 – Resultados no SDT utilizando o critério do SME em dados de umidade (DC=5) e Temperatura (DC=2)

Dispositivos	**Umidade**		**Temperatura**	
	TE	**TC**	**TE**	**TC**
Lapesd 1	3.83	96.76	5.26	83.31
Lapesd 2	3.22	99.63	3.53	99

Fonte – O autor.

Tabela 17 – Resultados do TSDT com $\beta = 1.0$ em mostras de umidade ($\delta_{max} = 15, \alpha = 0.01$) e temperatura ($\delta_{max} = 15, \beta = 0.01$) utilizando 100 e 1000 amostras de treinamento

	Umidade				**Temperatura**			
TE-TC	**100**		**1000**		**100**		**1000**	
Dispositivo	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**
Lapesd 1	3.92	97.19	4.63	99.14	9.9	96.61	9.51	97.47
Lapesd 2	2.91	99	3.91	99.9	4.21	99.82	6.64	99.98

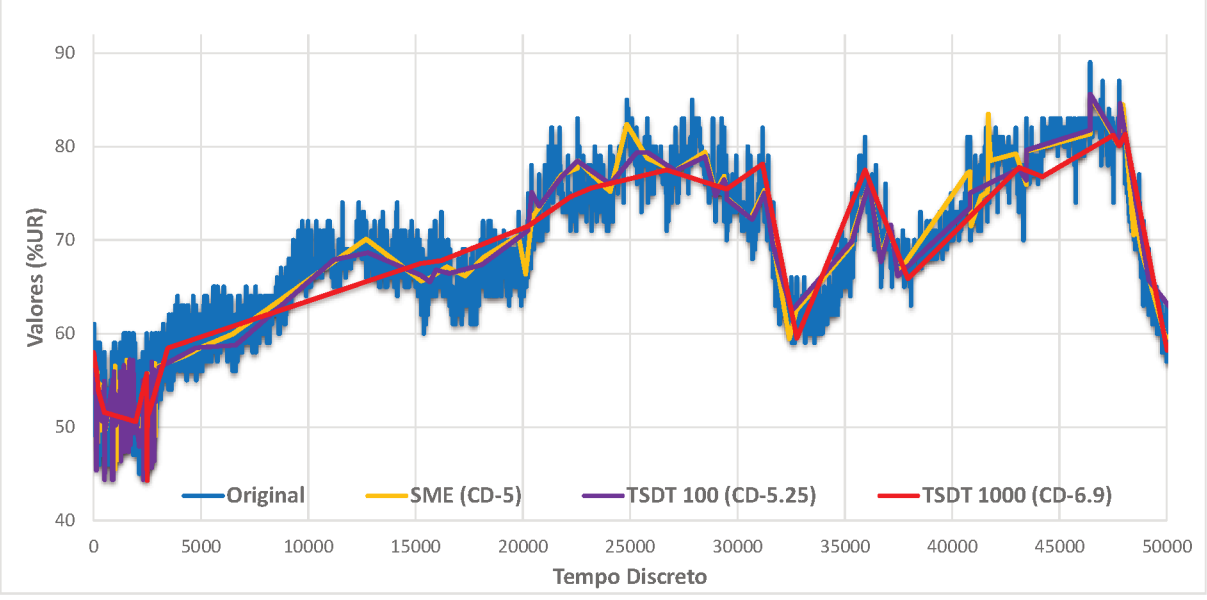
Fonte – O autor.

Tabela 18 – DC e CC no TSDT ($\beta = 1.0$) nos dados do estudo de caso LAPESD

Dispositivos	**Umidade**				**Temperatura**			
	100		**1000**		**100**		**1000**	
	DC	**CC**	**DC**	**CC**	**DC**	**CC**	**DC**	**CC**
Lapesd 1	5.25	97.23	6.9	96	4.6	96.71	4.91	98.58
Lapesd 2	3.85	95.77	6.35	96.45	3.6	96.71	5.9	98.71

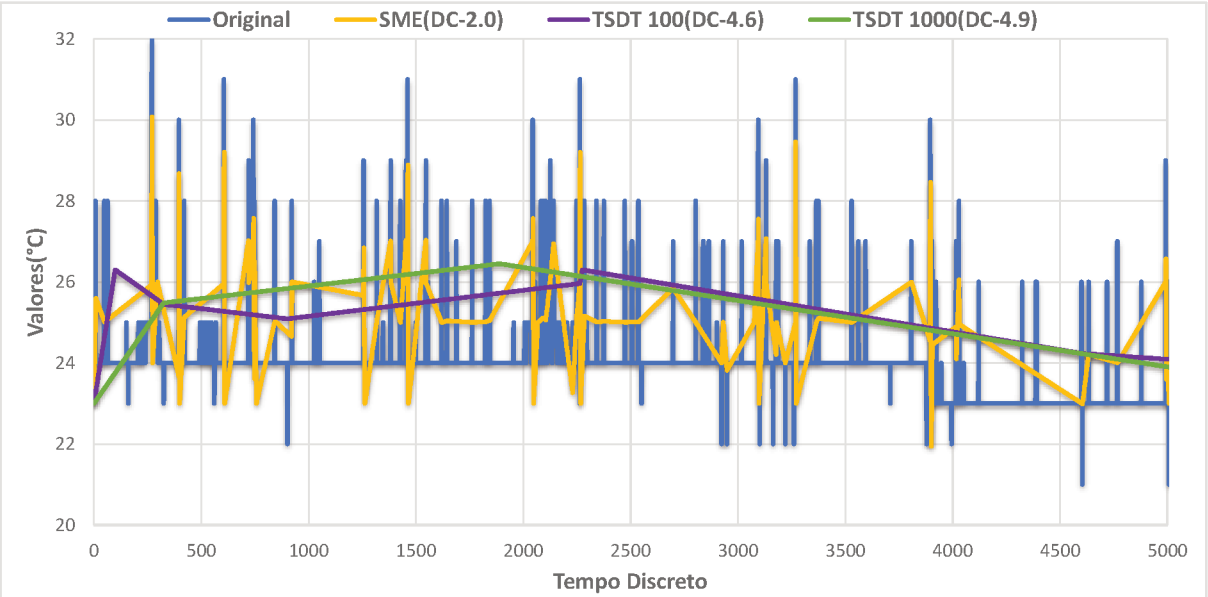
Fonte – O autor.

Figura 17 – Compressão de 50000 amostras de umidade do Dispositivo Lapesd 1



Fonte – O autor.

Figura 18 – Compressão de 5000 amostras de temperatura do Dispositivo Lapesd 1



Fonte – O autor.

5.4.2.2 *Caso de estudo: Intel Lab Data*

Na Tabela 19 encontram-se os resultados do SDT utilizando o critério de decisão do *Sensor Manufacture Error* (SME) e a Tabela 20 mostra os resultados da compressão utilizando o *Training Swinging Door Trending* (TSDT) com três valores diferentes de β. Quando o $\beta = 0.01$, a TC tem uma maior relevância, o que permite a seleção de valores de DC com TE maiores. O contrário acontece com $\beta = 10.0$, tendo maior relevância o TE. Comparando os resultados com o SME e o TSDT com $\beta = 1.0$, nas amostras de umidade, o SME conta com melhores resultados em termos de TC, porém todos seus TE estão acima de 4%, na seleção de DC do TSDT, os valores do TC são próximos aos do SME, além de haver um erro muito menor.

No caso da temperatura, o DC e o TE são pequenas, e pelo comportamento instável do sinal (conta com muitas amostras atípicas) o TC não retorna um bom resultado. O TSDT consegue resultados similares. Aumentar o DC para incrementar a TC não oferece uma ganho que valha o aumento no TE. O aumento do TC consegue-se com valores do δ_{max} altos, porém aumenta consideravelmente o TE.

Uma maior quantidade de amostras de treinamento melhora os resultados, porém requer um maior tempo de transmissão do dispositivo e de execução no servidor. O único parâmetro no começo era o α, porém na ausência de uma limitante do valor α, o algoritmo executa até que todos os valores de treinamento estiveram dentro da janela do SDT, o que causava um tempo de execução desconhecido que dependia do comportamento da sinal. O δ_{max} limita as iterações do algoritmo, além de permitir conhecer as execuções necessárias, valores maiores no DC depois de valores muitos altos não conseguem bons resultados de compressão, estes

Tabela 19 – Resultados no SDT utilizando o critério do SME em dados de umidade (DC=3.0) e Temperatura (DC=0.4)

Dispositivos	Umidade		Temperatura	
	TE	**TC**	**TE**	**TC**
Intel 1	4.68	93.54	0.73	77.87
Intel 2	4.28	93.79	0.68	68.49
Intel 3	4.53	92.27	0.77	76.44
Intel 4	4.54	95.66	0.58	70.77
Intel 5	4.12	99.8	1.1	67.89
Intel 6	5.41	99.03	1.24	95.18
Intel 7	4.39	97.58	0.57	68.37
Intel 8	4.42	94.84	0.67	75.7
Intel 9	4.4	89.82	0.78	77.22
Intel 10	4.14	95.37	0.66	73.31

Fonte – O autor.

aumentam o valor da TE sobre o crescimento da TC

Os valores do DC atribuídos pelo TSDT apresentam-se na Tabela 21. As figuras 19 e 20 mostram os resultados das compressões nos sinais de umidade e temperatura do Intel 3, no caso da umidade, o DC selecionado com 100 amostras e o SME representam a tendência geral das 200 amostras apresentadas. O resultado com 1000 amostras segue melhor a tendência do sinal, representando as mudanças significativas. No sinal de temperatura o SME representa a linha de tendência geral dos dados e o DC de 100 representa qualquer mudança. O DC selecionado com 1000 representa o sinal com linhas de tendência que representam as mudanças significativas. Uma boa compressão depende das necessidades da solução, porém o objetivo deste trabalho é atingir o equilíbrio entre a representação dos dados e a qualidade dos mesmos.

5.4.2.3 Análise de resultados

O CC funciona adequadamente como critério de seleção do valor inicial do DC, pois permite escolher uma tolerância que não requer limitar o erro máximo ou obrigar a cumprir um nível de compressão. Na literatura, encontraram-se propostas que priorizam uns dos níveis relacionados com a compressão com perda, neste caso se escolhe um valor equilibrado, com o qual se alcance um bom resultado de TC sem aumentar sem controle o TE

5.5 CONCLUSÕES DO CAPÍTULO

Este capítulo buscou avaliar a proposta sob o ponto de vista de diferentes estudos de casos, diferentes métricas e diferentes valores de média. Através dos resultados obtidos, ficou nítido que existe uma solução de compromisso na escolha de parâmetros de compressão. A escolha de um determinado parâmetro pode aumentar o desempenho com relação a uma deter-

Tabela 20 – Resultados do TSDT com diferentes valores de β em mostras de umidade ($\delta_{max} = 15, \alpha = 0.01$) e temperatura ($\delta_{max} = 15, \beta = 0.01$) utilizando 100 e 1000 amostras de treinamento

	Umidade				**Temperatura**			
β 0.1	**100**		**1000**		**100**		**1000**	
Dispositivo	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**
Intel 1	2.62	91.32	7.42	96.76	0.94	78.45	6.69	81.7
Intel 2	12.16	98.81	7.24	96.46	0.08	62.97	4.82	72.72
Intel 3	2.77	89.58	5.98	93.58	0.11	70.99	3.71	79.66
Intel 4	1.18	93.93	3.62	95.36	4.52	71.5	3.31	71.49
Intel 5	3.49	99.76	4.12	99.8	2.65	68.43	6.74	68.63
Intel 6	1.87	96.09	4.46	98.67	0.86	95.03	8.56	99.47
Intel 7	1.99	96.29	3.9	97.43	0.58	68.32	3.75	69.31
Intel 8	9	97.29	9.58	97.63	0.52	75.5	4.11	76.92
Intel 9	11.9	97.11	12.61	97.27	4.85	81.24	6.17	81.59
Intel 10	2.12	94.11	4.67	95.58	0.15	69.72	4.06	74.3
β 1.0	**100**		**1000**		**100**		**1000**	
Dispositivo	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**
Intel 1	2.62	91.32	0.96	89.18	0.1	71.35	0.38	76.34
Intel 2	0.76	87.33	4.28	93.79	0.08	62.97	1.15	69.91
Intel 3	2.77	89.58	0.82	86.28	0.11	70.99	0.66	76.02
Intel 4	1.18	93.93	0.69	93.19	0.1	67.83	0.24	69.81
Intel 5	3.49	99.76	0.93	99.02	0.21	65.42	0.58	67.2
Intel 6	1.87	96.09	0.49	94.11	0.86	95.03	0.45	94.24
Intel 7	1.99	96.29	1.19	95.5	0.58	68.32	0.21	67.83
Intel 8	1.04	92.64	1.56	93.11	0.52	75.5	0.23	74.81
Intel 9	4.19	89.44	1.79	86.66	0.83	77.4	1.01	78
Intel 10	2.12	94.11	1.09	92.3	0.15	69.72	0.34	72.29
β 10.0	**100**		**1000**		**100**		**1000**	
Dispositivo	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**	**TE**	**TC**
Intel 1	0.22	81.26	0.22	81.26	0.04	61.61	0.07	68.63
Intel 2	0.22	79.74	0.27	82.96	0.03	56.95	0.08	62.97
Intel 3	0.16	73.69	0.22	79.39	0.06	66.21	0.07	68.6
Intel 4	0.16	78.25	0.22	84.78	0.06	65.35	0.06	65.35
Intel 5	0.48	97.59	0.15	86.87	0.08	61.43	0.11	63.11
Intel 6	0.15	86.74	0.15	86.74	0.06	83.47	0.08	87.95
Intel 7	0.27	90.3	0.21	87.94	0.11	67.05	0.06	65.28
Intel 8	0.15	78.1	0.21	83.51	0.11	73.54	0.05	69.55
Intel 9	0.22	75.01	0.22	75.01	0.1	68.96	0.1	68.96
Intel 10	0.34	84.68	0.28	81.51	0.05	61.3	0.09	66.06

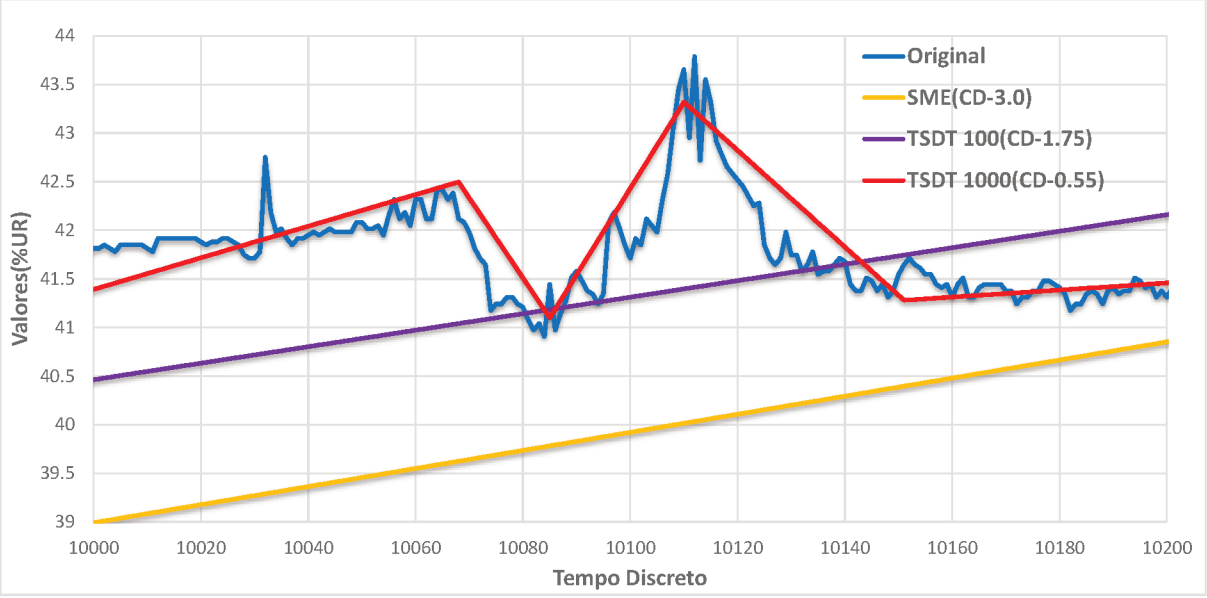
Fonte – O autor.

Tabela 21 – DC e CC no TSDT ($\beta = 1.0$)

Dispositivos	Umidade				Temperatura			
	100		1000		100		1000	
	DC	CC	DC	CC	DC	CC	DC	CC
Intel 1	1.7	94.25	0.65	93.85	0.06	95.63	0.22	98.93
Intel 2	0.55	92.91	3	94.75	0.05	95.15	0.69	97.11
Intel 3	1.75	93.25	0.55	92.28	0.06	91.89	0.35	98.46
Intel 4	0.85	96.31	0.5	96.15	0.07	97.45	0.17	99.36
Intel 5	2.5	98.1	0.65	99.05	0.08	98.14	0.22	99.25
Intel 6	1.2	97.1	0.35	96.73	0.29	98.54	0.16	99.41
Intel 7	1.35	97.14	0.9	97.13	0.41	98.46	0.15	99.29
Intel 8	0.8	95.69	1.15	95.7	0.31	93.39	0.14	98.62
Intel 9	2.75	92.51	1.25	92.08	0.42	96.88	0.51	98.92
Intel 10	1.65	95.96	0.85	95.49	0.09	93.91	0.21	98.86

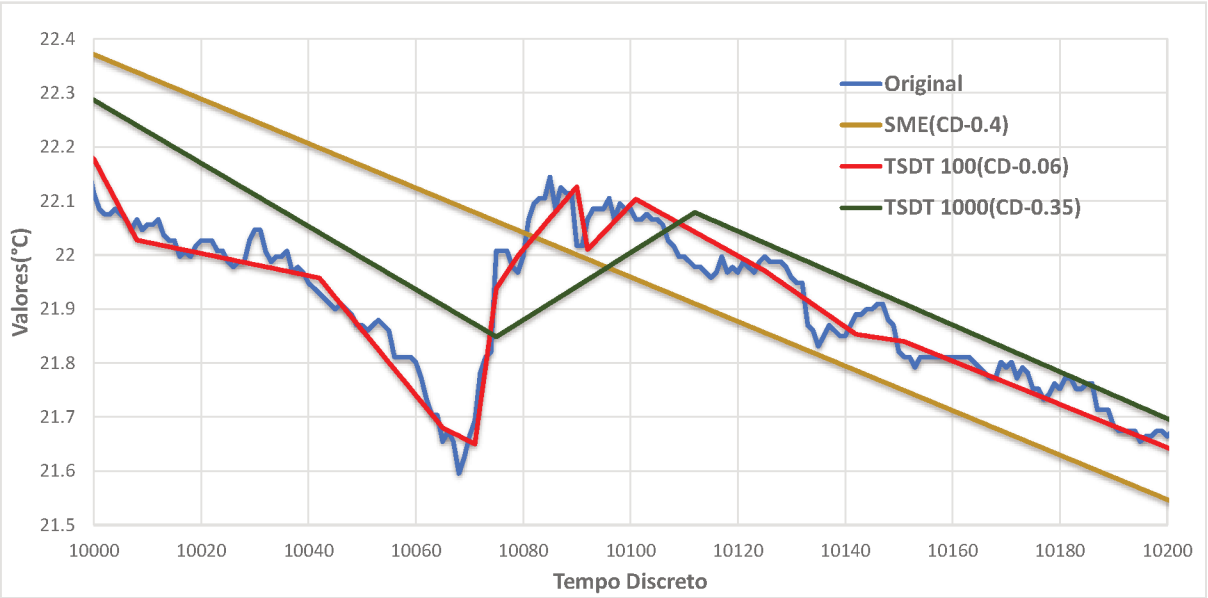
Fonte – O autor.

Figura 19 – Compressão de 200 amostras de umidade do Intel 3



Fonte – O autor.

Figura 20 – Compressão de 200 amostras de temperatura do Intel 3



Fonte – O autor.

minada métrica, mas pode degradar outra métrica. O Critério de Compressão (CC), proposto neste trabalho de dissertação, demostrou que pode ser uma métrica adequada para a escolha de um valor inicial do DC, após uma análise prévia de amostras de dados de sensores.

6 CONCLUSÕES E TRABALHOS FUTUROS

A *Internet of Things* (IoT) é uma visão que permite melhorar e criar produtos e serviços. Os dispositivos da *Internet of Things* (IoT) tendem a ter limitações computacionais, energéticas e de armazenamento, tornando necessária a optimização da utilização de seus recursos. A transmissão de dados coletados desde sensores é uma tarefa frequente para o monitoramento do ambiente, ou como fonte de amostras necessárias para decidir as ações de controle desde o servidor ou desde dispositivos com maior capacidades computacionais ou energéticas.

A redução da utilização do sistema de comunicação aumenta o tempo de vida do dispositivo, os mecanismos de Compressão de Dados (CD) reduzem a quantidade de dados que precisam ser transmitidos, o tempo de transmissão e a quantidade de pacotes. Um ponto forte de Compressão de Dados (CD) é a capacidade de reduzir a utilização do sistema de comunicação sem desconhecer o estado dos sinais coletados pelo sensor.

Existem dois tipos de métodos de Compressão de Dados (CD), os sem perda, que permitem realizar uma compressão completamente reversível, e os com perda que resultam em uma aproximação dos dados originais. Os algoritmos com perda tendem a ter uma menor complexidade computacional e conseguir uma compressão maior que os sem perda.

Os métodos de Compressão de Dados (CD) com perda são adequados para aplicações que aceitam um certo grau de erro ou para comprimir sinais com amostras redundantes que podem ser eliminadas ou representadas de uma maneira comprimida. Já os algoritmos com perda são adequados para aplicações com dados de sensores, porém faz-se necessário se considerar os requisitos funcionais e não funcionais da aplicação.

Os algoritmos de Compressão de Dados (CD) com perda contam com vários mecanismos para realizar a compressão. O mais comum é utilizar um nível de tolerância e realizar uma representação no domínio do tempo do sinal por meio de algum método de interpolação ou utilizando um algoritmo de codificação para representar as mudanças relevantes do sinal. A transformação dos dados a um domínio diferente, que requeira menor volume de dados (menos coeficientes ou amostras) ou algoritmos de Inteligência artificial (IA) que conseguem comprimir os dados, são utilizados também para reduzir a quantidade de dados que precisam ser transmitidos.

A compressão dos dados coletados de sensores reduz os custos de armazenamento de dados em sistemas de armazenamento, um algoritmo adequado pode conseguir uma redução considerável dos dados armazenados.

Os algoritmos de Compressão de Dados (CD) precisam ser avaliados considerando o nível de compressão alcançada e o erro provocado. Métricas de desempenho e avaliação como as apresentadas na Subseção 2.2.1 são utilizadas para representar os resultados da compressão. Ao utilizar algoritmos de compressão em dispositivos da *Internet of Things* (IoT) é importante considerar a complexidade computacional dos mesmos. Já no caso de aplicações de tempo real, é preciso considerar o tempo de execução no dispositivo, e avaliar sua viabilidade para cumprir os prazos e períodos estabelecidos.

Os algoritmos de Compressão de Dados (CD) com perda existentes podem ser melhorados utilizando um método de ajuste considerando o comportamento do sinal, melhorando a seleção dos parâmetros de configuração ou monitorando o efeito da compressão nos dados. Porém, essa melhoria pode aumentar a complexidade computacional, o que pode inviabilizar o uso do algoritmo em dispositivos da *Internet of Things* (IoT).

Neste trabalho foram apresentadas duas modificações do *Swinging Door Trending* (SDT) para utilizar-se nos dispositivos da *Internet of Things* (IoT). O *Self-definition Swinging Door Trending* (SSDT) que utiliza a diferença entre duas amostras consecutivas para ajustar o Desvio da Compressão (DC) no dispositivo, e o *Training Swinging Door Trending* (TSDT) que utiliza uma métrica de decisão para escolher um valor adequado do Desvio da Compressão (DC) para a compressão de dados. As melhorias do *Swinging Door Trending* (SDT) propostas atingem bons resultados de compressão, porém o *Self-definition Swinging Door Trending* (SSDT) é afetado pelos dados atípicos ou as mudanças grandes no sinal, o que pode baixar sua viabilidade. O Critério de Compressão (CC) é proposto como métrica de avaliação da compressão considerando a Taxa de Compressão (TC) e a Taxa de Erro (TE). Esta representa adequadamente a qualidade da compressão resultante. O Critério de Compressão (CC) permite modificar o nível de relevância da Taxa de Compressão (TC) ou a Taxa de Erro (TE), permitindo modificar o critério de qualidade dependendo das necessidades do sistema. O Critério de Compressão (CC) funciona adequadamente como critério de decisão do Desvio da Compressão (DC) no *Swinging Door Trending* (SDT), conseguindo um bom nível de custo benefício entre a Taxa de Compressão (TC) resultante e a Taxa de Erro (TE) causado. A limitação da Taxa de Erro (TE) ou a Taxa de Compressão (TC) como método pode limitar o potencial de compressão do sistema se não for considerado o comportamento dos sinais a comprimir.

As propostas de ajuste e definição de parâmetros que resultaram no *Self-definition Swinging Door Trending* (SSDT) e o *Training Swinging Door Trending* (TSDT) contam com uma complexidade computacional constante , tornando-as previsíveis e adequadas para ambientes de dispositivos da IoT. Entre as limitações da proposta encontram-se o efeito que geram os dados atípicos, e a perda dos dados originais necessários para avaliar a compressão.

Nas contribuições desta dissertação encontra-se uma revisão das propostas atuais encontradas na literatura e dos conceitos necessários para compreender os algoritmos de compressão com perda. Esta revisão resultou em uma síntese dos conceitos, desafios e mecanismos de compressão atuais.

Nos artigos, patentes e dissertações encontraram-se algumas dificuldades, principalmente a falta de informação sobre os algoritmos propostos e utilizados. Esta falta de informação limitada a avaliação da complexidade computacional de algumas propostas. Outra dificuldade foi um erro encontrado na patente do *Swinging Door Trending* (SDT), o que requereu um análise do processo lógico utilizado para criar este algoritmo.

Os resultados da dissertação apresentam como um importante trabalho futuro o controle dos dados atípicos ou *outliers*. Estes reduzem o Taxa de Compressão (TC) conseguido e aumentam o Taxa de Erro (TE), a utilização de mecanismos de detecção de *outliers* poderia

aumentar os níveis de compressão conseguidos ao utilizar-se com algoritmos de Compressão de Dados (CD). Outra possibilidade é a utilização de métricas de avaliação nos dispositivos para conseguir um ajuste adequado dos parâmetros com relação ao estado atual do sinal. A utilização de métodos de compressão sem perda no nodo de coleção ou servidor ajudaria a reduzir mais o espaço ocupado no sistema de armazenamento.

A Compressão de Dados (CD) tem um potencial de utilização grande em áreas de pesquisa relacionadas com dispositivos que têm limitações energéticas consideráveis. Os mecanismos de Compressão de Dados (CD) permitem otimizar o uso dos recursos energéticos ao reduzir a utilização do canal de comunicação . A agricultura de alta precisão, os veículos aéreos não tripulados, o monitoramento ambiental e climático são algumas das áreas que podem ser beneficiadas.

REFERÊNCIAS

Abu Alsheikh, M. et al. Rate-distortion balanced data compression for wireless sensor networks. **IEEE Sensors Journal**, v. 16, n. 12, p. 5072–5083, 06 2016. ISSN 1530-437X.

ADAFRUIT. **DHT11 Humidity and Temperature Sensor**. Adafruit, 2019. Disponível em: https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf.

AI-THINKER. **ESP-12E WiFi Module**. Ai-Thinker, 2015. V1.0. Disponível em: https://www.kloppenborg.net/images/blog/esp8266/esp8266-esp12e-specs.pdf.

ALSALAET, J. K.; ALI, A. A. Data compression in wireless sensors network using mdct and embedded harmonic coding. **ISA Transactions**, v. 56, p. 261 – 267, 2015. ISSN 0019-0578. Disponível em: http://www.sciencedirect.com/science/article/pii/S0019057814003012.

AZAR, J. et al. An energy efficient iot data compression approach for edge machine learning. **Future Generation Computer Systems**, v. 96, p. 168 – 175, 2019. ISSN 0167-739X.

AZAR, J. et al. An energy efficient iot data compression approach for edge machine learning. **Future Generation Computer Systems**, v. 96, p. 168 – 175, 2019. ISSN 0167-739X. Disponível em: http://www.sciencedirect.com/science/article/pii/S0167739X18331716.

BANSAL, M.; GIMPEL, K.; LIVESCU, K. Tailoring continuous word representations for dependency parsing. In: . Association for Computational Linguistics (ACL), 2014. v. 2, p. 809–815. ISBN 9781937284732. Cited By 134. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84906933205&partnerID=40&md5=2d7d6a96bff9f81f1562924967f28d9e.

BELLAVISTA, P. et al. A survey on fog computing for the internet of things. **Pervasive and Mobile Computing**, v. 52, p. 71 – 99, 2019. ISSN 1574-1192. Disponível em: http://www.sciencedirect.com/science/article/pii/S1574119218301111.

BISWAS, A.; GIAFFREDA, R. Iot and cloud convergence: Opportunities and challenges. In: . IEEE Computer Society, 2014. p. 375–376. Cited By 88. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84900405250&doi=10.1109\\%2fWF-IoT.2014.6803194&partnerID=40&md5=68714d4bfd6e3688e3bc94af8eef19ef.

BOSE, T. et al. Signal characteristics on sensor data compression in iot - an investigation. In: . Institute of Electrical and Electronics Engineers Inc., 2016. ISBN 9781509024292. Cited By 2. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85004006965&doi=10.1109\\%2fSECONW.2016.7746810&partnerID=40&md5=6798a134c7cc94b48b62b75bfa5b873e.

BOTTA, A. et al. Integration of cloud computing and internet of things: A survey. **Future Generation Computer Systems**, v. 56, p. 684 – 700, 2016. ISSN 0167-739X. Disponível em: http://www.sciencedirect.com/science/article/pii/S0167739X15003015.

BRISTOL, E. Swinging door trending. adaptive trend recording? In: ANON (Ed.). Publ by ISA Services Inc, Research Triangle Pk, NC, United States, 1990. v. 45, n. pt 2, p. 749–754. ISSN 00652814. Cited By 59. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0025544216&partnerID=40&md5=e04a6589fe3eb169e6fdf80e2c473472.

BRISTOL, E. H. **DATA COMPRESSION FOR DISPLAY AND STORAGE**. 1987. US Patent 4,669,097.

CAPO-CHICHI, E.; GUYENNET, H.; FRIEDT, J.-M. K-rle: A new data compression algorithm for wireless sensor network. In: . SENSORCOMM, 2009. p. 502–507. ISBN 9780769536699. Cited By 43. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-70449478172&doi=10.1109\\%2fSENSORCOMM.2009.84&partnerID=40&md5=051673db0396e679b0a6e10483a1a7c7.

CHEN, F. et al. Nonthreshold-based node level algorithm of data compression over the wireless sensor networks. In: . ICSPS, 2010. v. 2, p. V2223–V2227. ISBN 9781424468911. Cited By 4. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-77957286120&doi=10.1109\\%2fICSPS.2010.5555286&partnerID=40&md5=cf98bbc3d148446962ee2af60e763743.

CHEN, S. et al. A hierarchical adaptive spatio-temporal data compression scheme for wireless sensor networks. **Wireless Networks**, v. 25, n. 1, p. 429–438, 01 2019. ISSN 1572-8196. Disponível em: https://doi.org/10.1007/s11276-017-1570-6.

CHEN, Y.-H. et al. Dynamic bounded-error data compression and aggregation in wireless sensor network. In: . IEEE, 2012. ISBN 9781457717659. Cited By 2. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84873972824&doi=10.1109\\%2fICSENS.2012.6411224&partnerID=40&md5=4b12411fee31febd6facc0ecb27f2a55.

COETZEE, L.; EKSTEEN, J. The internet of things - promise for the future? an introduction. In: **2011 IST-Africa Conference Proceedings**. Washington, USA: IEEE, 2011. p. 1–9.

CORMEN, T. H. et al. **Introduction to Algorithms, Third Edition**. 3rd. ed. 1 Duchess St, Marylebone, London W1W 6AN, Reino Unido: The MIT Press, 2009. ISBN 0262033844, 9780262033848.

CORREA, J. D. A. et al. Swinging door trending compression algorithm for iot environments. In: **Anais do IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais**. Porto Alegre, RS, Brasil: SBC, 2019. p. 143–148. Disponível em: https://sol.sbc.org.br/index.php/sbesc_estendido/article/view/8650.

DI, S.; CAPPELLO, F. Fast error-bounded lossy hpc data compression with sz. In: . Institute of Electrical and Electronics Engineers Inc., 2016. p. 730–739. ISBN 9781509021406. Cited By 85. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84983332081&doi=10.1109\\%2fIPDPS.2016.11&partnerID=40&md5=e69035d1a5b32e99d277eedcc36f41c6.

DOLFUS, K.; BRAUN, T. An evaluation of compression schemes for wireless networks. In: . Washington, USA: IEEE, 2010. p. 1183–1188. ISBN 9781424472857. Cited By 11. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-79951474613&doi=10.1109\\%2fICUMT.2010.5676532&partnerID=40&md5=b881b1c4de466cf75f6f21a0091debbb.

DORSEMAINE, B. et al. Internet of things: A definition and taxonomy. In: N. AL-BEGAIN K., A.-B. K. A. (Ed.). Institute of Electrical and Electronics Engineers Inc., 2016. p. 72–77. ISBN 9781479986606. Cited By 38. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84964666301&doi=10.1109\\%2fNGMAST.2015.71&partnerID=40&md5=cb1afdeb94ca3e114ce100f7b3d2ed01.

ESPRESSIF. **ESP8266EX Datasheet**. Espressif Systems, 2018. V6.0. Disponível em: https://www.espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf.

FENG, X. et al. An improved process data compression algorithm. In: Y CAO X, G. L. X. (Ed.). IEEE, 2002. v. 3, p. 2190–2193. Cited By 9. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0036955370&partnerID=40&md5=6a3fea72bfe545a3d2e38bc1e6d9c64c.

Giorgi, G. A combined approach for real-time data compression in wireless body sensor networks. **IEEE Sensors Journal**, v. 17, n. 18, p. 6129–6135, 09 2017. ISSN 1530-437X.

GROKHOTKOV, I. **Arduino core for ESP8266 WiFi chip**. 2017. Disponível em: https://github.com/esp8266/Arduino.

GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, v. 29, n. 7, p. 1645 – 1660, 2013. ISSN 0167-739X. Disponível em: http://www.sciencedirect.com/science/article/pii/S0167739X13000241.

Harb, H.; Makhoul, A.; Abou Jaoude, C. A real-time massive data processing technique for densely distributed sensor networks. **IEEE Access**, v. 6, p. 56551–56561, 2018. ISSN 2169-3536.

HOSSEINI; MOHAMMAD. A survey of data compression algorithms and their applications. **.**, 01 2012.

Kasirajan, P.; Larsen, C.; Jagannathan, S. A new adaptive compression scheme for data aggregation in wireless sensor networks. In: **2010 IEEE Wireless Communication and Networking Conference**. Washington, USA: IEEE, 2010. p. 1–6. ISSN 1558-2612.

KITCHENHAM, B. Procedures for performing systematic reviews. **Keele, UK, Keele University**, v. 33, n. 2004, p. 1–26, 2004.

KOLO, J. et al. Energy-efficient adaptive data compression in wireless sensor networks. **International Journal of Sensor Networks**, v. 22, p. 229, 01 2016.

KOTHA, H.; TUMMANAPALLY, M.; UPADHYAY, V. Review on lossless compression techniques. In: . Institute of Physics Publishing, 2019. v. 1228, n. 1. ISSN 17426588. Cited By 0. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85067804395&doi=10.1088\\%2f1742-6596\\%2f1228\\%2f1\\%2f012007&partnerID=40&md5=b75109d53c6ff897d0c49f04a2c85a9d.

LEãO, E.; GUEDES, L.; VASQUES, F. An event-triggered smart sensor network architecture. In: . IEEE, 2007. v. 1, p. 523–528. ISBN 1424408644; 9781424408641. ISSN 19354576. Cited By 3. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-39749126221&doi=10.1109\\%2fINDIN.2007.4384812&partnerID=40&md5=f77746b0a213044de25d5d9d85aac9c3.

LEãO, E.; GUEDES, L.; VASQUES, F. Implementation of an event-triggered smart sensor network architecture based on the ieee 802.15.4 standard. In: . IFAC, 2007. v. 7, n. PART 1, p. 279–284. ISBN 9783902661340. ISSN 14746670. Cited By 1. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-79960997757&partnerID=40&md5=b2d6c1e9b700e8f9048d55933c1e61c4.

LI, J.; LI, J. Data sampling control and compression in sensor networks. In: JIA, X.; WU, J.; HE, Y. (Ed.). **Mobile Ad-hoc and Sensor Networks**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 42–51. ISBN 978-3-540-32276-4.

Li, M. et al. Error-bounded data compression using data, temporal and spatial correlations in wireless sensor networks. In: **2010 International Conference on Multimedia Information Networking and Security**. Washington, USA: IEEE, 2010. p. 111–115. ISSN 2162-8998.

LI, Y.; LIANG, Y. Temporal lossless and lossy compression in wireless sensor networks. **ACM Transactions on Sensor Networks**, v. 12, 10 2016.

Li, Y.; Loke, S. W.; Ramakrishna, M. V. Energy-saving data approximation for data and queries in sensor networks. In: **2006 6th International Conference on ITS Telecommunications**. Washington, USA: IEEE, 2006. p. 782–785.

LIGHT, R. A. Mosquitto: server and client implementation of the mqtt protocol. **Journal of Open Source Software**, 2017. Disponível em: http://joss.theoj.org/papers/10.21105/joss.00265.

LIU, J.; CHEN, F.; WANG, D. Data compression based on stacked rbm-ae model for wireless sensor networks. **Sensors**, v. 18, p. 4273, 12 2018.

LIU, X. et al. An energy-efficiency wireless sensing method for mechanical failure signals. **Key Engineering Materials**, v. 572, n. 1, p. 451–454, 2014. ISSN 10139826. Cited By 0. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84887974750&doi=10.4028\%2fwww.scientij1c.net\%2fKEM.572.451&partnerID=40&md5=13ee7b6396e0467a5bc0a3b3d9e57171.

LIU, X. C.; YU, L. Development of geolocation data compression for transportation target identification. **Transportation Research Record**, v. 2105, p. 71–82, 12 2009.

MADDEN, S. et al. Intel lab data. **Web page, Intel**, 2004. Disponível em: http://db.csail.mit.edu/labdata/labdata.html.

MAHDI, O.; MOHAMMED, M.; MOHAMED, A. J. Implementing a novel approach an convert audio compression to text coding via hybrid technique. **IJCSI**, 11 2013.

Marcelloni, F.; Vecchio, M. A simple algorithm for data compression in wireless sensor networks. **IEEE Communications Letters**, v. 12, n. 6, p. 411–413, 06 2008.

Mohamed, M. I.; Wu, W. Y.; Moniri, M. Adaptive data compression for energy harvesting wireless sensor nodes. In: **2013 10th IEEE INTERNATIONAL CONFERENCE ON NETWORKING, SENSING AND CONTROL (ICNSC)**. Washington, USA: IEEE, 2013. p. 633–638.

NETO LUIZ AUGUSTO, D. C. S. L. A. G. E. J. M. Adaptive swinging door trending: Um algoritmo adaptativo para compressao de dados em tempo real. **Anais do XX Congresso Brasileiro de Automática**, 2014.

Pattar, S. et al. Searching for the iot resources: Fundamentals, requirements, comprehensive review, and future directions. **IEEE Communications Surveys Tutorials**, v. 20, n. 3, p. 2101–2132, thirdquarter 2018. ISSN 1553-877X.

PAUL, P. V.; SARASWATHI, R. The internet of things — a comprehensive survey. In: **2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC)**. Washington, USA: IEEE, 2017. p. 421–426.

Perera, C. et al. Context aware computing for the internet of things: A survey. **IEEE Communications Surveys Tutorials**, v. 16, n. 1, p. 414–454, First 2014. ISSN 1553-877X.

Pham, N. D.; Le, T. D.; Choo, H. Enhance exploring temporal correlation for data collection in wsns. In: **2008 IEEE International Conference on Research, Innovation and Vision for the Future in Computing and Communication Technologies**. Washington, USA: IEEE, 2008. p. 204–208.

RAMIJAK, D.; PAL, A.; KANT, K. Pattern mining based compression of iot data. **ACM International Conference Proceeding Series**, p. 1–6, 01 2018.

RIJSBERGEN, C. J. V. **Information Retrieval**. 2nd. ed. Newton, MA, USA: Butterworth-Heinemann, 1979. ISBN 0408709294.

ROWELL, E. Big-O Cheat Sheet. 2019. Disponível em: https://www.bigocheatsheet.com/. Acesso em: 11 de novembro de 2019.

SASAKI; YUTAKA et al. The truth of the f-measure. **Teach Tutor mater**, v. 1, n. 5, p. 1–5, 2007.

SCHRICKTE, L. et al. Design and implementation of a 6LoWPAN gateway for wireless sensor networks integration with the internet of things. **International Journal of Embedded Systems**, v. 8, n. 5-6, p. 380–390, 2016.

SENSIRION. **Datasheet SHT1x (SHT10, SHT11, SHT15),Humidity and Temperature Sensor**. Sensirion, 2008. Disponível em: https://www.sparkfun.com/datasheets/Sensors/SHT1x_datasheet.pdf.

Sharma, R. A data compression application for wireless sensor networks using ltc algorithm. In: **2015 IEEE International Conference on Electro/Information Technology (EIT)**. Washington, USA: IEEE, 2015. p. 598–604. ISSN 2154-0373.

Shravana, K. P.; Veena, D. S. V. Review on lossless data compression using x-matchpro algorithm. In: **2017 2nd IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)**. Washington, USA: IEEE, 2017. p. 1095–1100.

Silva, I. M. D.; Guedes, L. A.; Vasques, F. Performance evaluation of a compression algorithm for wireless sensor networks in monitoring applications. In: **2008 IEEE Int. Conf. on Emerging Technologies and Factory Automation**. Washington, USA: IEEE, 2008. p. 672–678. ISSN 1946-0740.

SRISOOKSAI, T. et al. Practical data compression in wireless sensor networks: A survey. **Journal of Network and Computer Applications**, v. 35, n. 1, p. 37 – 59, 2012. ISSN 1084-8045. Collaborative Computing and Applications. Disponível em: http://www.sciencedirect.com/science/article/pii/S1084804511000555.

Stojkoska, B. R.; Nikolovski, Z. Data compression for energy efficient iot solutions. In: **2017 25th Telecommunication Forum (TELFOR)**. Washington, USA: IEEE, 2017. p. 1–4.

TUAMA, A. Y. et al. Recent advances of data compression in wireless sensor network. **Journal of Engineering and Applied Sciences**, Medwell Journals, v. 13, n. 21, p. 9002–9015, 2018.

Ud Din, I. et al. The internet of things: A review of enabled technologies and future challenges. **IEEE Access**, v. 7, p. 7606–7640, 2019. ISSN 2169-3536.

UTHAYAKUMAR, J.; VENGATTARAMAN, T.; DHAVACHELVAN, P. A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications. **Journal of King Saud University - Computer and Information Sciences**, 2018. ISSN 1319-1578. Disponível em: http://www.sciencedirect.com/science/article/pii/S1319157818301101.

V, B.; M, A.; O, T. Fuzzycat: A novel procedure for refining the f-transform based sensor data compression. In: . Association for Computing Machinery, Inc, 2015. p. 340–341. ISBN 9781450334754. Cited By 1. Disponível em: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84954121143&doi=10.1145\\%2f2737095.2742921&partnerID=40&md5=e2f6eaffd54ee88d838d2f76bbdeb602.

Wang, C. et al. Tree-structured linear approximation for data compression over wsns. In: **2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)**. Washington, USA: IEEE, 2016. p. 43–51. ISSN 2325-2944.

YILDIRIM, O.; TAN, R. S.; ACHARYA, U. R. An efficient compression of ecg signals using deep convolutional autoencoders. **Cognitive Systems Research**, v. 52, p. 198 – 211, 2018. ISSN 1389-0417. Disponível em: http://www.sciencedirect.com/science/article/pii/S1389041718302730.

Yu, W. et al. A survey on the edge computing for the internet of things. **IEEE Access**, v. 6, p. 6900–6919, 2018. ISSN 2169-3536.

ZHANG, J. et al. A self-adaptive regression-based multivariate data compression scheme with error bound in wireless sensor networks. **International Journal of Distributed Sensor Networks**, v. 9, n. 3, p. 913497, 2013. Disponível em: https://doi.org/10.1155/2013/913497.

ZHANG, Y.; LI, J. Wavelet-based vibration sensor data compression technique for civil infrastructure condition monitoring. **Journal of computing in civil engineering**, American Society of Civil Engineers, v. 20, n. 6, p. 390–399, 2006.

ZHAO, J. K. et al. In-network time-series data compression for electric internet of things. **Applied Mechanics and Materials**, v. 241-244, p. 3213–3223, 12 2012.

APÊNDICE A – EXEMPLO SWINGING DOOR TRENDING

0 *Swinging Door Trending* (SDT) é um algoritmo de compressão com perda que utiliza um mecanismo de tolerância ao erro para realizar uma representação linear dos dados a ser comprimidos. Este funciona como duas portas rotatórias, perpendiculares, que vão se abrindo para receber dentro de uma área de aceitação as amostras que vão chegando. A área criada é um paralelogramo e estas portas só podem continuar se abrindo até que estejam paralelas.

De acordo com a patente do SDT (BRISTOL, 1987; BRISTOL, 1990), este mecanismo conta com oito passos apresentados no Algoritmo 1:

1. Receber o primeiro ponto.
2. O pivô superior e inferior são estabelecidos.
3. Receber o ponto seguinte.
4. Calcular as inclinações atuais em relação ao pivô superior e inferior.
5. Comparar os declives atuais com os casos de declives extremos (S^s_{max}, S^i_{min}). Se S^s_{max} é maior que S^i_{min}, o algoritmo continua no sexto passo; caso contrário, retorna ao terceiro.
6. Localizar o ponto final dentro da área do paralelogramo. A inclinação entre este e o ponto atual é calculada, e a borda cruzada é ajustada para ser paralela à outra. Uma interseção entre a borda cruzada e a inclinação do ponto final e atual é calculada e definida como o novo valor de c.
7. Emitir o ponto final c.
8. O novo ponto inicial c é usado como inicio do paralelogramo, as inclinações extremas são restauradas, os pivôs são novamente definidos e o ponto atual é avaliado. Finalmente, o algoritmo volta ao terceiro passo.

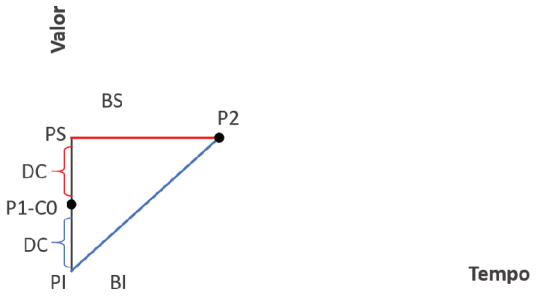
Nas Figuras 21-28 apresenta-se o passo a passo da Figura 3 contida na Seção 3.1. Na Figura 21 se recebe o primeiro ponto e se estabelecem os pivô superior e inferior. Na Figura 22 consta a segunda amostra, com esta são calculados os declives atuais em relação aos pivôs superior e inferior, estes declives são comparados com os extremos atuais, em caso de que as "portas"precisem ser "abertas", os extremos são trocados pelos atuais. Nas figuras s 23-25 segue o mesmo processo, porém as portas estão no limite estabelecido, a abertura máxima delas é até estar paralelas. Na Figura 26 o ponto esta fora do paralelogramo do SDT criado pelas portas, então segue ao passo 6, no qual se localiza este ponto final, este processo é apresentado na Figura 27. Primeiramente é gerada uma linha entre a ultima amostra dentro do paralelogramo e a ultima recebida, depois é encontrado o ponto em que ultrapassa uma das portas, que é posicionada paralela à outra porta, depois é subtraído pelo DC, parâmetro do SDT que define a largura das portas e o nível de tolerância ao erro. Na Figura 28 cria-se um novo paralelogramo,

as portas são reiniciadas, o ultimo ponto encontra-se dentro deste novo paralelogramo, e o processo é reiniciado.

Figura 21 – Chegada da amostra inicial SDT

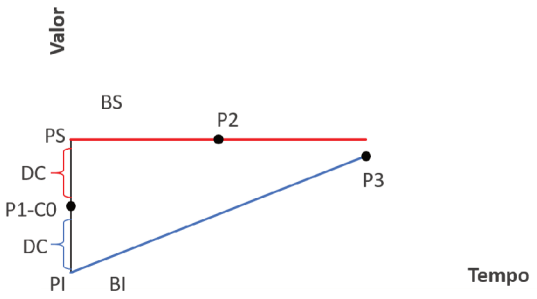
Fonte – O autor.

Figura 22 – Chegada da segunda amostra SDT



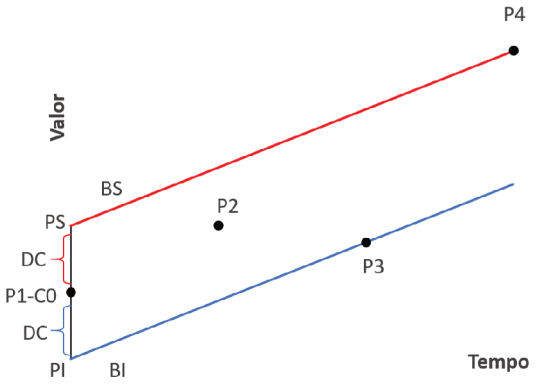
Fonte – O autor.

Figura 23 – Chegada da terça amostra SDT



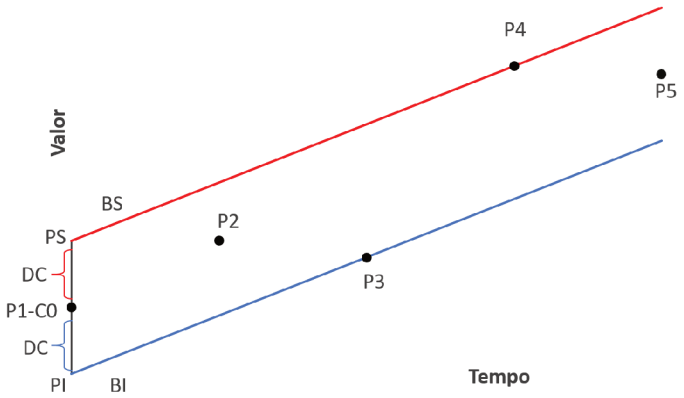
Fonte – O autor.

Figura 24 – Chegada da quarta amostra SDT



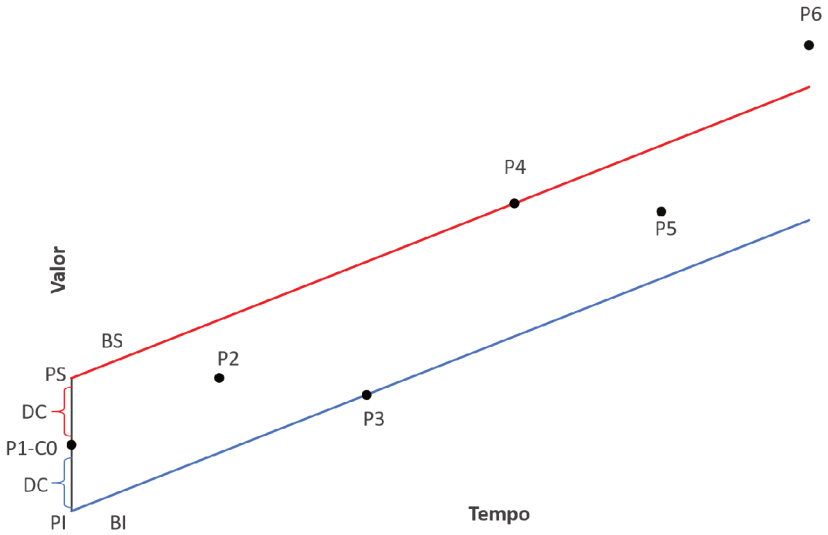
Fonte – O autor.

Figura 25 – Chegada da quinta amostra SDT



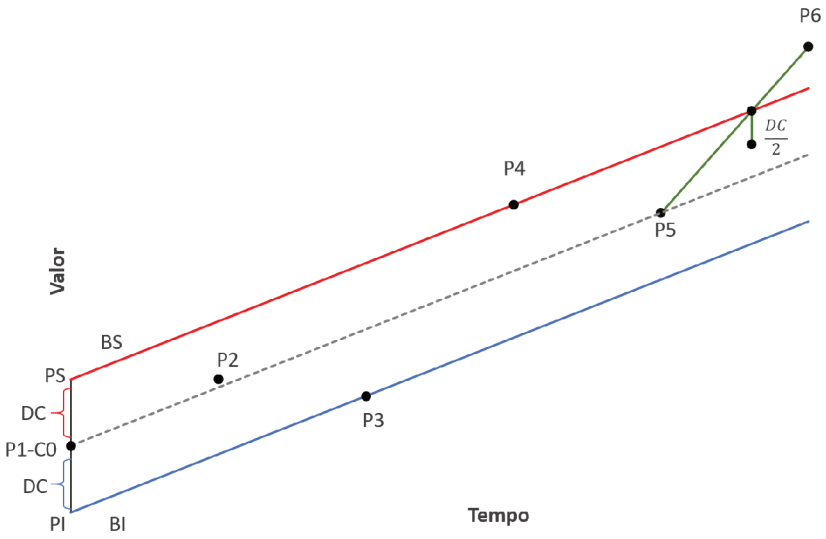
Fonte – O autor.

Figura 26 – Chegada da sexta amostra SDT



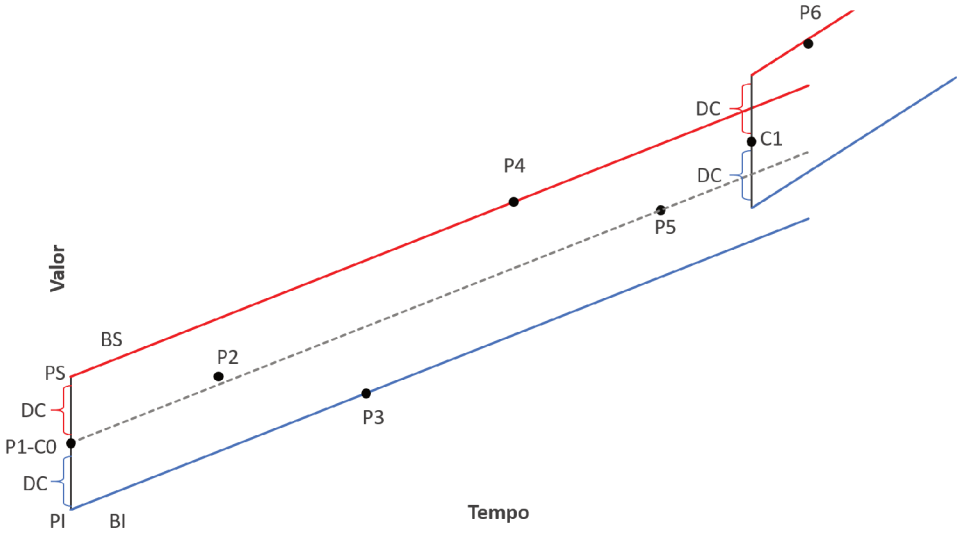
Fonte – O autor.

Figura 27 – Calculando o ponto final do paralelogramo e o ponto entregado



Fonte – O autor.

Figura 28 – Novo paralelogramo criado



Fonte – O autor.