

Redução de Dados em Redes de Sensores sem Fio Baseada em Análise de Dispersão

Samuel Silva de Oliveira^{1,2}, Janine Kniess²

¹ Departamento de Ciências Exatas e Tecnológicas
Universidade Federal do Amapá (UNIFAP) – Macapá, AP – Brasil

² Programa de Pós-Graduação em Computação Aplicada
Universidade do Estado de Santa Catarina (UDESC) – Joinville, SC – Brasil

samuel.oliveira@unifap.br, janine.kniess@udesc.br

Abstract. *In WSN monitoring applications, sensor nodes may rely on limited power sources. Some studies indicate that the main source of energy consumption in sensor nodes is related to data transmission. In this paper, we propose an approach to reduce transmission in sensor nodes based on sensor data dispersion analysis. This approach aims to avoid transmitting detections whose values demonstrate low dispersion. The conducted experiments using real sensor data-set and the Castalia Simulator show that the proposed approach has been able to achieve up to 84% transmission reduction while maintaining a low level of errors and low energy consumption.*

Keywords: *WSN, Dispersion Analysis, Data Reduction*

Resumo. *Nas aplicações de monitoramento com Redes de Sensores sem Fio (RSSF), os sensores podem depender de fontes de energia limitada. Estudos apontam que a principal fonte de consumo de energia em nós sensores está relacionada à transmissão de dados. Neste artigo, apresenta-se uma abordagem para redução de dados baseada na análise da dispersão dos dados de sensores, visando evitar o envio de detecções cujos valores sejam pouco dispersos. Os experimentos realizados com dados de sensores reais e com o simulador Castalia mostraram que a abordagem proposta atingiu uma redução maior que 84%, mantendo um baixo nível de erros e baixo consumo de energia.*

Palavras chave: *RSSF, Análise de Dispersão, Redução de Dados*

1. Introdução

A propagação das tecnologias de sistemas microeletromecânicos¹, comunicação sem fio e eletrônica digital impulsionaram o desenvolvimento de dispositivos de baixo custo com capacidades de sensoriamento, processamento e comunicação sem fio, normalmente chamados de nós sensores [Yick et al. 2008]. A utilização destes dispositivos em conjunto caracteriza a formação de Redes de Sensores sem Fio (RSSF), que são baseadas no esforço colaborativo de uma grande quantidade de nós.

¹Do Inglês: *Micro Electro Mechanical Systems* (MEMS)

Nas RSSF, comumente utilizadas no contexto da Internet das Coisas (IoT), os nós sensores podem possuir espaço de armazenamento limitado e consumir grande quantidade de energia. Por isso, é habitual que eles sejam programados para enviar os dados coletados de forma imediata para as estações bases, chamadas de nós sorvedouro², que são responsáveis pelo encaminhamento das mensagens até o seu destino final.

[Alsheikh et al. 2016] destaca que em determinadas RSSF, até 80% do consumo total de energia pode estar relacionado às atividades de transmissões de dados. Com base no exposto, é possível identificar que a minimização das transmissões de dados dos sensores nas RSSF podem prolongar o seu tempo de vida. Outrossim, [Alsheikh et al. 2016] destaca que a redução das transmissões de dados a partir dos sensores não apenas resulta em economia de energia, mas também na redução do consumo na largura de banda da rede, possibilitando o uso mais eficiente dos recursos disponíveis.

Neste trabalho, as contribuições apresentadas são: (i) **Método para Redução da Transmissão de Dados em RSSF** e (ii) **Algoritmo de Reconstrução de Dados**. O Método para a Redução de Dados tem como alvo os nós sensores atuando em aplicações de monitoramento, como por exemplo, medição de temperatura, controle de CO₂, dentre outros. O método é baseado na análise da dispersão dos dados detectados, cujo objetivo é identificar e deixar de enviar conjuntos de detecções com valores que demonstrem baixa dispersão, enviando apenas um valor normalizado (média) correspondente a todo o conjunto. Através dessa redução, pretende-se utilizar de forma mais eficiente os recursos da rede, como largura de banda e energia dos nós, mantendo alta acurácia nos dados detectados pelos sensores. O Método para a Redução de Dados inclui ainda um algoritmo para definir um valor que será utilizado como limite máximo aceitável para que duas ou mais detecções sejam consideradas pouco dispersas, e então, deixem se der transmitidas. Já o Algoritmo de Reconstrução de Dados visa reconstruir os dados otimizados pelos sensores a fim de que apresentem a quantidade correta de amostras no seu destino final.

Este artigo está organizado da seguinte forma: na Seção 2 apresentam-se alguns trabalhos relacionados. Na Seção 3, a abordagem proposta, composta por um algoritmo de análise da dispersão para a redução da transmissão de dados e um algoritmo para a reconstrução dos dados no destino final. Na Seção 4 apresentam-se os experimentos computacionais. Por fim, na Seção 5, as conclusões e os trabalhos futuros.

2. Fundamentos e Trabalhos Relacionados

Nesta seção serão discutidos alguns trabalhos da literatura que lidam com o problema da redução da transmissão de dados em aplicações de monitoramento, baseados em três principais categorias: Predição de Dados, Agregação de Dados e *Compressive Sensing*.

A predição de dados é uma técnica que visa estimar valores futuros com base em dados históricos. Baseado neste princípio, o histórico das detecções de um determinado nó sensor pode ser utilizado para que um nó sorvedouro realize a predição de detecções futuras. O objetivo principal é evitar que o nó sensor envie um dado que pode ser previsto no sorvedouro, levando em consideração um certo nível de acurácia da predição.

No trabalho proposto por [Dias et al. 2016], os autores adotam o esquema de dupla predição (no sensor e no sorvedouro) com a finalidade de prever o valor das detecções

²Em Inglês: Sink nodes

futuras. Quando a predição no sensor possui boa acurácia, o dado original não é enviado. Ao invés disso, o sorvedouro encaminha o valor resultante da predição, provendo assim menor sobrecarga na comunicação interna da rede. Na análise dos resultados, identificou-se que ao utilizar o esquema de dupla predição somado a técnicas de agregação de dados, é possível economizar até 92% da energia consumida pelas transmissões.

O trabalho proposto por [Fathy et al. 2018], contudo, é baseado no algoritmo de filtragem adaptativa *Least Mean Square* (LMS)³, combinando dois filtros LMS com diferentes tamanhos para estimar os valores detectados. Essa estimativa é realizada tanto nos nós sensores quanto nos nós sorvedouros, de modo que os nós sensores precisem transmitir apenas seus valores que se desviam significativamente dos valores previstos. Constatou-se que a abordagem proposta pelos autores alcançou alta redução nas transmissões de dados (em torno de 95%).

A técnica de Agregação de dados é utilizada em RSSF com o objetivo principal de agregar dados provenientes de diferentes nós. Nessa abordagem, os nós definidos como agregadores são responsáveis por coletar os dados de diferentes sensores e enviá-los de uma só vez, ao invés de simplesmente retransmiti-los. A consequência do uso desta técnica é a redução no número de pacotes transmitidos para a rede, bem como, maior economia de energia [Dhand and Tyagi 2016].

No trabalho de [Zegarra et al. 2016], é proposto o algoritmo chamado *Continuous Enhancement Routing* (CER), que utiliza o Algoritmo Genético *Biased Random Key Genetic Algorithm* (BRKGA) para computar a árvore de roteamento de uma RSSF. O CER busca encontrar árvores de roteamento próximas do ótimo, realizando a agregação de dados pelo maior tempo possível. Os resultados mostraram que o CER gera maior sobrecarga comparado ao algoritmo *Shortest Path Tree* (SPT), pois a alta quantidade de árvores de roteamento resulta em uma quantidade expressiva de pacotes de controle.

O trabalho de [Li et al. 2017], por sua vez, propõe um método balanceado de agregação de dados de longo prazo, denominado *Lifetime Balanced Data Aggregation* (LBA). O objetivo principal do LBA é prolongar o tempo de vida da rede considerando sua dinamicidade e heterogeneidade, garantindo o mínimo de atraso na entrega de dados. Os resultados apresentados demonstram que o LBA se mostra mais interessante quando todos os nós da rede iniciam atividade com a mesma quantidade de energia em suas baterias.

As técnicas de processamento de sinais também têm sido exploradas em RSSF, dentre elas, a técnica *Compressive Sensing* (CS). Esta técnica aplica uma matriz de detecções $\Phi \in \mathbb{R}^{m \times n}$ em um vetor de sinal n -dimensional $x = (x_1, \dots, x_n)^T$ a fim de obter um sinal m -dimensional $y \in \mathbb{R}^m$. A reconstrução dos dados, segundo [Zheng et al. 2017], pode ser realizada através da propriedade *Restricted Isometry Property* (RIP)⁴, que garante a recuperação de x a partir do sinal comprimido y se $m \geq k \log n/k$.

O trabalho de [Masoum et al. 2013], utiliza o princípio de amostragem adaptativa para transmitir apenas as detecções que indicam uma mudança significativa no ambiente sensoriado. Essa abordagem tira proveito da correlação espacial e temporal entre os dados coletados pelos sensores a fim de alcançar maior compressão. Para definir o conjunto

³Em Português: Mínimo Quadrado Médio

⁴Em Português: Propriedade de Isometria Restrita

de detecções a ser enviado pelo sensor, foi utilizado um modelo estocástico baseado na Inferência Bayesiana. A reconstrução dos dados é realizada pelo sorvedouro através do Algoritmo de Propagação de Crença. A abordagem atingiu melhores resultados que os métodos *Basis Pursuit* e *Joint-BP* em termos de acurácia da reconstrução e consumo de energia.

Já [Huang et al. 2017] propuseram um esquema compressivo e adaptativo de coleta de dados, que se trata do ajuste adaptativo da frequência em que as detecções são realizadas pelo sensor. Para a reconstrução dos dados, os autores propuseram uma versão adaptativa do algoritmo guloso chamado *Stagewise Orthogonal Matching Pursuit algorithm* (StOMP), que ajusta o limite de dispersão (parâmetro) baseado no erro da reconstrução. Os resultados demonstraram que o método alcança maior acurácia na reconstrução e menor consumo de energia quando comparado ao método *Principal Component Analysis* (PCA), também utilizado para reconstrução de sinais em *Compressive Sensing*.

Embora os trabalhos descritos apresentem bons resultados com relação à redução da transmissão de dados em RSSF, algumas limitações podem ser discutidas. Nos trabalhos baseados em Predição de Dados e em *Compressive Sensing*, espera-se que o sorvedouro tenha grande quantidade de energia e alto poder de processamento. Nos trabalhos baseados em agregação, o atraso pode ser um fator determinante, uma vez que um nó agregador deve aguardar a chegada de pacotes oriundos dos demais sensores para que sejam agregados. A técnica de redução de dados proposta neste trabalho visa alcançar a redução de dados explorando a análise da dispersão dos dados coletados pelos sensores, visando diminuir o consumo de energia, bem como o uso da largura de banda da RSSF. Além disso, o algoritmo de reconstrução de dados proposto neste trabalho, visa reconstruir os dados otimizados pelos sensores a fim de que apresentem a quantidade correta de amostras no seu destino final.

3. Abordagem Proposta

Nas aplicações que utilizam sensores para realizar monitoramento, os nós sensores geralmente realizam detecções de variáveis do ambiente físico com uma frequência pré-definida. O resultado disto é a criação de uma coleção ordenada de dados. Para exemplificar, essa coleção pode ser chamada de S , iniciando em um ponto no tempo t_0 , e seguindo até um tempo final t_n . Especificamente, $S = \{s_0, s_1, s_2, \dots, s_n\}$, na qual o n -ésimo elemento é um par $s_n = (V_n, t_n)$. O primeiro elemento do par, V_n , indica o valor da variável detectada pelo sensor (por exemplo: valor de temperatura), enquanto que o segundo elemento do par, t_n , indica o instante de tempo da ocorrência da leitura.

De acordo com [El-Telbany and Maged 2017], as detecções realizadas pelos sensores em aplicações de monitoramento frequentemente retornam valores muito semelhantes, ou até mesmo idênticos. O método proposto neste trabalho é baseado na análise da dispersão dos dados coletados por sensores, que visa identificar conjuntos de detecções com valores que demonstrem baixa dispersão, ou seja, valores próximos uns dos outros.

3.1. Algoritmo para Redução de Dados Baseado na Análise da Dispersão

A abordagem para redução na transmissão de dados em nós sensores funciona como exposto a seguir. Suponha que duas ou mais detecções de um determinado nó sensor demonstrem um certo nível de similaridade (baixa dispersão). O nó sensor mantém esses

valores em uma variável, de forma temporária, visando não transmitir todas as detecções com valores pouco dispersos. A seguir, é executado o cálculo da média dos valores com baixa dispersão, transmitindo esse valor ao sorvedouro em apenas uma mensagem. A partir de então, a variável utilizada para guardar os valores pouco dispersos pode ser apagada e a análise dos valores detectados é retomada a fim de encontrar novos conjuntos de detecções com baixa dispersão.

A medida de dispersão é representada por um número real não-negativo que é zero quando os dados são iguais. Esse número aumenta à medida em que os dados se divergem. No contexto deste trabalho, a análise da dispersão é realizada a cada detecção dos nós sensores. A medida de dispersão é representada pela Distância de *Manhattan*, expressa na Equação 3.1, que é aferida entre V_i e V_{i-1} . Justifica-se o uso da Distância de *Manhattan* pelo baixo custo computacional se comparada à outras medidas de distância, como a Distância Euclidiana.

$$D_{V_i, V_{i-1}} = |V_i - V_{i-1}| \quad (3.1)$$

Com o objetivo de permitir que o nó sensor seja capaz de identificar um conjunto de detecções com baixa dispersão, define-se um parâmetro, denominado D_{max} , que representa o limite máximo aceitável para a dispersão entre os valores detectados. No Algoritmo 3.1 apresenta-se a estratégia desenvolvida para redução de dados baseada na análise da dispersão.

Algoritmo 3.1 Algoritmo para Redução de Dados Baseado na Análise da Dispersão

```

1: FirstReadings[]                                ▷ Primeiras detecções do sensor (Fase de Inicialização)
2:  $N \leftarrow 50$                                   ▷ Número de detecções na fase inicial. Ex: 50
3:  $V_{i-1} \leftarrow \text{FirstReadings}[N - 1]$         ▷ Última detecção da Fase de Inicialização
4:  $N_V \leftarrow 1$                                   ▷ Contador de detecções armazenadas localmente
5:  $N_{max} \leftarrow 21$                               ▷ Máximo de detecções armazenadas localmente. Ex: 21
6: buffer  $\leftarrow 0$ 
7:  $D_{max} \leftarrow \text{GETDMAXVALUE}(\text{firstReadings}[], N)$     ▷ Define valor de  $D_{max}$ 
8: while True do
9:    $V_i \leftarrow \text{GETSENSORVALUE}$                     ▷ Realiza detecção ( $V_i$ )
10:  if  $|V_i - V_{i-1}| \leq D_{max}$  and  $N_V < N_{max}$  then
11:     $N_V \leftarrow N_V + 1$ 
12:    buffer  $\leftarrow \text{buffer} + V_i$ 
13:  else
14:    if buffer  $> 0$  then
15:       $\text{SENDTOSINK}((\text{buffer}/(N_V - 1)), N_V)$           ▷ Transmite Valor Normalizado
16:      buffer  $\leftarrow 0$ 
17:    else
18:       $\text{SENDTOSINK}(V_{i-1}, N_V)$                         ▷ Transmite Detecção Isolada
19:    end if
20:     $N_V \leftarrow 1$ 
21:  end if
22:   $V_{i-1} \leftarrow V_i$ 
23: end while

```

Se a distância entre o valor das detecções V_i e V_{i-1} for menor ou igual a D_{max} , então a detecção V_i não será transmitida, mas será armazenada temporariamente em uma variável, denominada *buffer* (linha 12). Além disso, um contador N_V que inicia

em 1 é incrementado (linha 11). Este procedimento será repetido enquanto a condição $D_{V_i, V_{i-1}} \leq D_{max}$ for satisfeita (linha 10). Em outras palavras, este procedimento será repetido até que a distância entre o valor de uma determinada detecção e o valor de sua detecção antecessora seja maior que D_{max} . Nesse caso, uma tupla é transmitida (linha 15), contendo a média dos valores armazenados na variável *buffer* e a quantidade de detecções que deixaram de ser enviadas, que é armazenada na variável (N_V), cujo valor será utilizado na fase de reconstrução, abordada na Seção 3.2.

O número de detecções com valores pouco dispersos em um nó sensor pode ser muito grande. Para evitar que um sensor armazene uma quantidade muito grande de detecções, bem como evitar um alto atraso no envio dos dados, define-se um parâmetro que estabelece um número máximo de detecções em um conjunto de dados com baixa dispersão, denominado N_{max} . A escolha desse parâmetro depende dos requisitos de cada aplicação de monitoramento com relação ao atraso na entrega das detecções. Quanto menor o atraso desejado, menor deve ser o valor de N_{max} . Uma taxonomia para a escolha de N_{max} de acordo com os diferentes tipos de aplicação é proposta na Seção 3.3.

A fase inicial do Algoritmo 3.1 compreende a coleta das primeiras detecções do nó sensor. Esses dados são utilizados para encontrar o valor ideal para D_{max} através da função GETDVALUE() (linha 6), cujos detalhes serão abordados na Seção 3.3.

3.2. Algoritmo para Reconstrução de Dados

Uma questão importante é garantir que, no seu destino, os dados enviados pelos nós sensores apresentem a frequência correta em que foram detectados. Dessa forma, o nó sorvedouro deverá repetir o valor normalizado de acordo com o valor de N_V , a fim de que seja mantida a frequência do conjunto de dados.

O algoritmo de reconstrução dos dados enviados pelos sensores é apresentado no Algoritmo 3.2. O principal objetivo deste algoritmo é atuar no nó sorvedouro, a fim de reconstruir os dados enviados pelos nós sensores.

Algoritmo 3.2 Algoritmo para Reconstrução dos Dados no Sorvedouro

```

1: dataset = []
2: while True do
3:    $V_i \leftarrow \text{RECEIVESENSORDATA}$                                 ▷ Recebimento da mensagem no sorvedouro
4:    $j = 0$ 
5:   while  $j < V_i[1]$  do
6:     dataset.append( $V_i[0]$ )                                       ▷ Repetição do valor normalizado (média)
7:      $j \leftarrow j + 1$ 
8:   end while
9:
10: end while

```

Supondo que os nós sensores enviem suas detecções utilizando tuplas, conforme descrito no Algoritmo 3.1, o primeiro item $V_i[0]$ (valor normalizado) irá se repetir de acordo com o valor do segundo item $V_i[1]$ (linha 5). Dessa forma o conjunto de dados reconstruído será composto pela mesma quantidade de detecções do sensor, embora nem todas tenham sido transmitidas.

3.3. Fase de Inicialização: Escolha do Valor de D_{max}

Para que a escolha de D_{max} (parâmetro do Algoritmo 3.1 para análise da dispersão) seja realizada de forma adequada, é proposto um algoritmo que visa analisar as primeiras detecções de um nó sensor e realizar esta escolha de forma automática (Algoritmo 3.3). Para isso, é definido um parâmetro denominado *acceptable_EAM*, que representa o erro máximo aceitável na reconstrução dos dados. Tal parâmetro deve ser ajustado de acordo com os requisitos de cada aplicação.

Com base nas primeiras detecções do nó sensor, o algoritmo para escolha do valor de D_{max} gera um conjunto de dados reduzido através da função *DATAREDUCTION()* que é uma adaptação do Algoritmo 3.1. Nesse caso, o parâmetro D_{max} deve iniciar com um valor baixo, a fim de que seja otimizado até encontrar um valor ideal. Em seguida, esse novo conjunto de dados é reconstruído através da função *DATA RECONSTRUCTION()* (adaptação do Algoritmo 3.2) localmente no sensor, e então compara-se o conjunto de dados reconstruído com o conjunto de dados inicial. Caso o erro obtido na reconstrução dos dados seja menor que o erro máximo aceitável, repete-se o procedimento elevando o valor de D_{max} . Ou seja, o valor de D_{max} será elevado até que o erro obtido na reconstrução seja menor ou igual ao erro máximo aceitável.

O erro obtido na reconstrução dos dados é determinado através do cálculo do Erro Absoluto Médio (EAM) [Willmott and Matsuura 2005]. Quando o valor do EAM obtido for igual ou maior que o erro máximo aceitável, o valor final de D_{max} será retornado. O algoritmo proposto para a escolha do valor de D_{max} na fase de inicialização é descrito no Algoritmo 3.3.

Algoritmo 3.3 Definição do valor de D_{max}

```
1: function GETDMAXVALUE(FirstReadings[], N)
2:    $D_{max} \leftarrow 0.01$  ▷ Valor inicial de  $D_{max}$ . Ex: 0,01
3:    $EAM \leftarrow 0$  ▷ Erro Absoluto Médio
4:    $acceptable\_EAM \leftarrow 0.5$  ▷ Erro máximo aceitável. Ex: 0,5
5:   while  $EAM \leq acceptable\_EAM$  do
6:      $ReducedData \leftarrow \text{DATAREDUCTION}(FirstReadings[], N, D_{max})$ 
7:      $ReconstructedData \leftarrow \text{DATA RECONSTRUCTION}(ReducedData[], N)$ 
8:      $EAM = (\sum_{i=0}^n |FirstReadings[i] - ReconstructedData[i]|) / n$ 
9:      $D_{max} \leftarrow D_{max} + 0.01$ 
10:  end while
11:  return  $D_{max}$ 
12: end function
```

A definição de um valor para o erro máximo aceitável é crucial para que o valor de D_{max} seja escolhido. [Karim 2017] destaca que algumas aplicações de monitoramento são críticas e exigem alta acurácia e baixo nível de atraso na entrega das mensagens. Com base no exposto, foi criada uma taxonomia que classifica diferentes tipos de aplicação de acordo com seus requisitos, atribuindo valores específicos para N_{max} e *acceptable_EAM*. Especificamente, foram definidas três classes de aplicações de monitoramento (0, 1 e 2) que são baseadas em seus requisitos de atraso na entrega das mensagens e no erro máximo aceitável na reconstrução dos dados. A classe 0 é menos tolerante a atrasos e exige maior acurácia nos dados reconstruídos. Já a classe 2, exige menor nível de acurácia e é mais tolerante a atrasos. A classe 1 possui valores intermediários entre as classes 0 e 2.

Na Seção 4, apresenta-se um exemplo de classificação dos *datasets* reais utilizados nos experimentos de acordo com a taxonomia desenvolvida. Na Tabela 1 são listados os parâmetros e as variáveis utilizadas nos Algoritmos 3.1, 3.2 e 3.3 da abordagem proposta para redução de dados em RSSF.

Tabela 1. Parâmetros e Variáveis: Algoritmos 3.1, 3.2 e 3.3

D_{max}	Distância máxima para que duas detecções sejam consideradas pouco dispersas
N_{max}	Quantidade máxima de detecções em um conjunto de detecções de baixa dispersão
$acceptable_EAM$	Erro máximo aceitável (Varia de acordo com os requisitos de cada aplicação)
$FirstReadings[]$	Vetor com as primeiras detecções
N	Quantidade de detecções em uma lista
N_V	Contador de detecções em um conjunto de detecções de baixa dispersão
$buffer$	Armazena os valores das detecções de baixa dispersão
V_i	Valor da detecção (s_i) do nó sensor no instante t_i
$dataset[]$	Lista que armazena os dados reconstruídos
$ReducedData[]$	Vetor com número reduzido de detecções
$ReconstructedData[]$	Vetor com dados reconstruídos
EAM	Erro absoluto médio

4. Simulações, Resultados e Análises

De acordo com o exposto na seção anterior, a abordagem proposta neste trabalho é baseada na análise da dispersão dos dados coletados por sensores, deixando de enviar detecções pouco dispersas. Nesta seção, apresenta-se uma avaliação de desempenho dos algoritmos de Redução e Reconstrução de dados em relação às seguintes métricas: **Consumo de Energia e Número de Pacotes Transmitidos**.

A avaliação de desempenho foi dividida em duas etapas. Na primeira etapa avaliou-se o potencial da redução de dados da abordagem proposta utilizando 5 (cinco) *datasets* reais disponíveis ao público. Ainda na primeira etapa, realizou-se uma comparação com um trabalho do estado-da-arte baseado no esquema de dupla predição com filtros *Least Mean Square* (LMS), proposto por [Fathy et al. 2018]. A escolha deste trabalho para comparação com a abordagem proposta se deu pelo fato de ser um trabalho recente, com resultados que demonstram alta redução na transmissão de dados em sensores, superando outras abordagens do estado-da-arte encontradas na literatura, como [Jaber et al. 2017], [Dias et al. 2016] e [Santini and Romer 2006].

Na segunda etapa, avaliou-se o consumo de energia de nós sensores no simulador Castalia⁵, com a aplicação da abordagem para a redução de dados baseada na análise da dispersão e sem a mesma. Mais detalhes sobre as simulações de rede serão abordadas na Subseção 4.3.

⁵<https://github.com/boulis/Castalia>

4.1. Caracterização dos Conjuntos de Dados (*Datasets*)

Nos experimentos da abordagem proposta, foram utilizados cinco conjuntos de dados (*datasets*) reais de aplicações que realizam monitoramento através de sensores. São eles:

1. Dados de sensores de qualidade do ar - Universidade da Califórnia em Irvine (UCI) [Vito et al. 2008]. Disponível no sítio da *UCI Machine Learning Repository*⁶ que contém uma coleção de bases de dados disponibilizadas pela Universidade da Califórnia em Irvine;
2. Dados do medidor inteligente de energia elétrica (*smartmeter*) do Projeto *Low Carbon London* [UK Power Networks 2015]. Dados de medidores inteligentes de 5.567 lares participantes em Londres - Reino Unido, no período de Novembro de 2011 a Fevereiro de 2015;
3. Dados de monitoramento da qualidade da água do Rio Burnett (Austrália) [Queensland Government 2015]. Este *dataset* coleta leituras de parâmetros sobre a qualidade da água estauriana a cada meia hora no período de 2016 a 2018;
4. Dados de temperatura capturados no Laboratório de Pesquisa da Intel na Universidade da Califórnia em Berkeley [Madden 2004]. Este conjunto de dados provém de 54 sensores Mica2Dot que monitoram os seguintes parâmetros: temperatura, umidade, luminosidade e tensão;
5. Monitoramento de compressores de ar de cadeiras odontológicas [Castañeda 2016]. Dados de sensores de umidade aplicados em compressores de cadeiras odontológicas para medir a umidade interna dos compressores e evitar que as cadeiras fiquem fora de operação, inviabilizando o atendimento odontológico.

Na Tabela 2 apresenta-se uma classificação dos 5 (cinco) *datasets* reais de acordo com a taxonomia descrita na Seção 3.3. Os *datasets* são divididos em classes (0, 1 e 2) e os parâmetros N_{max} e *acceptable_EAM* são ajustados para cada *dataset*.

Tabela 2. *Datasets* de Acordo com sua Classificação

#	<i>Dataset</i>	Classe	N_{max}	<i>acceptable_EAM</i>
#1	Umidade interna de compressores [Castañeda 2016]	0	15	0,25
#2	Dados de consumo de energia [UK Power Networks 2015]	0	15	0,25
#3	Turbidez da água do Rio Burnett [Queensland Government 2015]	1	30	0,35
#4	Dados de temperatura (Intel) [Madden 2004]	2	45	0,50
#5	Dados de Qualidade do ar (Umidade Relativa) [Vito et al. 2008]	2	45	0,50

4.2. Análise da Redução e Reconstrução de Dados

A implementação dos Algoritmos 3.1, 3.2 e 3.3 foi realizada na linguagem de programação Python e executada em um computador com as seguintes características: processador AMD Phenom(tm) II X4 B93 com 4 núcleos e 4Gb de memória RAM.

Os *datasets* compõe a entrada do programa e o resultado final é um *dataset* reconstruído. Dessa forma é possível identificar a quantidade de transmissões evitadas em

⁶<https://archive.ics.uci.edu/ml/datasets.html>

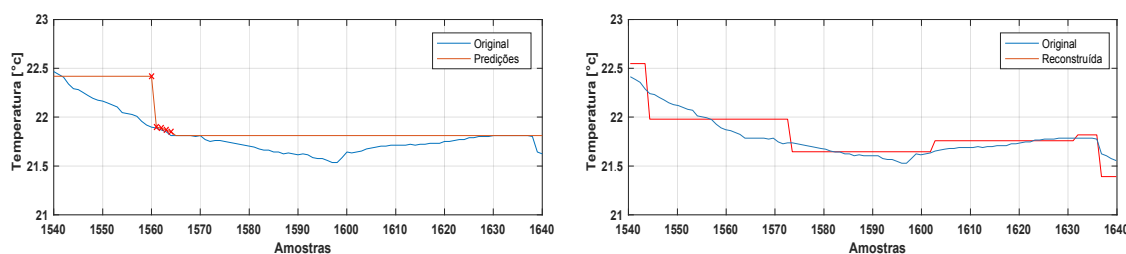
cada um dos *datasets*, bem como aferir a acurácia dos dados reconstruídos. Na Tabela 3 são descritos os resultados encontrados nos experimentos em relação à redução de dados. O valor de D_{max} foi ajustado automaticamente conforme o Algoritmo 3.3 (ver Seção 3.3).

Tabela 3. Resultados com o Algoritmo de Redução

Dataset	D_{max}	Número de detecções antes da redução	Número de detecções após a redução	Redução		EAM
				Quantidade	Porcentagem	
#1	0,01	3.970.215	359.768	3.610.447	93,85%	0,008
#2	7,0	17.458	14.762	2.696	15,42%	0,176
#3	1,5	14.332	5.739	8.593	59,95%	0,35
#4	0,85	30.506	689	29.817	97,74%	0,102
#5	2,17	9.357	5.156	4.201	44,89%	0,495

Através da análise dos resultados apresentados na Tabela 3, pode-se perceber que a quantidade de redução a ser alcançada depende do comportamento da série temporal em questão. Séries temporais com valores muito dispersos alcançam menor redução, como é o caso do *dataset* de consumo de energia (#2) que atingiu apenas 15.42% de redução. Por outro lado, séries temporais menos dispersas alcançam maior redução, como os *datasets* de umidade interna em compressores (#1) e de sensores de temperatura da Intel (#4) que atingiram 93,85% e 97,74% de redução, respectivamente.

No experimento representado pelas Figuras 1(a) e 1(b), comparou-se a abordagem proposta neste trabalho com o método *Adaptive Method for Data Reduction* (AMDR), proposto por [Fathy et al. 2018]. Utilizou-se o mesmo *dataset*, especificamente, o *dataset* de sensores de temperatura da Intel (#4). Assim como no trabalho de [Fathy et al. 2018] foram utilizados os dados de 4 sensores (1, 11, 13 e 49) do *dataset* da Intel. À semelhança de [Fathy et al. 2018], ajustou-se o erro máximo aceitável para 0,5. Na Figura 1(a) demonstram-se os resultados obtidos com o método AMDR. Na Figura 1(b) demonstram-se os resultados com a abordagem proposta neste trabalho.



(a) AMDR [Fathy et al. 2018]

(b) Abordagem proposta

Figura 1. Comparação entre o AMDR (a) e a Abordagem Proposta (b)

Na Figura 1(a), as marcações em vermelho representam as transmissões realizadas pelo nó sensor em virtude do valor previsto apresentar um desvio maior que o erro máximo aceitável com relação ao valor original. Os resultados expostos nas Figuras 1(a) e 1(b), demonstram que os dados reconstruídos pela abordagem proposta apresentam um

desvio médio de 0,10, enquanto, no trabalho de [Fathy et al. 2018], o desvio médio foi de 0,5. Com relação a redução, a abordagem proposta apresentou um percentual de 97,74% e o método AMDR [Fathy et al. 2018], um percentual de 95%.

Nota-se na Figura 1(a) que os valores previstos pelo método AMDR são mantidos acima dos valores reais, no geral. Já a abordagem proposta realiza a reconstrução dos dados gerando valores tanto abaixo quanto acima dos valores reais, visando obter maior acurácia possível.

4.3. Análise do Consumo de Energia

Com a finalidade de avaliar o impacto energético da abordagem proposta em nós sensores, foram realizadas simulações no simulador Castalia⁷. Nesse sentido, foi reproduzida uma RSSF com 55 nós equipados com o rádio CC2420 2.4 GHz IEEE 802.15.4 *ZigBee-Ready RF Transceiver*⁸, sendo apenas 1 nó sorvedouro. Os sensores foram distribuídos de forma aleatória através do parâmetro *SN.deployment*, em uma área de 150m x 150m. O sorvedouro, por sua vez, foi posicionado no centro.

Com relação ao rádio CC2420, seu raio de alcance é de 50m e a largura de banda é de 250 Kbps. No modo RX, são consumidos 0,062 *joules/segundo*. Já no modo TX são consumidos 0,057 *joules/segundo* e no modo *Idle*, são consumidos 0,014 *joules/segundo*.

O simulador Castalia é dividido em módulos. Dentre eles, destacam-se: aplicação, comunicação e de camada física. No módulo de comunicação, utilizou-se a implementação do algoritmo de roteamento *Low Energy Adaptive Clustering Hierarchy* (LEACH). No módulo de aplicação, foi desenvolvida uma aplicação denominada neste ponto, *DataReduction*, que contém a implementação dos Algoritmos 3.1, 3.2 e 3.3. O consumo de energia da aplicação *DataReduction*, foi comparado ao da aplicação *Value-Reporting (General)*. Esta aplicação está disponível no Castalia, porém, nenhuma técnica de redução de dados é aplicada. No módulo de camada física, por sua vez, foram incluídos os dados dos sensores de temperatura da Intel (Dataset #4), do dia 06 a 09 de Março de 2004, totalizando em média 8.640 leituras para cada um dos 55 sensores.

Na simulação, foi definido um intervalo de 12 segundos entre o envio das leituras através do parâmetro *minSampleInterval*. Dessa forma, para que todos os dados sejam enviados, o tempo definido para cada simulação foi de 103.680 segundos e a energia inicial dos nós foi ajustada em 7.000 Joules. Foram executadas 30 simulações para cada experimento. Na Figura 2 são apresentados os resultados com relação ao consumo energético dos sensores na utilização das aplicações *DataReduction* e *ValueReporting (General)*.

É importante ressaltar que, nas simulações, quando a quantidade de transmissão de dados é reduzida, os sensores tendem a permanecer por mais tempo no modo de recepção (RX), o que pode ocasionar um leve aumento no consumo de energia. Ainda assim, os resultados apresentados na Figura 2 mostraram que através da abordagem *DataReduction* foi possível reduzir o consumo de energia em 76,1%, em média, se comparado com a aplicação *ValueReporting (General)*. Dessa maneira, conclui-se que a estratégia de redução de dados com base na análise de dispersão, pode reduzir o consumo de energia, contribuindo para estender o tempo de vida da rede.

⁷<https://github.com/boulis/Castalia>

⁸<http://www.ti.com/product/CC2420>

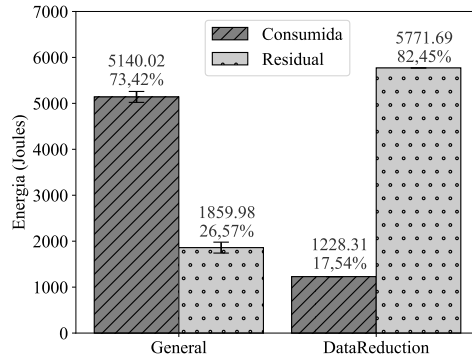
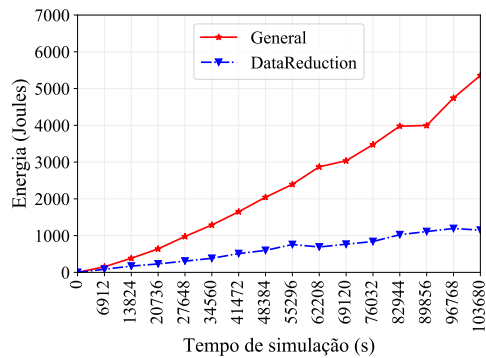
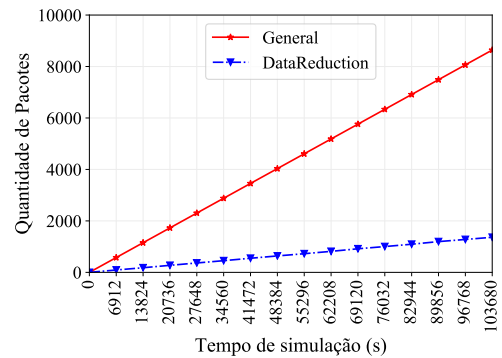


Figura 2. Energia Consumida × Residual

Na Figura 3, avaliou-se o consumo de energia e o comportamento da redução na transmissão de pacotes ao longo do tempo. De acordo com os resultados obtidos, observou-se que a abordagem proposta apresentou um desempenho melhor a medida que o tempo de simulação aumenta, e consequentemente o número de mensagens que trafegaram na rede. A redução, em média, alcançou um percentual de 84,32%.



(a) Consumo de Energia



(b) Transmissão de Pacotes

Figura 3. (a) Consumo de energia × Tempo (b) Pacotes Transmitidos × Tempo

5. Conclusão e Trabalhos Futuros

Neste trabalho, foi proposta uma abordagem para a redução da transmissão de dados em nós sensores atuando em aplicações de monitoramento no contexto de Redes de Sensores sem Fio. O objetivo principal dessa abordagem é evitar que nós sensores transmitam detecções cujos valores demonstram baixa dispersão (variabilidade/espalhamento) dentro de um limite máximo aceitável.

Para alcançar este objetivo, três algoritmos foram propostos. São eles: Algoritmo para redução de dados baseado na análise da dispersão dos dados coletados por sensores, algoritmo para reconstrução dos dados no destino final e algoritmo para a fase de inicialização (ajuste de parâmetros).

O algoritmo para redução de dados baseado na análise da dispersão contém com um parâmetro crítico denominado D_{max} que representa a dispersão máxima aceitável

para que as transmissões de dados sejam evitadas. Dessa forma, quanto maior o valor de D_{max} , maior redução é alcançada, porém, a acurácia da reconstrução dos dados é mais comprometida. Consequentemente, existe um *trade-off* entre quantidade de redução e nível de acurácia na reconstrução a ser avaliado.

Os experimentos basearam-se no uso de *datasets* reais de sensores e foram conduzidos em duas etapas. A primeira etapa objetivou a avaliação do desempenho dos algoritmos de redução e reconstrução a fim de aferir a quantidade de redução alcançada em cada *dataset*, bem como o erro obtido nas reconstruções. Na segunda etapa, avaliou-se o consumo de energia dos sensores através da aplicação da abordagem proposta no simulador Castalia. Verificou-se através dos resultados apresentados na Seção 4, que o método proposto neste trabalho alcançou maior nível de redução nas transmissões de dados se comparado ao trabalho de [Fathy et al. 2018] (97,74%), sendo que o Erro Absoluto Médio (EAM) dos dados reconstruídos foi 0,10. Em relação às simulações, os resultados mostraram que a abordagem proposta obteve redução nas transmissões de dados superior a 84%, bem como baixo consumo de energia dos nós, o que representou a economia de 76,1%, se comparado com a aplicação *ValueReporting*.

Como perspectivas de trabalhos futuros, sugere-se a implementação de uma abordagem híbrida para redução de dados em RSSF, basendo-se tanto na análise da dispersão, quanto na predição de valores futuros, a fim de alcançar maior nível de redução nas transmissões, e consequentemente, maior economia de energia dos nós. Além disso, sugere-se que sejam investigadas novas formas de reconstrução dos dados a fim de propor um método mais eficaz que torne possível diminuir ainda mais o erro obtido na fase da reconstrução.

Referências

- Alsheikh, M. A., Lin, S., Niyato, D., and Tan, H.-P. (2016). Rate-distortion balanced data compression for wireless sensor networks. *IEEE Sensors Journal*, 16(12):5072–5083.
- Castañeda, W. A. C. (2016). *Metodologia de gestão ubíqua para tecnologia médico-hospitalar utilizando tecnologias pervasivas*. PhD thesis, Universidade Federal de Santa Catarina.
- Dhand, G. and Tyagi, S. (2016). Data aggregation techniques in WSN:survey. *Procedia Computer Science*, 92:378–384.
- Dias, G. M., Bellalta, B., and Oechsner, S. (2016). Using data prediction techniques to reduce data transmissions in the IoT. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE.
- El-Telbany, M. E. and Maged, M. A. (2017). Exploiting sparsity in wireless sensor networks for energy saving: A comparative study. *International Journal of Applied Engineering Research*, 12(4):452–460.
- Fathy, Y., Barnaghi, P., and Tafazolli, R. (2018). An adaptive method for data reduction in the internet of things. In *Proceedings of IEEE 4th World Forum on Internet of Things*. IEEE.
- Huang, Z., Li, M., Song, Y., Zhang, Y., and Chen, Z. (2017). Adaptive compressive data gathering for wireless sensor networks. In *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pages 362–367.

- Jaber, A., Taam, M. A., Makhoul, A., Jaoude, C. A., Zahwe, O., and Harb, H. (2017). Reducing the data transmission in sensor networks through kruskal-wallis model. In *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE.
- Karim, S. (2017). Energy efficiency in wireless sensor networks, through data compression. Master's thesis, University of Oslo.
- Li, Z., Zhang, W., Qiao, D., and Peng, Y. (2017). Lifetime balanced data aggregation for the internet of things. *Computers & Electrical Engineering*, 58:244–264.
- Madden, S. (2004). Intel Lab Data. <http://db.lcs.mit.edu/labdata/labdata.html>. [Online; acessado em 08/03/2018].
- Masoum, A., Meratnia, N., and Havinga, P. J. (2013). A distributed compressive sensing technique for data gathering in wireless sensor networks. *Procedia Computer Science*, 21:207 – 216. The 4th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2013) and the 3rd International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH).
- Queensland Government (2015). Ambient estuarine water quality monitoring data (includes near real-time sites) - 2012 to present day. <https://data.qld.gov.au/dataset/ambient-estuarine-water-quality-monitoring-data-near-real-time-sites-2012-to-present-day>. [Online; acessado em 08/03/2018].
- Santini, S. and Romer, K. (2006). An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proceedings of the 3rd international conference on networked sensing systems (INSS 2006)*, pages 29–36.
- UK Power Networks (2015). SmartMeter Energy Consumption Data in London Households. <https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households>. [Online; acessado em 08/03/2018].
- Vito, S. D., Massera, E., Piga, M., Martinotto, L., and Francia, G. D. (2008). On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757.
- Willmott, C. J. and Matsuura, K. (2005). Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82.
- Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330.
- Zegarra, E. T., Schouery, R. C. S., Miyazawa, F. K., and Villas, L. A. (2016). A continuous enhancement routing solution aware of data aggregation for wireless sensor networks. In *2016 IEEE 15th International Symposium on Network Computing and Applications (NCA)*, pages 93–100.
- Zheng, H., Li, J., Feng, X., Guo, W., Chen, Z., and Xiong, N. (2017). Spatial-temporal data collection with compressive sensing in mobile sensor networks. *Sensors*, 17(11):2575.