CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

GERENCIAMENTO INTELIGENTE DE FALHAS APLICADO A REDE DE SENSORES SEM FIO COM INTEGRAÇÃO À INTERNET DAS COISAS

TÚLIO PHILIPE FERREIRA E VIEIRA

BELO HORIZONTE
MAIO DE 2017

TÚLIO PHILIPE FERREIRA E VIEIRA

GERENCIAMENTO INTELIGENTE DE FALHAS APLICADO A REDE DE SENSORES SEM FIO COM INTEGRAÇÃO À INTERNET DAS COISAS

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional

Linha de pesquisa: Sistemas Inteligentes

Orientador: Paulo Eduardo Maciel de Almeida

Coorientadora: Magali Rezende Gouvêa Meireles
PUC-Minas

BELO HORIZONTE
MAIO DE 2017

Vieira, Túlio Philipe Ferreira e Vieira
V658g Gerenciamento inteligente de falhas aplicado a rede de sensores sem fio com integração à internet das coisas. / Túlio Philipe Ferreira e Vieira. – – Belo Horizonte, 2017.
xiii, 93 f. : il.

Dissertação (mestrado) – Centro Federal de Educação Tecnológica de Minas Gerais, Programa de Pós-Graduação em Modelagem Matemática e Computacional, 2017.
Orientador: Profº Drº Paulo Eduardo Maciel de Almeida
Coorientadora: Profª Drª Magali Rezende Gouvêa Meireles

Bibliografia

1. Inteligência Computacional. 2. Kalman, Filtragem de. 3.Redes Neurais (Computação). I. Almeida, Paulo Eduardo Maciel de. II. Centro Federal de Educação Tecnológica de Minas Gerais. III. Título

CDD 001.64

Elaboração da ficha catalográfica pela Biblioteca-Campus II / CEFET-MG

Agradecimentos

Primeiramente, agradeço a Deus por toda força e pelas bençãos que têm derramado em minha vida.

Agradeço aos meus pais Leandro de Lourdes Vieira e Vânia Lúcia Ferreira e Vieira, por serem esta constante de carinho e sabedoria na minha vida e por todo apoio desde o início deste trabalho.

Aos meus irmãos Talita Hélen Ferreira e Vieira e Tássio Henrique Ferreira e Vieira, pela amizade e incentivo em todas as etapas da minha vida e construção deste trabalho.

A minha namorada Thayza Soares Dupim, pelo amor, companheirismo e compreensão em todos os momentos de angústia e ansiedade que foram compartilhados nesta caminhada;

Ao meu orientador Paulo Almeida e coorientadora Magali Meireles, pela sabedoria, confiança, aprendizado e por terem acreditado no meu trabalho;

Aos colegas do MMC, pela companhia durante esse período, especialmente a Samara, o Wagner, o Ramon, o Rondy e a Carolina pelos momentos de estudo, aprendizado e diversão.

Ao CEFET-MG e a CAPES pela estrutura e financiamento fornecido durante a execução deste trabalho;

A todos os envolvidos, muito obrigado.

Resumo

Este trabalho consiste no desenvolvimento de um sistema centralizado de gerenciamento de falhas que possibilita a integração das redes de sensores sem fio com a Internet das Coisas mediante a redução dos efeitos danosos causados por falhas de *hardware*. As falhas de *hardware* são as principais responsáveis pela ocorrência de falhas de dados. Quando um erro nos *dados* não é identificado e prontamente corrigido, todo o monitoramento de uma rede de sensores é comprometido. Ao integrar estes sensores à Internet das Coisas, os danos causados são potencializados, pois muitos outros dispositivos se utilizam destes dados para incrementar a qualidade de monitoramento do sistema e tomar decisões que podem gerar resultados negativos. A abordagem centralizada, proposta nesta dissertação, adota o protocolo 6LoWPAN como interface de rede para a comunicação. Este sistema é baseado em dois níveis de detecção de falhas. No primeiro nível, cada nó da rede realiza uma inferência estatística sobre o seu próprio estado de funcionamento. No segundo nível, realizado pela estação base, é usado um comitê composto por duas redes neurais artificiais do tipo Multilayer Perceptron (MLP). Uma destas redes é constantemente treinada enquanto a outra é utilizada como classificador da falhas. Em intervalos variados de tempo ocorre o chaveamento entre estas redes. As entradas da rede MLP são as saídas de um filtro de Kalman, que prevê o comportamento dinâmico da temperatura medida, e o nível de impacto sofrido pelo nó, medido por um acelerômetro. Os resultados experimentais demonstram que esta abordagem reduz o consumo de energia de um sistema centralizado em até 35,0% e mantém a precisão do processo de detecção de falhas em 97,0%.

Palavras-chave: Inteligência Computacional, 6LoWPAN, Filtro de Kalman, Acelerômetro.

Abstract

This work consists of the development of a centralized fault management system that enables the integration of wireless sensor networks with the Internet of Things by reducing the damaging effects caused by hardware errors. When a hardware error is not identified and promptly corrected, all monitoring of a sensor network is compromised. By integrating these sensors into the Internet of Things, the effects caused by failures are enhanced because many other devices use this data to increase the quality of system monitoring and make decisions that can lead to negative results. The centralized approach, proposed in this dissertation, adopts the 6LoWPAN protocol as a network interface for communication. This system is based on two levels of fault detection. At the first level, each node of the network makes a statistical inference about its own operating state. In the second level, performed by the base station, a committee composed of two artificial neural networks of the type Multilayer Perceptron (MLP) is used. One of these networks is constantly trained while the other is used as a fault classifier. At different time intervals, the change between these networks occurs. The inputs of the MLP network are the outputs of a Kalman filter, which predicts the dynamic behavior of the measured temperature, and the level of impact suffered by the node as measured by an accelerometer. Experimental results demonstrate that this approach reduces the energy consumption of a centralized system by up to 35.0% and maintains the accuracy of the fault detection process by 97.0%.

Keywords: Computational Intelligence, 6LoWPAN, Kalman Filter, Accelerometer.

Lista de Figuras

Figura 1 – Matriz de relacionamento da IoT 2
Figura 2 – Representação esquemática da arquitetura básica. 10
Figura 3 – Modelo de arquitetura em RSSF. 11
Figura 4 – Comparação entre diferentes abordagens para arquitetura proposta em IoT. 13
Figura 5 – Modelos de integração entre RSSF e a Internet. 15
Figura 6 – Diferenças entre as pilhas de protocolos IP e 6LoWPAN. 17
Figura 7 – Arquitetura 6LoWPAN. 18
Figura 8 – Gerenciamento de falhas. 20
Figura 9 – Exemplo modelo *cluster head*. 21
Figura 10 – Comportamento temporal das medidas realizadas por sensores. 22
Figura 11 – Ilustração do processo de associação em uma RNA. 25
Figura 12 – Ilustração do processo de classificação em uma RNA. 25
Figura 13 – Exemplo de um neurônio artificial. 26
Figura 14 – Estrutura de uma RNA composta por apenas uma camada. 26
Figura 15 – Exemplo de funções utilizadas como limiar (ativação) nos neurônios. . . 27
Figura 16 – Exemplo problemas linearmente separáveis (16a e 16b) e não linearmente separáveis (16c). 28
Figura 17 – Exemplo *MLP* com duas entradas, uma camada escondida com quatro neurônios e uma saída. 29
Figura 18 – Exemplo de funções de base radial comumente utilizadas. 31
Figura 19 – Comparação gráfica entre o mapeamento das redes RBF e MLP. 33
Figura 20 – Modelo genérico de um SIF. 35
Figura 21 – Modelo genérico de um SNF. 35
Figura 22 – Exemplo de arquitetura para o modelo ANFIS. 36
Figura 23 – Comparação entre as duas principais abordagens em gerenciamento de falhas. 47
Figura 24 – Representação do SCIGF com integração à Internet 48
Figura 25 – Estratégia estatística adotada para a fase local (primeiro nível) do SCIGF. 49
Figura 26 – Exemplo das trocas de mensagens realizadas no treinamento para o processo de detecção da falha do SCIGF. 50
Figura 27 – Exemplo das trocas de mensagens realizadas no diagnóstico da falha para o SCIGF. 50
Figura 28 – Representação da terceira fase do SCIGF. 53
Figura 29 – Etapas metodológicas do trabalho. 57
Figura 30 – Configuração do modelo de simulação de rede. 58
Figura 31 – Resposta à consulta por meio do *browser*. 59

Figura 32 – Consumo simulado das redes para solução do problema XOR. 62
Figura 33 – Experimento realizado para simulação de impactos. 64
Figura 34 – Modelo de dados encontrados após o experimento E_4. 65
Figura 35 – Níveis escolhidos segundo a variação expressa pela equação 15. 66
Figura 36 – Níveis simulados como entrada para a RNA. 66
Figura 37 – Curva ROC para o primeiro nível de detecção/diagnóstico da falha. . . . 68
Figura 38 – Matriz de confusão para o nível próprio do SCIGF. 68
Figura 39 – Filtro de Kalman: diferença entre os valores previstos e medidos. 69
Figura 40 – Curva ROC para o segundo nível de classificação. 70
Figura 41 – Matriz de confusão para o segundo nível de classificação. 71
Figura 42 – Comparação entre o consumo de energia do SCIGF e uma abordagem centralizada típica. 72
Figura 43 – Teste de Tukey para a comparação entre os gastos de energia do SCIGF e uma abordagem típica de gerenciamento centralizado de falhas. . . . 73
Figura 44 – Representação da aplicação de uma arquitetura IoT no monitoramento e controle de mudas em casas de vegetação. 74
Figura 45 – Modelo de interação com usuário através de *e-mail*: 45a. 77

Lista de Tabelas

Tabela 1 – Dados utilizados para simulação E_2. 60
Tabela 2 – Exemplo de dados gerados pelo *PowerTrace*: CPU indica o processador, *Load Program Memory* (LPM), TX representa o transmissor e RX o receptor. Os valores em cada célula representam o acumulador de pulsos 61
Tabela 3 – Consumo total: comparação entre cada uma das redes. 61
Tabela 4 – Exemplo de dados: temperatura na cidade de Diamantina (07/01/2017). 63
Tabela 5 – Teste Kolmogorov-Smirnov: H_0 não rejeitada. 63
Tabela 6 – Exemplo de dados utilizados para o treinamento da rede MLP. 67
Tabela 7 – Precisão na detecção de falhas: comparação entre algumas abordagens. 71

Lista de Algoritmos

Algoritmo 1 – FILTRO DE KALMAN - adaptado de Kovvali, Banavar e Spanias (2013). 24

Algoritmo 2 – ERROR BACK-PROPAGATION - adaptado Silva e Maia (2004). 30

Algoritmo 3 – TREINAMENTO REDE RBF - adaptado de Haykin (2001). 32

Lista de Abreviaturas e Siglas

6LoWPAN	*Low power Wireless Personal Area Networks*
AIS	*Artificial Immune System*
ANFIS	*Adaptive Neuro-Fuzzy Inference System*
ARM	*Architectural Reference Model*
BEA	*Bureau d'Enqueretês et d'Analyses*
BGM	*Brace-Gatarek-Muastrini*
CD	Células Dendríticas
CH	*Cluster Head*
CoAP	*Constrained Application Protocol*
CPU	*Central Processing Unit*
CSMA	*Carrier Sense Multiple Access*
DDS	*Data Distribution Service*
DFD	*Distributed Fault Detection*
DPV	*Disjoint Path Vector*
DNS-SD	*DNS-based Service Discovery*
FOP	*First-Order Perturbation*
FTSHM	*Fault-Tolerance in SHM*
GPS	*Global Position System*
IEEE	*Institute of Electrical and Electronics Engineers*
IoT	*Internet of Things*
IP	*Internet Protocol*
IPv6	*Internet Protocol Version 6*
LMS	*Least Mean Square*
M2M	*Machine-to-Machine*

MAC	*Media Access Control*
MANET	*Mobilr Ad hoc Networks*
mDND	*multicast Domain Name System*
MLP	*Multilayer Perceptron*
MQTT	*Message Queue Telemetry Transport*
OSI	*Open Systems Interconnection*
PCA	*Principal Component Analysis*
ping6	*Packet Internet Network Grouper IPv6*
QoS	*Quality of Service*
RAM	*Random Access Memory*
RBF	*Radial Basis Function*
RFID	*Radio-Frequency IDentification*
RNA	Redes Neurais Artificiais
ROC	*Receiver Operation Characteristic*
ROM	*Read-Only Memory*
RPL	*Routing Protocol for Low Power Networks*
RSSF	Redes de Sensores Sem Fio
RSSI	*Received Signal Strength Indicator*
SHM	*Structural Health Monitoring*
SIF	Sistema de Inferência Fuzzy
SMS	*Short Message Service*
SNF	Sistema Neuro-Fuzzy
SO	Sistema Operacional
SOM	*Self Organization Map*
SSW	*Statistics Sliding Windows*
SVM	*Support Vector Machine*
WPAN	*Wireless Personal Area*

Sumário

1 – Introdução 1
1.1 Justificativa 4
1.2 Definição do Problema 4
1.3 Objetivos 5
1.4 Organização Deste Trabalho 6

2 – Fundamentação Teórica 7
2.1 Sistemas Autonômicos 7
2.2 Falhas em RSSF 8
2.3 Redes de Sensores Sem Fio (RSSF) 9
2.3.1 Arquitetura das RSSF 9
2.3.2 Padrões em RSSF 10
2.4 Internet das Coisas (IoT) 11
2.4.1 Arquitetura de Aplicações em IoT 12
2.5 Integração entre RSSF e IoT 14
2.5.1 Oportunidades Geradas Pela Integração Entre RSSF e IoT 14
2.5.2 Protocolo 6LoWPAN 16
2.5.3 Arquitetura 6LoWPAN 17
2.6 Gerenciamento de Falhas 18
2.7 Filtro de Kalman 21
2.8 Redes Neurais Artificiais 24
2.9 Tipos de Redes Neurais Artificiais 27
2.9.1 Perceptron 27
2.9.2 Multilayer Perceptron 28
2.9.3 Redes de Função de Base Radial 30
2.10 Redes Neuro *Fuzzy* ANFIS 33
2.11 Modelagem e Simulação de RSSF 36

3 – Trabalhos Relacionados 38
3.1 Abordagem Centralizada 38
3.2 Abordagem Descentralizada 40
3.3 Detecção e Classificação de Falhas em Outros Sistemas 43
3.4 Detecção de Falhas Por Meio do Acelerômetro 44

4 – Modelo 46
4.1 Modelo de Gerenciamento de Falhas Proposto 46

4.2 SCIGF - Processo de Detecção de Falhas 48
4.3 SCIGF - Processo de Diagnóstico de Falhas 50
4.4 SCIGF - Processo de Recuperação da Falha 52

5 – Metodologia 54
5.1 Hipóteses 54
5.2 Planejamento de Experimentos 54

6 – Resultados e Discussões 58
6.1 Experimento E_1 58
6.2 Experimento E_2 59
6.3 Experimento E_3 62
6.4 Experimento E_4 64
6.5 Experimento E_5 67
6.6 Experimento E_6 71
6.7 Estudo de Caso 73

7 – Conclusão 78
7.1 Discussões Finais 78
7.2 Contribuições 81
7.3 Propostas de Continuidade 82

Referências 84

1 Introdução

Menos de duas décadas após Week (1999) anunciar que as redes compostas por microssensores estavam entre as 21 (vinte e uma) invenções mais importantes do século XXI, estamos vivenciando a comprovação desta afirmação. Com o desenvolvimento dos sistemas embarcados de baixo consumo energético, crescente progresso das tecnologias que possibilitam a miniaturização de dispositivos eletrônicos, evolução dos protocolos e infraestrutura de comunicação, as Redes de Sensores Sem Fio (RSSF) tornaram-se cada vez mais presentes em nossas vidas. Não é difícil encontrar redes de monitoramento remoto utilizadas no acompanhamento de pacientes (NEVES; STACHYRA; RODRIGUES, 2008), na previsão e alerta de desastres naturais (CHEN et al., 2013), no gerenciamento e conservação da biodiversidade natural (MAINWARING et al., 2002), na automação de cidades (NANDURY; BEGUM, 2015) e edifícios inteligentes (YEH; WANG; TSENG, 2009) além do monitoramento industrial (YAMAJI et al., 2008) e militar (WINKLER et al., 2012). Quando não apenas as RSSF mas também os próprios nós da rede (sensores e atuadores) e objetos do nosso cotidiano (*smartphones*, geladeiras, televisores, máquinas industriais, meios de transporte e etc) são conectados à Internet temos, o início de um novo paradigma conhecido como: Internet das Coisas (do inglês *Internet of Things* - IoT). Segundo Minerva, Biru e Rotondi (2015), a IoT é vista como a Internet dos itens (e seus componentes sensores e atuadores) conectados à Internet.

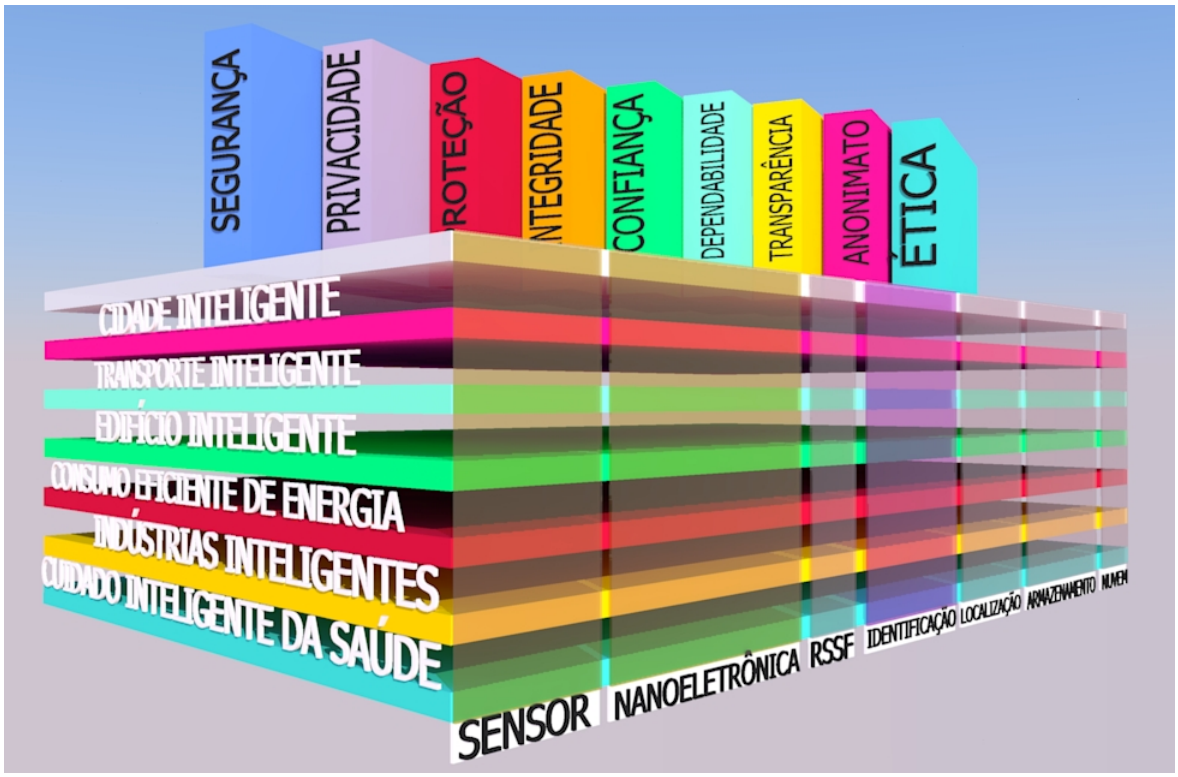
Uma rede de sensores sem fio pode ser vista como um conjunto de nós utilizados para sentir e interagir com o ambiente em que estão instalados (BRÖRING et al., 2011). Uma estrutura em IoT utiliza-se das RSSF para relacionar-se com o ambiente. Assim, algumas características inerentes ao desenvolvimento de uma rede de sensores sem fio são compartilhadas com o conceito mais amplo da IoT.

IoT pressupõe a materialização de objetos inteligentes, pois o monitoramento e a interação entre sistemas passa a acontecer de maneira ubíqua e autônoma. Aplicações de IoT encontram-se nas mais variadas tarefas. Segundo Vermesan e Friess (2014), algumas atividades de destaque são o consumo inteligente de energia, monitoramento eficiente da saúde de pacientes, consumo adequado de insumos industriais e transporte de cargas e pessoas, além da construção de casas, edifícios e cidades inteligentes. A Figura 1 apresenta a matriz em 3 (três) dimensões de uma possível estrutura em IoT. Neste trabalho, o destaque será atribuído às RSSF.

Com o desenvolvimento das tecnologias empregadas em RSSF, as pesquisas e esforços estão sendo conduzidos no sentido de integrar estas redes à Internet. A utilização de identificadores únicos por meio dos protocolos IPv6 (*Internet Protocol Verion 6*) e o

6LoWPAN (*Low power Wireless Personal Area Networks*) vem possibilitando a união destes dois paradigmas – RSSF e IoT. Ao integrar estas diferentes redes, torna-se factível a incorporação dinâmica de sensores e atuadores à Internet sem a necessidade de intervenção humana. Esta integração possibilita uma cooperação mútua no sentido de melhorar o desempenho da aplicação. Por exemplo, um ar condicionando pode interagir com sensores de temperatura instalados em diferentes pontos de uma sala a fim de ajustar de maneira precisa a temperatura ideal para o ambiente. Com esta integração, um veículo, quando identificar o período de revisão, pode automaticamente marcar um horário na concessionária para realizá-la, ou os dispositivos instalados em um paciente monitorado à distância podem, automaticamente, informar ao médico sobre a necessidade ou não da realização de exames complementares. Estas são apenas algumas aplicações geradas pela união das RSSF com a IoT.

Figura 1 – Matriz de relacionamento da IoT



Fonte: Adaptado de Vermesan e Friess (2014)

Quando estes dois paradigmas são integrados, falhas ocorridas nas RSSF podem ser desastrosas para todo o funcionamento de uma estrutura em IoT. Por exemplo, sensores responsáveis por monitorar e informar a iminência de um desastre natural, como um tsunami, quando acometidos por uma falha na leitura dos dados, podem gerar alarmes falsos e acarretar um sentimento de pânico generalizado em toda uma população. Caso deixem de funcionar, podem não avisar as autoridades sobre a iminência de um desastre. Portanto, assegurar o funcionamento adequado desta rede de sensores é fundamental para

a concepção da IoT e para uma maior confiabilidade de seus propósitos.

Segundo Kaur e Kaur (2015), as RSSF, e, consequentemente, IoT, estão sujeitas a falhas de *hardware*, pois são utilizadas em ambientes severos, estando submetidas à ação de uma variedade de agentes químicos e biológicos, além de erros devido ao gerenciamento de energia. Isso ocorre pois, geralmente, os dispositivos sensores possuem apenas baterias como fonte de alimentação, que muitas vezes não podem ser substituídas. As falhas de *hardware* são as principais responsáveis pelas falhas relativas aos dados. As falhas de dados ocorrem quando os sensores realizam medidas equivocadas sobre uma variável monitorada. Outras falhas também ocorrem em decorrência, não apenas do mal funcionamento do *hardware*, mas também da dificuldade de comunicação, uma vez que a intensidade dos sinais é afetada por interferências e obstáculos.

Quando uma falha de dados ocorre, todo o sistema criado pode ser comprometido (KOUSHANFAR; POTKONJAK; SANGIOVANNI-VINCENTELLI, 2009). Um exemplo de falha no sensoriamento remoto que causou grande repercussão ocorreu, em 2009, no voo 447 da Air France. Neste acidente aéreo, um Airbus A330-200 partiu do Rio de Janeiro com destino a Paris, entretanto, caiu no Oceano Atlântico com 228 pessoas a bordo. Segundo o relatório do BEA (Bureau d'Enquetês et d'Analyses), órgão francês responsável por investigar os acidentes aéreos, as velocidades medidas ficaram incoerentes, presumidamente, como resultado da obstrução das sondas Pitot devido ao acúmulo de cristais de gelo. Os tubos de Pitot são elementos sensores utilizados para medir a diferença de pressão em um fluido (BALBINOT; BRUSAMARELLO, 2000) e também são utilizadas para medir a velocidade de aeronaves ou do vento em estações fixas. Este exemplo demonstra o quão sérias podem ser as consequências de falhas em leituras de sensores. Portanto, garantir que a própria rede consiga realizar o gerenciamento desta falha é de suma importância para assegurar a viabilidade das RSSF e possibilitar a integração segura e eficaz com a IoT. Esta característica de autogerenciamento de falhas, somada a auto-configuração, atribui a RSSF um caráter de sistema autonômico. Segundo Computing et al. (2006), a computação autonômica é a utilização da tecnologia para o autogerenciamento de um sistema, a fim de auxiliar na redução da complexidade.

Um sistema de gerenciamento de falhas pode ser dividido em 3 (três) etapas distintas: detecção da falha, diagnóstico da falha e recuperação da falha (YU; MOKHTAR; MERABTI, 2007). A detecção da falha consiste na primeira fase do processo de gerenciamento, quando o erro inesperado do sistema é detectado. A segunda fase, diagnóstico da falha, consiste na identificação correta da falha, ou seja, distinção entre alarmes suspeitos e irrelevantes. A última fase diz respeito à recuperação da falha, que corresponde ao momento em que o sistema é reestruturado e reconfigurado.

Na literatura, muitos são os algoritmos utilizados a fim de identificar, diagnosticar e recuperar o sistema. Alguns autores propõem algoritmos baseados em lógica *Fuzzy*,

por exemplo, Behzadi e Azad (2014). Redes Neurais Artificiais (RNA) são utilizadas por Bokareva, Bulusu e Jha (2006). A aplicação de algoritmos probabilísticos é encontrada no trabalho de Lau, Ma e Chow (2014). Técnicas bioinspiradas para o gerenciamento de falhas estão presentes no trabalho de Salmon et al. (2013). Entretanto, nenhuma das técnicas propostas, é realizada com o objetivo de propor a integração das RSSF à IoT.

1.1 Justificativa

Alcaraz et al. (2010) indagam sobre qual seria a real necessidade de integração total entre as RSSF e a IoT. Este questionamento é realizado, pois são elencados inúmeros obstáculos que necessitam ser ultrapassados para uma integração eficiente entre estas redes. Partynski e Koo (2013) também apresentam vários desafios inerentes a esta integração que indicam a dificuldade em promover a união destas duas redes. Entretanto, nestes mesmos artigos, os autores relatam as possibilidades que uma integração eficiente entre estes dois paradigmas pode gerar. Não é difícil encontrar na literatura trabalhos que exemplificam as oportunidades criadas por esta integração como: Chui, Löffler e Roberts (2010), Liu e Wassell (2011), Gubbi et al. (2013) e Vermesan e Friess (2014). Assim, esforços no sentido de possibilitar a integração adequada destas redes são necessários a fim de garantir um monitoramento ubíquo e conveniente do ambiente.

Neste contexto, criar um sistema autonômico de gerenciamento de falhas que visa integrar de maneira eficiente a RSSF com a Internet, por meio de protocolos modernos de comunicação, é de suma importância para a utilização eficiente da IoT. Portanto, este trabalho busca contribuir com a tarefa de tornar possível algumas das oportunidades criadas pela IoT.

1.2 Definição do Problema

Devido às características de concepção e aplicação de uma RSSF, Gaynor et al. (2004), Tejasvit (2014), Partynski e Koo (2013) elegem as falhas dos dados como um dos principais desafios existentes para a integração das RSSF com a Internet e, consequentemente, com a IoT. Isso ocorre pois as RSSF são, normalmente, empregadas em ambientes severos e de difícil acesso, além de os nós responsáveis pelo monitoramento serem constituídos por *hardware* de baixo custo, exígua capacidade de processamento e fonte de energia limitada.

Na literatura, existem modelos de detecção de falhas classificados como centralizados e descentralizados. Os sistemas centralizados são vistos, normalmente, como sistemas de menor eficiência energética, pois apresentam um maior número de mensagens trocadas na rede. Neste contexto, tem-se preferido utilizar sistemas descentralizados de gerencia-

mento de falhas. Entretanto, uma das grandes vantagens de uma abordagem centralizada, quando comparada com o modelo descentralizado, encontra-se na menor complexidade de implementação. Vários autores como Elshaikh et al. (2014), Alam, De e Ray (2015), Boonsongsrikul, Kocijancic e Suppharangsan (2013) relatam que a concepção de um modelo centralizado de baixo consumo energético constitui um desafio para os modelos de gerenciamento de falhas em RSSF. Quando uma RSSF integra uma aplicação IoT, a quantidade de mensagens trocadas entre os nós aumenta, pois não é apenas o usuário da aplicação que realiza consultas aos nós sensores. Cada nó (objeto físico, humano, máquina etc) conectado à rede local por meio da Internet (e que possua permissão para consumir o serviço) pode solicitar os dados. Logo, em todo instante, requisições são recebidas e enviadas. A cada dia, o número de dispositivos conectados à Internet sofre um aumento considerável. Estima-se que, em 2020, existirão mais de 50 bilhões de dispositivos conectados (EVANS, 2011). Devido à heterogeneidade destes dispositivos (sensores, atuadores, utensílios domésticos, bicicletas, carros etc) e serviços, a complexidade de todo este sistema também tende a aumentar. Portanto, implementar uma abordagem de reduzida complexidade computacional auxilia a integração de redes heterogêneas ao modelo de IoT sem a necessidade de aumentar a obscuridade do sistema. Neste contexto, um tratamento centralizado de falhas que garanta taxas de falsos positivos e negativos semelhantes a outros modelos, assegurando um consumo de energia equivalente a uma abordagem descentralizada, apresenta-se como um desafio para integração das RSSF com a IoT.

Segundo Computing et al. (2006), outro aliado na redução da complexidade corresponde a adoção de um sistema autonômico. Por meio de um sistema autonômico é possível realizar o autogerenciamento de falhas sem a necessidade da interferência humana. Assim, conceber um sistema autonômico de gerenciamento de falhas de dados é de grande importância para o desenvolvimento da IoT.

Muitos são os trabalhos que investigam e sugerem algoritmos de gerenciamento de falhas em RSSF, mas poucos abordam o tema de integração desta abordagem com a Internet, principalmente no que diz respeito aos efeitos destes algoritmos sobre os protocolos de comunicação mais recentes, como o 6LoWPAN. Assim, este trabalho sugere a adoção de uma arquitetura centralizada de gerenciamento de falhas que possibilite uma integração confiável das RSSF com a IoT.

1.3 Objetivos

Como expresso anteriormente, a incorreta identificação de falhas em protocolos sem fio pode ser catastrófica para todo um sistema, especialmente quando uma RSSF está integrada à IoT. Na literatura, existem muitos algoritmos desenvolvidos a fim de promover a detecção, diagnóstico e recuperação de falhas em uma RSSF. Contudo, estes algoritmos

não são empregados de maneira eficiente na integração com redes em IoT. Assim, este trabalho tem como objetivo geral propor um algoritmo centralizado de gerenciamento de falhas aplicado a uma rede de sensores sem fio com integração a IoT. A fim de alcançar este objetivo geral, são traçados os seguintes objetivos específicos:

- Propor um algoritmo centralizado eficiente e eficaz para gerenciamento de falhas (detecção, diagnóstico e recuperação da falha);
- Propor um mecanismo que permita reduzir as trocas de mensagens no gerenciamento centralizado de falhas;
- Aplicar o algoritmo proposto em aplicações simuladas e compará-lo com os modelos existentes na literatura;
- Apresentar uma aplicação de arquitetura IoT que utiliza-se dos protocolos de comunicação mais recentes e que possibilite a integração com o sistema centralizado de gerenciamento de falhas proposto.

1.4 Organização Deste Trabalho

Este documento é organizado em outros 7 (sete) capítulos. O Capítulo 2 apresenta uma introdução aos conceitos utilizados neste trabalho. O Capítulo 3 ilustra e discuti o estado da arte no gerenciamento de falhas aplicados em RSSF e IoT. O Capítulo 4 apresenta a modelagem do sistema de gerenciamento de falhas proposto. O Capítulo 5 apresenta a metodologia utilizada, indicando quais são os experimentos realizados e como foram conduzidos. No Capítulo 6, os resultados obtidos, após a realização dos testes executados, são apresentados. O Capítulo 7 apresenta as considerações finais sobre o trabalho, apontando as dificuldades encontradas, as contribuições e os possíveis trabalhos futuros.

2 Fundamentação Teórica

Este capítulo apresenta os conceitos necessários para o desenvolvimento deste trabalho, uma vez que apresenta as definições e características de cada elemento presente no texto com foco nos conceitos centrais para o entendimento desta dissertação. É realizada uma revisão dos conceitos presentes na literatura a fim de facilitar o entendimento do objetivo geral deste texto e da metodologia empregada neste projeto de pesquisa.

2.1 Sistemas Autonômicos

Em 2001, a *International Bussines Machine* (IBM) publicou um manifesto, apontando que o principal obstáculo para o desenvolvimento da Tecnologia da Informação (TI) seria a crise gerada pelo aumento da complexidade dos sistemas. Para a IBM, as dificuldades de gerenciamento de um sistema tendem a aumentar, consideravelmente, a medida que o desenvolvimento tecnológico introduz novos níveis de complexidade. Isto ocorre pois é necessário lidar com a heterogeneidade dos novos sistemas e aplicações.

Para a IBM, a criação de sistemas autonômicos, apresenta uma poderosa ferramenta para a redução da complexidade de gerenciamento de uma aplicação. Os sistemas autonômicos representam um conjunto de elementos de programação interligados com a capacidade de autogerenciamento e adaptação a mudanças imprevisíveis, o que permite a redução da complexidade de gerenciamento e melhor utilização dos recursos (KEPHART; CHESS, 2003).

Um sistema autonômico possui 4 (quatro) principais características (LALANDA; MCCANN; DIACONESCU, 2013):

- Autoconfiguração: capacidade do sistema em definir e redefinir os seus parâmetros internos de modo a estar em conformidade com as condições iniciais de implantação, adaptando-se às mudanças do ambiente;
- Autorrecuperação: o próprio sistema deve detectar, isolar e reparar os componentes defeituosos, a fim de aumentar a confiabilidade;
- Auto-otimização: o sistema, de maneira proativa, deve se esforçar para otimizar o aproveitamento de recursos a fim de alcançar os objetivos iniciais traçados;
- Autoproteção: o sistema deve, antecipadamente, identificar e prevenir vários tipos de ameaças a fim de preservar sua integridade e segurança.

Estas características são interdependentes, por exemplo, a autoconfiguração é responsável por iniciar todos demais serviços, e a tarefa de autorrecuperação, disponibiliza para a tarefa de autoproteção, elementos que auxiliarão na tarefa de identificar possíveis

ameaças.

Com a adoção de sistemas autonômicos, é possível tratar a crescente complexidade dos sistemas computacionais, por meio da automatização do processo de gerenciamento. Ao realizar o processo de gerenciamento de maneira automática, a necessidade de intervenção humana é reduzida. A adoção de um sistema automático, normalmente, tem por objetivo:

- Melhorar a experiência da interação com usuário, pois a qualidade do serviço é melhorada;
- Reduzir os custos de manutenção, uma vez que diminui a necessidade de intervenção humana;
- Diminuir os custos de utilização, pois existe um melhor gerenciamento dos recursos.

Estas qualidades de um sistema autonômico são fundamentais para a concepção da IoT, já que, lidar com a heterogeneidade de bilhões de dispositivos, é um dos principais desafios à expansão da IoT. Com o gerenciamento eficiente de recursos e diminuição da necessidade de intervenção humana a complexidade da tarefa de administração do sistema tende a ser reduzida, o que é essencial para o desenvolvimento da IoT.

2.2 Falhas em RSSF

Koushanfar, Potkonjak e Sangiovanni-Vincentell (2002) indicam que um erro é a manifestação da falha dentro de um programa. Assim é possível entender que o erro pode se manifestar fora do local onde ocorreu a falha. Por exemplo, uma falha de um tubo de Pitot pode gerar medidas erradas para um sistema de controle de altitude em uma aeronave e ocasionar, em casos extremos, a queda desta aeronave. Assim, uma falha de *hardware* gerou o colapso do sistema. Falha é um estado incorreto do *hardware* ou programa em decorrência de incapacidade funcional momentânea ou permanente de um componente.

As falhas em RSSF podem ser classificadas em 4 (quatro) diferentes grupos de acordo com o local onde ocorrem: falhas relacionadas à rede, aos nós, ao *sink-node* ou roteador de borda e falhas relativas aos dados (PARADIS; HAN, 2007; JURDAK et al., 2011).

- Falhas relativas à rede: este tipo de falha ocorre quando existe algum tipo de erro de comunicação dentro da rede. Este tipo de falha pode ocorrer devido a vários fatores como interferências, colisões de mensagens, perdas de rota ou problemas nos rádios receptores ou transmissores.
- Falhas relativas aos nós: ocorrem quando um ou vários nós de uma rede apresentam falhas. Os nós podem ser destruídos por animais, pessoas ou variações climáticas. As fontes de alimentação podem ser degradadas. Falhas de *software*, como vazamento

de memória, podem também levar a falhas nos nós.

- Falhas relativas ao roteador de borda: decorrem dos mesmos fatores apresentados para as falhas em nós, entretanto, acometem os nós com maior capacidade de processamento de uma rede (roteadores de borda) responsáveis por manter a conectividade e gerenciamento do sistema. Falhas deste tipo podem comprometer toda a estrutura de uma rede.
- Falhas relativas aos dados: são evidenciadas por meio da discrepância entre os dados coletados pelos sensores e os dados reais. Neste tipo de falhas as principais causas são as falhas de *hardware*, calibração incorreta dos sensores ou interferências na comunicação.

Como pode ser visto, a identificação da origem da falha em uma RSSF é uma tarefa bastante dispendiosa, pois um mesmo erro pode ser advindo de falhas de *hardware*, *software* ou retirada do nó da rede. Assim, o Sistema Centralizado Inteligente de Gerenciamento de Falhas (SCIGF) proposto, visa tratar as falhas relativas aos nós, aos roteadores de borda e aos dados, pois estas falhas estão intimamente relacionas a problemas de mal funcionamento de *hardware*. Falhas de comunicação não são tratadas, pois é considerado que a troca de mensagens entre os nós ocorra sem interferências e que os dados recebidos representam, realmente, o dado lido pelo nó.

2.3 Redes de Sensores Sem Fio (RSSF)

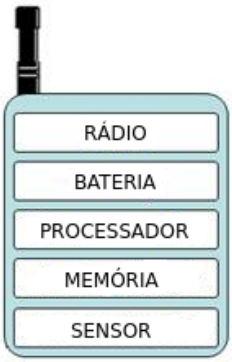
Segundo Fahmy (2016), RSSF são redes compostas por dezenas a milhares de dispositivos autônomos com capacidade de colher e disseminar dados por meio de comunicação sem fios. Uma RSSF é concebida como resultado da combinação das técnicas de sensoriamento, sistemas embarcados, processamento distribuído da informação e mecanismos de comunicação. Em uma RSSF, usualmente, os sensores apresentam baixas capacidades de processamento, possuem baterias como fonte de alimentação e, portanto, disponibilidade limitada de energia, além de estarem, normalmente, instalados em ambientes perigosos ou de difícil acesso.

2.3.1 Arquitetura das RSSF

A arquitetura básica de RSSF proposta por Fahmy (2016) é composta por sensores e/ou atuadores, estação base (*sink*), pelo fenômeno de interesse, ou seja, a aplicação e o observador. Os nós sensores de uma rede, devido às características de funcionalidade, podem ser vistos como dispositivos de baixa potência, equipados com um ou mais sensores e/ou atuadores, um processador, memória, fonte de alimentação, uma interface de rádio e outros componentes adicionais destinados à aplicação. A estrutura de *hardware* de um nó e um exemplo de dispositivo real podem ser vistos na Figura 2.

Figura 2 – Representação esquemática da arquitetura básica.

(a) Modelo de *hardware* em um nó sensor.



(b) Modelo real de um nó sensor.

Fonte: Zolertia (2016).

Segundo Kahn, Katz e Pister (1999), além destas características de construção, um nó sensor deve seguir algumas especificações, como:

- Possuir baixo consumo energético;
- Possuir baixo custo de produção e ser de fácil substituição (dependendo da aplicação);
- Possuir características de autonomia, por exemplo, quando identificado com falhas, automaticamente, desligar-se da rede;
- Ser adaptado ao ambiente, por exemplo, um nó subaquático deve ser hermeticamente protegido a fim de evitar danos aos componentes eletrônicos.

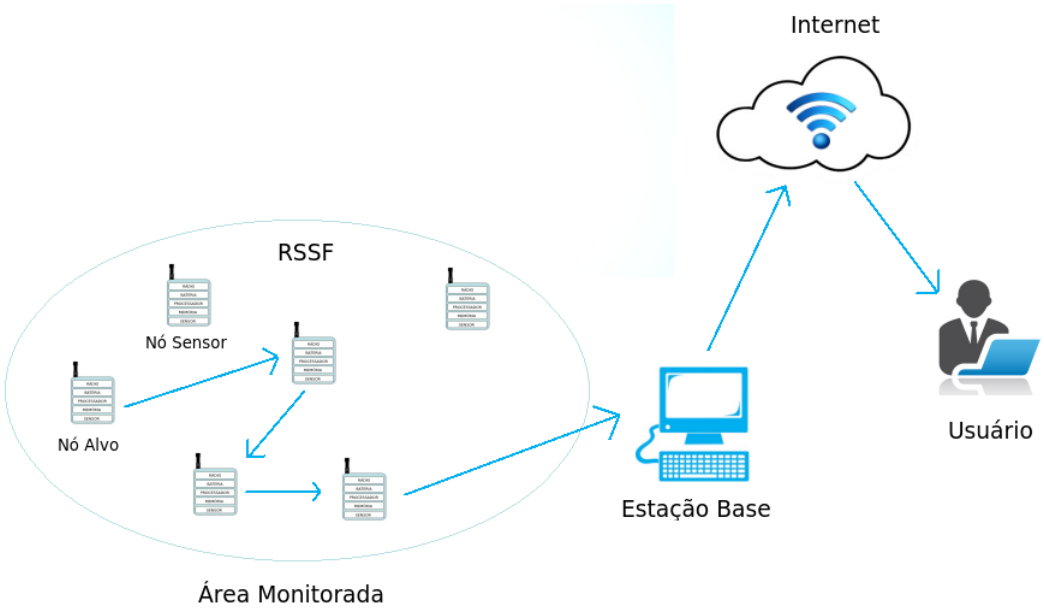
Pela Figura 3, é possível ver um exemplo de RSSF com os principais componentes. Uma estação base (ou *sink*) corresponde ao elemento responsável por estabelecer a comunicação com o usuário seja via Internet ou satélite. Normalmente, essa estação é mantida nas proximidades da rede. O observador é o usuário que deseja obter informações sobre a aplicação.

2.3.2 Padrões em RSSF

Segundo Ferreira (2004), a palavra “padrão” significa uma base de comparação consagrada como modelo de consenso geral ou por determinado órgão oficial. Em RSSF, vários são os padrões que definem as funções e protocolos necessários para assegurar uma interoperabilidade entre as diferentes configurações de redes e produtos disponibilizados no mercado.

Especialmente em RSSF, esta padronização tem concentrado esforços no sentido de prover sistemas de baixo consumo energético. Dentre os grupos de trabalhos para construção de padrões internacionais destaca-se o *Institute of Electrical and Electronics Engineers* (IEEE) 802.15. Esse grupo é responsável por definir diretrizes para *Wireless*

Figura 3 – Modelo de arquitetura em RSSF.



Fonte: Adaptado de Fahmy (2016).

Personal Area Network (WPAN) ou rede pessoal sem fio. Seguindo-se o modelo de referência *Open Systems Interconnection* (OSI) em todos os padrões conhecidos como IEEE 802.15.X, são definidas apenas as camadas física e de enlace. Exemplos de alguns IEEE 802.15.X são: 802.15.4 (utilizada por ZigBee (ALLIANCE et al., 2013) e 6LoWPAN (SHELBY; BORMANN, 2011)) e 802.15.1 (WPAN/Bluetooth).

O padrão IEEE 802.15.4 define o canal físico de comunicação em baixa potência por meio de radiofrequências a 2,4 GHz, 915,0 MHz e 868,0 MHz e, mais recentemente, introduziu comunicações em frequências de 3,1-10,6 GHz. Para a camada de enlace, propõe taxas de transmissão de 20-250 kbits/s, dependento da frequência utilizada, e acesso ao canal de comunicação utilizando-se *Carrier Sense Multiple Access* (CSMA). Molisch et al. (2004) apresentam outras informações sobre o padrão IEEE 802.15.4.

2.4 Internet das Coisas (IoT)

Desde a primeira aparição do termo IoT, em Brock (2001), pesquisadores e indústria têm reunido esforços no sentido de implementar este novo paradigma. Estima-se que este mercado vai gerar, até o ano de 2020, mais de US$ 9 trilhões apenas com vendas de dispositivos aptos a integrarem a IoT (NEDELTCHEV, 2014). No decorrer dos anos, o conceito formal deste paradigma tem sofrido modificação, seja por pesquisadores ou estratégias de *marketing* em vendas. A primeira definição, presente em Brock (2001), elucida a IoT como uma estrutura inteligente capaz de conectar objetos, informação e pessoas por

meio da rede mundial de computadores. Esta definição sugere que máquinas e humanos consigam acessar, de maneira ubíqua, qualquer informação necessária de um sistema, uma vez que os objetos físicos estão inteiramente conectados à Internet.

Quando seres humanos estão conectados à Internet, questões relativas à segurança, privacidade, integridade dos dados, padronização de protocolos e identificação única (*Internet Protocol* - IP) são características primordiais para uma comunicação eficiente. Ao conectar objetos físicos do nosso cotidiano à Internet, além das características mencionadas anteriormente, algumas questões específicas de concepção deste modelo de comunicação (entre objetos/objetos e pessoas/objetos) passam a ser relevantes. Problemas relativos à interoperabilidade, escalabilidade, confiabilidade, auto-organização, latência (especialmente em aplicações de tempo real) e, principalmente, tolerância a falhas tornam-se pontos centrais na concepção da IoT (BREIVOLD; SANDSTR et al., 2015).

2.4.1 Arquitetura de Aplicações em IoT

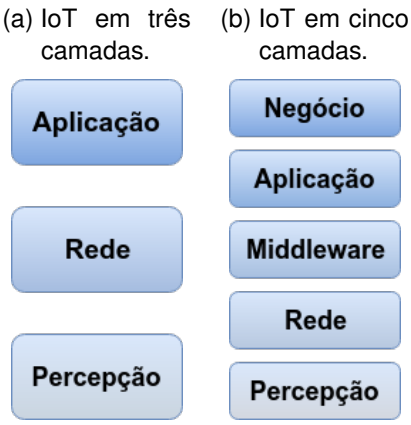
Assim como acontece com a definição do paradigma IoT, existe uma diversidade de arquiteturas propostas a fim de indicar como as diferentes tecnologias associadas à IoT estão relacionadas. O padrão de arquitetura em IoT pode ser muito complexo, dependendo da aplicação, pois envolve modelos de interconexão entre vários protocolos e serviços de camadas distintas além da união entre diferentes redes de comunicação. Uma aplicação IoT deve ser capaz de conectar bilhões de dispositivos heterogêneos. Yang et al. (2011) propõem uma arquitetura para IoT em 3 (três) camadas: camada de percepção, camada de rede e camada de aplicação.

1. Camada de percepção: esta camada é composta pelos dispositivos sensores e aturadores. Possui como principal atribuição coletar os dados do ambiente (temperatura, umidade, tensão etc) e possibilitar a interação por meio dos atuadores.
2. Camada de rede: nesta camada, são realizadas as operações de transporte da informação. O troca de informação pode ocorrer dentro de uma rede local, entre redes locais ou entre uma rede e a Internet.
3. Camada de aplicação: a principal tarefa realizada nesta camada é mostrar aos usuários (humanos, máquinas ou objetos) quais os serviços estão disponíveis e possibilitar a utilização destes serviços.

Uma abordagem mais completa e atual sobre a arquitetura em IoT é proposta por Khan et al. (2012). Nesse tratamento, são introduzidas outras duas camadas: camada de *middleware* e camada de negócio, além de propor algumas mudanças na concepção das outras camadas. A Figura 4 apresenta uma comparação entre estas arquiteturas propostas.

No modelo proposto por Khan et al. (2012), cada uma das 5 (cinco) camadas apresenta características bem definidas e alguns dos protocolos mais utilizados também

Figura 4 – Comparação entre diferentes abordagens para arquitetura proposta em IoT.



Fonte: Adaptado de Khan et al. (2012).

são listados. As funções definidas por cada uma das camadas são:

1. Camada de percepção: também conhecida como camada de dispositivos. Assim como no modelo anterior, é composta pelos objetos físicos responsáveis pelo sensoriamento do ambiente. Essa camada lida, basicamente, com a identificação única dos dispositivos e coleta de informação do ambiente. Também é responsabilidade desta camada transmitir os dados para a camada de rede. A grande massa de dados, ou *Big Data*, é gerada por essa camada. Os protocolos mais utilizados são o *Radio-Frequency IDentification* (RFID) e uCode.
2. Camada de rede: também é conhecida como camada de transmissão. Essa camada garante a transmissão segura da informação dos dispositivos para o sistema de processamento da informação. O meio de transmissão pode ser cabeado ou sem fio. Os protocolos mais utilizados são o *Bluetooth Low Energy*, infravermelho, ZigBee e 6LoWPAN.
3. Camada middleware: também conhecida como camada de gerenciamento de serviço. É responsável por gerenciar as requisições de serviços sendo encarregada de realizar a entrega dos serviços solicitados por meio do endereçamento e nomes dos dispositivos. Esta camada também é a responsável por manter a transparência entre a heterogeneidade das redes, uma vez que o programador não necessitará programar para um sistema de *hardware* específico. Também é tarefa dessa camada receber os dados da camada de rede e os armazenar nos bancos de dados. Exemplos de protocolos utilizados são *Multicast Data Distribution Service* (mDNS) e *Data Distribution Service - Service Discovery* (DNS-SD).
4. Camada de aplicação: proporciona o gerenciamento global da aplicação baseada nas informações processadas pela camada de gerenciamento de serviço. É responsável por entregar os serviços aos consumidores. Por exemplo, disponibilizar os dados de temperatura e umidade para uma estação meteorológica. Exemplos de protocolos

utilizados são o *Constrained Application Protocol* (CoAP), o *Message Queue Telemetry Transport* (MQTT) e o (*Data Distribution Service* (DDS).

5. Camada de negócio: é responsável por construir o modelo de negócio, gráficos, diagramas de fluxo etc, por meio dos dados recebidos da camada de aplicação. Essa camada provê suporte a decisão baseada na análise do *Big Data*, além de realizar o gerenciamento e o monitoramento das camadas inferiores. As análises de alto nível são realizadas por essa camada. Normalmente, é hospedada em dispositivos com maior poder de processamento.

A arquitetura em 5 (cinco) camadas será utilizada nesta dissertação para demonstrar a integração do sistema de gerenciamento de falhas proposto com a IoT. A escolha dessa arquitetura deve-se ao fato dela possibilitar a maior integração entre os diferentes protocolos e garantir uma abstração mais abrangente para o usuário final.

2.5 Integração entre RSSF e IoT

Por meio da integração entre as RSSF e a IoT, como expresso na seção 1.1, torna-se possível o desenvolvimento de várias aplicações que, anos atrás, estavam presentes apenas em livros de ficção. Atividades de monitoramento em tempo real da cadeia de suprimentos e equipamentos, rastreamento e acompanhamento de ecossistemas degradados ou ameaçados de extinção e desenvolvimento de redes com microssensores capazes de medir a concentrações de elementos químicos no corpo humano e diagnosticar problemas são apenas algumas aplicações favorecidas com a integração entre estas duas redes.

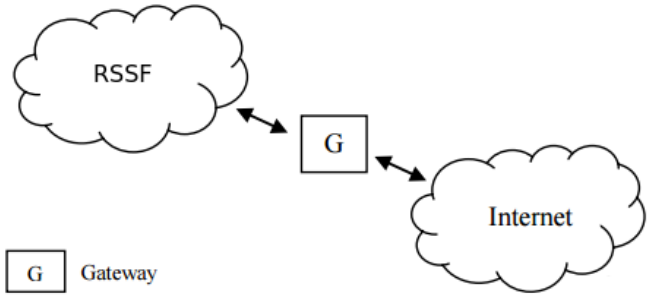
Como apresentado no artigo de Christin et al. (2009) e exposto nas Figuras 5a, 5b e 5c, são evidenciadas algumas estratégias de integração entre RSSF e a Internet. Na Figura 5a a integração ocorre de maneira independente por meio de um *gateway*. As Figuras 5b e 5c exemplificam a integração por meio de modelos mais complexos. Na Figura 5b, cada nó da rede está conectado à Internet mediante um *gateway* e, no modelo da Figura 5c, a comunicação é realizada por meio de uma rede IEEE 802.15.4 como *access point*. Portanto, em 5c, múltiplos nós podem acessar a Internet por meio de um único nó. Entretanto, como exposto na seção 1.1, Alcaraz et al. (2010) e Christin et al. (2009), indicam que esta abordagem não pode ser vista como fixa ou incontestável, pois, em determinadas aplicações, este tratamento não apresenta uma estrutura adequada. Uma aproximação, recentemente utilizada, versa sobre a união destas duas redes por meio de protocolos que integram as camadas de rede das RSSF e a Internet.

2.5.1 Oportunidades Geradas Pela Integração Entre RSSF e IoT

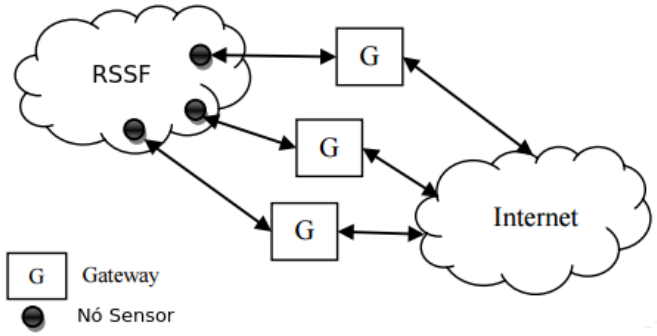
A quantidade de dados gerados pelos sensores espalhados pelo mundo, realizando as mais variadas funções e controlando uma enormidade de processos tende a crescer

Figura 5 – Modelos de integração entre RSSF e a Internet.

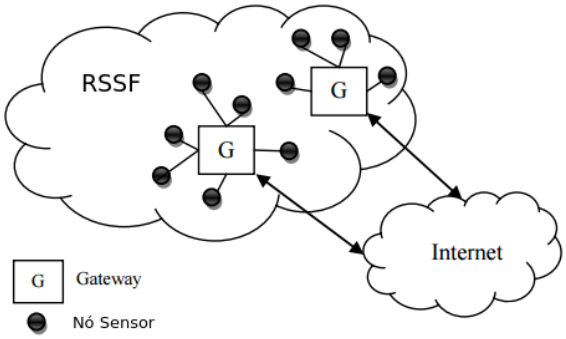
(a) Modelo de integração - rede independente.



(b) Modelo de integração - rede híbrida.



(c) Modelo de integração - rede com *access point*.



Fonte: Adaptado de Christin et al. (2009).

exponencialmente devido a maior utilização de dispositivos inteligentes. Quando uma RSSF não está conectada à Internet, algumas dificuldades como o armazenamento da informação, a acessibilidade de recursos e análise em tempo real não são, eficazmente, ultrapassadas (LIU; WASSELL, 2011).

Ao integrar uma RSSF com a IoT é possível que exista um armazenamento remoto da informação, por meio da *Cloud Computing*, por exemplo. Uma vez que, os sensores de uma rede, possuem disponibilidade reduzida de recursos, a capacidade de armazenamento de grandes quantidades de dados passa a ser essencial para o monitoramento de um processo. Quando o acesso a estes dados ocorre de maneira segura, por meio da Internet, não

existe a necessidade da aplicação e o usuário estarem em uma mesma região geográfica, o que facilita a troca de informação. Ao disponibilizar estes dados, é possível que o gerente da aplicação possa tomar decisões e atuar no sistema em tempo real.

Segundo Chui, Löffler e Roberts (2010), a junção de sensores e dispositivos de comunicação a objetos do nosso cotidiano ocasiona o surgimento de um novo cenário produtivo, em que são criadas novos modelos de negócio, melhorias nos processos, redução dos custos e riscos na produção. Para Chui, Löffler e Roberts (2010), a união das RSSF com a IoT ocasiona o rastreamento adequado de produtos, melhora a compreensão do ambiente, subsidia a tomada de decisões, otimiza processos e consumo de recursos, além de possibilitar a criação de sistemas autônomos complexos.

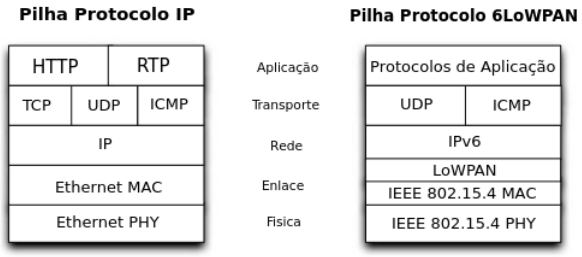
Com o rastreamento de produtos, as companhias podem analisar os movimentos das mercadorias e também monitorar a interação entre estes e seus consumidores. De posse destes dados, é possível sintonizar o modelo de negócio segundo as necessidades do cliente. Quando sensores estão integrados, o monitoramento ambiental torna-se mais eficiente, uma vez que cada elemento pode utilizar-se de dados de outros sensores para melhorar a acurácia de um processo, o que dá suporte ao planejamento e concepção de aplicações cada vez mais complexas. Com a adoção da IoT, uma RSSF pode, por meio de um mecanismo de realimentação, fornecer informação sobre a disponibilidade de recursos (energia, por exemplo) e assim alterar os padrões de consumo a fim de reduzir, automaticamente, o gasto desnecessário.

Segundo Vermesan e Friess (2014), a IoT tem o potencial de mudar, radicalmente, nossa vida cotidiana, pois, praticamente, não existe um cenário em que, o emprego da IoT, não gera a quebra de paradigmas. Mas, para ocasionar essa ruptura, é necessário que algumas barreiras como a eficiente integração com as redes de comunicação, ocorra de maneira adequada. Assim, integrar de modo eficiente as RSSF com a IoT é essencial para o desenvolvimento de todo o potencial gerado pela união destas das tecnologias.

2.5.2 Protocolo 6LoWPAN

6LoWPAN é um padrão capaz de utilizar eficientemente o protocolo IPv6 sobre uma plataforma de baixo consumo energético em redes sem fio. O 6LoWPAN é empregado, principalmente, em dispositivos embarcados simples, por meio de adaptações de algumas camadas e otimização de alguns protocolos (SHELBY; BORMANN, 2011). Como dito anteriormente, o 6LoWPAN utiliza-se do padrão IEEE 802.15.4 para definir as camadas físicas e de enlace, mas com uma pequena modificação na segunda camada *Media Access Control* (MAC). Esta diferença, presente na Figura 6, deve-se à introdução de uma subcamada entre as camadas MAC e de rede. Esta modificação visa possibilitar o encapsulamento dos pacotes IPv6, ocasionando uma redução do consumo de energia e a integração entre as camadas superiores da pilha de protocolos IPv6. Segundo Shelby e Bormann (2011), a utili-

Figura 6 – Diferenças entre as pilhas de protocolos IP e 6LoWPAN.



Fonte: Adaptado de Shelby e Bormann (2011).

zação deste protocolo deve-se, principalmente, às seguintes características de integração com o protocolo IP:

- A concepção de dispositivos baseados em protocolos IP, que podem ser conectados, facilmente, a outras redes IP sem a necessidade de utilização de um *gateway*, por exemplo;
- A concepção de dispositivos baseados em IP possibilita a utilização de toda a estrutura existente para a rede Internet;
- Como a tecnologia IP é especificada como um protocolo aberto, o desenvolvimento de novos produtos e tecnologias é facilitado;
- Ferramentas que são empregadas na rede mundial de computadores (por meio do IPv6) são bem conhecidas e utilizadas. Logo, podem ser empregadas sem grandes dificuldades.

2.5.3 Arquitetura 6LoWPAN

Devido às características de utilização, o protocolo 6LoWPAN foi concebido para ser empregado em sistemas que possuem uma baixa capacidade de processamento, disponibilidade reduzida de energia e que necessitem ser conectadas à Internet. Portanto, este é um protocolo a ser utilizado quando uma ilha em rede de sensores sem fio necessita ser conectada à Internet ou a outra rede. No protocolo 6LoWPAN, existem três configurações distintas de distribuição e conexão dos dispositivos. A Figura 7 apresenta estes três níveis de interação.

A rede LoWPAN simples é conectada por meio de um roteador de borda (*Border Router*) a outra rede IP. Em uma configuração do tipo LoWPAN estendida, a rede é conectada à Internet por meio de múltiplos roteadores de borda. Diferentemente das duas primeiras configurações de redes anteriores, em uma configuração LoWPAN *Ad hoc*, a rede não é conectada à Internet e, portanto, existe apenas o tráfego de mensagens internas.

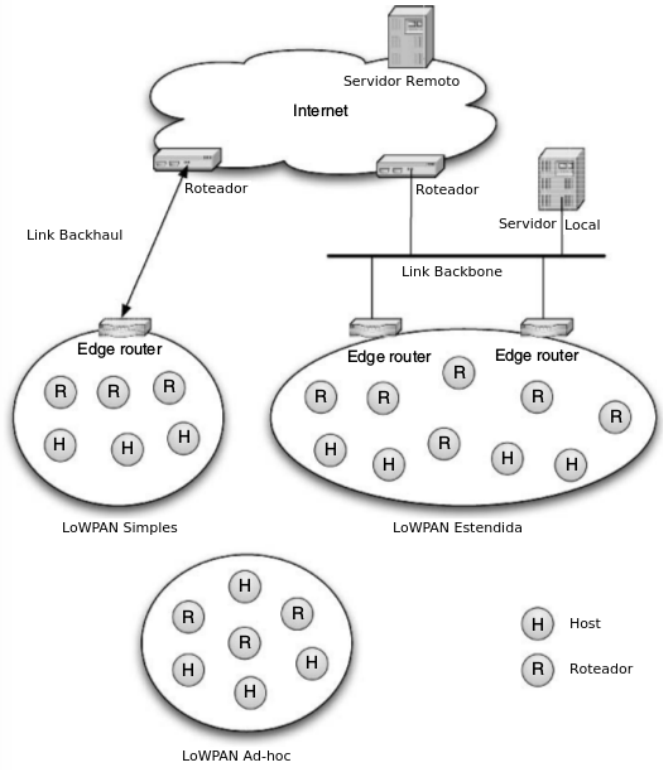
Na Figura 7, é possível verificar a existência de três elementos fundamentais na concepção do modelo 6LoWPAN : o *Host*, o roteador (*Router*) e o roteador de borda (*Edge*

Router ou *Border Router*). O *Host* corresponde ao elemento sensor que está no nível mais baixo do sistema, normalmente, com menor capacidade de processamento e responsável apenas por obter dados sobre a aplicação e enviá-los ao roteador. O roteador, como o nome sugere, possui as tarefas de obter dados do ambiente e rotear a troca de mensagens até o roteador de borda. O roteador de borda é o principal elemento responsável por integrar uma rede LoWPAN isolada à Internet. O roteador de borda é o dispositivo com maior capacidade de processamento da rede e é por essa razão que as funções de configuração, sincronização, autenticação dos demais nós e encapsulamento dos pacotes são realizadas por este elemento.

2.6 Gerenciamento de Falhas

Mesmo sendo um protocolo bastante robusto e integrado às mais recentes tecnologias, o 6LoWPAN não define um modelo de gerenciamento de rede. Esta característica se deve ao fato do comportamento imprevisível de uma rede de sensores sem fio e ao caráter particular da aplicação na qual uma RSSF é empregada (LOUREIRO et al., 2003). Por exemplo, uma rede de sensores sem fio desenvolvida para o monitoramento de um sistema subterrâneo possui características distintas de um sistema de sensoriamento oceânico.

Figura 7 – Arquitetura 6LoWPAN.



Fonte: Adaptado de Shelby e Bormann (2011).

Esta diferença substancial entre estas duas redes é evidenciada pelo ambiente em que a RSSF será inserida (ambiente subterrâneo ou subaquático).

Portanto, desenvolver um conjunto de serviços que visem promover a gestão, acompanhamento e controle adequado do comportamento de uma RSSF, além de integrar de maneira adequada o sensoriamento remoto com o usuário, pode ser de suma importância para a utilização das RSSF. A esse conjunto de serviços dá-se o nome de gerenciamento da rede (LEE; DATTA; CARDELL-OLIVER, 2006). Um sistema de gerenciamento em RSSF integra configuração, operação, administração, segurança e manutenção de todos os elementos e serviços de uma rede de sensores.

Os sistemas de gerenciamento são divididos em várias áreas funcionais como: gestão de configuração, segurança, desempenho e falhas. Como expostos na seção 1, o gerenciamento de falhas é uma das áreas funcionais de maior importância para a integração de sensores à IoT.

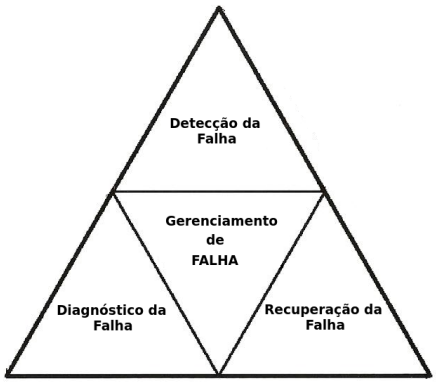
Segundo Loureiro et al. (2003), o gerenciamento de falhas consiste em prover um conjunto de funções que possibilitem à rede detectar, isolar e corrigir um comportamento anormal. As RSSF são compostas por dispositivos de baixo custo, com dimensões reduzidas, baixa capacidade de processamento, memória modesta e fonte limitada de energia (usualmente baterias). Somando-se a estas características dos nós sensores o fato de as RSSF serem redes autônomas, normalmente empregadas em ambientes de difícil acesso (redes aquáticas, subterrâneas e monitoramento de produtos químicos) e necessitarem permanecer em funcionamento por um longo período de tempo, tem-se um cenário em que a probabilidade de falha em uma RSSF é maior que a verificada em outras redes. Portanto, o gerenciamento desta falha é fundamental para o funcionamento correto de toda a rede.

Khan et al. (2010) elencam as principais características necessárias para um sistema eficiente de gerenciamento de falhas em RSSF:

- Definir os limites para as condições potenciais de falhas;
- Monitorar, constantemente, o sistema e o nível de utilização;
- Promover diagnósticos gerais do sistema;
- Notificar qualquer ocorrência de erros ou mal funcionamento;
- Indicar a região onde ocorreu este mal funcionamento;
- Corrigir autonomamente os problemas;
- Manter a probabilidade de alarmes falsos a menor possível;
- Recuperar o sistema após possíveis falhas.

A Figura 8 apresenta as três principais ações a serem realizadas pelo sistema de gerenciamento de falhas. A detecção da falha é a primeira etapa para o gerenciamento, pois é nesta fase que a falha inesperada é identificada. A segunda fase corresponde ao diagnóstico da falha que é equivalente ao processo de distinguir se esta falha é realmente

Figura 8 – Gerenciamento de falhas.

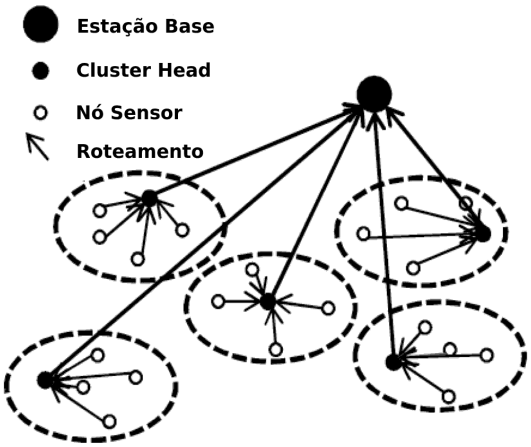


Fonte: Adaptado de Khan et al. (2010)

relevante para o funcionamento do sistema (distinção entre o próprio e o não próprio). A última etapa corresponde à recuperação do sistema, que é o processo de reestruturação e reconfiguração da rede após a falha.

A abordagem de um sistema de gerenciamento de falhas pode ocorrer de 4 (quatro) diferentes maneiras: centralizada, distribuída, distribuída baseado em agente móvel (*agent-based*) e hierárquica (KHAN et al., 2010). Na abordagem centralizada, a tarefa de monitorar e gerenciar possíveis falhas de toda a rede é realizada pela estação base. Esse tratamento possui como principal vantagem a baixa complexidade de implementação. Entretanto, gera um maior consumo de energia, pois todas as mensagens trocadas na rede necessitam ser transmitidas para análise na estação base. Em um gerenciamento distribuído, a rede é dividida em grupos de gerência que, cooperativamente, realizam a gestão das falhas. A maior vantagem desta abordagem, quando comparada com o modelo centralizado, está na redução da troca de mensagens dentro da rede, o que reduz o consumo de energia. Todavia, a complexidade de implementação é maior. Por meio do tratamento distribuído baseado em agente móvel (*agent-based*), um agente (nó com funções definidas de monitoramento) é utilizado para percorrer a rede e coletar informações sobre o real estado do sistema. A abordagem hierárquica possui características dos tratamentos centralizados e distribuídos. Nessa abordagem, a rede é dividida em níveis de tratamento em que entidades de um nível inferior coletam informações e as transferem para entidades de maior nível hierárquico, que gerenciam determinada região da rede. Um exemplo deste tipo de enfoque é o gerenciamento em *cluster*, em que um nó sensor (chamado de *Cluster Head*) é utilizado para gerenciar um conjunto de outros nós de uma região específica da rede, como mostra a Figura 9.

Figura 9 – Exemplo modelo *cluster head*.



Fonte: Adaptado de Khan et al. (2010).

2.7 Filtro de Kalman

Nesta seção, são abordadas algumas definições necessárias para o entendimento do processo de estimação de estados em um sistema dinâmico por meio da utilização do Filtro de Kalman. Estas definições são utilizadas durante todo o processo de estimação de parâmetros para o modelo de gerenciamento de falhas proposto.

Como nenhum sensor, na prática, realiza a medida exata de uma variável, temos uma incerteza inerente a todo processo de sensoriamento. Por exemplo, a temperatura ambiente da cidade de Belo Horizonte, no dia 27/02/2017, às 15:30, foi registrada como 25 °C e o fabricante do instrumento de medição afirma que existe uma incerteza de 0,5 °C para mais ou para menos. Assim, o valor real da temperatura será dado por:

$$T_{observada} = T_{medida} + incerteza \tag{1}$$

Quando se conhece a dinâmica de um sistema (variação temporal da temperatura), pode-se estimar a temperatura atual baseando-se nos estados anteriores do sistema. Por exemplo, sabendo-se que a temperatura de Belo Horizonte do dia 27/02/2017 sofre uma queda a taxa constante de 5% resulta que:

$$T_{prevista} = T_{anterior} \times 0{,}95 + incerteza \tag{2}$$

Assim, há duas equações com características distintas, mas com o mesmo propósito de conhecer a temperatura local. O filtro de Kalman (ou Estimador Linear Quadrático) tem por finalidade integrar estas duas equações (estado real e previsão) a fim de encontrar um valor mais próximo da temperatura real, minimizando-se o valor da incerteza (ruído). Como o estado atual do sistema (temperatura medida) apresenta ruído a previsão do estado

futuro é o estado presente com alguma correção, devida ao modelo, acrescida de um ruído inerente ao processo. Considerando-se que esta incerteza (erro na medição) possui uma distribuição normal ao redor do valor medido, tem-se, na Figura 10, uma representação gráfica da variação temporal de uma medida. Em cada passo, existe uma incerteza que está representada pelas curvas normais em cada ponto.

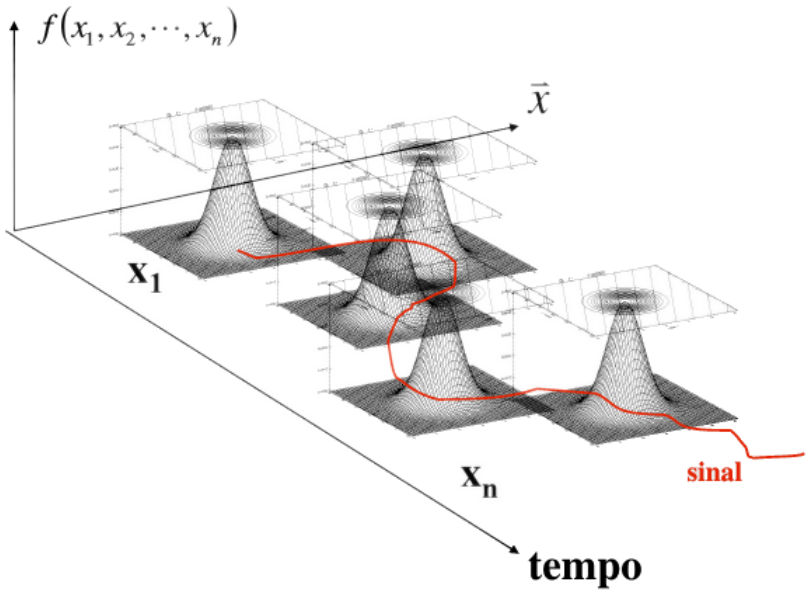
Kalman et al. (1960) e, posteriormente, Kalman e Bucy (1961) propuseram um algoritmo que, sequencialmente, estima estes estados de um sistema dinâmico. O processo de filtragem proposto consiste em uma solução Bayesiana para o método de estimação, na qual a evolução dos estados e as medidas obtidas são ambas lineares e Gaussianas (KOVVALI; BANAVAR; SPANIAS, 2013). Para a explicação matemática do filtro considere o modelo de espaço de estados discretos:

$$\begin{cases} \mathbf{x_n} = \mathbf{F_n x_{n-1}} + \mathbf{v_n}, \\ \mathbf{y_n} = \mathbf{H_n x_n} + \mathbf{w_n}, \end{cases} \tag{3}$$

onde:

$\mathbf{F_n}$: é a matriz de transição de estados *D* x *D* ,
$\mathbf{v_n}$: é o vetor ruído *D* x *1* com distribuição Gaussiana de média zero e matriz de covariância $\mathbf{Q_n}$ associada a $\mathbf{F_n}$,
$\mathbf{H_n}$: é a matriz de medidas do processo *M* x *D*,
$\mathbf{w_n}$: é o vetor ruído *M* x *D* com distribuição Gaussiana de média zero e matriz de covariância $\mathbf{R_n}$ associada a $\mathbf{H_n}$.

Figura 10 – Comportamento temporal das medidas realizadas por sensores.



Considerando que o sistema de equações 3 representa a evolução temporal de estados e medidas de um sistema dinâmico linear e com função de distribuição de probabilidade Gaussiana, pode ser representado como:

$$\begin{aligned} p(\mathbf{x_n}|\mathbf{x_{n-1}}) &\equiv N(\mathbf{x_n};\mathbf{F_n}\mathbf{x_{n-1}},\mathbf{Q_n}), \\ p(\mathbf{y_n}|\mathbf{x_n}) &\equiv N(\mathbf{y_n};\mathbf{H_n}\mathbf{x_n},\mathbf{R_n}), \end{aligned} \tag{4}$$

onde: $p(\mathbf{x_n}|\mathbf{x_{n-1}})$ e $p(\mathbf{y_n}|\mathbf{x_n})$: são, respectivamente, as funções de distribuição de probabilidade para os estados e medidas; $N(.;\mathbf{m},\mathbf{P})$: denota a função de distribuição de probabilidade com média $\mathbf{m}$ e covariância $\mathbf{P}$.

Utilizando-se o processo recursivo de estimação Bayesiano a função de distribuição de probabilidade do posterior $p(\mathbf{x_n}|\mathbf{Y_n})$ no tempo n também pode ser vista como uma representação Gaussiana:

$$p(\mathbf{x_n}|\mathbf{Y_n}) \equiv N(\mathbf{x_n};\mathbf{m_{n|n}},\mathbf{P_{n|n}}) \tag{5}$$

onde: notação $n|n$ indica a função de distribuição de probabilidade associada ao vetor de estados $\mathbf{x_n}$ no tempo n, calculada utilizando-se as medidas obtidas até o passo n.

Assim, o processo pode ser entendido como o cálculo recursivo da média $\mathbf{m_{n|n}}$ e covariância $\mathbf{P_{n|n}}$ a cada passo n:

$$\left\{\mathbf{m_{0|0}},\mathbf{P_{0|0}}\right\} \xrightarrow{\text{previsão}} \left\{\mathbf{m_{1|0}},\mathbf{P_{1|0}}\right\} \xrightarrow{\text{atualização}} \left\{\mathbf{m_{1|1}},\mathbf{P_{1|1}}\right\} \xrightarrow{\text{previsão}} .... \xrightarrow{\text{atualização}} \left\{\mathbf{m_{n|n}},\mathbf{P_{n|n}}\right\}$$

As equações que descrevem o filtro de Kalman são mostradas abaixo:

$$\mathbf{m_{n|n-1}} = \mathbf{F_n}\mathbf{m_{n-1|n-1}}, \tag{6}$$

$$\mathbf{P_{n|n-1}} = \mathbf{F_n}\mathbf{P_{n-1|n-1}}\mathbf{F_n^T} + \mathbf{Q_n}, \tag{7}$$

$$\mathbf{S_n} = \mathbf{H_n}\mathbf{P_{n|n-1}}\mathbf{H_n^T}\mathbf{R_n}, \tag{8}$$

$$\mathbf{K_n} = \mathbf{P_{n|n-1}}\mathbf{H_n^T}\mathbf{S_n^{-1}}, \tag{9}$$

$$\mathbf{m_{n|n}} = \mathbf{m_{n|n-1}} + \mathbf{K_n}(\mathbf{y_n} - \mathbf{H_n}\mathbf{m_{n|n-1}}), \tag{10}$$

$$\mathbf{P_{n|n}} = \mathbf{P_{n|n-1}} - \mathbf{K_n}\mathbf{H_n}\mathbf{P_{n|n-1}}, \tag{11}$$

As equações 6 e 7 representam o passo para previsão enquanto as equações 8 a 11 são utilizadas na etapa de atualização dos estados. O algoritmo para a realização do filtro de Kalman pode ser resumido como se segue:

Algoritmo 1: FILTRO DE KALMAN - adaptado de Kovvali, Banavar e Spanias (2013).

Entrada: Espaço de estados $\{eq.3\}$ e conjunto de medidas do processo $\{y_1,y_2,...y_N\}$.

Saída: Estimativa dos estados $\{x_1,x_2,...x_N\}$ nos tempos $n = 1,2,...,N$.

início

Inicializa a média e covariância para a distribuição Gaussiana dos estados $N(\mathbf{x_0};\mathbf{m_{0|0}},\mathbf{P_{0|0}})$ para o passo $n = 0$.

para *Passos de* $n = 1,2,...,N$ **faça**

$$\mathbf{m_{n|n-1}} = \mathbf{F_n}\mathbf{m_{n-1|n-1}}$$

$$\mathbf{P_{n|n-1}} = \mathbf{F_n}\mathbf{P_{n-1|n-1}}\mathbf{F_n^T} + \mathbf{Q_n}$$

$$\mathbf{S_n} = \mathbf{H_n}\mathbf{P_{n|n-1}}\mathbf{H_n^T}\mathbf{R_n}$$

$$\mathbf{K_n} = \mathbf{P_{n|n-1}}\mathbf{H_n^T}\mathbf{S_n^{-1}}$$

$$\mathbf{m_{n|n}} = \mathbf{m_{n|n-1}} + \mathbf{K_n}(\mathbf{y_n} - \mathbf{H_n}\mathbf{m_{n|n-1}})$$

$$\mathbf{P_{n|n}} = \mathbf{P_{n|n-1}} - \mathbf{K_n}\mathbf{H_n}\mathbf{P_{n|n-1}}$$

fim

fim

Portanto, este algoritmo demonstra que, para a previsão dos estados, necessita-se apenas conhecer o passo anterior. A dinâmica do sistema é armazenada nas matrizes de covariância e a variação destas matrizes ocorre por meio de mudanças no ganho do filtro ($\mathbf{K_n}$).

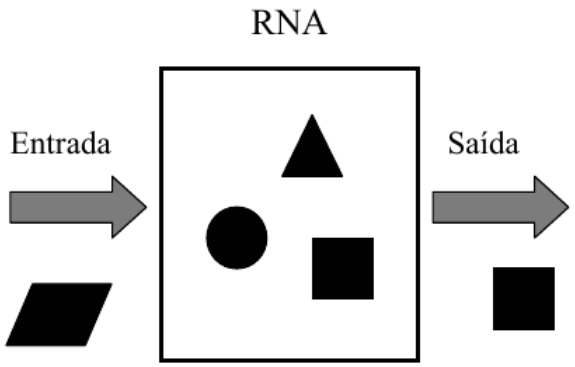
2.8 Redes Neurais Artificiais

Segundo Zurada (1992), uma RNA é simplesmente uma classe de algoritmos matemáticos, pois, uma RNA, pode ser considerada essencialmente como uma notação gráfica, para uma grande variedade de algoritmos. Estes algoritmos produzem soluções para uma série de problemas específicos. Comparando-se com os organismos vivos, uma RNA, tenta copiar as características de funcionamento de uma rede neural biológica.

O processo computacional responsável por encontrar a saída desejada (**s**), dado um conjunto de entradas (**x**), é conhecido como *recall*. O *recall* é o processo responsável por recuperar, através da decodificação, toda a informação armazenada pela rede anteriormente. Após uma RNA ser apresentada a um conjunto de dados (ou padrões), é possível, após o treinamento, associar, novos dados de entrada, ao conjunto de padrões previamente armazenados. Este processo é conhecido como associação. A Figura 11 ilustra o processo

de associação de uma RNA.

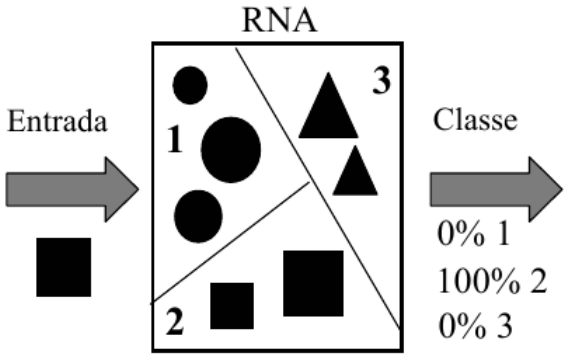
Figura 11 – Ilustração do processo de associação em uma RNA.



Fonte: Adaptado de Zurada (1992).

Outro processo computacional, realizado pelas RNA, é denominado classificação. Neste processo, um conjunto de padrões de entrada é dividido por classes ou categorias. Quando um determinado valor de entrada é apresentado, a RNA, indica a qual conjunto de dados (ou classe) a entrada pertence. A Figura 12 ilustra o processo de classificação.

Figura 12 – Ilustração do processo de classificação em uma RNA.

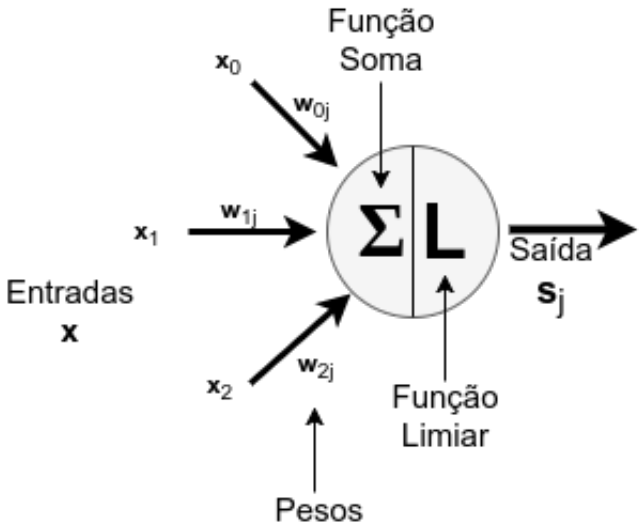


Fonte: Adaptado de Zurada (1992).

O aprendizado de uma RNA ocorre por meio dos ajustes dos pesos. Estes ajustes têm por finalidade possibilitar o melhor mapeamento dos valores de entrada com as saídas desejadas. Os algoritmos de aprendizagem possuem a função de alterar os pesos sinápticos das unidades fundamentais da rede (neurônios). Por meio destes ajustes, a RNA, consegue adquirir o conhecimento necessário para realizar uma tarefa desejada. Os neurônios são os elementos responsáveis por realizar o processamento da rede. A cada processamento, é gerado apenas um valor de saída que é repassado a todos os neurônios adjacentes por

meio de conexões. O valor de saída corresponde a uma função das entradas e dos pesos sinápticos. A Figura 13 apresenta um exemplo de neurônio.

Figura 13 – Exemplo de um neurônio artificial.

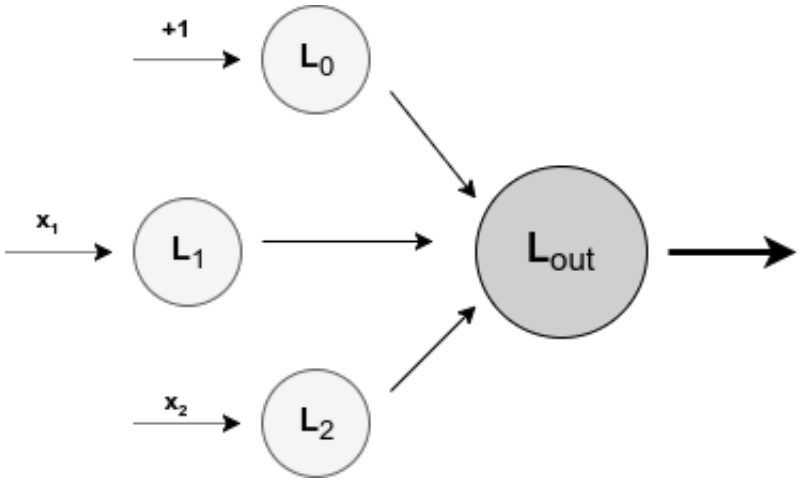


Fonte: Adaptado de Zurada (1992).

Em uma RNA, os neurônios estão conectados de tal modo que possam ser realizados ajustes nos pesos e modificações nas forças de conexão entre cada elemento da rede. Por meio destes ajustes sucessivos, torna-se possível capturar o conhecimento do ambiente. A Figura 14 mostra um exemplo de RNA.

A interação entre os neurônios acontece por meio das funções de limiar (ativação), uma vez que estas são utilizadas para indicar a forma e intensidade com que os sinais são trocados entre cada um dos neurônios. As principais funções de limiar (ativação)

Figura 14 – Estrutura de uma RNA composta por apenas uma camada.

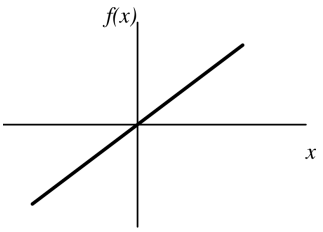


Fonte: Adaptado de Zurada (1992).

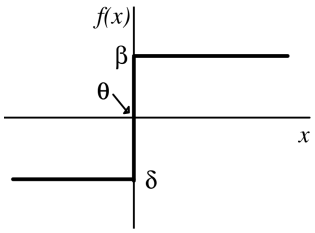
utilizadas são: linear, degrau, rampa, sigmoide e gaussiana. Ilustrações destas funções estão presentes na Figura 15. As alterações dos pesos da rede são realizadas por meio das funções de aprendizado que normalmente são definidas para cada tipo específico de RNA.

Figura 15 – Exemplo de funções utilizadas como limiar (ativação) nos neurônios.

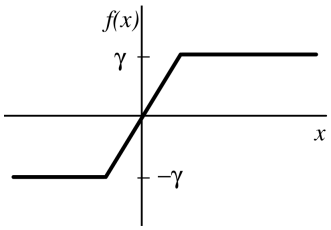
(a) Função linear.



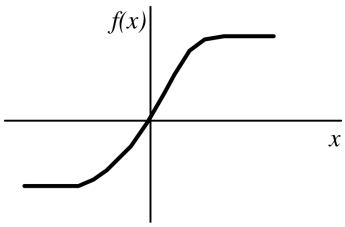
(b) Função degrau: o limiar determina o limite de transferência.



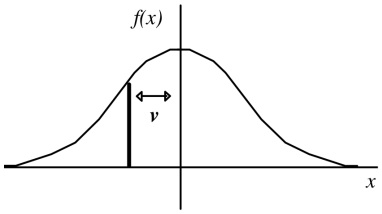
(c) Rampa: transição gradual entre os extremos.



(d) Sigmoide: transição suavizada.



(e) Gaussiana: transição normal com valor médio e desvio padrão.



Na tarefa de adquirir o conhecimento, são apresentados para a RNA padrões de dados de entrada. A cada um destes padrões de dados de entrada está associado um padrão relativo à saída. A interligação física entre cada neurônio é realizada por meio de conexões que possuem um peso característico. São estes pesos os responsáveis por armazenar os padrões apresentados à rede.

2.9 Tipos de Redes Neurais Artificiais

Existem, na literatura, várias arquiteturas de RNA. Nesta dissertação serão listados apenas 3 (três) modelos de maior interesse para a concepção do modelo de gerenciamento de falhas proposto.

2.9.1 Perceptron

É o modelo de RNA mais simples existente. O perceptron é um classificador binário, em que os valores de entrada são mapeados para encontrar uma saída desejada. A propagação das mudanças de pesos ocorre no sentido direto com alimentação a frente e

aprendizado supervisionado. Essa RNA é composta por uma única camada de elementos processadores. A equação 12 descreve matematicamente o modelo da Figura 13.

$$\mathbf{s_j} = \mathbf{L}(net_j) = \mathbf{L}\left(\sum_i \mathbf{x_i w_{ij}} + \theta_\mathbf{i}\right) = \begin{cases} 1 & net_j > 0 \\ 0 & net_j \leq 0 \end{cases} \tag{12}$$

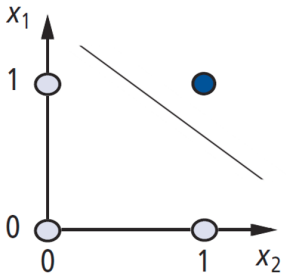
na qual:

$\mathbf{s_j}$: saída da rede;
$\mathbf{x_i}$: entradas da rede;
$\mathbf{w_{ij}}$: pesos da rede;
$\theta_\mathbf{i}$: coeficiente de limiar.

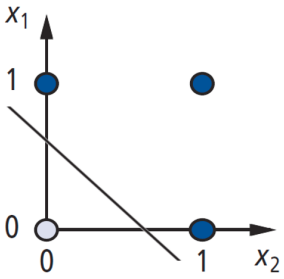
A grande vantagem da utilização desta arquitetura de RNA está na maior simplicidade de implementação, pois existem poucos parâmetros para serem ajustados. Entretanto, com esta abordagem não é possível tratar de sistemas que não sejam linearmente separáveis, como é o caso da função XOR. A Figura 16 exemplifica a diferença entre problemas linearmente separáveis e não separáveis.

Figura 16 – Exemplo problemas linearmente separáveis (16a e 16b) e não linearmente separáveis (16c).

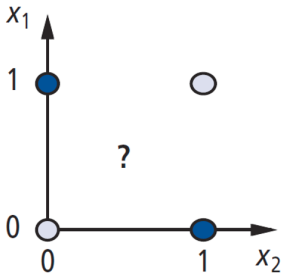
(a) Representação da porta lógica $X_1 and X_2$.



(b) Representação da porta lógica $X_1 or X_2$.



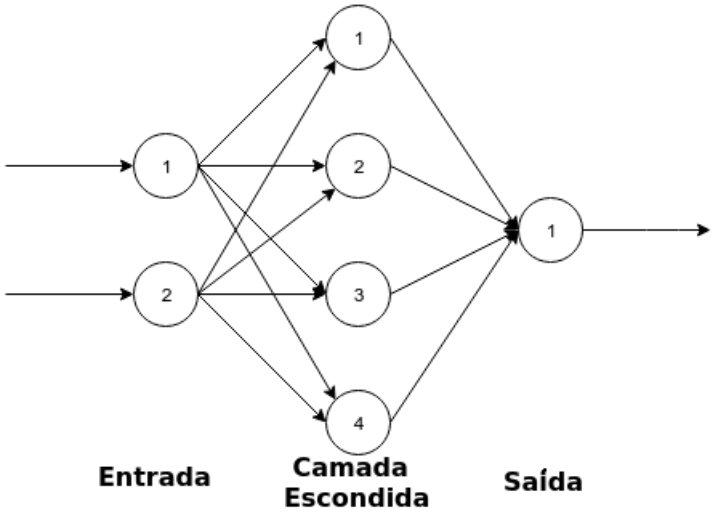
(c) Representação da porta lógica $X_1 xor X_2$.



2.9.2 Multilayer Perceptron

Quando uma RNA é composta por várias camadas de perceptrons, é atribuída a denominação *MultiLayer Perceptron* (MLP). Esta rede apresenta uma camada inicial onde são encontrados os nós sensoriais (nós de fonte), conhecida como camada de entrada, uma ou mais camadas intermediárias (camadas ocultas), responsáveis por realizar a maior parte do processamento da rede, e uma camada de saída onde são computadas as respostas da rede. Quando um sinal de entrada é apresentado para uma *MLP*, os estímulos gerados por estes sinais são propagados camada por camada até que sejam direcionados à camada de saída. A Figura 17 apresenta um modelo de rede *MLP* de duas entradas, uma camada escondida composta por 4 (quatro) neurônios e uma saída.

Figura 17 – Exemplo *MLP* com duas entradas, uma camada escondida com quatro neurônios e uma saída.



O algoritmo de treinamento supervisionado mais comumente utilizado em uma MLP é conhecido como algoritmo de retropropagação de erro (do inglês *error back-propagation*). Este algoritmo, como o nome sugere, pode ser dividido em dois passos distintos e complementares. O primeiro passo corresponde ao processo de propagação dos sinais de entrada. Neste passo, são apresentados para os nós fontes os padrões de sinais de entrada e estes estímulos são propagados camada por camada até que sejam encontradas as saídas iniciais da rede. Neste primeiro momento, não são ajustados os pesos sinápticos, já que estes serão refinados no passo seguinte. No segundo passo (*retropropagação*), as saídas encontradas no processo anterior são comparadas com as saídas desejadas (saídas alvo) e este sinal de erro é propagado para trás (no sentido contrário ao realizado no primeiro passo). Assim, os pesos sinápticos podem ser ajustados a fim de tornar a saída encontrada mais próxima da saída real da rede.

Uma rede MLP é uma ferramenta poderosa para resolver problemas complexos, pois, por meio da *retropropagação*, é eliminada a limitação presente no perceptron com uma camada (impossibilidade de tratar problemas não linearmente separáveis). O algoritmo de treinamento *error back-propagation* é apresentado, resumidamente pelo Algoritmo 2. Pelo algoritmo 2, é possível verificar a relativa simplicidade de implementação e a característica supervisionada da rede MLP. Estas características de concepção da rede MLP são interessantes para a construção do modelo centralizado de gerenciamento de falhas proposto. Isso ocorre, pois manter o sistema com reduzida complexidade é um dos principais objetivos do modelo proposto, visto que os nós de uma RSSF, normalmente, são compostos por dispositivos com reduzida disponibilidade de recursos. Além disso, quando vários dispositivos com características heterogêneas são conectados por meio da Internet, a complexidade da rede, naturalmente, tende a aumentar. Assim, esta abordagem pretende não elevar o grau de complexidade do sistema.

Algoritmo 2: ERROR BACK-PROPAGATION - adaptado Silva e Maia (2004).

Entrada: Vetores de entradas (x_{real}) e saídas reais (y_{real}).

Saída: Estimativa do vetor de saída.

início

1. Inicialização dos pesos e parâmetros da rede neural com valores aleatórios.
2. Apresentar um vetor (x_{real}) na entrada da rede neural e observar o vetor de saída y_{saida} gerado.
3. Comparar o vetor de saída y_{saida} gerado pela rede com o vetor y_{real} da resposta desejada.
4. Adaptar os pesos w_{ij} através do algoritmo recursivo :
   a. Iniciando pela camada de saída até a entrada :

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x'_i$$

   onde:
   η: velocidade de ajustes dos pesos;
   Para um neurônio pertencente a camada de saída:

$$\delta_j = y_j(1-y_j)(d_j - y_j)$$

   onde:
   d_j: representa a saída encontrada e y_j: representa a saída real;
   Para todos os neurônios pertencentes as camadas ocultas:

$$\delta_j = x'_j(1-x'_j)\sum \delta_k w_{jk}$$

   onde:
   k: representa todos os elementos processadores acima de j.
5. Retorno ao passo 2 até que a condição de parada seja encontrada.

fim

2.9.3 Redes de Função de Base Radial

Uma rede de Função de Base Radial, RBF do inglês (*Radial-Bases Function*), trata os problemas de inteligência computacional como um problema de ajuste de curvas (HAYKIN, 2001). Com uma rede RBF ,estamos interessados em encontrar uma superfície (em um espaço multidimensional) que consiga separar as classes de dados e, assim, indicar a que classe os novos dados apresentados pertencem.

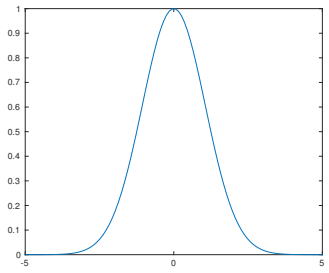
Uma rede RBF é constituída, basicamente, por três camadas distintas: camada de entrada, camada escondida e camada de saída. Assim como nas redes MLP a camada de entrada tem a função de conectar a rede ao mundo externo, uma vez que nesta camada são apresentados os padrões de entrada da rede. Como uma MLP, a camada oculta é intermediária entre as camadas de entrada e saída. Entretanto, existe apenas uma camada oculta que é responsável por aplicar uma transformação não-linear das entradas. Na camada

mais externa (saída), são realizadas as transformações lineares que fornecem as saídas da rede no mesmo padrão dos sinais apresentados à camada de entrada.

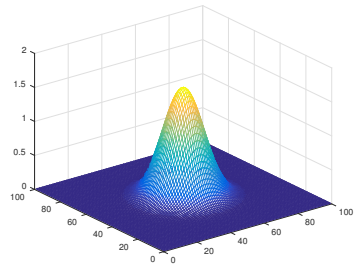
Funções de bases radial são uma classe especial de funções que apresentam papel fundamental para o aprendizado de uma rede RBF. A principal característica de uma função de base radial está no fato da resposta gerada diminuir (ou aumentar) monotonicamente com a distância de um ponto central. Como em uma rede RBF, estamos interessados em encontrar superfícies do espaço multidimensional que separem as classes de dados. A característica de uma variação monotônica da classe de funções de base radial torna-se primordial para o aprendizado da rede. Algumas funções de base radial estão apresentadas na Figura 18.

Figura 18 – Exemplo de funções de base radial comumente utilizadas.

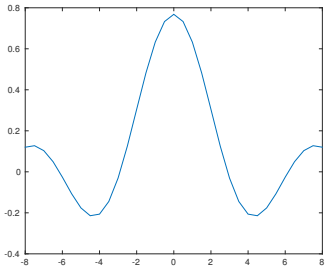
(a) Função gaussiana em duas dimensões.



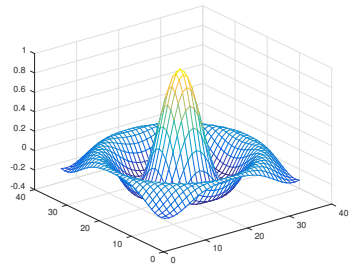
(b) Função gaussiana em três dimensões.



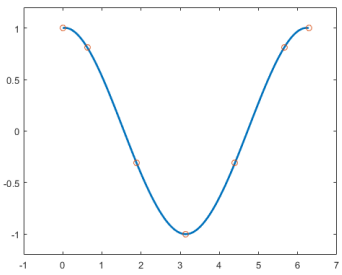
(c) Função sombrero mexicano em duas dimensões.



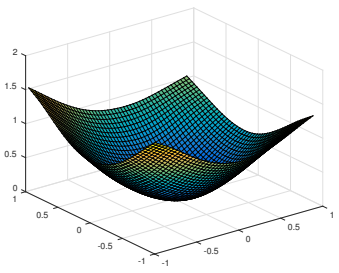
(d) Função sombrero mexicano em três dimensões.



(e) Função *thin-plate spline* em duas dimensões.



(f) Função *thin-plate spline* em três dimensões.



Assim como nas demais RNA apresentadas até o momento, nas redes RBF, o

treinamento ocorre de modo supervisionado. Portanto, o treinamento visa reduzir o erro entre os padrões apresentado à rede e os valores de saída obtidos. A redução da distância entre os dados apresentados e os encontrados ocorre por meio de ajustes nos parâmetros livres na rede RBF. Os principais parâmetros a serem ajustados são: os centros das regiões, o espalhamento e os pesos das rede. Um exemplo de treinamento é apresentado no Algoritmo 3.

Algoritmo 3: TREINAMENTO REDE RBF - adaptado de Haykin (2001).

Entrada: Vetores de entradas (x_{real}) e saídas reais (y_{real}).

Saída: Estimativa do vetor de saída.

início

1. Os centros são selecionados estatisticamente a partir de um conjunto de treinamento
2. O espalhamento dos dados são escolhidos pela normalização das distâncias (σ):

$$\sigma = \frac{d_{max}}{\sqrt{m_1}}$$

onde:

d_{max}: distância máxima em dois centros quaisquer.

m_1: número de centros.

3. Escolha das funções de base radial:

$$\varphi_i(\|x - t_i\|^2) = exp\left(\frac{-m_1}{d_{max}^2}\|x - t_i\|^2\right)$$

onde:

φ_i: função de base radial escolhida.

$\|x - t_i\|$: distância entre a entrada x e o centro t, com $i \in [1,m_1]$.

4. Ajuste dos pesos $\boldsymbol{w}$ por meio do método da pseudo-inversa :

$$\boldsymbol{w} = \boldsymbol{\varphi}^{+}\boldsymbol{d}$$

onde:

$\varphi+$: pseudo-inversa de $\boldsymbol{\varphi}$.

$\boldsymbol{d}$: resposta desejada.

com:

$$\boldsymbol{\varphi} = \{\varphi_{ij}\} \quad \textbf{e} \quad \varphi_{ji} = exp\left(\frac{-m_1}{d_{max}^2}\|x_j - t_i\|^2\right)$$

para:

$j = 1,2,...N$ e $i = 1,2,...,m_1$

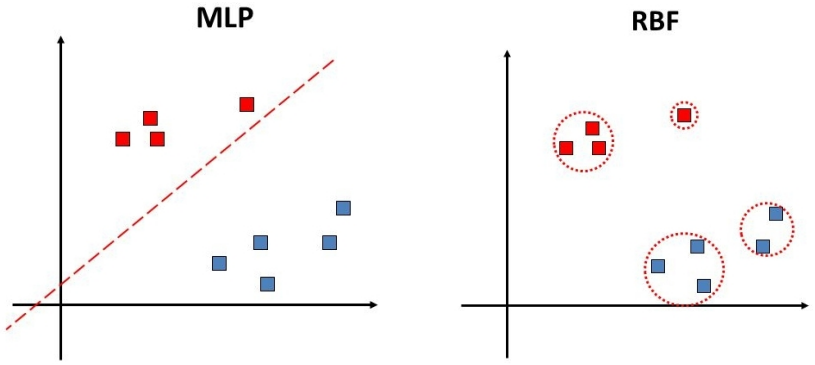
fim

Comparando-se as redes RBF com as redes MLP pode-se verificar algumas simi-

laridades e diferenças. Ambas tem em comum o fato de serem aproximadores universais e apresentam camadas com saídas não lineares. Entretanto, para uma RBF a camada escondida é não linear enquanto a camada de saída é linear. Já em uma rede MLP as camadas escondidas e de saídas são, geralmente, do tipo linear. O argumento para a função de ativação em uma rede RBF é a distância Euclidiana entre os dados de entrada e o centro de uma região. Nas redes MLP o argumento da função de ativação é a soma dos produtos de cada uma das entradas pelos pesos sinápticos do neurônio. Devido a estas características, o mapeamento das regiões de classificação é realizado de maneira distinta por cada uma destas redes. Uma rede RBF constrói aproximações locais para o mapeamento entra-saída, enquanto uma rede MLP realiza aproximações globais para encontrar as regiões de classificação. A Figura 19 mostra uma representação gráfica desta diferença de abordagem.

Figura 19 – Comparação gráfica entre o mapeamento das redes RBF e MLP.

(a) Mapemento global de uma rede MLP. (b) Mapemento local de uma rede RBF.



Fonte: Adaptado de Jang, Sun e Mizutani (1997).

2.10 Redes Neuro *Fuzzy* ANFIS

Como visto nas seções anteriores, a utilização das RNA possuem um papel essencial na extração de conhecimento por meio de observações. Esse aprendizado ocorre, geralmente, por meio da apresentação de padrões de entrada e saída à rede a ser treinada. Após o treinamento, a RNA consegue mapear novos valores de entradas e encontrar as saídas correspondentes. Entretanto, normalmente, após este treinamento não é possível obter maiores informações sobre como a RNA modelou o problema em questão, ou seja, o processo de aprendizado da rede é visto como um modelo caixa-preta. A fim de reduzir esta dificuldade em compreender o comportamento do sistema as RNA são utilizadas em conjunto com Sistemas de Inferência Fuzzy (SIF) o que forma os Sistemas Neuro-Fuzzy (SNF). Um exemplo de SNF bastante utilizado é a rede ANFIS (do inglês Adaptative-Network-

Based Fuzzy Inference System) (JANG, 1993). A rede ANFIS apresenta características funcionais das redes RBF e sistemas *fuzzy* do tipo Takagi–Sugeno–Kang (TSK) (JANG; SUN; MIZUTANI, 1997).

Um SIF, consiste na utilização da teoria de conjuntos *fuzzy*, regras de inferência do tipo SE-ENTÃO e raciocínio *fuzzy* para a modelagem de sistemas complexos. Em um SIF, as entradas também são mapeadas para as saídas, entretanto, o conhecimento não advém do treinamento da rede, ele é incorporado por meio de modelos matemáticos ou linguísticos. O conhecimento é adquirido por meio da definição das regras SE-ENTÃO e funções de pertinência. A diferença entre estes modelos está no modo como a consequência (ENTÃO) da regra é tratado. Existem, basicamente, duas fontes de conhecimento utilizadas na construção dos modelos *fuzzy*: especialistas e dados.

Na concepção da inferência baseada no conhecimento de especialistas, as regras são adicionadas, ao sistema computacional, por meio do conhecimento de um especialista humano. Um exemplo desta abordagem, é modelo de inferência Mamdani (MAMDANI; ASSILIAN, 1975). Já os modelos *fuzzy* baseados em dados, utilizam-se dos dados históricos armazenados para extrair padrões e criar as regras de inferência (REZENDE, 2003).

A construção do modelo TSK é realizada por meio da análise da massa de dados armazenada. Diferentemente do modelo Mamdani, um sistema de inferência TSK, utiliza-se, como consequente das regras, equações paramétricas a fim de relacionar os dados de entrada com as saídas desejadas. A Equação 13 apresenta um exemplo de regras, utilizadas para a inferência em modelos do tipo Mamdani, e a Equação 14 exemplifica a regra empregada ao sistema TSK.

$$\textbf{Se} \quad x_1 = A_1 \quad \textbf{E} \quad x_2 = A_2 \quad ... \quad \textbf{E} \quad x_p = A_p \quad \textbf{Então } y = B \tag{13}$$

onde: x_i são as entradas do sistema; A_i são os termos primários definidos nas partições *fuzzy* das entradas e presentes na regra i; y é a variável de saída e B é o termo primário definido na partição *fuzzy*.

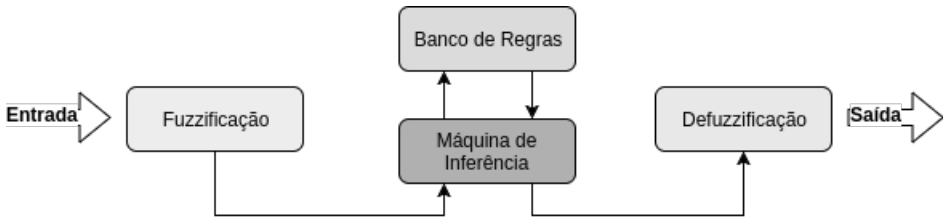
$$\textbf{Se} \quad x_1 = A_i \quad \textbf{E} \quad x_2 = A_j \quad ... \quad \textbf{E} \quad x_p = A_m \quad \textbf{Então } y = \phi(x_1,x_2,...,x_p) \tag{14}$$

onde: x_i são as entradas do sistema; A_i são os termos primários definidos nas partições *fuzzy* das entradas e presentes na regra i; y é a variável de saída e ϕ_i é a saída da regra i, geralmente, um polinômio relativo a entrada x.

Um SIF é composto, basicamente, por 4 (quatro) diferentes módulos: módulo de *fuzzificação*, banco de regras, máquina de inferência e *defuzzificação*. O módulo de *fuzzifica-*

ção corresponde a interface de entrada do sistema e, portanto, é responsável por receber os valores numéricos e convertê-los em conjuntos *fuzzy* equivalentes. Após esta conversão, a máquina de inferência processa as regras existentes no banco de regras e gera um conjunto *fuzzy* a partir das regras disparadas. Uma regra é disparada quando a combinação dos antecessores das regras geram valores diferentes de zero. A *defuzzificação*, corresponde ao processo de receber o conjunto *fuzzy* para cada variável de saída convertê-lo em um valor escalar. A Figura 20, apresenta o modelo de concepção genérico para um SIF. Em um SIF do tipo TSK, o módulo de *defuzzificação*, pode ser suprimido, pois a saída do SIF do tipo TSK corresponde a uma função paramétrica dos valores de entrada. Assim, após a inferência é gerada uma saída numérica.

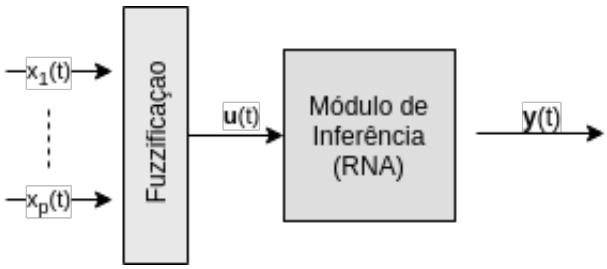
Figura 20 – Modelo genérico de um SIF.



Fonte: Adaptado de Rezende (2003).

A arquitetura geral de um SNF é apresentada na Figura 21. Nesta figura é possível identificar dois módulos principais: *fuzzificação* e inferência. O módulo de *fuzzificação* é responsável por realizar os cálculos das funções de pertinências, já o módulo de inferência é composto por uma RNA que recebe como entrada o vetor de ativação e gera como saída os valores de pertinência de cada um dos conjuntos *fuzzy*. A Figura 22, exemplifica, de forma mais detalhada, o modelo genérico de uma arquitetura ANFIS. Esta figura apresenta um SNF com duas entradas e com duas regras.

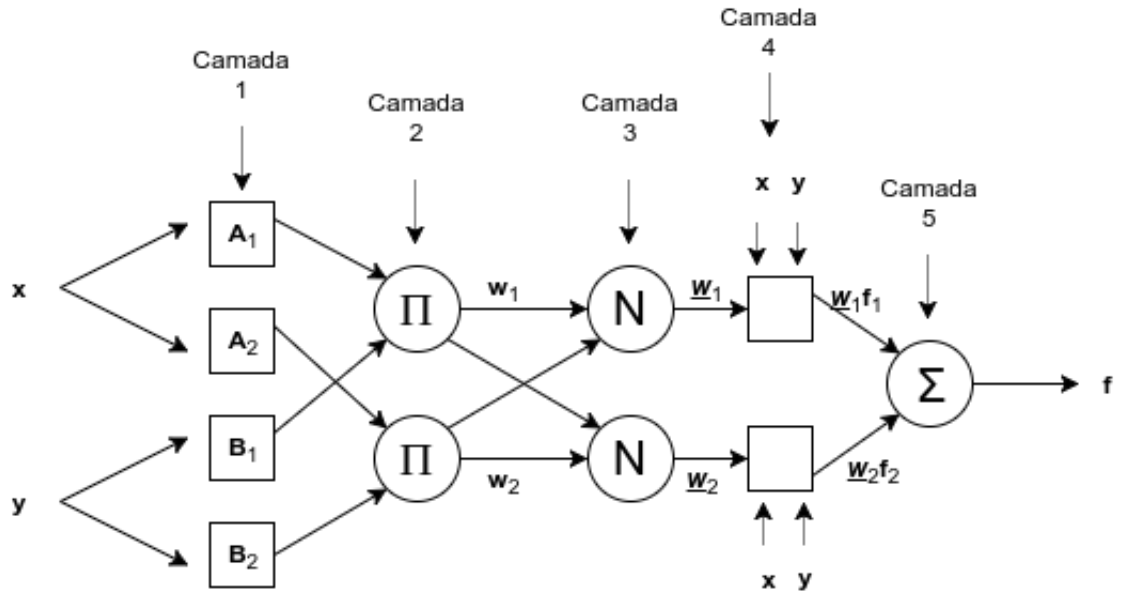
Figura 21 – Modelo genérico de um SNF.



Fonte: Adaptado de Rezende (2003).

Por meio da Figura 22 é possível identificar a existência de 5 (cinco) camadas para o modelo de ANFIS, as principais funções realizadas em cada uma destas camadas é apresentada a seguir:

Figura 22 – Exemplo de arquitetura para o modelo ANFIS.



Fonte: Adaptado de Jang, Sun e Mizutani (1997).

- Camada 1: são calculados os graus de pertinência para cada uma das entradas (semelhante ao processo de *fuzzificação*);
- Camada 2: a cada nó é atribuída uma regra (figura 22: 2 (dois) nós). Cada nó calcula o grau de ativação para cada regra. O grau de ativação de uma regra corresponde ao peso atribuído a cada consequente da regra;
- Camada 3: é realizada a normalização, neste processo, cada nó calcula a razão entre o nível de disparo de uma regra e a soma dos níveis de disparo de todas as regras;
- Camada 4: as saídas dos neurônios são calculadas pelo produto entre os níveis de disparo normalizados e o valor do consequente da regra em questão;
- Camada 5: é calculada a saída do modelo, por meio da soma de todos os sinais, já normalizados, de entrada do nó. O processamento realizado pela camada 5 é similar ao processo de *defuzzificação*.

Dentre as principais aplicações do modelo ANFIS, encontra-se a aproximação de funções e a identificação de sistemas complexos. O reconhecimento de sistemas não lineares é, particularmente, importante para o controle de processos industriais (KURNAZ; CETIN; KAYNAK, 2010; AKCAYOL, 2004).

2.11 Modelagem e Simulação de RSSF

O crescente aumento das pesquisas em IoT ocasionou uma maior demanda por sistemas que possibilitem a implementação, simulação e integração das RSSF. A fim de suprir esta demanda, foram desenvolvidas várias ferramentas de *software* como TinyOS

(LEVIS et al., 2003) e Contiki (DUNKELS; GRONVALL; VOIGT, 2004). Estes sistemas têm por finalidade possibilitar o desenvolvimento e a simulação de aplicações em IoT, especialmente, em RSSF sem a necessidade de adquirir todos os componentes físicos para implementação da rede.

O Contiki é um Sistema Operacional (SO) em código aberto desenvolvido para possibilitar a conexão de microcontroladores pequenos, com fonte restrita de energia e baixo custo com a Internet (DUNKELS; GRONVALL; VOIGT, 2004). Assim, o Contiki torna-se uma ferramenta valiosa para a implementação da IoT. Esse sistema operacional, escrito em linguagem C, é baseado em eventos implementados por meio de Protothreads. Esta abordagem, por meio de Protothreads, reduz a utilização de memórias RAM (apenas 2 kilobytes) e ROM (40 kilobytes) (FAROOQ; KUNZ, 2011). No Contiki várias pilhas de protocolos de comunicação, incluindo o 6LoWPAN, já estão implementadas. Somando-se a este fato a existência do simulador gráfico Cooja, tem-se uma ferramenta de *software* poderosa para o desenvolvimento das pesquisas e aplicações reais em IoT.

Cooja é um simulador, presente no SO Contiki, baseado em Java e desenvolvido para simulação de RSSF (OSTERLIND et al., 2006). Neste simulador, podem ser criados diferentes tipos de redes, uma vez que cada um dos nós podem possuir características próprias de *software* e *hardware*. Para o Cooja, existem vários *plugins* que possibilitam a análise e integração da rede com a Internet. Uma vez que o Cooja possui uma interface gráfica, as informações de projeto (zonas de cobertura e penumbra do sinal, IP dos nós, localização, diagramas de comunicação e trocas de mensagens) podem ser acessadas de maneira direta, o que auxilia no estudo das características da rede projetada. Assim, o conjunto destes sistemas (Contiki e Cooja) é utilizado como ferramental para construção e simulação do modelo de gerenciamento de falhas proposto nesta dissertação.

3 Trabalhos Relacionados

Nesta parte do texto, são listados e discutidos os trabalhos relacionados ao tema gerenciamento de falhas. Em cada parágrafo, é apresenta uma visão geral do trabalho em questão, a principal contribuição e o relacionamento com os objetivos apresentados na seção 1.3 deste trabalho.

3.1 Abordagem Centralizada

A abordagem realizada em Jäger et al. (2014) emprega as RNA para detectar 3 (três) tipos diferentes de possíveis falhas (*offset* constante, zero constante e *outlier*). Jäger et al. (2014) demonstram que os dois primeiros tipos de falhas foram detectados com erros inferiores a 10%, mas os *outliers* não foram corretamente evidenciados. Neste sistema, a possível falha é apenas detectada e diagnosticada, entretanto nenhuma ação é realizada a fim de contornar este erro. Como nenhuma ação no sentido de mitigar os potenciais danos causados pelo falha não são empregados, não é possível dizer que, nesta abordagem, existe um gerenciamento de falhas.

A aplicação do autoaprendizado na detecção de falhas em RSSF dentro de ambientes industriais é o foco do trabalho realizado por Liu et al. (2013). Neste artigo, é utilizada a técnica conhecida como *Statistics Sliding Windows* (SSW). Nesta abordagem, janelas temporais são aplicadas ao conjunto de dados e, por meio da função Gaussiana, são comparadas com os dados dos nós em uma mesma região da rede. Caso um valor medido seja superior ao desvio padrão desta função, o nó sensor é marcado como faltoso. A partir desta análise dos dados, é gerado um modelo de regressão que possibilita à rede identificar as próximas falhas sem a necessidade de analisar novamente todo o conjunto de dados. Liu et al. (2013) atestam que a taxa de falsos positivos é inferior a 5%. Esta técnica foi aplicada em 3 (três) tipos diferentes de falhas (picos, *outlier* e zeros constante). Entretanto, não são realizados experimentos relativos ao consumo de energia da rede e a integração com a Internet das Coisas também não é realizada. Como não existem experimentos relativos ao consumo de energia, não é possível verificar qual seria o impacto da adoção deste sistema em uma RSSF.

Shih e Wei (2014) utilizam-se do mecanismo BGM (*Brace-Gatarek-Muastrini*) a fim de propor um sistema de detecção de falhas em uma rede M2M (*Machine-to-Machine*). Os resultados encontrados demonstram que esta abordagem propiciou a localização de quase 100% das falhas simuladas. O trabalho de Shih e Wei (2014), mesmo apresentando resultados satisfatórios, também não integra a rede M2M à Internet além de ser executado apenas em comunicação direta ponto a ponto. Uma vez que a interação ocorre apenas

entre dois dispositivos, a integração deste sistema com a IoT torna-se inviável, pois em uma arquitetura IoT existe a necessidade de comunicação entre todos os dispositivos da rede.

O diagnóstico e previsão de falhas aplicado à IoT é o ponto central discutido por Wang, Vo e Ni (2015). Neste artigo, por meio da análise dos dados, o processo de diagnóstico do erro é dividido em 2 (duas) fases. Inicialmente, é traçada a relação causal entre os dados e a falha e, posteriormente, por meio do modelo de relação criado, torna-se possível detectar novos erros na rede. São 2 (dois) os algoritmos empregados neste trabalho: *Conditional Probability Analisys* e *Apriori Analysis*, ambos implementados no sistema *SAP HANA Plataform*. Os resultados encontrados demonstraram que esta abordagem foi efetiva para previsão de falhas em IoT aplicadas a ambiente industriais. Uma vez que este modelo de detecção utiliza uma plataforma proprietária, a aplicação em larga escala deste método torna-se inviável devido ao custo envolvido.

Técnicas de macroprogramação são desenvolvidas por Nogueira (2015) a fim de prover uma abstração em alto nível e aumentar a capacidade de tolerância a falhas de uma RSSF utilizando, especificamente, a ferramenta *Srijan* para RSSF heterogêneas. A abordagem utilizada possibilitou o desenvolvimento de um sistema em programação de alto nível, entretanto, esta técnica está restrita a uma plataforma.

Outro trabalho que utiliza-se de técnicas estatísticas para promover a detecção de falhas em redes de sensores sem fio é realizado por Lau, Ma e Chow (2014). Nesta abordagem, é utilizado o *framework Naive Bayes*. Este *framework* é um classificador que emprega o teorema de Bayes e o estimador de máxima verossimilhança (do inglês *Maximum Likelihood Estimation*) para estimar a função de densidade de probabilidade (do inglês *Probality Density Function*). Com esta estimação, o classificador consegue distinguir o que são falhas reais e o que é funcionamento normal do sistema. Os resultados mantiveram um nível baixo de falsos positivos, entretanto, também não é realizado o gerenciamento da falha, pois ela apenas é identificada.

A lógica *Fuzzy* é empregada por Raajini, Kumar e Indumathi (2015) para encontrar falhas em *links* de transmissão em RSSF dinâmicas. A estimação da qualidade destes *links* de transmissão ocorre por meio de regras *Fuzzy*, e, caso sejam evidenciados baixos índices de qualidade do sinal, uma possível falha é detectada. Esta abordagem apresenta um pequeno conjunto de regras e os resultados obtidos estão abaixo dos encontrados em outras técnicas. Tratamentos similares, mas com melhores resultados, são dados por Behzadi e Azad (2014) e Babu e Masheswari (2014) pois é utilizada a combinação da lógica *Fuzzy* com técnicas estatísticas (*Naive Bayes*). Assim, a taxa de falsos positivos é reduzida. Porém, não são realizados experimentos relativos ao consumo de energia, além de não serem utilizados protocolos de integração das RSSF com a IoT.

3.2 Abordagem Descentralizada

Chudzikiewicz, Furtak e Zielinski (2015) propõem um mecanismo de detecção, diagnóstico e recuperação de falhas em Internet das Coisas Militares. O sistema proposto consiste em dividir a rede em *clusters* e, por meio da avaliação dos dados gerados pelos nós de um mesmo *cluster*, é possível indicar se existem grandes desvios entre os dados coletados. Caso seja constatado este desvio, um alerta de possível erro é gerado pelo *Cluster Head*. Além desta aplicação estar restrita ao ambiente militar, neste modelo, a comunicação dentro de toda a rede é realizada pelo protocolo ZigBee, não possibilitando total integração com a Internet. Outra característica desta abordagem diz respeito ao fato de, após uma falha ser identificada, toda a informação deste erro ser perdida, o que obriga a rede a desempenhar um monitoramento constante dos dados nos nós vizinhos. Desse modo, o consumo de energia tende a ser elevado.

O trabalho de Goudar e Potkonjak (2014) divide uma rede de sensores aplicadas ao monitoramento móvel de paciente em regiões bastante correlacionadas. Dentro de cada uma destas regiões, os dados gerados são comparados, e caso algum *outlier* seja detectado, uma possível falha é informada. Nesta abordagem, diferentemente dos *clusters*, não existe um CH. Sistemas que utilizam-se deste mecanismo não possuem uma memória sobre as falhas já detectadas, o que ocasiona a troca desnecessária de mensagens dentro da rede. Uma das características fundamentais de uma RSSF eficiente reside na capacidade de manter um consumo reduzido de energia. Segundo Rhee, Seetharam e Liu (2004), Torres (2006), Dâmaso et al. (2013), as tarefas, realizadas por nós da rede, que consomem maior quantidade de energia são as etapas de recebimento e envio de mensagens.

Zhang, Dragoni e Wang (2015) propõem a construção de um *framework* a fim de realizar a detecção de falhas em RSSF por meio de um tratamento mais abstrato focado no consumo eficiente de energia. O *framework* proposto é constituído por 3 (três) módulos: *Model stablishment*, *Information collection* e *Decision making*. Estes módulos têm a função de reduzir a troca desnecessária de mensagens dentro da rede, assim, será realizado um consumo eficiente de energia. Neste artigo, apenas a base de construção deste sistema é proposta, mas a implementação deste algoritmo não é realizada. Assim, não é possível verificar a qualidade real desta abordagem.

No trabalho de Zhou, Lin e Shih (2015), é empregada uma estrutura hierárquica para monitoramento de aglomerados (*clustering*) em um sistema de IoT a fim de promover a comunicação com baixo gasto de energia. O algoritmo desenvolvido utiliza o problema do caixeiro viajante múltiplo para encontrar melhores rotas após a detecção da falha, com a finalidade de reduzir os custos de comunicação. Apesar dos resultados alcançados mostrarem que a heurística utilizada obteve melhores frutos que outras abordagens, não é mencionado se esta aplicação está integrada à Internet ou se utiliza o protocolo 6LoWPAN.

Jiang (2009) aborda a detecção de falhas em RSSF por meio do algoritmo *Distributed Fault Detection* (DFD) modificado. Em um DFD comum, os dados enviados pelos nós vizinhos a um nodo, de um mesma região, devem ser próximos o bastante dos dados anteriores para que o nodo seja classificado como sadio. Na técnica empregada por Jiang (2009), a comparação entre os dados não é realizada apenas com os nós mais próximos. A comparação é feita com uma grande porção da rede e caso menos da metade dos nós apresentem medidas semelhantes ao nodo em análise, uma possível falha é identificada. Com esta modificação, os resultados alcançados melhoraram, entretanto, a troca de mensagens na rede também será elevada, o que gera um consumo maior de energia.

Outra sugestão de arquitetura para detecção de falhas em redes de sensores sem fio é realizada por Bokareva, Bulusu e Jha (2006). Nesta proposta, os autores dividem a rede em grupos de sensores responsáveis por monitoramento do ambiente e nós responsáveis pela comunicação, monitoramento de outros nós e detecção de falhas. Para possibilitar o aprendizado da rede, são utilizadas as RNA com treinamento não supervisionado *Self Organization Map* (SOM). Não são apresentados resultados, pois esta arquitetura não é implementada.

Prakash et al. (2014) empregam a abordagem em *cluster* adaptativo para sugerir um sistema tolerante a falhas que garante a QoS (*Quality of Service*) e redução do consumo de energia. Para a utilização desta abordagem, os autores assumem algumas premissas como: todos os nós conseguirem medir o *Received Signal Strength Indicator* (RSSI) da mensagem recebida e a estação base possuir suprimento ilimitado de energia. Em uma RSSF real, estas premissas não são completamente atendidas. Assim, mesmo este algoritmo conseguindo ser tolerante a falhas, a aplicação em um ambiente real é comprometida.

A fim de reduzir o esforço computacional na análise dos dados e, consequentemente, minimizar o consumo de energia, Xie, Chen e Zhao (2012) utilizam a técnica estatística *Principal Component Analysis* (PCA) para encontrar as combinações mais relevantes para a representação da base de dados. Esta técnica é empregada juntamente com a teoria *First-Order Perturbation* (FOP) a fim de possibilitar a atualização, recursivamente, dos autovalores e autovetores do PCA. Os resultados encontrados mostram que a combinação destas duas técnicas pode ser utilizada para detecção de falhas em RSSF. Contudo, estes resultados não foram alcançados com a integração à IoT.

Uma abordagem tolerante a falhas e com consumo eficiente de energia é apresentada no trabalho de Azharuddin e Jana (2015). Neste trabalho, os autores desenvolvem um algoritmo de roteamento para RSSF descentralizadas sem alterar, significativamente, o consumo de energia da rede. Com este algoritmo, falhas no CH, são contornadas a fim de manter a comunicação entre os membros da RSSF. Os resultados mostram que o sistema consegue manter a RSSF em funcionamento mesmo após a falha de um CH.

Entretanto, mesmo com resultados promissores, esta abordagem é empregada em um contexto diferente desta dissertação. Isso porque, Azharuddin e Jana (2015), focam em falhas de comunicação em redes descentralizadas.

Wu, Tan e Xiong (2016), também utilizam uma abordagem descentralizada por meio de *cluster*. A fim de garantir a tolerância a falhas, é desenvolvido um *framework* que combina a previsão e compressão de dados. Na etapa de previsão dos dados, é implementado o algoritmo *Least Mean Square* (LMS). Assim, o CH pode identificar possíveis falhas nos dados coletados. A compressão dos dados e recuperação da falha é realizada por meio da utilização do PCA. Com o PCA, apenas os dados relevantes para o tratamento da falha são enviados, assim o consumo de energia tende a ser mitigado. Apesar dos resultados alcançados demonstrarem a eficiência e aplicabilidade do *framework*, este modelo também apresenta uma abordagem descentralizada.

Bagci, Korpeoglu e Yazıcı (2015) introduzem o algoritmo distribuído, tolerante a falhas, chamado de *Disjoint Path Vector* (DPV). O DPV tem como objetivo possibilitar o controle da topologia de RSSF heterogêneas. As RSSF heterogêneas são compostas por nós com capacidade reduzida de recursos (fonte de energia e poder de processamento limitado) e por nós com recursos ilimitados. Os experimentos demonstram que, o DPV, apresenta resultados similares aos sistemas centralizados no controle da topologia de RSSF heterogêneas. A principal contribuição do DPV é a redução do número de mensagens trocadas dentro da RSSF. Com a diminuição dos processos de transmissão e recepção, o consumo de energia, tende a ser reduzido.

Outro sistema tolerante a falhas é proposto por Bhuiyan et al. (2015). Nesta abordagem, as RSSF, são empregadas no monitoramento da saúde de estruturas, do inglês *Structural Health Monitoring* (SHM). Estas redes são utilizadas para monitorar as condições de operação de estruturas como pontes e edifícios. O algoritmo chamado *Fault-Tolerance in SHM* (FTSHM) é empregado, por meio de uma abordagem distribuída, em RSSF. A tolerância a falhas é alcançada com a utilização de sensores de *backup*. Quando uma falha é detectada, em uma região do *cluster*, os sensores de *backup* são adicionados a RSSF, a fim de manter o monitoramento adequado da estrutura. Nesta abordagem são utilizados novos conjuntos de sensores que, inicialmente, não pertencem a RSSF. Assim, o custo de construção do sistema, pode sofre um acréscimo considerável.

Técnicas bioinspiradas para a detecção de intrusos e autorrecuperação de uma rede em IoT são apresentadas por Salmon et al. (2013) e Almeida, Ribeiro e Moreno (2015). Estes trabalhos utilizam-se da Teoria do Perigo e Células Dendríticas para identificar, rastrear e recuperar a rede após uma tentativa de invasão. Apesar de serem aplicados na detecção de invasores à IoT, estas técnicas podem ser empregadas para o gerenciamento de falha em IoT.

Koushanfar, Potkonjak e Sangiovanni-Vincentelli (2004) sumarizam alguns métodos

para detecção de falhas em sistemas de sensoriamento e comunicação, especialmente em RSSF. Yu, Jia e Xi-yuan (2011), Lau, Ma e Chow (2014), Haque, Rahman e Aziz (2015) são exemplos de utilização de métodos estatísticos para detecção de anomalias. Yu, Jia e Xi-yuan (2011) utilizam-se da técnica conhecida como *Node Self Detection by History Data and Neighbours* (NDHN) a fim de identificar, de maneira descentralizada, possíveis falhas em RSSF. Dados coletados pelo sensor são comparados estatisticamente com os dados históricos de nós vizinhos a fim de encontrar possíveis falhas. Haque, Rahman e Aziz (2015) empregam as técnicas estatísticas em RSSF específicas para a o monitoramento de pacientes e toda a análise e conseguente detecção de falhas é realizada por uma estação base que, por meio da inspeção dos dados históricos, identifica possíveis falhas dos sensores.

Uma abordagem que se utiliza do filtro de Kalman para indicar a confiabilidade de dados gerados por sensores de uma RSSF, é apresentada no trabalho de Dogan e Avincan (2017). No sistema proposto, por exemplo, caso o sinal de GPS seja perdido, por meio da previsão dos dados realizada pelo filtro de Kalman, novas rotas para troca de mensagens são calculadas. A confiabilidade dos dados gerados pelos nós, também é verificada por meio da análise da diferença entre os valores previstos e medidos. Entretanto, este sistema, é empregado em RSSF descentralizadas e considerações sobre o consumo de energia não são realizadas.

3.3 Detecção e Classificação de Falhas em Outros Sistemas

Jim e Gregory (2016) realizam uma revisão sobre a utilização de sistemas imunológicos artificiais (AIS do inglês *Artificial Immune System*) como técnicas para promover a segurança em redes *Mobile Ad hoc Networks* (MANET). Algumas das estratégias utilizadas merecem destaque, pois podem ser empregadas na detecção e diagnóstico de uma falha em RSSF, além de possibilitarem a recuperação da rede após um erro. Dentre os principais trabalhos listados, estão Mohamed e Abdullah (2010), Mazhar e Farooq (2008) e Barani (2014). Mohamed e Abdullah (2010) utilizam-se das técnicas de AIS, por meio de um agente móvel (*Mobile Agent*) para conseguir detectar, classificar, bloquear/isolar uma tentativa de invasão e, após estas etapas, possibilitar a própria rede recuperar-se do ataque. No trabalho de Mazhar e Farooq (2008), é empregada a técnica bioinspirada conhecida como células dendríticas (CD). As CD são células responsáveis por apresentar os possíveis antígenos ao sistema de tratamento. Caso este possível antígeno seja identificado como uma falha pelo sistema, uma reação de combate ao agente danoso é gerada. Outra técnica interessante a ser mencionada é apresentada por Barani (2014). Neste trabalho, juntamente com o AIS, são utilizados algoritmos genéticos a fim de possibilitar a detecção de anomalias na rede. Todos estes trabalhos citados apresentam resultados promissores para a garantia de segurança em uma rede MANET, entretanto, estas técnicas não foram utilizadas a fim de

garantir o gerenciamento de falhas.

O sistema para gerenciamento de falha utilizado por Cheng e Wang (2011) consiste em, por meio dos dados de *Global Position System* (GPS) e mensagens *Short Message Service* (SMS), automaticamente, identificar, julgar e tratar uma possível falha. Cheng e Wang (2011) empregam esta abordagem para detecção de erros em um transformador, a fim de reduzir os períodos de manutenção e, consequentemente, diminuir os gastos com estes serviços. Segundo Cheng e Wang (2011), este enfoque minimizou o desembolso de recursos. Contudo, mesmo empregando-se o termo Internet das Coisas, Cheng e Wang (2011) não realizam a integração deste sistema com a Internet, além desta abordagem não ser empregada em RSSF.

Shao e Lunetta (2012), demonstram que o algoritmo *Support Vector Machine* (SVM) apresenta bons resultados como classificador de diferentes tipos de cobertura do solo utilizando-se um conjunto reduzido de dados para treinamento. Alguns trabalhos de aplicações em bioinformática (ZIEN et al., 2000), reconhecimento de face em imagens (HEASRT et al., 1998) e detecção de movimentos dos olhos (SINGLA et al., 2011) também demonstram que os resultados de classificação encontrados com o algoritmo SVM são superiores aos obtidos pelas RNA. Entretanto, por ser um método não-paramétrico, não possui tamanho fixo o que pode ocasionar um consumo expressivo de memória para os nós de uma rede.

3.4 Detecção de Falhas Por Meio do Acelerômetro

Medidas do acelerômetro são utilizadas para obter informação sobre erros em muitas áreas. O diagnóstico próprio de falhas em uma RSSF é apresentado por Harte, Rahman e Razeeb (2005), em que as falhas de *hardware* são detectadas por meio da análise das medidas do acelerômetro. Huang e Su (2015) utilizam também o acelerômetro para detectar e isolar sensores que estão com falhas a fim de melhorar a navegação de veículos inteligentes. Outro exemplo de aplicação encontra-se no trabalho de Lewicki, Lambert e Wagoner (2015) em que são detectadas possíveis falhas nos dentes de máquinas rotativas por meio da análise da vibração do rotor. Outros exemplos de aplicações da abordagem de detecção de falhas por meio de acelerômetros podem ser encontrados nos trabalhos de Kojima, Tsunashima e Matsumoto (2006), Marcal, Hatakeyama e Susin (2006), Jia (2009), Raj, Natarajan e Girikumar (2013).

Como pode ser visto pela coletânea de trabalhos relacionados ao tema proposto, existem algumas lacunas que os procedimentos realizados neste trabalho vêm investigar. Muitos destes trabalhos não estão voltados, especificamente, para a integração das RSSF com a IoT, pois as RSSF são utilizadas apenas para o monitoramento da aplicação na qual estão inseridas. Assim, a possibilidade de outros sistemas obterem estes dados de maneira autônoma, por meio da Internet, é cerceada. Em outros trabalhos, não é construído um

sistema de gerenciamento das falhas, uma vez que apenas a detecção ou o diagnóstico da falha é realizado. Outra questão reside no fato de as abordagens empregadas serem, majoritariamente, realizadas por meio de algoritmos descentralizados ou por meio de tratamentos mais complexos. Portanto, este projeto se destaca por propor a concepção e a aplicação de um sistema de gerenciamento de falhas centralizado e que possibilite a integração das RSSF à IoT.

4 Modelo

Nesta seção, são apresentadas as características do modelo para o gerenciamento de falhas proposto. São expostos os contornos do problema, ou seja, quais as tarefas a serem realizadas pelo sistema de gerenciamento de falhas e quais os tipos de erros tratados.

4.1 Modelo de Gerenciamento de Falhas Proposto

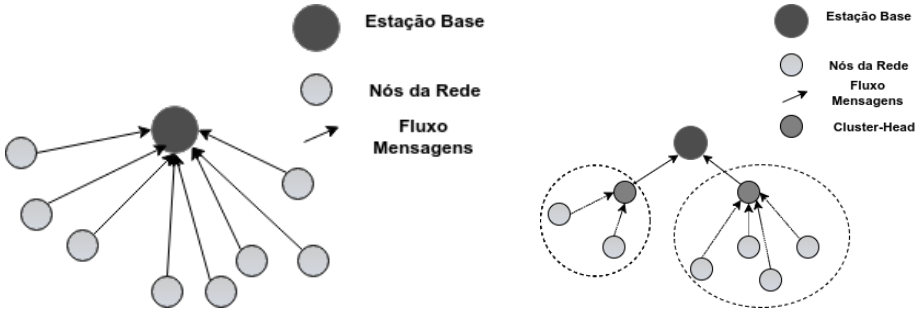
A arquitetura de gestão de erros é dividida em três fases. Como apresentado na seção 2.6, existem 4 (quatro) diferentes abordagens para o gerenciamento de falhas em redes sem fio, sendo dois grandes grupos (centralizado e descentralizado) e outros dois tratamentos híbridos (*agent-based* e hierárquica). O modelo proposto corresponde a um sistema centralizado inteligente de gerenciamento de falhas denominado SCIGF. A principal características de uma abordagem centralizada é a existência de uma entidade (estação) principal, normalmente, com maior capacidade de processamento, que é responsável por analisar todo o tráfego de informação dentro da rede e indicar possíveis falhas dos sensores. Também é papel da estação base realizar o tratamento deste erro, seja por meio da tentativa de correção da falha ou da eliminação do nó faltoso. Nessa tarefa de gerenciamento de falhas, a estação base, necessita analisar cada uma das mensagens trocadas pelos nós da rede, mantendo-se a todo momento em um estado de funcionamento pleno.

Quando existe a necessidade de cada nó da rede enviar pelo menos uma mensagem, periodicamente, para a estação base, são gerados múltiplos processos de envio e recebimento de informação de maneira desnecessária. Muitas destas mensagens carregam dados úteis apenas para o gerenciamento de falhas. Como em vários momentos os transmissores e receptores dos nós da rede necessitam estar em funcionamento, o consumo de energia tende a aumentar de maneira significativa. Por este motivo, um modelo centralizado, normalmente, é preterido em relação a uma abordagem descentraliza. A Figura 23 exemplifica os modelos centralizado e descentralizado de gerenciamento de falhas. Ao analisar a Figura 23, pode ser visto que a estação base não necessita monitorar todo o fluxo da rede, pois o pré-processamento é realizado pelo *cluster-head*. Além disso, o intervalo de troca de mensagens para gerenciamento de falhas dentro de um *cluster* é definido localmente, portanto, os nós da rede, não necessitam estar em funcionamento pleno em todo instante.

A principal vantagem em adotar um sistema de gerenciamento centralizado reside no fato deste modelo apresentar uma abordagem de implementação simplificada quando comparada com modelos descentralizados. Por exemplo, no tratamento em *clusters*, existe a necessidade de dividir a RSSF em regiões com características similares. Nesse processo de

Figura 23 – Comparação entre as duas principais abordagens em gerenciamento de falhas.

(a) Modelo centralizado. (b) Modelo descentralizado.



Fonte: Adaptado de Venkatesan e Shanmugavel (2014).

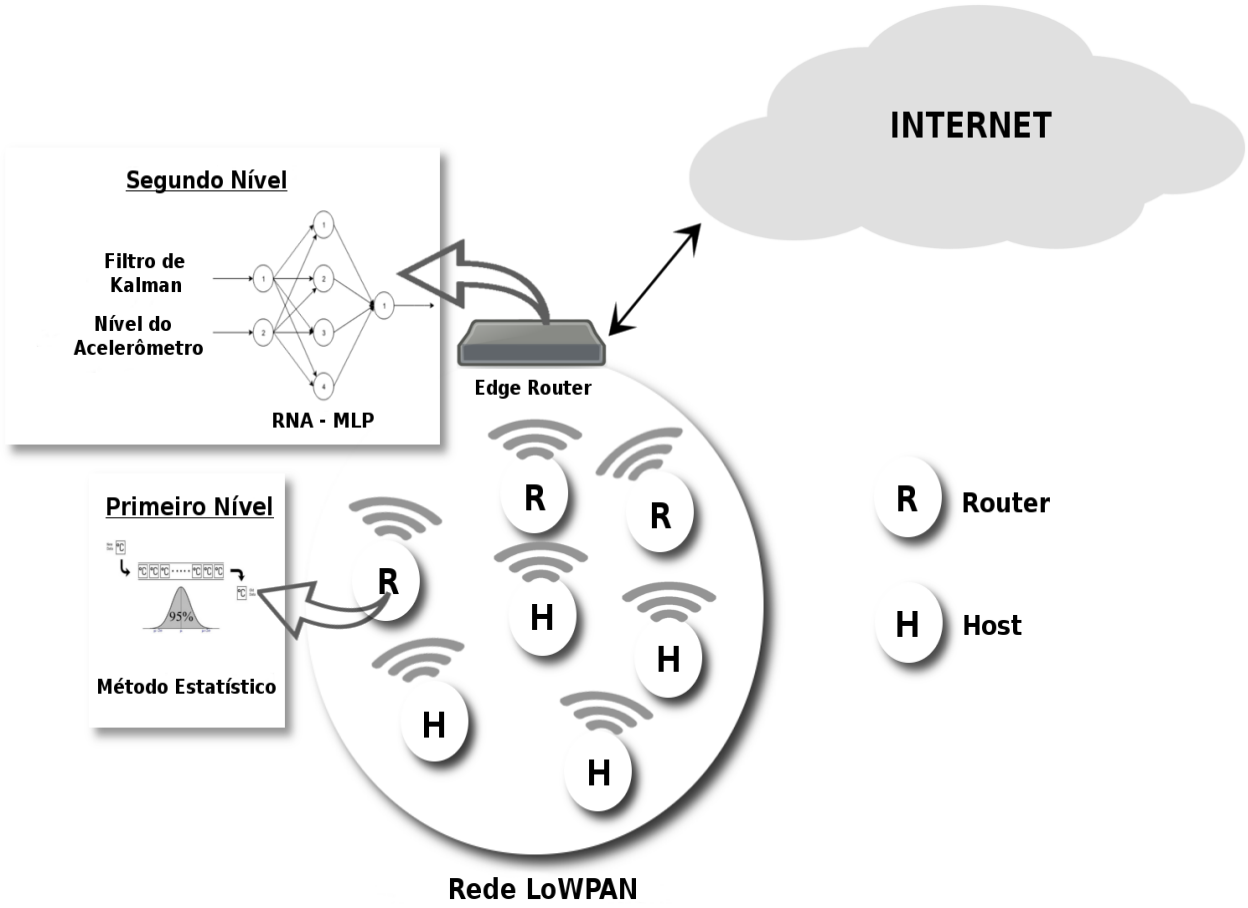
definição de regiões, os *cluster-head* devem ser formados por nós com maior capacidade de processamento, pois algumas variáveis (localização geográfica, quantidade de dados, tipo de dados medidos etc) devem ser levadas em consideração para a definição da região. Na abordagem descentralizada por meio de *agent-based*, a escolha do *agent-based* também é dispendiosa. Isso ocorre, já que o agente necessita possuir características especiais como maior capacidade de armazenamento de energia e possibilidade de percorrer cada um dos nós presentes na rede. Assim, quando comparado com outros modelos de gerenciamento de falhas em RSSF, o tratamento centralizado apresenta reduzida complexidade de implementação.

O SCIGF visa manter a principal característica de um sistema centralizado, exposta no parágrafo anterior, mas, garantindo níveis de consumo de energia inferiores às abordagens centralizadas comumente apresentadas na literatura. Outra característica do SCIGF é manter a taxa de falsos positivos e falsos negativos nos mesmos níveis das abordagens presentes na literatura, sejam elas centralizadas, descentralizadas ou híbridas.

O SCIGF apresenta 3 (três) principais inovações quando comparado a outros modelos encontrados na literatura. A primeira diferença deve-se ao fato deste modelo ser construído sobre o protocolo 6LoWPAN, o que possibilita a completa integração da RSSF com a Internet. A segunda característica, que diverge dos tratamentos anteriores, corresponde à utilização de 2 (duas) RNA com características distintas, mas com o mesmo propósito de classificar as falhas da RSSF. Por último, o SCIGF, emprega um modelo estatístico para um pré-diagnostico local de possíveis erros de dados sem a necessidade da escolha de *clusters* ou definição de agentes com maior capacidade de processamento. Assim, o SCIGF apresentado nesta seção utiliza algumas técnicas existentes na literatura, diferenciando-se das abordagens usuais por proporcionar a integração das RSSF com a IoT sem comprometer o desempenho de ambas as redes.

O SCIGF é utilizado para gerenciar falhas relativas aos dados, uma vez que este tipo de erro está, estreitamente, relacionado à falhas de *hardware* que acometem os nós de

Figura 24 – Representação do SCIGF com integração à Internet

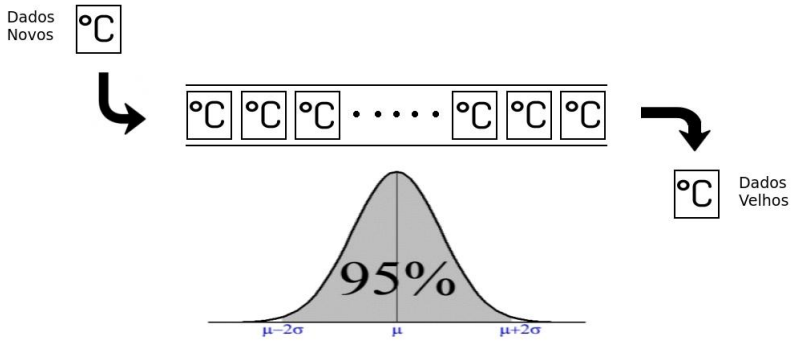


uma RSSF. A Figura 24 apresenta o esquema geral de funcionamento do modelo proposto com a distinção de cada uma das etapas do sistema. Assim, com esta configuração de sistema de gerenciamento de falhas, é possível realizar a integração entre a RSSF e a IoT sem que o desempenho de ambas seja prejudicado.

4.2 SCIGF - Processo de Detecção de Falhas

O gerenciamento de falhas apresentado é composto por dois níveis de detecção: detecção própria e análise pela estação base. O primeiro nível, denominado detecção própria, é realizado por cada um dos nós da rede. A existência deste nível inicial de detecção de falhas visa reduzir a taxa de falsos positivos e negativos assim como restringir a quantidade de mensagens a serem trocadas pelo sistema de gerenciamento de falhas. O segundo nível, conhecido como detecção da estação base, ocorre no nó de maior capacidade de processamento (estação base, roteador de borda ou *edge-router*). Esta etapa é responsável pela decisão sobre a classificação do evento recebido, indicar se o dado captado representa uma falha ou é apenas uma variação normal no comportamento do sistema. A junção destes dois níveis tem por objetivo aumentar a confiabilidade na classificação dos eventos recebidos.

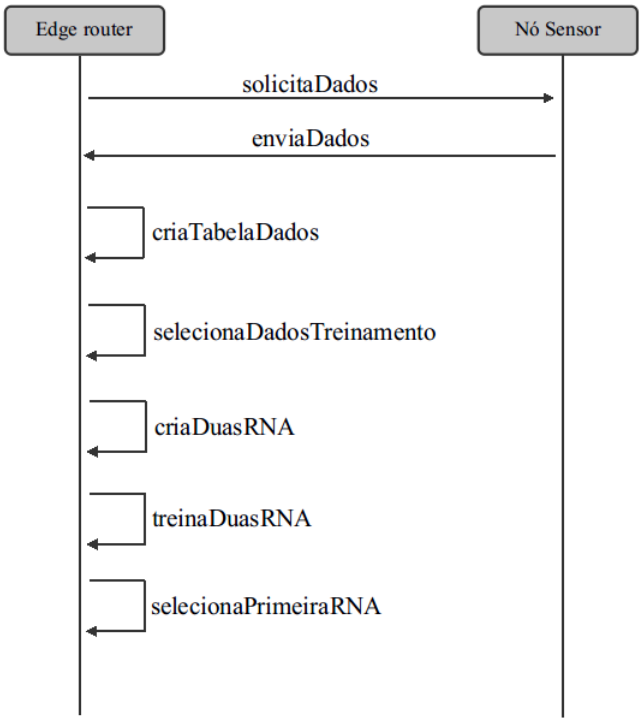
Figura 25 – Estratégia estatística adotada para a fase local (primeiro nível) do SCIGF.



A detecção própria ocorre por meio da adoção de uma inferência estatística, onde o dado coletado por um nó é comparado com um histórico de dados coletados pelo próprio nó. Esta etapa consiste em coletar, localmente, os dados medidos pelo próprio nó sensor e armazená-los em uma fila. Quando 30 (trinta) destas medidas (para o teste, será utilizada a temperatura) são reunidas, é criado um modelo de distribuição normal destes dados (são encontrados a média e o desvio padrão). Assim, cada uma das novas medidas são adicionadas a esta fila de dados e comparadas, estatisticamente, com o modelo normal criado, como pode ser visto na Figura 25. Caso estas novas medidas estejam distantes de mais de 2 (dois) desvios padrões da média do conjunto de dados coletados, esta medida é enviada para análise na RNA central presente no *edge router*. A fim de evitar o problema deste conjunto de dados coletados estarem em uma região faltosa (o nó sensor está com uma falha e todas as 30 (trinta) medidas estarem normalmente distribuídas ao redor desta falha), periodicamente, um intervalo aleatório de tempo é disparado e uma medida coletada é enviada para análise no *edge router* mesmo que não seja identificada como falha pelo nó sensor. Esta estratégia é similar à empregada por Liu et al. (2013), mas com um enfoque diferente, uma vez que o próprio nó torna-se responsável pela análise dos dados coletados.

Como apresentado na seção 2.6, um sistema de gerenciamento de falhas é composto por 3 (três) processos (ou fases): detecção da falha, diagnóstico da falha e recuperação do sistema. A Figura 26 apresenta o fluxo de informação presente no primeiro processo: Detecção da falha. Neste processo, o *edge router* colhe informação sobre a variável medida por cada um dos nós sensores presentes em seu *whiteboard*, realiza um pré-processamento destes dados e treina a rede neural. Esta análise inicial tem por objetivo selecionar os dados para treinamento da rede de modo a minimizar ambiguidades, cobrir amplamente o problema, adicionando o maior número de exceções, além de reduzir o tamanho dos vetores de entrada de dados. Nesse momento são configuradas as duas redes neurais.

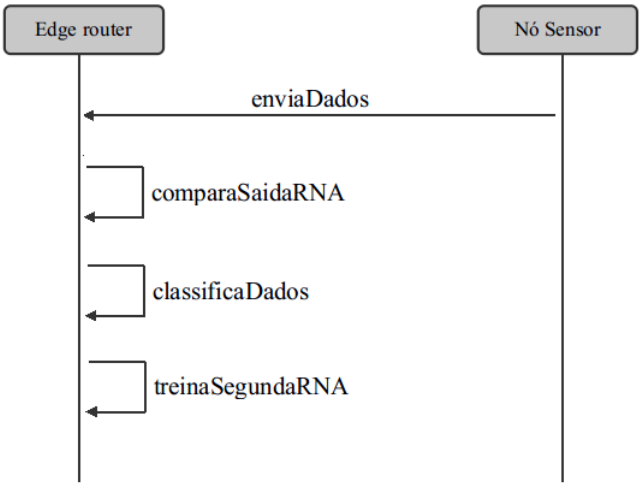
Figura 26 – Exemplo das trocas de mensagens realizadas no treinamento para o processo de detecção da falha do SCIGF.



4.3 SCIGF - Processo de Diagnóstico de Falhas

Nesta fase, após o treinamento da RNA, é realizado o monitoramento do funcionamento de toda a rede. Os dados coletados são classificados como próprios ou não-próprios, identificando-se, assim, se os valores coletados estão dentro dos limites de funcionamento normal da rede ou se são identificados como falhas dos dados. Na Figura 27, estão expressas as informações necessárias para a etapa de diagnóstico da falha.

Figura 27 – Exemplo das trocas de mensagens realizadas no diagnóstico da falha para o SCIGF.



A segunda fase é realizada pela estação base. A estação base é escolhida para a realização deste processo, pois, normalmente, é o elemento da rede que possui maior poder de processamento e uma fonte de energia com capacidade de armazenamento superior aos demais nós. As duas RNA criadas na etapa de detecção de falhas são do tipo MLP (esta escolha deve-se aos resultados alcançados pelo experimento 2). A fim de aproveitar uma das vantagens de um sistema centralizado de gerenciamento de falhas (baixa complexidade computacional) escolheu-se redes neurais do tipo MLP, mesmo sabendo-se da existência de outros classificadores com características diferentes. A abordagem em SVM, por exemplo, poderia ser utilizada, entretanto por ser um método não-paramétrico não possui tamanho fixo o que pode ser desastroso para a aplicação, uma vez que a estação base, mesmo com uma maior capacidade de processamento, ainda possui memória reduzida. Outra característica do algoritmo SVM, é que o ajuste do modelo, normalmente, é mais complexo que o treinamento de uma rede MLP. Uma comparação entre o consumo de energia, para a solução do problema XOR, em alguns tipos de RNA é apresentado na seção 6.

As redes MLP utilizadas para o diagnóstico da falha apresentam 2 (duas) entradas e apenas uma saída. Na camada escondida existem 4 (quatro) neurônios. As duas entradas destas redes são: 1) a diferença entre o valor previsto, obtido pela saída do filtro de Kalman, e o valor real e 2) o nível do acelerômetro. A classificação da rede MLP é do tipo binária (0 ou 1). A saída do tipo 1 (um) indica que a medida recebida foi classificada como uma variação normal do sistema e o valor 0 (zero) mostra que a medida representa uma falha do sistema de *hardware*.

A fim de ser possível rastrear a dinâmica da rede de sensores, é utilizado o filtro de Kalman como um previsor das medidas realizadas pelo sensor. Analisando as medidas atuais e realizando a previsão por meio do filtro (ajuste do ganho), é possível indicar qual deveria ser a próxima medida a ser realizada pelo sensor, pois é assumido que flutuações nas medidas são inerentes ao processo de sensoriamento. Essa diferença absoluta entre o valor provável de uma medida e o valor realmente aferido pelo sensor é utilizada como uma das entradas da rede MLP. Com esta entrada, se espera que a rede MLP possa ter uma medida da distância que o valor provável encontra-se da medida realizada pelo sensor.

A segunda entrada é a classificação da intensidade de um possível impacto sofrido pelo nó da rede. O acelerômetro é utilizado para medir a variação do vetor aceleração resultante do impacto, uma vez que danos físicos causados aos nós são uma das maiores causas de falhas de *hardware* (HARTE; RAHMAN; RAZEEB, 2005). Para a geração da segunda entrada da RNA foram utilizadas as medidas dos três eixos do acelerômetro (eixos x, y e z). Em cada um destes eixos, são medidas as acelerações instantâneas em relação à gravidade. Em cada eixo, é calculada a diferença entre o valor atual da aceleração e o valor imediatamente anterior. Para calcular o efeito desta variação, em cada um dos eixos utiliza-se a equação 15 a fim de indicar a intensidade total do impacto sofrido por um nó. A

intensidade desse possível choque é classificada em 6 (seis) níveis de ordem crescente (0, 1, 2, 3, 4, 5), onde o nível 0 (zero) indica que nenhum choque significante ocorreu e o nível 5 (cinco) representa um choque significativo. Experimentalmente, é traçado um limiar para cada um dos níveis de impacto.

$$magnitude = \sqrt{(a_{x(k-1)} - a_{xk})^2 + (a_{y(k-1)} - a_{yk})^2 + (a_{z(k-1)} - a_{zk})^2} \quad (15)$$

na qual:
a_{xk}: aceleração relativa no eixo x no tempo k;
a_{yk}: aceleração relativa no eixo y no tempo k;
a_{zk}: aceleração relativa no eixo z no tempo k.

O chaveamento entre cada uma das duas redes MLP criadas ocorre quando mais de 10% dos nós da mesma RSSF foram detectados com falhas ou quando mais de 90% das medidas analisadas foram classificadas como normais. Este chaveamento visa minimizar a degradação do processo de reconhecimento de padrões ao longo da vida da RSSF.

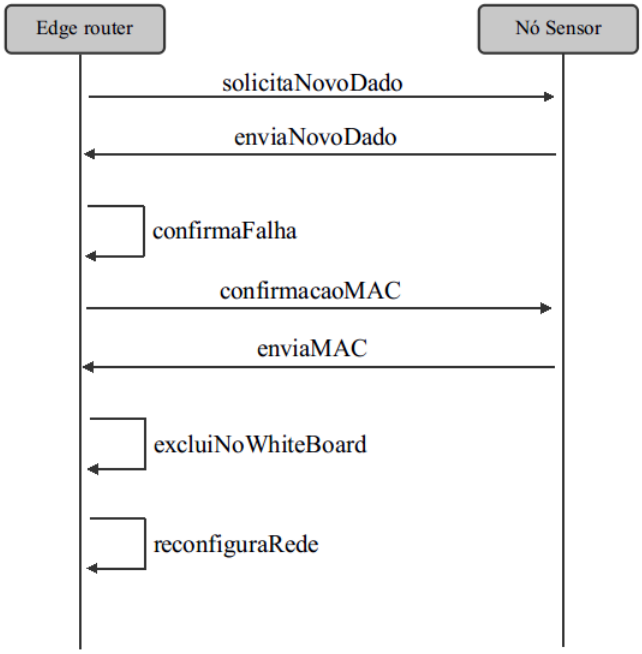
4.4 SCIGF - Processo de Recuperação da Falha

A última etapa do processo de gerenciamento de falhas (recuperação) também é realizado pela estação base. Para possibilitar o fácil acesso à Internet e integrar o sistema a uma rede em IoT, o modelo proposto é concebido utilizando-se o protocolo 6LoWPAN. Uma vez que é utilizado o 6LoWPAN, quando uma falha é diagnosticada, o nó responsável pelo erro é removido da *Whiteboard* por meio da exclusão do respectivo IP e envio de e-mail para o usuário responsável. Assim, o nó deixará de ser membro da RSSF até que seja substituído ou seja realizada a manutenção. A *Whiteboard* é onde o *edge router* armazena todos os endereços IP dos sensores pertencentes a LoWPAN. O nó marcado como faltoso não é excluído imediatamente da rede, pois quando uma falha é identificada o nó é enviado para quarentena. Assim, um contador de novas falhas é incrementado e o *edge router* envia um total de 30 (trinta) mensagens em intervalos aleatórios de tempo a fim de garantir que o nó está realmente apresentando erro nas medições. Após ser comprovado que o nó, realmente, apresenta falha este é definitivamente removido da *Whiteboard*. Novas rotas de comunicação são, então, criadas (por meio do próprio protocolo 6LoWPAN) e novos nós são escolhidos para realizar o treinamento das redes na primeira etapa de gerenciamento (detecção da falha). A Figura 28 mostra o fluxo de informação para a última etapa de gerenciamento que é a recuperação da falha.

Como pode ser visto, a ultima etapa do gerenciamento de falhas está intimamente relacionado ao protocolo 6LoWPAN da camada de rede. Entretanto, para adaptar o SCIGF

a outros protocolos de rede é necessário apenas que a forma de exclusão do nó sensor seja modificada, assim todas as demais funcionalidades do SCIGF serão preservadas.

Figura 28 – Representação da terceira fase do SCIGF.



5 Metodologia

Gil (2002) classifica os tipos de pesquisa quanto aos objetivos e aos procedimentos. Quanto aos objetivos, as pesquisas se dividem em exploratória, descritiva e explicativa. Assim a pesquisa proposta neste projeto pode ser classificada como exploratória, pois tem por objetivo aprimorar a aplicação de técnicas de gerenciamento de falhas já existentes ao contexto de Internet das Coisas. Quanto aos procedimentos a serem utilizados, esta dissertação pode ser classificada como experimental, uma vez que é proposto um algoritmo centralizado de gerenciamento de falhas dos dados para integração de RSSF à IoT, utilizando-se o protocolo 6LoWPAN para comunicação. Os procedimentos experimentais utilizados serão apresentados na seção 5.2.

5.1 Hipóteses

A hipótese pode ser vista como uma ideia testável com possibilidade de vir a ser a solução para um problema (GIL, 2002). Neste contexto, para o presente projeto de pesquisa são formuladas as seguintes hipóteses:

O SCIGF não apresenta taxa de classificação correta de falhas superior a 95%.

O consumo de energia do SCIGF, com o primeiro nível de autodetecção, não é 50% inferior ao SCIGF sem este nível.

5.2 Planejamento de Experimentos

Uma descrição de como será conduzido o processo de experimentação e os materiais utilizados para a construção dos experimentos é apresentada nesta seção. Esses experimentos visam verificar a aplicabilidade real do modelo proposto e compará-lo com os trabalhos presentes na literatura. São realizados 6 (seis) diferentes experimentos. Também é apresentado um estudo de caso para a aplicação do SCIGF.

Todos os experimentos foram realizados por meio da utilização de uma máquina virtual para a execução do SO Contiki. As simulações e implementações foram realizadas dentro do ambiente COOJA. Como apresentado na seção 2.11, estas ferramentas são as

mais utilizadas para o simulação de RSSF, pois e possível que sejam emulados diferentes tipos de nós sensores. Para todos os processos de simulação foram escolhidos os nós do tipo Zolertia Z1. A escolha do Zolertia Z1, para os nós da rede, deve-se ao fato deste ser o *hardware* com maior capacidade de processamento presente no COOJA.

Inicialmente, é construída a RSSF 6LoWPAN (Simples). Para testar o funcionamento correto desta rede, é necessário verificar a comunicação desta com a Internet. O experimento inicial consiste em realizar o comando de consulta de rota a fim de verificar a conectividade entre o *browser* e um nó específico da rede 6LoWPAN construída. Testar a conectividade da RSSF com a Internet é necessário, pois é fundamental que o SCIGF possua integração com IoT.

Como uma das premissas para a concepção de um modelo aplicável a RSSF é a capacidade do sistema em manter um consumo reduzido de energia, o segundo experimento visa comparar o consumo de energia de alguns algoritmos de inteligência computacional comumente utilizados como classificadores. O segundo experimento tem por objetivo analisar qual das RNA possui o menor consumo de energia para a solução do problema XOR de classificação neural. As abordagens selecionadas para esta aplicação foram: redes MLP, redes RBF e redes ANFIS. A escolha destas arquiteturas de classificação deve-se ao fato de serem amplamente utilizados na literatura como classificadores neurais. A métrica de desempenho utilizada será o consumo estimado de energia do processador (CPU – do inglês *Central Processing Unit*). Para esta simulação, será utilizado o *PowerTrace*, ferramenta desenvolvida por Dunkels et al. (2011), que permite simular o consumo de energia em diferentes plataformas de *hardware*. Foi mostrado, experimentalmente, por Cherifi et al. (2015), que a precisão das simulações com o *PowerTrace* alcança 94% de uma aplicação real.

O terceiro experimento tem a finalidade de verificar o classificador estatístico criado (primeiro nível de detecção de falhas). Inicialmente, é necessário realizar um teste de normalidade para a distribuição dos dados a fim de ser possível utilizar o método estatístico proposto. É necessário realizar o teste de normalidade, pois o SCIGF utiliza-se desta característica para detectar possíveis falhas no primeiro nível de detecção, além disso, o filtro de Kalman, possui como requisito de utilização a normalidade dos dados. O desempenho deste classificador é avaliado por meio da curva *Receiver Operating Characteristic* (ROC) e da matriz de confusão. Assim, é possível identificar a eficiência desta etapa de classificação. A curva ROC é traçada utilizando-se as taxas de positivos corretos e de falsos positivos (VUK; CURK, 2006). Por meio da analise da matriz de confusão, é possível ter uma visão geral da eficiência do classificador utilizado (DAVIS; GOADRICH, 2006).

Uma vez que os dados do acelerômetro são utilizados como uma das entradas do classificador neural, e deseja-se ter uma simulação mais próxima da realidade, existe a necessidade de um experimento adicional. Assim, o quarto experimento é realizado a fim de

gerar dados para a simulação de impactos sofridos por um nó sensor em um ambiente real. Estes dados são obtidos por meio da realização de impactos de diferentes intensidades a um dispositivo que possui um acelerômetro de três eixos. O dispositivo utilizado para esta simulação corresponde a um aparelho celular. Como pode ser visto nos trabalhos de Harte, Rahman e Razeeb (2005), Huang e Su (2015) e Lewicki, Lambert e Wagoner (2015), esta técnica é bastante utilizada em várias áreas para medir as vibrações em determinado equipamento a fim de indicar possíveis falhas de funcionamento.

O quinto experimento consiste em identificar o funcionamento correto do sistema de detecção/diagnóstico de falhas. Este sistema é constituído de duas RNA e um classificador estatístico responsáveis por identificar o funcionamento próprio e não-próprio da rede separando falhas dos dados de situações de operação normal do sistema. O experimento que verifica a operação correta da rede consiste em, após o treinamento da RNA, realizar a validação dos resultados, utilizando-se novos dados gerados pela rede que não foram previamente aplicados ao treinamento. Testes relativos à análise de comportamento, por exemplo, emprego de entradas especiais (valores em extremos), também são utilizados, além da verificação dos pesos encontrados a fim de identificar o dimensionamento correto da rede. Os resultados destes testes também serão apresentados por meio da curva ROC e da matriz de confusão. A construção da curva ROC e da matriz de confusão, é realizada utilizando-se o *software* Matlab. A escolha desta ferramenta para a confecção destes gráficos, deve-se a maior precisão e facilidade de criação dos mesmos. Por último, serão comparados os dados encontrados para o modelo de gerenciamento de falhas sugerido com alguns sistemas presentes na literatura. Mesmo considerando que os modelos de outros autores não são específicos para a integração das RSSF com a IoT. Esta comparação é realizada a fim verificar a aplicabilidade real do modelo apresentado na seção 4.1.

O sexto experimento tem como finalidade estimar o nível de energia consumido pelo sistema de gerenciamento de falhas. Neste experimento, também, é investigada a diferença de consumo energético obtido entre uma abordagem centralizada que, não inclui a detecção de falhas em primeiro nível (modelo centralizado padrão), e o tratamento sugerido pelo presente trabalho. O modelo centralizado padrão consiste no envio periódico de mensagens para análise na estação base sem o auxílio de uma classificação prévia destes dados, como a realizada pelo primeiro nível de detecção presente no SCIGF.

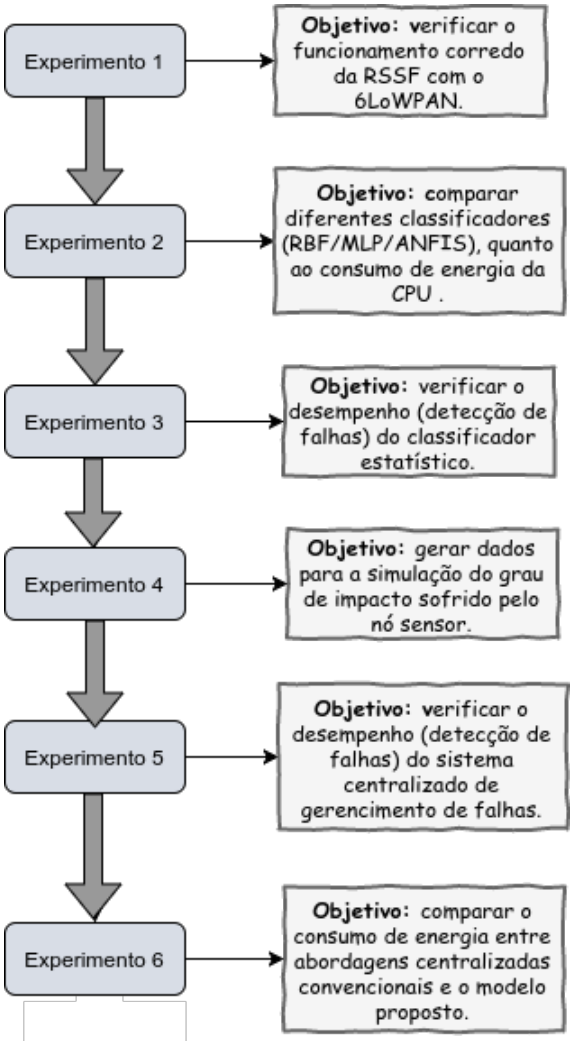
O estudo de caso consiste em exemplificar uma aplicação para uma arquitetura em IoT e na simulação de três diferentes cenários que objetivam identificar se o sistema proposto é capaz de ser utilizado na integração com a IoT. Neste sentido, uma arquitetura IoT é empregada para o monitoramento do cultivo de mudas em casas de vegetação. É exemplificado como cada um dos níveis da arquitetura deve ser implementado, destacando-se os protocolos e serviços a serem empregados, além de indicar como o SCIGF está relacionado com a aplicação. Também neste experimento, uma rede simulada é conectada à

Internet e um usuário (também simulado) deve receber um *e-mail* (em tempo real) indicando possíveis falhas e *status* da rede, quando:

1. Uma quantidade igual ou superior a 20% dos nós for identificada com falha;
2. Um mesmo nó for colocado em quarentena por um número superior a 2 (duas) vezes;
3. Um mesmo nó sofra impactos classificados como nível 5.

A Figura 29 apresenta um diagrama com a sequência das etapas metodológicas do trabalho. Também, são apresentados os objetivos de cada um dos experimentos conduzidos.

Figura 29 – Etapas metodológicas do trabalho.



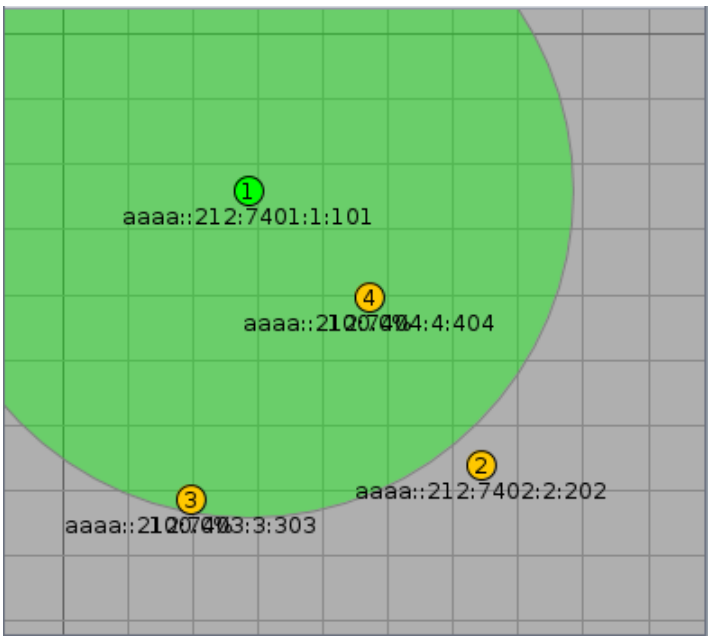
6 Resultados e Discussões

Nesta seção, são apresentados os resultados encontrados após a construção do modelo proposto na seção 4. Estes resultados foram alcançados por meio de simulações no ambiente virtual *Cooja*. Como apresentado, anteriormente, na seção 5, foram realizados 6 (seis) experimentos nomeados, respectivamente, como E_1, E_2, E_3, E_4, E_5, E_6. Também é apresentado o estudo de caso referente a aplicação do SCIGF ao cultivo de mudas em casas de vegetação.

6.1 Experimento E_1

Este primeiro experimento tem por finalidade apenas apresentar o modelo de simulação utilizado. A abordagem proposta consiste em uma rede composta por 4 (quatro) elementos. Estes elementos estão numerados de 1 a 4, sendo que o nó 1 (um) corresponde ao *edge router*, o nó 4 (quatro) é o *router* e os demais representam os nós sensores da rede. Colocando-se estes elementos constituintes da rede em ordem crescente de capacidade de processamento, resulta: nós sensores, *router* e *edge router*. A Figura 30 apresenta a rede simulada.

Figura 30 – Configuração do modelo de simulação de rede.

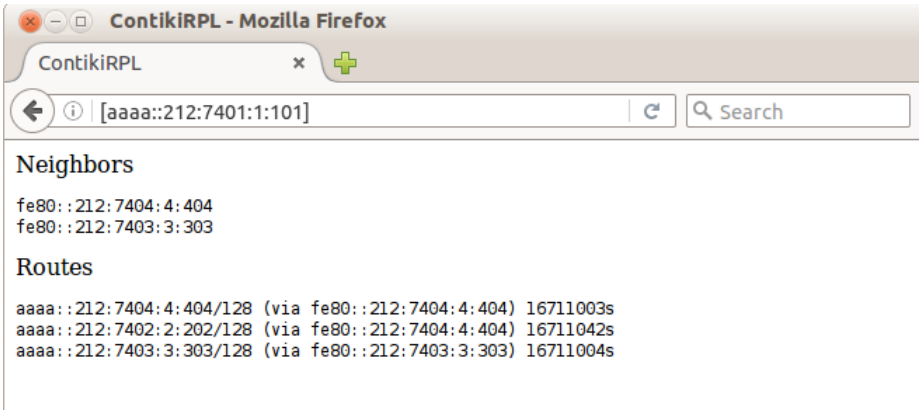


Na Figura 30, é possível visualizar alguns círculos concêntricos e centralizados sobre o nó 1 (um). O círculo na cor verde simboliza a região em que os sinais enviados pelo elemento 1 (um) são completamente recebidos pelo destinatório. A região externa a

este círculo identifica a demarcação do espaço em penumbra de sinal. Assim, alguns sinais enviados podem ser perdidos. Todos os nos, presentes na Figura 30, são do tipo Zolertia Z1. O no 1 (um) ,estação base, também realiza a medidas de temperatura.

A Figura 31 exprime uma resposta de consulta à rede *Routing Protocol for Low Power Networks* (RPL) criada por meio do protocolo 6LoWPAN. Nesta figura é possível verificar as rotas criadas entre o nó 1 (um) e os demais elementos da rede.

Figura 31 – Resposta à consulta por meio do *browser*.



Como pode ser visto pela Figura 31, os resultados demonstram que a configuração inicial para simulação da rede ocorreu de maneira correta. Assim, para os demais experimentos é utilizado o modelo de rede presente na Figura 30.

6.2 Experimento E_2

Este segundo experimento tem por objetivo apresentar alguns dos motivos para a escolha de uma rede MLP como classificador do modelo. O experimento E_2 foi conduzido da seguinte maneira. Foram escolhidas 3 (três) abordagens de inteligência computacional utilizadas como classificadores binários: rede MLP, rede RBF e rede ANFIS. A escolha por estas redes deve-se ao fato de serem comumente relatadas como classificadores universais e apresentarem características similares de aplicação. Entretanto, cada uma delas possui particularidades no processo de implementação.

Para a simulação, implementou-se um problema com regiões não linearmente separáveis conhecido como problema XOR. Para encontrar a região do espaço que separa o agrupamento de saída de uma porta lógica XOR, cada uma das 3 (três) redes, anteriormente citadas, foi implementada utilizando-se os algoritmos mais simples existentes. Por algoritmos simples entende-se que nenhuma modificação visando a otimização dos processos de aprendizado foi empregada. Esta abordagem mais simplificada visa manter o nível de complexidade na implementação destas redes o mais próximo possível dos objetivos desta dissertação.

Tabela 1 – Dados utilizados para simulação E_2.

Entradas		**Saídas**
0	0	0
0	1	1
1	0	1
1	0	0

A rede MLP simulada apresenta 4 (quatro) neurônios na camada escondida, possui duas entradas e uma saída. Os dados para treinamento utilizados em todas as redes estão apresentados na Tabela 1. A rede RBF utilizada, também, apresenta 2 (duas) entradas, 1 (uma) saída e apenas uma camada escondida. A escolha dos centros dos dados é realizada de maneira aleatória e a função de base radial empregada é do tipo Gaussiana. Para a rede ANFIS, é escolhido um sistema *Fuzzy* com 4 (quatro) regras. O critério de parada empregado para o treinamento em todas as simulações é um erro (diferença entre a resposta obtida e real) inferior a 0,01%.

Todas estas redes foram implementadas em linguagem C e simuladas utilizando-se o *Contiki*/*Cooja* com a utilização do nó Zolertia Z1 (ZOLERTIA, 2016). A ferramenta empregada para as medidas do consumo de energia é o *PowerTrace*. Como o *PowerTrace* realiza a simulação do consumo de energia por meio da contagem dos pulsos realizados pelo processador em um intervalo de tempo, é necessário, para que o gasto de energia em Watts (W) seja calculado, empregar a equação 16. Uma vez que, neste experimento, não é realizado nenhum tipo de comunicação entre os nó da rede, apenas os dados referentes ao consumo do processador serão considerados. A Tabela 2 apresenta um dado tipico gerado pela ferramenta *PowerTrace*.

$$consumo(mW) = \frac{\Delta Tr \times A \times V}{RS \times Runtime} \tag{16}$$

na qual:
ΔTr: é a diferença entre contagens consecutivas do *clocktimer*;
A: é a corrente de alimentação do dispositivo;
V: é a tensão de alimentação do dispositivo;
RS: é o número de oscilações por segundo;
$Runtime$: é o intervalo de tempo em que as informações são coletadas.

Um exemplo de utilização da fórmula para um nó do tipo Zolertia Z1 e com os dados apresentados na Tabela 2 é apresentado a seguir pela Equação 17. Este nó possui 32.768 como número de oscilações por segundo, corrente de alimentação do processador igual a 0,33A, 3,0V de alimentação do processador e intervalo de coleta de informações igual a 10s (ZOLERTIA, 2010)

$$consumo(mW) = \frac{(8042 - 6154) \times 0{,}33 \times 3{,}0}{32768 \times 10} = 0{,}0057mW. \quad (17)$$

Tabela 2 – Exemplo de dados gerados pelo *PowerTrace*: CPU indica o processador, *Load Program Memory* (LPM), TX representa o transmissor e RX o receptor. Os valores em cada célula representam o acumulador de pulsos

CPU	LPM	TX	RX
6154	321736	2651	1968
8042	647531	2651	3888
14977	968273	5481	5856
16854	1294077	5481	7776
19115	1619494	5481	9696
20859	1945432	5481	11616

A Figura 32 apresenta a variação do consumo de energia para as redes MLP, RBF e ANFIS. Os dados foram coletados durante um intervalo de 2 min (120 segundos). O consumo total para cada uma das abordagens é expresso na Tabela 3.

Como pode ser verificado pela análise da Figura 32 e da Tabela 3, a abordagem para a solução do problema XOR que mais demanda processamento e, consequentemente, maior consumo de energia corresponde ao algoritmo ANFIS. Este maior consumo é justificado pela necessidade em ser calculada, durante o seu processo de aprendizagem, uma matriz pseudo-inversa, o que demanda maior processamento. O consumo intermediário ocorre no caso em que é utilizado o modelo RBF. Como o processo de clusterização ocorre de maneira aleatória, é necessário um número maior de iterações para encontrar o erro desejado. O menor processamento ocorre no caso de utilização da rede MLP, que é cerca de 20% do consumo apresentado pela abordagem RBF. Esta diferença já era esperada, pois para o treinamento *backpropagation* são realizados apenas cálculos simples para ajuste dos pesos, o que não demanda grande esforço computacional e consumo de memória.

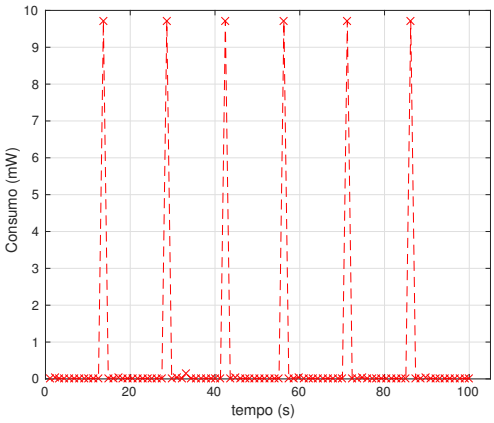
Tabela 3 – Consumo total: comparação entre cada uma das redes.

Rede	Consumo Total (mWh)
ANFIS	68,97
RBF	50,02
MLP	10,98

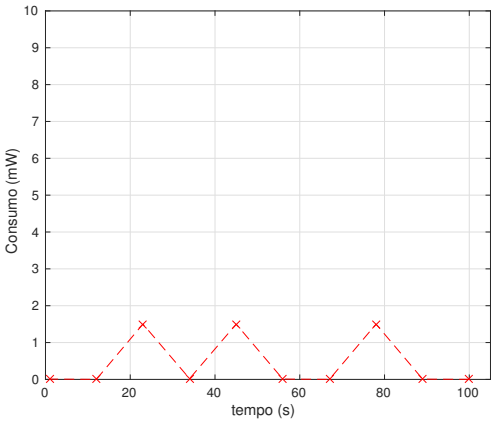
A solução do problema XOR apresenta baixa complexidade quando comparada a outros existentes na literatura. Portanto, em aplicações complexas, os resultados podem ser diferentes. Por estes resultados, têm-se indícios de que o classificador com menor consumo de energia é a RNA MLP. Devido a esta característica, no SCIGF é utilizado como classificador neural a RNA MLP.

Figura 32 – Consumo simulado das redes para solução do problema XOR.

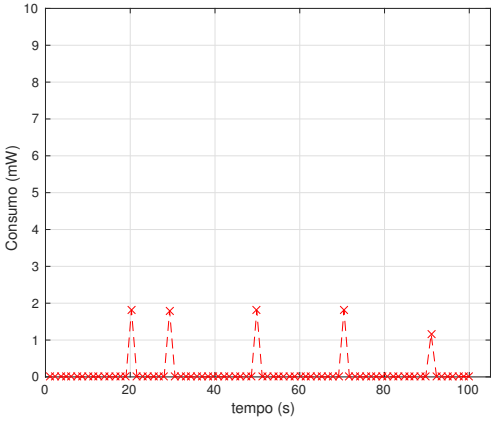
(a) Consumo rede ANFIS.



(b) Consumo rede RBF.



(c) Consumo rede MLP.



6.3 Experimento E_3

Este experimento tem por finalidade verificar a aplicabilidade do nível inicial no processo de detecção/diagnóstico de falhas. Esta etapa é executada por cada um dos nós

presentes na RSSF. Como este primeiro nível de detecção/diagnóstico utiliza-se de uma comparação estatística para julgar se uma medida realizada pelo próprio nó apresenta erro, algumas premissas devem ser atendidas.

Em primeiro lugar, os dados (medidas) utilizados para comparação devem constituir uma distribuição normal com a possibilidade de conhecer o seu valor médio e desvio padrão. Em segundo lugar, esta amostra de dados deve ser suficiente para inferir sobre o comportamento temporal da variável medida. Outra importante questão a ser investigada com este experimento é a premissa para a utilização do filtro de Kalman, uma vez que os dados devem ser normalmente distribuídos para que as precisões ocorram sem prejuízo ao SCIGF.

Para a simulação da detecção própria, são utilizados dados de temperatura. Estes dados foram obtidos de 4 (quatro) cidades de regiões distintas do estado de Minas Gerais, Diamantina, Montes Claros, São Lourenço e Belo Horizonte entre os dias 07/01/2017 e 08/01/2017. Estes dados foram coletados pelo *site* do Climatempo (2017) a cada 15 minutos. Um exemplo da base de dados utilizada pode ser visto na Tabela 4 para a cidade de Diamantina.

Tabela 4 – Exemplo de dados: temperatura na cidade de Diamantina (07/01/2017).

Cidade: Diamantina	
Hora do Dia	**Temperatura (.$^{\circ}C$)**
00:00	21,0
00:15	21,0
00:30	20,5
00:45	20,5

A fim de verificar a normalidade dos dados, é realizado o teste de *Kolmogorov-Smirnov*. Este teste utiliza como hipótese nula a afirmação de que os dados provêm de uma distribuição normal. O nível de confiança, utilizado para o teste, é de 95% (LILLIEFORS, 1967). A Tabela 5 apresenta o resultado encontrado após a realização do teste de normalidade.

Tabela 5 – Teste Kolmogorov-Smirnov: H_0 não rejeitada.

Hipótese Nula	**Nível de Confiança**	**Resultado**
Dados provêm de uma distribuição normal	95.0%	0.0

Por meio da análise da Tabela 5, verifica-se que a hipótese nula não é rejeitada. Assim, admite-se que os dados provêm de uma distribuição normal. Como estes dados estão normalmente distribuídos, é possível utilizar a abordagem de comparação entre a variabilidade das medidas. Esta variação é inerente a todo processo de sensoriamento

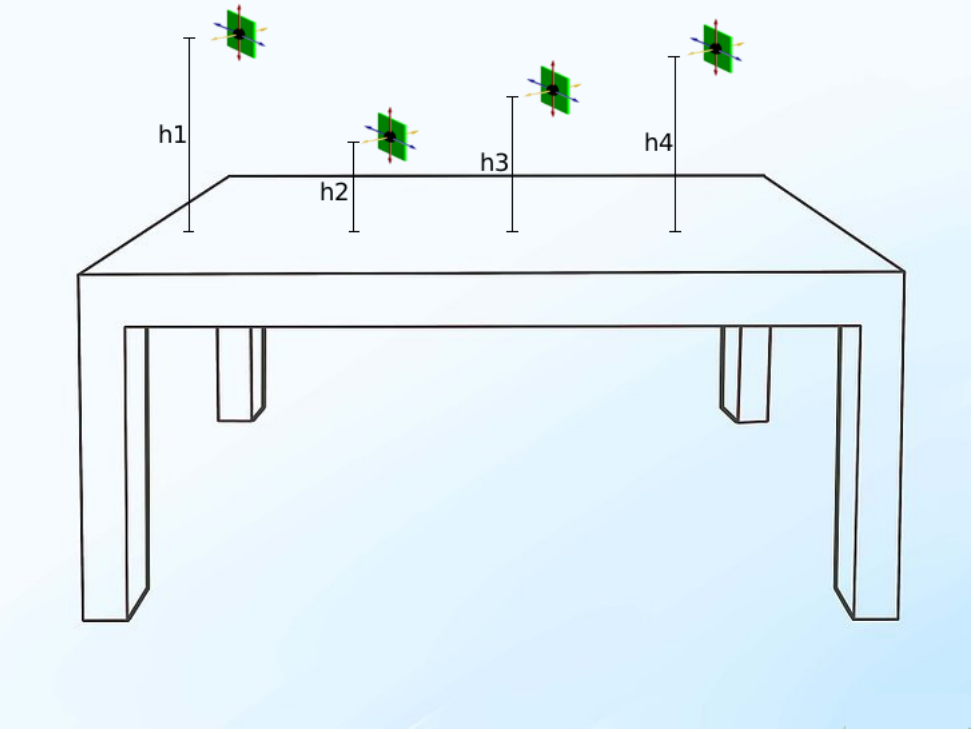
e deve ser ignorada por um sistema de gerenciamento de falhas, pois não deve ser classificada como uma falha dos dados. Assim, apenas medidas realizadas que diferem da média estimada por mais de 2 (dois) desvios padrões devem ser enviadas para análise, contribuindo para a redução do consumo de energia do sistema.

6.4 Experimento E_4

O quarto experimento indica como os dados do acelerômetro foram gerados para a simulação de impactos sofridos pelos nós de uma rede. Foram utilizados dados reais, pois é necessário identificar os limiares de cada um dos níveis que serão empregados como uma das entradas da RNA. Além disso, este experimento visa gerar uma aproximação dos impactos reais que possam acometer um nó.

Os dados foram coletados por meio de um acelerômetro de 3 (três) eixos como presente em *smartphones*. A variação de aceleração relativa foi simulada deixando-se o acelerômetro em queda livre de diferentes alturas sobre uma superfície sólida e plana (mesa de testes). A Figura 33 exemplifica o experimento de simulação de impacto.

Figura 33 – Experimento realizado para simulação de impactos.



Na Figura 34 estão expostos os resultados obtidos após a realização do experimento presente na Figura 33. Cada uma das linhas do gráfico representa a variação total das acelerações em cada um dos 3 (três) eixos. Os limiares que separam cada um dos 6 (seis) níveis utilizados como entrada da RNA são obtidos por meio da variação destes eixos. Por exemplo, valores entre o intervalo $18{,}42m/s^2$, $-6{,}55m/s^2$ e $1{,}14m/s^2$ dos eixos Z, X, e

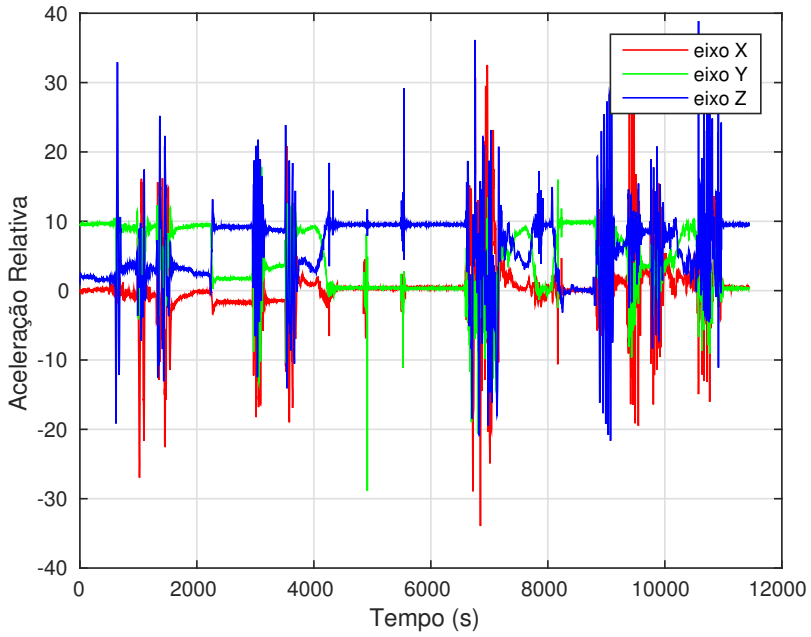
Y no tempo 4.264s são classificados como nível 2, pois representam a queda sobre uma superfície de com maior capacidade de dissipação de energia e, consequentemente, menor variação de aceleração relativa. Pelo gráfico presente na Figura 35, é possível identificar diferentes níveis de impactos (variações nos eixos x, y e z) calculados pela Equação 15.

$$magnitude = \sqrt{(a_{x(k-1)} - a_{xk})^2 + (a_{y(k-1)} - a_{yk})^2 + (a_{z(k-1)} - a_{zk})^2} \quad (15)$$

na qual:
a_{xk}: aceleração relativa no eixo x no tempo k;
a_{yk}: aceleração relativa no eixo y no tempo k;
a_{zk}: aceleração relativa no eixo z no tempo k.

Como o acelerômetro mede a aceleração relativa à gravidade (aproximadamente $9{,}8m/s^2$ no local dos testes), é realizado um truncamento no valor de $9{,}8m/s^2$. Assim, a Figura 35 apresenta o exemplo de alguns níveis simulados.

Figura 34 – Modelo de dados encontrados após o experimento E_4.



De posse de cada um desses limiares, é possível estabelecer os níveis de impactos por meio da equação 15. Por exemplo, dados com uma magnitude entre 7,5 e 12,5 são classificados como nível 2 e magnitudes entre 17,5 e 22,5 pertencem ao nível 4. A Figura 36 apresenta os níveis utilizados como uma das entradas da rede neural MLP para o processo de detecção de falha na estação base.

Como foi apresentado anteriormente pelas Figuras 34, 35 e 36, a coleta dos dados para a simulação de impactos sofridos pelos nós da rede, foi realizada para de obter com maior fidelidade a variação de acelerações acometidas aos dispositivos que são alvo de

Figura 35 – Níveis escolhidos segundo a variação expressa pela equação 15.

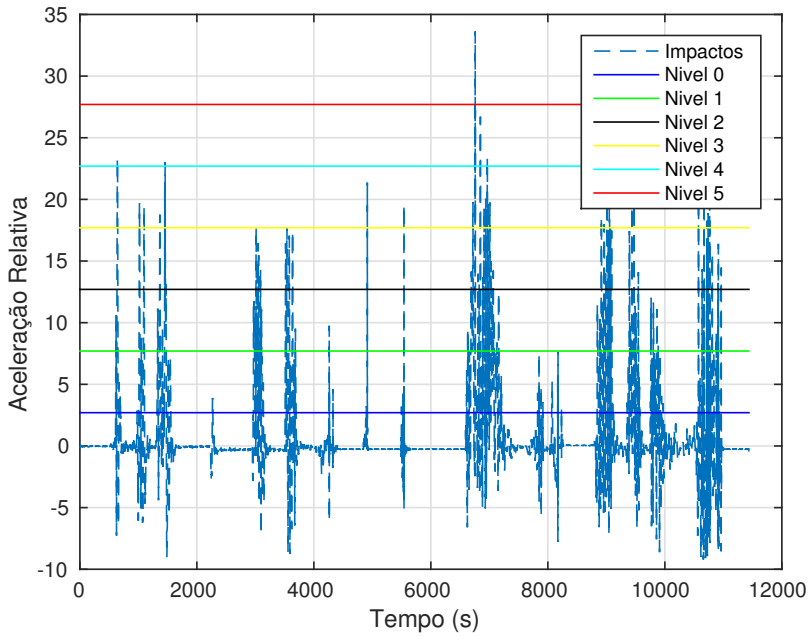
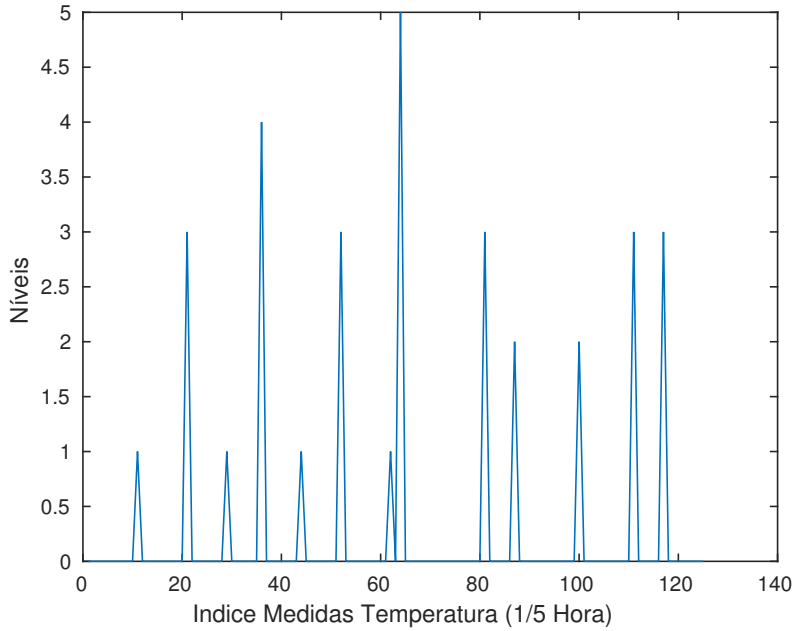


Figura 36 – Níveis simulados como entrada para a RNA.



choques mecânicos. Como os impactos físicos representam uma das maiores causas de falhas em elementos de uma RSSF, pela classificação desses choques, é possível que a rede neural MLP possa ter informações sobre o grau de integridade do equipamento após a colisão. Assim, a classificação das falhas dos dados pode ocorrer com maior precisão.

6.5 Experimento E_5

O experimento E_5 é dividido em duas partes a fim de verificar o funcionamento correto do modelo proposto. Na primeira parte, são analisados os resultados obtidos pelos dois níveis de detecção/diagnóstico da falha. São apresentados os resultados da detecção própria (realizada por cada nó da rede) e pela detecção da rede MLP (realizada pela estação base). Na segunda parte, estes dados são comparados com alguns modelos de gerenciamento existentes. Essa comparação visa verificar se as taxas encontradas por meio da abordagem proposta são similares as identificadas na literatura. Os dados de interesse para este experimento são as taxas encontradas para falso positivos e falsos negativos. A Tabela 6 apresenta um exemplo de um conjunto de dados utilizados como treinamento da RNA. Estes dados são referentes apenas a um nó específico da RSSF.

Tabela 6 – Exemplo de dados utilizados para o treinamento da rede MLP.

Entradas MLP		**Saída MLP**
Saída Filtro de Kalman	**Nível do Acelerômetro**	**Classificação**
0,099	0	0
2,241	4	1
0,605	1	0
1,170	2	1

No primeiro nível de detecção, como exposto na seção 4, cada nó da rede realiza um inferência estatística sobre as medidas realizadas por ele mesmo. Este nível inicial tem por finalidade reduzir a troca desnecessária de mensagens dentro da RSSF, pois não é necessário que todas as medidas coletadas sejam repassadas para tratamento na estação base. Apenas as medidas que forem julgadas como não pertencentes à distribuição normal dos dados devem ser endereçadas para a classificação no segundo nível de detecção. A Figura 37 apresenta a curva ROC para o primeiro nível de detecção.

A curva ROC é um gráfico de sensibilidade. No eixo das abscissas, encontra-se a taxa de falsos positivos (classificação incorreta) e no eixo das ordenadas estão representadas as taxas de positivos verdadeiros (classificação correta). A linha diagonal presente no gráfico representa uma classificação aleatória (área 0,5) e é utilizada como um modelo padrão de comparação para o teste realizado. Quanto maior a taxa de verdadeiros positivos maior será a área sob a curva e mais próximo do valor 1 (um) estará o modelo. A Figura 38 apresenta a matriz de confusão para a classificação própria. A matriz presente na Figura 38 mostra de maneira detalhada como a classificação das falhas é realizada.

Na Figura 38, as duas primeiras células da diagonal principal indicam o percentual correto de classificação das falhas. Como pode ser visto, de um lado temos que 40 medidas foram corretamente classificadas como falhas, que correspondem a 30,1% do total dos dados coletados e 88 medidas foram acertadamente classificadas como normais (66,2%

Figura 37 – Curva ROC para o primeiro nível de detecção/diagnóstico da falha.

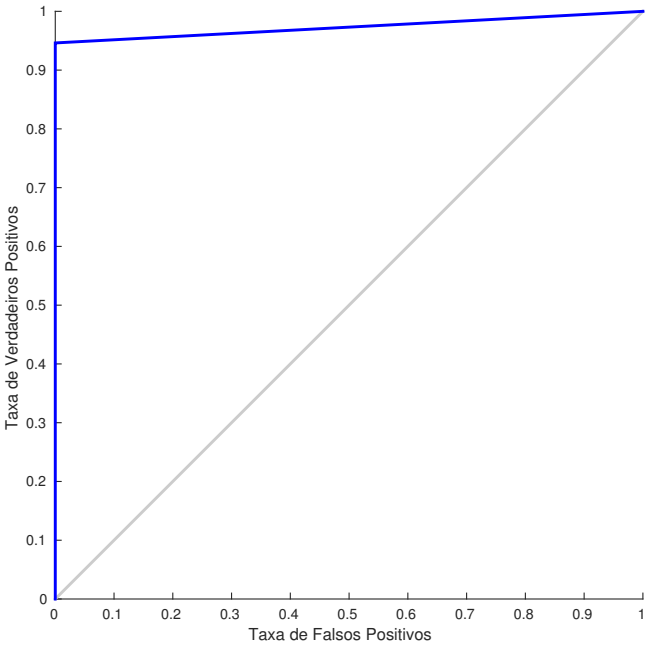
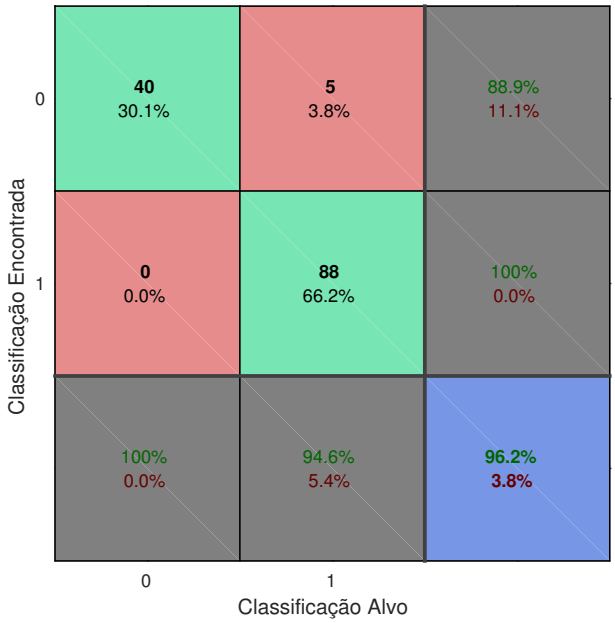


Figura 38 – Matriz de confusão para o nível próprio do SCIGF.

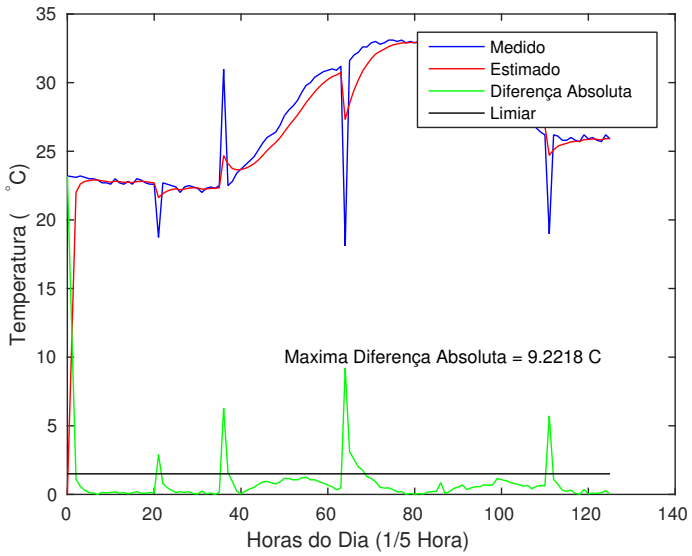


do total). Por outro lado, 5 (cinco) medidas foram erroneamente classificadas como falhas quando deveriam representar um funcionamento normal do sistema. Todas as medidas que representavam uma falha foram corretamente classificadas. Este é um resultado significativo para a solução do problema de gerenciamento de falhas, pois toda a decisão final de

classificação é realizada pela estação base. Uma vez que, a estacão base, é o elemento de rede com maior capacidade de processamento e possui o classificador neural, assim é fundamental que a decisão final de diagnostico de falha seja realizada pela estação base no segundo nível do SCIGF. Portanto, este primeiro nível de detecção reduz apenas a troca de mensagens dentro da rede, não mascarando possíveis falhas de dados do dispositivo. Como resultado final, resulta que 96,2% de toda a análise realizada pela detecção própria foi corretamente classificada.

No segundo nível de detecção de falhas, a rede MLP, utilizada como classificador, apresenta 2 (duas) entradas: 1) diferença absoluta entre o valor medido e o previsto (saída do filtro de Kalman) e 2) nível do acelerômetro. A Figura 39 apresenta os dados obtidos para o filtro de Kalman construído. A diferença absoluta entre os valores presentes no gráfico (linhas vermelha e azul) correspondem a uma das entradas da rede (linha verde).

Figura 39 – Filtro de Kalman: diferença entre os valores previstos e medidos.

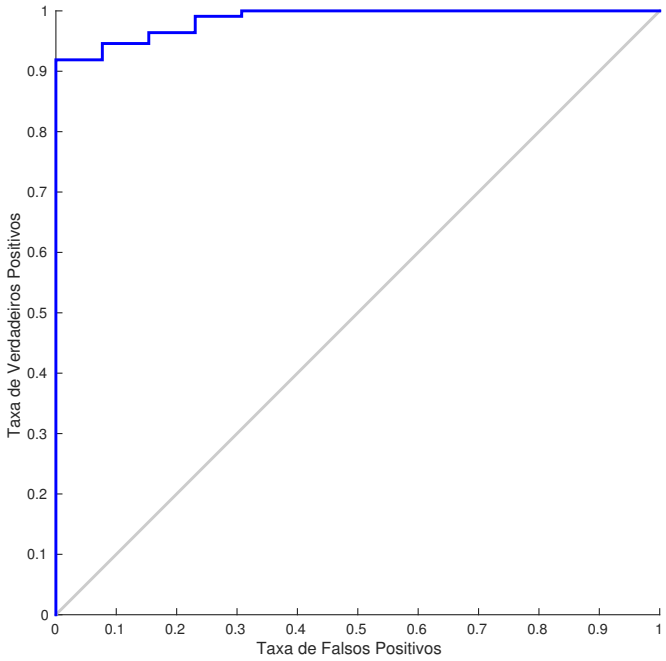


Uma característica do filtro de Kalman que o torna interessante para aplicação em RSSF é o fato da previsão ser realizada utilizando-se apenas a medida, imediatamente, anterior. Isso ocorre pois a matriz de covariância é responsável por armazenar a memória do sistema, assim não é necessário que todos os dados sejam armazenados, logo a utilização de memória pode ser reduzida. Como pode ser visto, comparando-se os gráficos presentes nas Figuras 36 e 39, as maiores diferenças entre os valores previstos e medidos ocorrem, normalmente, nos pontos em que o impacto sofrido apresentou maiores proporções. É possível verificar também, pela Figura 39, que o tempo de resposta para o filtro apresenta um valor reduzido.

Utilizando-se a classificação dos níveis de impacto sofrido pelo nó e a diferença absoluta entre os valores previstos e medidos pelo filtro de Kalman como entradas da rede MLP, encontra-se a curva ROC (Figura 40) e a matriz de confusão (Figura 41) para o

segundo nível de detecção. Como pode ser verificado pela Figura 40, os resultados estão próximos do canto superior esquerdo do gráfico, o que demonstra alta sensibilidade e menor proporção de falsos positivos. A matriz de confusão presente na Figura 41 corrobora com esta afirmação, pois informa que apenas 1 (um) evento normal foi classificado como falha (0,8%) e que 76,9% das falhas foram encontradas utilizando-se esta abordagem. Por estes dados, é possível inferir que o método proposto apresenta um desempenho total de 96,8% para a classificação correta de falhas de dados.

Figura 40 – Curva ROC para o segundo nível de classificação.



Sistemas de gerenciamento de falhas aplicados em RSSF são objetos de estudo há bastante tempo, seja por centros acadêmicos ou por industrias. Zhang, Dragoni e Wang (2015) fazem uma compilação sobre as principais abordagens existentes na literatura, identificando o desempenho de cada uma delas. Assim, o comportamento destes sistemas na classificação de possíveis falhas pode ser comparado. A Tabela 7 sumariza estes dados.

Comparando-se os resultados obtidos com algumas abordagens sumarizadas pela Tabela 7, verifica-se que o modelo SCIGF apresenta um desempenho similar aos relatados na literatura. Alguns modelos, como apresentados na Tabela 7, não expõem resultados referentes ao consumo de energia. Assim, questões relativas à aplicabilidade destas abordagens não são investigadas. O próximo experimento investiga o consumo de energia do modelo proposto.

Figura 41 – Matriz de confusão para o segundo nível de classificação.

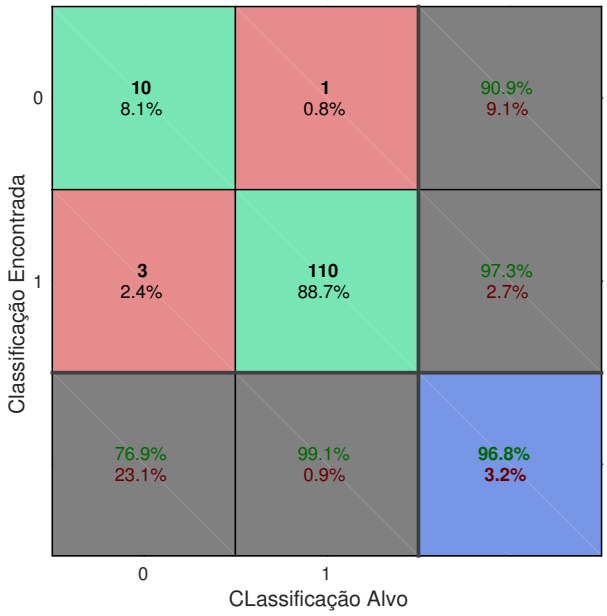


Tabela 7 – Precisão na detecção de falhas: comparação entre algumas abordagens.

Trabalho	**Ano**	**Precisão de Detecção**
Nie et. Al. (NIE; MA; MO, 2012)	2012	>0.90
Ma et. Al. (MA et al., 2012)	2012	>0.86
Kamal et Al. (KAMAL; BLEAKLEY; DOBSON, 2013)	2013	0.88 - 0.98
SCIGF	2017	$\sim$**0.97**

Fonte: Adaptado de (ZHANG; DRAGONI; WANG, 2015).

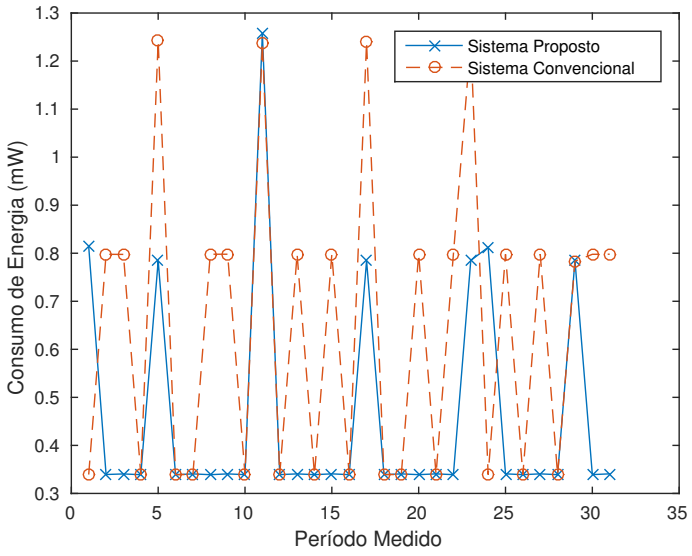
6.6 Experimento E_6

Uma das principais características das RSSF é a existência de suprimento reduzido de energia. Portanto, é imprescindível que qualquer sistema utilizado para detecção de falhas seja construído levando-se em consideração este requisito a fim de prolongar o tempo de vida de rede. A finalidade deste experimento é identificar se o modelo de gerenciamento centralizado de falhas proposto apresenta consumo de energia inferior às abordagens centralizadas usuais. Como exposto em seções anteriores, tratamentos centralizados, normalmente, necessitam de um maior número de mensagens para detectar uma falha, visto que a estação base necessita ser informada, a todo momento, sobre o funcionamento de cada nó da rede.

Para este experimento, foi implementada uma abordagem centralizada típica: cada

um dos nós da rede necessita enviar regularmente uma mensagem para a estação base, informando o valor da media realizada. Após receber esta mensagem, a estação base inicia o processo de classificação do dado (falha ou não) a fim de identificar possíveis falhas. Em uma RSSF, como exposto anteriormente, os processos que consomem maior energia são as sequências de troca de mensagens. Assim, para facilitar a interpretação gráfica, apenas o processo de transmissão será apresentado. A Figura 42 mostra a comparação entre a abordagem centralizada típica, sem o primeiro nível de detecção de falhas, e o SCIGF, com o nível de detecção estatístico. Os dados de consumo total (CPU, Transmissor, Receptor e LPM) apresentados são de apenas um dos nós da rede.

Figura 42 – Comparação entre o consumo de energia do SCIGF e uma abordagem centralizada típica.

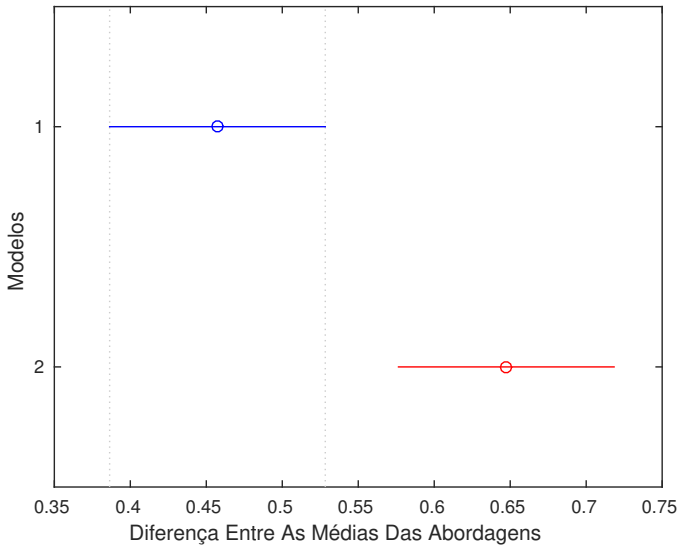


Pela análise dos gráficos apresentados na Figura 42, foi identificado o consumo no intervalo de 310,0s de 140,5mWH e 211,0mWH, respectivamente, para o modelo proposto e uma abordagem típica. Como podem existir diferenças entre cada uma destas medidas (processos executados em segundo plano e eventos aleatórios), é necessária uma análise estatística para a comparação entre os modelos. Após várias execuções em intervalos aleatórios, pode-se utilizar o teste de Tukey para confrontar o consumo de cada abordagem. Com intervalos de confiança de 95%, a Figura 43 apresenta os resultados encontrados pelo teste de Tukey. É possível verificar que a diferença média entre o consumo do SCIGF e de uma abordagem típica para gerenciamento de falhas centralizado chega a ser de 37,0%.

Neste cenário especifico de gerenciamento de falhas, com a utilização de dados de temperatura para simulação, constata-se que a utilização do primeiro nível de detecção reduz, consideravelmente, o consumo de energia em modelos centralizados. Assim, é possível concluir que pode-se empregar este primeiro nível de detecção de falhas a outros modelos de gerenciamento centralizado, pois, além de reduzir o consumo de energia ele

possibilita que a classificação final da falha seja realizada pela estação base. Portanto, não irá influenciar nas decisões de diagnóstico das falhas dos dados.

Figura 43 – Teste de Tukey para a comparação entre os gastos de energia do SCIGF e uma abordagem típica de gerenciamento centralizado de falhas.



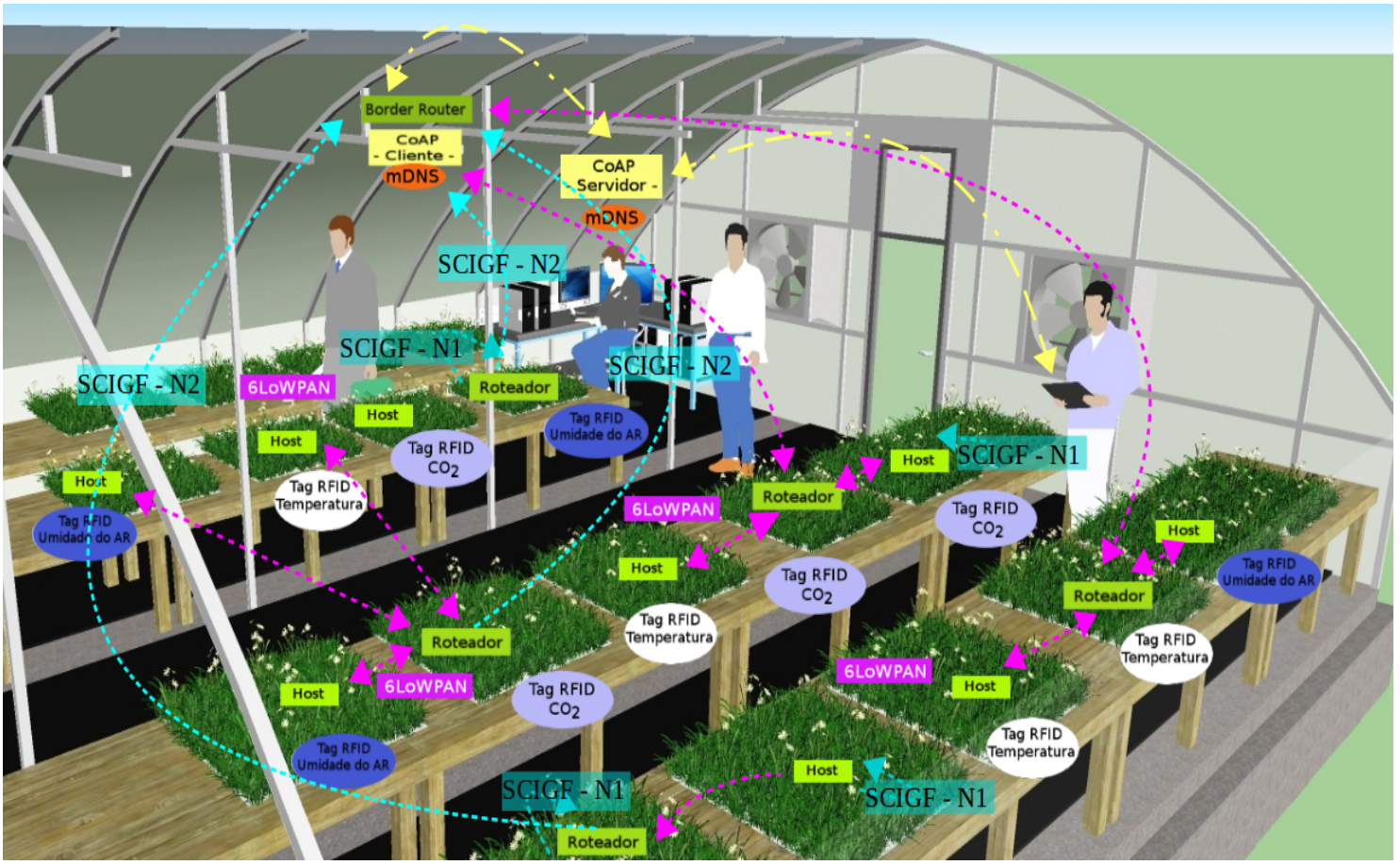
6.7 Estudo de Caso

Este estudo de caso corresponde à apresentação de um exemplo de aplicação que visa mostrar como o sistema de gerenciamento de falhas proposto é utilizado em uma arquitetura de IoT. Seguindo o modelo genérico para uma arquitetura IoT expresso por Khan et al. (2012), são definidos alguns dos protocolos empregados para integração do sistema com a Internet e exemplificado o funcionamento geral de uma rede IoT. Também são simulados 3 (três) cenários de utilização do modelo sugerido para gerenciamento de falhas. A arquitetura seguida apresenta 5 (cinco) níveis de abstração (camadas), pois, como demonstrado pelo trabalho de Al-Fuqaha et al. (2015), esta abordagem possui vantagens quando comparada a um modelo constituído por um número menor de níveis. Isso ocorre pois o nível de abstração expresso em uma abordagem com 5 (cinco) níveis possibilita a definição e a interpretação de maneira mais natural dos serviços disponibilizados e consumidos em cada camada. Além disso, o modelo presente na Figura 4b propicia, com maior clareza, a definição dos protocolos utilizados.

A aplicação que representa a utilização e a integração do modelo com a IoT consiste em um sistema de monitoramento da produção de mudas em casas de vegetação. As casas de vegetação são viveiros permanentes compostos por estruturas metálicas (ou construídas em madeira) cobertas com plásticos, objetivando fornecer um ambiente climático interno controlado. Estas instalações são comumente utilizadas para a produção controlada de

mudas (TERUEL, 2010). Junior, Araújo e Guimarães (2015) realizam um estudo sobre a automação do controle e a utilização da robótica nestes viveiros. Teruel (2010) destaca a importância do controle de algumas variáveis no processo de crescimento das plantas em casas de vegetação. Dentre as variáveis preponderantes para um desenvolvimento sadio das mudas, destacam-se a temperatura (influencia diretamente no metabolismo das plantas), a umidade relativa do ar (influencia na transpiração e controle térmico das plantas) e a concentração de CO_2 (crescimento das plantas), uma vez que o controle destas variáveis é primordial para o desenvolvimento correto das mudas. Assim, com a utilização da IoT, é possível monitorar o microclima existente no cultivo e assegurar um controle adequado de toda a produção das mudas. A Figura 44 apresenta a integração dos diferentes protocolos para a concepção da arquitetura proposta, destacando-se o papel do gerenciamento de falhas neste cenário. Nesta figura, são apresentados alguns dos protocolos mais utilizados e a interação com o SCIGF. SCIGF-N1, corresponde ao sistema centralizado inteligente de gerenciamento de falhas no nível 1. Já o SCIGF-N2, representa o nível 2 do sistema centralizado inteligente de gerenciamento de falhas.

Figura 44 – Representação da aplicação de uma arquitetura IoT no monitoramento e controle de mudas em casas de vegetação.



O protocolo sugerido para o primeiro nível da arquitetura (percepção) deve ser definido a fim de possibilitar uma identificação única dos dispositivos sensores. Neste

sentido, o protocolo RFID apresenta grandes vantagens de utilização, pois proporciona uma identificação de baixo custo para os elementos constituintes da rede. Para a camada de rede, é utilizada a comunicação sem fio por meio do protocolo 6LoWPAN, já que favorece a integração completa da rede com a Internet. Isso ocorre porque, a cada um dos nós, é atribuída uma identificação única para recebimento de mensagens via IPv6 e todo o roteamento interno à rede LoWPAN é realizado por meio do protocolo RPL. O sistema de gerenciamento de falhas proposto está localizado entre as camadas de percepção e rede, já que consome os dados provenientes da camada de percepção e utiliza-se dos protocolos de rede para a comunicação interna à LoWPAN. Quando o sensoriamento de uma variável (temperatura, umidade etc) é realizado, o nó sensor analisa os dados no primeiro nível de detecção de falhas (detecção própria). Após identificar algum possível erro, utilizando-se do protocolo 6LoWPAN, são enviadas as mensagens para a estação base (*edge router*) a fim de ser realizada a segunda etapa de detecção de falhas. A camada de gerenciamento de serviço é composta pelo protocolo mDNS. Assim, torna-se possível identificar cada uma das funções e a capacidade dos nós internos da rede. Quando um serviço é solicitado, pode ser prontamente atendido. O procolo CoAP é utilizado na camada de aplicação. O CoAP emprega o diálogo entre cliente-servidor do tipo um-para-um. Assim, a comunicação ocorre diretamente entre estes elementos sem a necessidade de distribuição de mensagens entre os clientes (como ocorre no protocolo MQTT). Outra característica do CoAP é o fato de ter sido concebido a fim de possibilitar a interconexão dos dispositivos de baixo processamento com a Internet, pois são utilizados pacotes de tamanho reduzido e cada pacote é transmitido no espaço de memória RAM que foi utilizado pelo pacote anterior (trocas *in place*). Para exemplificar a integração do sistema de gerenciamento de falhas proposto com a camada de aplicação da IoT são implementados 3 (três) cenários de interação entre o sistema e o usuário final:

1. Uma falha de *hardware* foi detectada em um nó da RSSF;
2. O número de nós da rede detectados com falhas corresponde a mais de 20% da rede;
3. Um mesmo nó é levado mais de 2 (duas) vezes para análise em quarentena.

O SCIGF pode utilizar o protocolo RFID a fim de atribuir, de modo automático, um identificador único para cada sensor. Ao atribuir esta identificação, é possível que sejam definidas classes de sensores, por exemplo, separar todos os elementos responsáveis por medir a temperatura, umidade ou concentração de CO_2, assim fica mais fácil o administrador do sistema realizar a manutenção dos equipamentos. O SCIGF utiliza o protocolo 6LoWPAN, da camada de rede, para realizar a troca de mensagens entre os nós sensores e a estação base, além de realizar a recuperação da falha por meio da exclusão do nó da whiteboard. Como são utilizados protocolos das camadas de percepção e rede é possível dizer que este é implementado como uma subcamada da camada de rede.

A Figura 45 apresenta as mensagens recebidas pelo usuário final, por meio da Inter-

net, a fim de informar sobre o funcionamento do sistema. Como o controle da temperatura em casas de vegetação representa um fator primordial para o cultivo adequado de mudas, qualquer variação da capacidade de monitoramento da RSSF deve ser alertada. Quando os dados lidos apresentam falhas, todo o cultivo pode ser prejudicado, inviabilizando-se a produção. Para a camada de negócio, podem ser implementados métodos de mineração de dados e geração de conhecimento a fim de auxiliar no melhor manejo da produção, como: análise do crescimento das mudas em função dos recursos disponibilizados, exame da qualidade dos insumos utilizados ou pesquisa da influência da variação climática externa no desenvolvimento vegetal. Por meio do monitoramento constante e da análise dos dados, é possível aumentar a produtividade e economizar recursos.

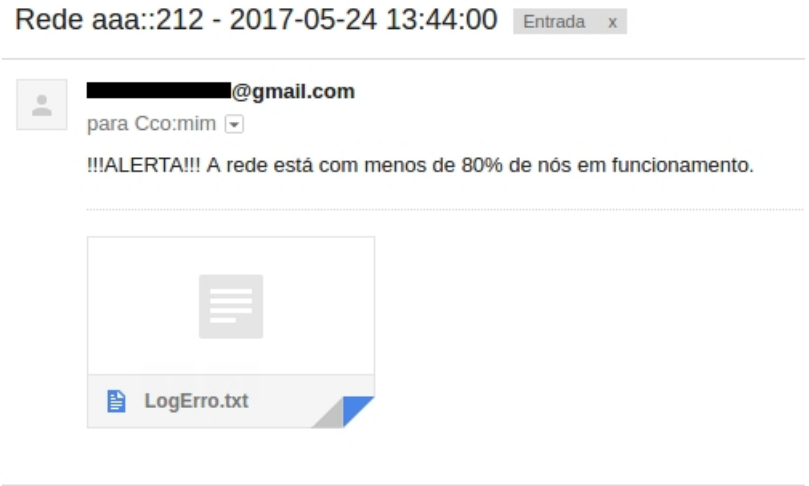
O exemplo de aplicação do modelo de gerenciamento de falhas em um sistema real mostra que os protocolos e serviços utilizados, atualmente, para a concepção da IoT podem ser inteiramente interligados com o gerenciamento de falhas proposto. Por meio desta aplicação, é possível identificar a importância de um gerenciamento eficiente de falhas e constatar que a abordagem sugerida utiliza-se das mais recentes tecnologias para possibilitar a integração eficiente entre uma RSSF e a IoT.

Figura 45 – Modelo de interação com usuário através de *e-mail*: 45a.

(a) Impacto nível 5 (cinco) sofrido.

(b) Nó constantemente em quarentena.

(c) Rede com 80% de nós em funcionamento.



7 Conclusão

Este capítulo revê e resume a pesquisa apresentada, identificando os principais métodos empregados e discute suas implicações na concepção da IoT. Além disso, são abordados os desdobramentos futuros que visam aprofundar e ampliar a aplicação do gerenciamento de falhas proposto.

7.1 Discussões Finais

A concepção da IoT emprega o conceito: *Anywhere, Anytime and with Anything* (em todos os lugares, a todo momento e com todas as coisas). Assim, deve existir uma interação entre todos os objetos do mundo físico por meio da comunicação sem fio e sistemas embarcados. Pesquisas apontam que, em 2020, existirão mais de 20 bilhões de dispositivos conectados à Internet, realizando as mais variadas tarefas (controle, monitoramento, comunicação etc). Grande parte deste dispositivos fará parte da Internet das Coisas. Mas, conectar cada um destes elementos requer uma enorme quantidade de dispositivos sensores com características próprias de capacidade de processamento, comunicação, disponibilidade de energia e tamanho reduzido. Normalmente, estes dispositivos, possuem uma capacidade reduzida de processamento e apresentam fontes com exígua disponibilidade de energia. Estas características impõem restrições aos modelos de gerenciamento de falhas, pois os sistemas devem ser capazes de manter a baixa taxa de falsos positivos e alto índice de precisão sem que, para isso, consumam grande quantidade de energia ou necessitem de alto poder de processamento. Devido a estas características de concepção, estes elementos com recursos limitados estão sujeitos a maiores probabilidades de falhas.

Neste contexto, as falhas de *hardware* apresentam-se como uma das principais ocorrências que geram erros ao sistema. Quando uma falha de *hardware* não é prontamente identificada e ações de contorno não são realizadas em tempo hábil, estas falhas podem acarretar medidas errôneas sobre o funcionamento de um sistema e, assim, subsidiar decisões equivocadas. Por exemplo, no monitoramento de pacientes, a partir de uma falha de *hardware*, o médico responsável pode ser informado sobre um comportamento anormal dos sinais vitais do paciente e, assim, em casos extremos, tomar decisões que não são adequadas para a recuperação do empalamado. Logo, detectar, diagnosticar e corrigir dinamicamente esta falha é fundamental para o funcionamento correto do sistema. Devido às características de concepção, grande parte dos elementos que integram a IoT estão mais suscetíveis a falhas. Como as falhas de *hardware* apresentam uma das principais causas de erros de dados em RSSF, um sistema de gerenciamento capaz de detectar e, prontamente, mitigar esse erro é de fundamental para a concepção da Internet das Coisas.

Dentre os modelos de gerenciamento de falhas propostos, destacam-se as abordagens centralizadas e descentralizadas. Uma abordagem centralizada possui como maior vantagem a característica inerente de menor complexidade computacional, pois a maior carga de trabalho na detecção e correção da falha é realizada pela estação base. Entretanto, esta abordagem é energeticamente ineficaz, pois a estação base deve receber mensagens de todos os nós da rede a fim de monitorar o real estado dos dispositivos. Diferentemente deste modelo, o tratamento descentralizado apresenta maior eficiência energética mas com um maior custo computacional. Portanto, conceber um sistema que possa se beneficiar das principais características destas duas diferentes abordagens (concepção e baixo consumo) é um desafio para a integração das RSSF com a Internet das Coisas.

Neste contexto, é sugerido um modelo de gerenciamento centralizado de falhas com redução das mensagens trocadas dentro da RSSF, denominado sistema centralizado inteligente de gerenciamento de falhas (SCIGF). A atenuação da tráfego da rede decorre do fato de os processos de transmissão de recepção de mensagens serem responsáveis pelo consumo da maior quantidade de energia em um nó. O modelo SCIGF tem por objetivo manter as características de reduzida complexidade computacional de uma abordagem centralizada em união com a maior eficiência energética de um tratamento descentralizado mantendo taxas de falsos positivos e negativos equivalentes. Os experimentos realizados comprovam que o sistema proposto alcançou estes objetivos desejados. O modelo SCIGF é composto por 2 (dois) níveis de detecção/diagnóstico da falha (nível de detecção próprio e nível de detecção na estação base). O primeiro nível, chamado de detecção própria, utiliza-se da inferência estatística para informar uma possível falha e reduzir a necessidade da troca de mensagens dentro da rede. No segundo nível, as medidas supostamente faltosas são analisadas pela estação base. O diagnóstico desta falha ocorre por meio de uma rede MLP que possui como entradas a saída de um filtro de Kalman (diferença entre a previsão da temperatura e o seu valor real) e a intensidade de impacto sofrido pelo nó (níveis de variação do acelerômetro). São realizados 6 (seis) experimentos e realizado um estudo de caso a fim de demonstrar a eficácia do método exposto. O estudo de caso apresenta a aplicação da IoT no auxílio à produção de mudas em casas de vegetação, indicando como a adoção do modelo de gerenciamento de falhas possibilita a integração eficiente e sensível a falhas dos dispositivos sensores com a Internet.

Os experimentos mostraram que o SCIGF utiliza, dentre os classificadores pesquisados (MLP, RBF e ANFIS), a RNA de menor consumo energético, para a solução de problemas simples de classificação, e com reduzida complexidade. Os resultados também mostraram que o nível de detecção própria apresentou uma eficiência de 96,2%, sendo que todas as possíveis falhas foram enviadas para análise na estação base. O segundo nível apresentou precisão total de 96,8%, evidenciando que os resultados relativos à classificação das falhas mantém níveis de precisão e acurácia compatíveis com outras abordagens presentes na literatura. A adoção do primeiro nível de detecção de falhas apresentou

uma redução de mais de 35,0% do consumo de energia de uma abordagem centralizada típica. Todos os resultados foram comprovados por métodos estatísticos de análise de experimentos.

O modelo proposto apresenta taxas de falsos positivos e negativos similares às abordagens apresentadas na literatura. Ao testar as hipótese, verifica-se que o a primeira hipótese pode ser refutada, uma vez que o SCIGF apresentou uma precisão de aproximadamente 97,0%. A hipótese referente ao consumo de energia não pode ser refutada, pois a utilização do primeiro nível de detecção ocasionou a redução do consumo de energia, em média, de 40,0%.

Os resultado alcançados pelo SCIGF, mostram que este apresenta características de um sistema autonômico, já que é possível detectar, diagnosticar e recuperar de falhas relativas aos dados. Assim, o gerenciamento eficiente de uma RSSF, integrada à Internet, pode ser realizada de maneira a não comprometer o desempenho de uma aplicação já existente. Os resultados obtidos pelos experimentos 3 (três), 4 (quatro) e 5 (cinco) atestam que o gerenciamento de falhas, realizado pelo SCIGF, apresenta desempenho similar às abordagens com as melhores taxas de classificação de falhas.

Ao analisar o experimento 6 (seis), verifica-se que o consumo de energia do SCIGF, para o cenário específico simulado, é inferior a um modelo centralizado convencional. Como não foi implementado um sistema de gerenciamento de falhas descentralizado e nos trabalhos pesquisados não existem resultados sobre o gasto energético, não foi possível comparar o consumo do SCIGF com uma abordagem descentralizada. Assim, uma comparação mais precisa do desempenho energético do SCIGF com os modelos de melhor eficiência energética não pôde ser realizada.

Como a aplicação, retratada no estudo de caso, não foi implementada em um ambiente real, não é possível quantificar a redução dos custos de manutenção e consumo desnecessário de recursos. Entretanto, para a introdução do SCIGF em uma aplicação já existente é necessário apenas que a camada de rede esteja implementada por meio do 6LoWPAN, sem a necessidade de *hardware* adicional. Portanto, é possível inferir que os custos de operação não serão alterados e os gastos devido a medidas equivocadas devem ser reduzidos drasticamente. Logo, a adoção do SCIGF, auxilia no melhor aproveitamento de recursos.

Uma questão a ser discutida, corresponde a escalabilidade do SCIGF. Como o SCIGF utiliza a criação de duas RNA MLP para cada um dos nos da aplicação, em cenários nos quais são utilizados muitos nós, pode existir o problema de *overhead*. Assim, para reduzir os efeitos deste excesso de armazenamento, é possível que a estação base agrupe os sensores que realizam medidas similares em uma mesma RNA. Por exemplo, no estudo de caso apresentado, vamos supor que existam 80 nós na aplicação. Assim, seria necessária a criação de 160 RNA MLP. Agrupando-se a RSSF em elementos que realizam

medidas de temperatura, nós que medem CO2 e umidade, por exemplo, a quantidade de redes criadas seria reduzida drasticamente. Esta abordagem é diferente de um sistema descentralizado, pois não existe a necessidade de criação de *cluster head*. Já que ocorre apenas o agrupamento de sensores com objetivos similares, a fim de possibilitar uma economia de armazenamento de dados.

Não foram utilizados outros níveis de detecção, pois, em um sistema centralizado, existem, basicamente, dois tipos diferentes de elementos (nós sensores e estação base). Assim, atribuir novos níveis para a detecção de falhas torna-se inviável, uma vez que, ao acrescentar novas etapas de classificação, a complexidade do sistema tende a ser elevada. Como estamos interessados em manter a complexidade do SCIGF a mais baixa possível, adotar as características de autonomia são essenciais.

O SCIGF pode ser visto como um sistema autonômico, pois apresenta características de autogerenciamento, autoproteção, autorecuperação e otimização. O autogerenciamento é realizado pela estação base, uma vez que por meio do protocolo 6LoWPAN, que implementa o RPL, é realizada a configuração inicial e administração da rede. Novos nós podem ser inseridos ou retirados sem a necessidade de reconfigurar manualmente o sistema. A autoproteção e autorecuperacão são realizadas inteiramente pelo SCIGF, uma vez que as falhas de dados são identificadas e o elemento que é marcado como faltoso é excluído da rede, evitando que novas medidas erradas sejam inseridas. A otimização do gerenciamento da rede ocorre a medida que o processo de monitoramento, por meio do SCIGF, visa manter os níveis de precisão das medidas o mais próximo possível do projeto inicial. Isso ocorre pois, através da identificação e exclusão das falhas, o SCIGF, de maneira automática, analisa os nós, a fim de manter a confiabilidade inicial da rede de monitoramento.

7.2 Contribuições

As principais contribuições deste trabalho podem ser identificadas no meio acadêmico, tecnológico e social. Para o meio acadêmico este trabalho coloca novamente em foco os modelos centralizados de gerenciamento de falhas, uma vez que, normalmente, este tipo de abordagem é visto como inferior aos modelos descentralizada, pois apresenta maior consumo de energia. Por meio da adoção de um nível inferior é possível reduzir a necessidade da troca de mensagens dentro da rede e, consequentemente, diminuir o gasto de energia. Outra contribuição refere-se a utilização de um método para capturar a dinâmica do sistema (filtro de Kalman) sem a necessidade de armazenamento de todos os dados históricos da rede. Neste contexto, a publicação do artigo Vieira, Almeida e Meireles (2017), sugere que os resultados obtidos são relevantes para o desenvolvimento deste modelo sugerido.

No âmbito tecnológico, a adoção de um modelo de gerenciamento de falhas que

utiliza-se dos mais modernos protocolos (6LoWPAN) para integração de sensores à Internet das Coisas possibilita o desenvolvimento de aplicações de monitoramento e controle em diferentes áreas. Além disso, o modelo proposto pode ser inserido em aplicações já existentes sem interferir, dramaticamente, nos no consumo de energia do sistema.

Para a sociedade, a maior contribuição deste trabalho está no fato de um sistema confiável de gerenciamento de falhas poder subsidiar de maneira mais assertiva as decisões realizadas por meio da análise de dados dos sensores. Assim, decisões baseadas em aplicações que afetam o usuário podem ser tomadas com um maior nível de confiabilidade.

7.3 Propostas de Continuidade

Como possíveis desdobramentos deste trabalho, deve-se investigar modelos de pré-processamento dos dados no primeiro nível de detecção, a fim de ser possível alcançar uma redução de consumo superior a 50,0%. Com esta redução de consumo espera-se que o tempo de vida de uma RSSF não sofra grandes impactos com a adoção de um sistema centralizado de gerenciamento de falhas e, portanto, seja possível emprega-lo em diferentes aplicações.

Outro proposta de continuidade é averiguar os impactos da utilização de outros classificadores neurais ou outros algoritmos classificadores citados na literatura como alternativas para a classificação de falhas. A análise de outros métodos tem por objetivo verificar se a relação de compromisso entre desempenho e consumo de recursos pode ser encontrada com a utilização de outros classificadores. É de fundamental importância que estas novas abordagens sejam implementadas em nós reais de um RSSF, pois devem ser empregadas em sistemas com baixa disponibilidade de recursos e com características particulares de uma RSSF.

A implementação do SCIGF em aplicações com diferentes características também deve ser investigada, com o propósito de avaliar a sua versatilidade. Uma possível aplicação para o SCIGF é o monitoramento de pacientes, pois, neste tipo de aplicação, as falhas de *hardware* são substancialmente prejudiciais, já que podem subsidiar decisões equivocadas e acarretarem um diagnóstico incorreto a cerca da saúde do paciente. Esta investigação é necessária, pois deve-se comprovar a aplicabilidade do modelo em cenários distintos, uma vez que a heterogeneidade de dispositivos, protocolos e serviços é inerente ao contexto de IoT.

Outra investigação a ser conduzida, corresponde a aplicação do SCIGF em cenários que demandam um tratamento de falhas em tempo real. Explorar este panorama é de fundamental importância, pois em aplicações como o controle de processos, as restrições de tempo são primordiais para um bom desempenho do sistema. Uma vez que, o SCIGF é implementado localmente, é possível que os resultados alcançados sejam relevantes para

o diagnóstico e correção de falhas em sistemas de tempo real.

Referências

AKCAYOL, M. A. Application of adaptive neuro-fuzzy controller for srm. **Advances in Engineering software**, Elsevier, v. 35, n. 3, p. 129–137, 2004. Citado na página 36.

AL-FUQAHA, A. et al. Internet of things: A survey on enabling technologies, protocols, and applications. **IEEE Communications Surveys & Tutorials**, IEEE, v. 17, n. 4, p. 2347–2376, 2015. Citado na página 73.

ALAM, S.; DE, D.; RAY, A. Analysis of energy consumption for iarp, rip and star routing protocols in wireless sensor networks. In: IEEE. **Advances in Computing and Communication Engineering (ICACCE), 2015 Second International Conference on**. [S.l.], 2015. p. 11–16. Citado na página 5.

ALCARAZ, C. et al. Wireless sensor networks and the internet of things: Do we need a complete integration? In: **1st International Workshop on the Security of the Internet of Things (SecIoT '10)**. [S.l.: s.n.], 2010. Citado 2 vezes nas páginas 4 e 14.

ALLIANCE, Z. et al. **Zigbee specification**. 2013. Citado na página 11.

ALMEIDA, F. M. de; RIBEIRO, A. d. R. L.; MORENO, E. D. An architecture for self-healing in internet of things. **UBICOMM 2015**, p. 89, 2015. Citado na página 42.

AZHARUDDIN, M.; JANA, P. K. A distributed algorithm for energy efficient and fault tolerant routing in wireless sensor networks. **Wireless Networks**, Springer, v. 21, n. 1, p. 251–267, 2015. Citado 2 vezes nas páginas 41 e 42.

BABU, L.; MASHESWARI, N. Energy efficient error detection in data aggregation in wsn. **International Journal of Innovative Research in Science, Engineering and Technology**, v. 3, 2014. Citado na página 39.

BAGCI, H.; KORPEOGLU, I.; YAZICI, A. A distributed fault-tolerant topology control algorithm for heterogeneous wireless sensor networks. **IEEE transactions on parallel and distributed systems**, IEEE, v. 26, n. 4, p. 914–923, 2015. Citado na página 42.

BALBINOT, A.; BRUSAMARELLO, V. J. **Instrumentação E Fundamentos de Medidas. Volume 2**. [S.l.]: Grupo Gen-LTC, 2000. Citado na página 3.

BARANI, F. A hybrid approach for dynamic intrusion detection in ad hoc networks using genetic algorithm and artificial immune system. In: **Intelligent Systems (ICIS), 2014 Iranian Conference on**. [S.l.: s.n.], 2014. p. 1–6. Citado na página 43.

BEHZADI, S.; AZAD, M. Fault-tolerant in wireless sensor networks using fuzzy logic. **International Research Journal of Applied and Basic Sciences**, v. 8, n. 9, p. 1276–1282, 2014. Citado 2 vezes nas páginas 4 e 39.

BHUIYAN, M. Z. A. et al. Deploying wireless sensor networks with fault-tolerance for structural health monitoring. **IEEE Transactions on Computers**, IEEE, v. 64, n. 2, p. 382–395, 2015. Citado na página 42.

BOKAREVA, T.; BULUSU, N.; JHA, S. An immunology inspired approach to robust sensor networking. 2006. Citado 2 vezes nas páginas 4 e 41.

BOONSONGSRIKUL, A.; KOCIJANCIC, S.; SUPPHARANGSAN, S. Effective energy consumption on wireless sensor networks: Survey and challenges. In: IEEE. **Information & Communication Technology Electronics & Microelectronics (MIPRO), 2013 36th International Convention on**. [S.l.], 2013. p. 469–473. Citado na página 5.

BREIVOLD, H. P.; SANDSTR, K. et al. Internet of things for industrial automation–challenges and technical solutions. In: IEEE. **2015 IEEE International Conference on Data Science and Data Intensive Systems**. [S.l.], 2015. p. 532–539. Citado na página 12.

BROCK, D. L. The electronic product code (epc). **Auto-ID Center White Paper MIT-AUTOID-WH-002**, 2001. Citado na página 11.

BRÖRING, A. et al. New generation sensor web enablement. **Sensors**, Molecular Diversity Preservation International, v. 11, n. 3, p. 2652–2699, 2011. Citado na página 1.

CHEN, D. et al. Natural disaster monitoring with wireless sensor networks: a case study of data-intensive applications upon low-cost scalable systems. **Mobile Networks and Applications**, Springer, v. 18, n. 5, p. 651–663, 2013. Citado na página 1.

CHENG, X. hui; WANG, Y. The remote monitoring system of transformer fault based on the internet of things. In: **Computer Science and Network Technology (ICCSNT), 2011 International Conference on**. [S.l.: s.n.], 2011. v. 1, p. 84–87. Citado na página 44.

CHERIFI, N. et al. Energy consumption of networked embedded systems. In: IEEE. **Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on**. [S.l.], 2015. p. 639–644. Citado na página 55.

CHRISTIN, D. et al. Wireless sensor networks and the internet of things: selected challenges. **Proceedings of the 8th GI/ITG KuVS Fachgespräch Drahtlose Sensornetze**, p. 31–34, 2009. Citado 2 vezes nas páginas 14 e 15.

CHUDZIKIEWICZ, J.; FURTAK, J.; ZIELINSKI, Z. Fault-tolerant techniques for the internet of military things. In: IEEE. **Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on**. [S.l.], 2015. p. 496–501. Citado na página 40.

CHUI, M.; LÖFFLER, M.; ROBERTS, R. The internet of things. **McKinsey Quarterly**, v. 2, n. 2010, p. 1–9, 2010. Citado 2 vezes nas páginas 4 e 16.

CLIMATEMPO. **Previsao do tempo**. 2017. <https://www.climatempo.com.br/previsao-do-tempo/cidade>. Citado na página 63.

COMPUTING, A. et al. An architectural blueprint for autonomic computing. **IBM White Paper**, v. 31, 2006. Citado 2 vezes nas páginas 3 e 5.

DÂMASO, A. et al. Evaluating the power consumption of wireless sensor network applications using models. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 13, n. 3, p. 3473–3500, 2013. Citado na página 40.

DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: ACM. **Proceedings of the 23rd international conference on Machine learning**. [S.l.], 2006. p. 233–240. Citado na página 55.

DOGAN, G.; AVINCAN, K. Multiprotru: A kalman filtering based trust architecture for two-hop wireless sensor networks. **Peer-to-Peer Networking and Applications**, Springer, v. 10, n. 1, p. 278–291, 2017. Citado na página 43.

DUNKELS, A. et al. **Powertrace: Network-level power profiling for low-power wireless networks**. [S.l.]: Swedish Institute of Computer Science, 2011. Citado na página 55.

DUNKELS, A.; GRONVALL, B.; VOIGT, T. Contiki-a lightweight and flexible operating system for tiny networked sensors. In: IEEE. **Local Computer Networks, 2004. 29th Annual IEEE International Conference on**. [S.l.], 2004. p. 455–462. Citado na página 37.

ELSHAIKH, M. et al. Energy consumption optimization with ichi taguchi method for wireless sensor networks. In: IEEE. **Electronic Design (ICED), 2014 2nd International Conference on**. [S.l.], 2014. p. 493–498. Citado na página 5.

EVANS, D. The internet of things: How the next evolution of the internet is changing everything. **CISCO white paper**, v. 1, n. 2011, p. 1–11, 2011. Citado na página 5.

FAHMY, H. M. A. **Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis**. [S.l.]: Springer, 2016. Citado 2 vezes nas páginas 9 e 11.

FAROOQ, M. O.; KUNZ, T. Operating systems for wireless sensor networks: A survey. **Sensors**, Molecular Diversity Preservation International, v. 11, n. 6, p. 5900–5930, 2011. Citado na página 37.

FERREIRA, A. d. H. **Novo dicionário Aurélio da língua portuguesa**. [S.l.]: Editora Positivo, 2004. Citado na página 10.

GAYNOR, M. et al. Integrating wireless sensor networks with the grid. **IEEE Internet Computing**, IEEE, v. 8, n. 4, p. 32–39, 2004. Citado na página 4.

GIL, A. C. Como elaborar projetos de pesquisa. **São Paulo**, v. 5, p. 61, 2002. Citado na página 54.

GOUDAR, V.; POTKONJAK, M. Low-power semantic fault-detection in multi-sensory mobile health monitoring systems. In: IEEE. **Internet of Things (WF-IoT), 2014 IEEE World Forum on**. [S.l.], 2014. p. 32–36. Citado na página 40.

GUBBI, J. et al. Internet of things (iot): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, Elsevier, v. 29, n. 7, p. 1645–1660, 2013. Citado na página 4.

HAQUE, S. A.; RAHMAN, M.; AZIZ, S. M. Sensor anomaly detection in wireless sensor networks for healthcare. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 15, n. 4, p. 8764–8786, 2015. Citado na página 43.

HARTE, S.; RAHMAN, A.; RAZEEB, K. Fault tolerance in sensor networks using self-diagnosing sensor nodes. In: IET. **Intelligent Environments, 2005. The IEE International Workshop on (Ref. No. 2005/11059)**. [S.l.], 2005. p. 7–12. Citado 3 vezes nas páginas 44, 51 e 56.

HAYKIN, S. **Redes Neurais–Princípios e Prática**. [S.l.]: Porto Alegre: Bookman, 2001. Citado 3 vezes nas páginas ix, 30 e 32.

HEASRT, M. et al. Trends and controversies-support vector machines. **IEEE Intelligent Systems**, v. 13, n. 4, p. 18–28, 1998. Citado na página 44.

HUANG, W.; SU, X. Design of a fault detection and isolation system for intelligent vehicle navigation system. **International Journal of Navigation and Observation**, Hindawi Publishing Corporation, v. 2015, 2015. Citado 2 vezes nas páginas 44 e 56.

JÄGER, G. et al. Assessing neural networks for sensor fault detection. In: IEEE. **2014 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)**. [S.l.], 2014. p. 70–75. Citado na página 38.

JANG, J.-S. Anfis: adaptive-network-based fuzzy inference system. **IEEE transactions on systems, man, and cybernetics**, IEEE, v. 23, n. 3, p. 665–685, 1993. Citado na página 34.

JANG, J.-S. R.; SUN, C.-T.; MIZUTANI, E. Neuro-fuzzy and soft computing; a computational approach to learning and machine intelligence. 1997. Citado 3 vezes nas páginas 33, 34 e 36.

JIA, N. Detecting human falls with a 3-axis digital accelerometer. **Analog Dialogue**, v. 43, n. 7, p. 1–7, 2009. Citado na página 44.

JIANG, P. A new method for node fault detection in wireless sensor networks. **Sensors**, Molecular Diversity Preservation International, v. 9, n. 2, p. 1282–1294, 2009. Citado na página 41.

JIM, L. E.; GREGORY, M. A. A review of artificial immune system based security frameworks for manet. **International Journal of Communications, Network and System Sciences**, Scientific Research Publishing, v. 9, n. 1, p. 1, 2016. Citado na página 43.

JUNIOR, L. A. Z.; ARAÚJO, V. M. de; GUIMARÃES, A. M. Automação em casas de vegetação: Aplicação de controles automáticos e robótica. **Anais SULCOMP**, v. 7, 2015. Citado na página 74.

JURDAK, R. et al. Wireless sensor network anomalies: Diagnosis and detection strategies. In: **Intelligence-Based Systems Engineering**. [S.l.]: Springer, 2011. p. 309–325. Citado na página 8.

KAHN, J. M.; KATZ, R. H.; PISTER, K. S. Next century challenges: mobile networking for ?smart dust? In: ACM. **Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking**. [S.l.], 1999. p. 271–278. Citado na página 10.

KALMAN, R. E.; BUCY, R. S. New results in linear filtering and prediction theory. **Journal of basic engineering**, v. 83, n. 3, p. 95–108, 1961. Citado na página 22.

KALMAN, R. E. et al. A new approach to linear filtering and prediction problems. **Journal of basic Engineering**, v. 82, n. 1, p. 35–45, 1960. Citado na página 22.

KAMAL, A. R. M.; BLEAKLEY, C.; DOBSON, S. Packet-level attestation (pla): A framework for in-network sensor data reliability. **ACM Transactions on Sensor Networks (TOSN)**, ACM, v. 9, n. 2, p. 19, 2013. Citado na página 71.

KAUR, E. J.; KAUR, E. P. A survey on fault detection and recovery techniques in wireless sensor networks. **International Journal of Engineering Research and General Science**, v. 3, n. 3, 2015. Citado na página 3.

KEPHART, J. O.; CHESS, D. M. The vision of autonomic computing. **Computer**, IEEE, v. 36, n. 1, p. 41–50, 2003. Citado na página 7.

KHAN, M. Z. et al. A fault-tolerant network management architecture for wireless sensor networks. In: **11th Annual postgraduate Symposium on the Convergence of Telecommunication, Networking and Broadcasting, ISBN**. [S.l.: s.n.], 2010. p. 978–1. Citado 3 vezes nas páginas 19, 20 e 21.

KHAN, R. et al. Future internet: the internet of things architecture, possible applications and key challenges. In: IEEE. **Frontiers of Information Technology (FIT), 2012 10th International Conference on**. [S.l.], 2012. p. 257–260. Citado 3 vezes nas páginas 12, 13 e 73.

KOJIMA, T.; TSUNASHIMA, H.; MATSUMOTO, A. Fault detection of railway track by multiresolution analysis. 2006. Citado na página 44.

KOUSHANFAR, F.; POTKONJAK, M.; SANGIOVANNI-VINCENTELL, A. Fault tolerance techniques for wireless ad hoc sensor networks. In: IEEE. **Sensors, 2002. Proceedings of IEEE**. [S.l.], 2002. v. 2, p. 1491–1496. Citado na página 8.

KOUSHANFAR, F.; POTKONJAK, M.; SANGIOVANNI-VINCENTELLI, A. Fault tolerance in wireless sensor networks. **Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems**, 2004. Citado na página 42.

KOUSHANFAR, F.; POTKONJAK, M.; SANGIOVANNI-VINCENTELLI, A. **Fault Tolerance in Wireless Sensor Networks**. [S.l.]: Chapter 4, 2009. Citado na página 3.

KOVVALI, N.; BANAVAR, M.; SPANIAS, A. An introduction to kalman filtering with matlab examples. **Synthesis Lectures on Signal Processing**, Morgan & Claypool Publishers, v. 6, n. 2, p. 1–81, 2013. Citado 3 vezes nas páginas ix, 22 e 24.

KURNAZ, S.; CETIN, O.; KAYNAK, O. Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. **Expert Systems with Applications**, Elsevier, v. 37, n. 2, p. 1229–1234, 2010. Citado na página 36.

LALANDA, P.; MCCANN, J. A.; DIACONESCU, A. **Autonomic Computing**. [S.l.]: Springer, 2013. Citado na página 7.

LAU, B. C.; MA, E. W.; CHOW, T. W. Probabilistic fault detector for wireless sensor network. **Expert Systems with Applications**, Elsevier, v. 41, n. 8, p. 3703–3711, 2014. Citado 3 vezes nas páginas 4, 39 e 43.

LEE, W. L.; DATTA, A.; CARDELL-OLIVER, R. Network management in wireless sensor networks. **Handbook of Mobile Ad Hoc and Pervasive Communications: American Scientific Publishers**, 2006. Citado na página 19.

LEVIS, P. et al. Tossim: Accurate and scalable simulation of entire tinyos applications. In: ACM. **Proceedings of the 1st international conference on Embedded networked sensor systems**. [S.l.], 2003. p. 126–137. Citado na página 37.

LEWICKI, D. G.; LAMBERT, N. A.; WAGONER, R. S. Evaluation of mems-based wireless accelerometer sensors in detecting gear tooth faults in helicopter transmissions. 2015. Citado 2 vezes nas páginas 44 e 56.

LILLIEFORS, H. W. On the kolmogorov-smirnov test for normality with mean and variance unknown. **Journal of the American statistical Association**, Taylor & Francis Group, v. 62, n. 318, p. 399–402, 1967. Citado na página 63.

LIU, R.; WASSELL, I. J. Opportunities and challenges of wireless sensor networks using cloud services. In: ACM. **Proceedings of the workshop on Internet of Things and Service Platforms**. [S.l.], 2011. p. 4. Citado 2 vezes nas páginas 4 e 15.

LIU, Y. et al. A self-learning sensor fault detection framework for industry monitoring iot. **Mathematical problems in engineering**, Hindawi Publishing Corporation, v. 2013, 2013. Citado 2 vezes nas páginas 38 e 49.

LOUREIRO, A. A. et al. Redes de sensores sem fio. In: SN. **Simpósio Brasileiro de Redes de Computadores (SBRC)**. [S.l.], 2003. p. 179–226. Citado 2 vezes nas páginas 18 e 19.

MA, Q. et al. Sherlock is around: Detecting network failures with local evidence fusion. In: IEEE. **INFOCOM, 2012 Proceedings IEEE**. [S.l.], 2012. p. 792–800. Citado na página 71.

MAINWARING, A. et al. Wireless sensor networks for habitat monitoring. In: ACM. **Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications**. [S.l.], 2002. p. 88–97. Citado na página 1.

MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. **International journal of man-machine studies**, Elsevier, v. 7, n. 1, p. 1–13, 1975. Citado na página 34.

MARCAL, R. M.; HATAKEYAMA, K.; SUSIN, A. Managing incipient faults in rotating machines based on vibration analysis and fuzzy logic. **Electrical Engineering Department**, p. 1–6, 2006. Citado na página 44.

MAZHAR, N.; FAROOQ, M. A sense of danger: Dendritic cells inspired artificial immune system for manet security. In: **Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2008. (GECCO '08), p. 63–70. ISBN 978-1-60558-130-9. Disponível em: <http://doi.acm.org/10.1145/1389095.1389105>. Citado na página 43.

MINERVA, R.; BIRU, A.; ROTONDI, D. Towards a definition of the internet of things (iot). **IEEE Internet Initiative, Torino, Italy**, 2015. Citado na página 1.

MOHAMED, Y. A.; ABDULLAH, A. B. Implementation of ids with response for securing manets. In: **2010 International Symposium on Information Technology**. [S.l.: s.n.], 2010. v. 2, p. 660–665. ISSN 2155-8973. Citado na página 43.

MOLISCH, A. F. et al. Ieee 802.15. 4a channel model-final report. **IEEE P802**, Citeseer, v. 15, n. 04, p. 0662, 2004. Citado na página 11.

NANDURY, S. V.; BEGUM, B. A. Smart wsn-based ubiquitous architecture for smart cities. In: IEEE. **Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on**. [S.l.], 2015. p. 2366–2373. Citado na página 1.

NEDELTCHEV, P. The internet of everything is the new economy. **Online]: http://www. cisco. com/c/en/us/solutions/collateral/enterprise/cisco-oncisco/Cisco_IT_Trends_IoE_Is_the_New_Economy. html**, 2014. Citado na página 11.

NEVES, P.; STACHYRA, M.; RODRIGUES, J. Application of wireless sensor networks to healthcare promotion. **Journal of Communications Software and Systems**, Croatian Communications and Information Society, p. 181–190, 2008. Citado na página 1.

NIE, J.; MA, H.; MO, L. Passive diagnosis for wsns using data traces. In: IEEE. **2012 IEEE 8th International Conference on Distributed Computing in Sensor Systems**. [S.l.], 2012. p. 273–280. Citado na página 71.

NOGUEIRA, G. de M. **Improving Fault Tolerance Support in Wireless Sensor Network Macroprogramming**. Tese (Doutorado) — Universidade de São Paulo, 2015. Citado na página 39.

OSTERLIND, F. et al. Cross-level sensor network simulation with cooja. In: IEEE. **Local computer networks, proceedings 2006 31st IEEE conference on**. [S.l.], 2006. p. 641–648. Citado na página 37.

PARADIS, L.; HAN, Q. A survey of fault management in wireless sensor networks. **Journal of Network and systems management**, Springer, v. 15, n. 2, p. 171–190, 2007. Citado na página 8.

PARTYNSKI, D.; KOO, S. G. Integration of smart sensor networks into internet of things: Challenges and applications. In: IEEE. **Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing**. [S.l.], 2013. p. 1162–1167. Citado na página 4.

PRAKASH, T. S. et al. Fault tolerant qos adaptive clustering for wireless sensor networks. In: SPRINGER. **Proceedings of Ninth International Conference on Wireless Communication and Sensor Networks**. [S.l.], 2014. p. 167–175. Citado na página 41.

RAAJINI, M.; KUMAR, R.; INDUMATHI, P. Fuzzy based faulty link isolation technique in dynamic wireless sensor networks. **WSEAS Transactions on Computers**, 2015. Citado na página 39.

RAJ, V. P.; NATARAJAN, K.; GIRIKUMAR, T. Induction motor fault detection and diagnosis by vibration analysis using mems accelerometer. In: IEEE. **Emerging Trends in Communication, Control, Signal Processing & Computing Applications (C2SPCA), 2013 International Conference on**. [S.l.], 2013. p. 1–6. Citado na página 44.

REZENDE, S. O. **Sistemas inteligentes: fundamentos e aplicações**. [S.l.]: Editora Manole Ltda, 2003. Citado 2 vezes nas páginas 34 e 35.

RHEE, S.; SEETHARAM, D.; LIU, S. Techniques for minimizing power consumption in low data-rate wireless sensor networks. In: IEEE. **Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE**. [S.l.], 2004. v. 3, p. 1727–1731. Citado na página 40.

SALMON, H. M. et al. Intrusion detection system for wireless sensor networks using danger theory immune-inspired techniques. **International journal of wireless information networks**, Springer, v. 20, n. 1, p. 39–66, 2013. Citado 2 vezes nas páginas 4 e 42.

SHAO, Y.; LUNETTA, R. S. Comparison of support vector machine, neural network, and cart algorithms for the land-cover classification using limited training data points. **ISPRS Journal of Photogrammetry and Remote Sensing**, Elsevier, v. 70, p. 78–87, 2012. Citado na página 44.

SHELBY, Z.; BORMANN, C. **6LoWPAN: The wireless embedded Internet**. [S.l.]: John Wiley & Sons, 2011. v. 43. Citado 4 vezes nas páginas 11, 16, 17 e 18.

SHIH, Y.-K.; WEI, H.-Y. A soft fault detection mechanism with high accuracy on machine-to-machine communication networks. In: IEEE. **Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom), IEEE and Cyber, Physical and Social Computing (CPSCom), IEEE**. [S.l.], 2014. p. 340–343. Citado na página 38.

SILVA, R. M.; MAIA, M. Redes neurais artificiais aplicadas a detecção de intrusos em redes tcp/ip. **SSI 2004 Simpósio de Segurança da Informação**, 2004. Citado 2 vezes nas páginas ix e 30.

SINGLA, R. et al. Comparison of svm and ann for classification of eye events in eeg. **Journal of Biomedical Science and Engineering**, Scientific Research Publishing, v. 4, n. 01, p. 62, 2011. Citado na página 44.

TEJASVIT, K. Challenges in integrating wireless sensor networks into the internet. **International Journal of Engineering and Management Sciences**, p. 7–11, 2014. Citado na página 4.

TERUEL, B. J. Controle automatizado de casas de vegetação: Variáveis climáticas e fertigação. **Revista Brasileira de Engenharia Agricola e Ambiental-Agriambi**, v. 14, n. 3, 2010. Citado na página 74.

TORRES, M. G. C. **Energy consumption in wireless sensor networks using GSP**. Tese (Doutorado) — University of Pittsburgh, 2006. Citado na página 40.

VENKATESAN, L.; SHANMUGAVEL, S. Reliable energy efficient fault tolerant clustering in wireless sensor network. **Asian Journal of Scientific Research**, Asian Network for Scientific Information (ANSINET), v. 7, n. 1, p. 33, 2014. Citado na página 47.

VERMESAN, O.; FRIESS, P. **Internet of things-from research and innovation to market deployment**. [S.l.]: River Publishers Aalborg, 2014. Citado 4 vezes nas páginas 1, 2, 4 e 16.

VIEIRA, T. P.; ALMEIDA, P. E.; MEIRELES, M. R. Intelligent fault management system for wireless sensor networks with reduction of power consumption. In: IEEE. **Industrial Electronics (ISIE), 2017 IEEE 26th International Symposium on**. [S.l.], 2017. p. 1521–1527. Citado na página 81.

VUK, M.; CURK, T. Roc curve, lift chart and calibration plot. **Metodoloski zvezki**, v. 3, n. 1, p. 89–108, 2006. Citado na página 55.

WANG, C.; VO, H. T.; NI, P. An iot application for fault diagnosis and prediction. In: IEEE. **2015 IEEE International Conference on Data Science and Data Intensive Systems**. [S.l.], 2015. p. 726–731. Citado na página 39.

WEEK, B. 21 ideas for the 21st century. **August**, v. 30, p. 78–167, 1999. Citado na página 1.

WINKLER, M. et al. Wireless sensor networks for military purposes. In: **Autonomous Sensor Networks**. [S.l.]: Springer, 2012. p. 365–394. Citado na página 1.

WU, M.; TAN, L.; XIONG, N. Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications. **Information Sciences**, Elsevier, v. 329, p. 800–818, 2016. Citado na página 42.

XIE, Y.; CHEN, X.; ZHAO, J. Adaptive and online fault detection using rpca algorithm in wireless sensor network nodes. In: IEEE. **Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on**. [S.l.], 2012. p. 1371–1374. Citado na página 41.

YAMAJI, M. et al. Wireless sensor network for industrial automation. In: IEEE. **Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on**. [S.l.], 2008. p. 253–253. Citado na página 1.

YANG, Z. et al. Study and application on the architecture and key technologies for iot. In: IEEE. **Multimedia Technology (ICMT), 2011 International Conference on**. [S.l.], 2011. p. 747–751. Citado na página 12.

YEH, L.-W.; WANG, Y.-C.; TSENG, Y.-C. ipower: an energy conservation system for intelligent buildings by wireless sensor networks. **International Journal of Sensor Networks**, Inderscience Publishers, v. 5, n. 1, p. 1–10, 2009. Citado na página 1.

YU, M.; MOKHTAR, H.; MERABTI, M. Fault management in wireless sensor networks. **IEEE Wireless Communications**, IEEE, v. 14, n. 6, p. 13–19, 2007. Citado na página 3.

YU, P.; JIA, S.; XI-YUAN, P. A self detection technique in fault management in wsn. In: IEEE. **Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE**. [S.l.], 2011. p. 1–4. Citado na página 43.

ZHANG, Y.; DRAGONI, N.; WANG, J. A framework and classification for fault detection approaches in wireless sensor networks with an energy efficiency perspective. **International Journal of Distributed Sensor Networks**, Taylor & Francis, Inc., v. 2015, p. 2, 2015. Citado 3 vezes nas páginas 40, 70 e 71.

ZHOU, S.; LIN, K.-J.; SHIH, C.-S. Device clustering for fault monitoring in internet of things systems. In: IEEE. **Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on**. [S.l.], 2015. p. 228–233. Citado na página 40.

ZIEN, A. et al. Engineering support vector machine kernels that recognize translation initiation sites. **Bioinformatics**, Oxford Univ Press, v. 16, n. 9, p. 799–807, 2000. Citado na página 44.

ZOLERTIA, W. **Datasheet: Z1 platform**. [S.l.]: Electronic Publication, 2010. Citado na página 60.

ZOLERTIA, Z. **Low-power wireless module for IoT and WSN**. 2016. <http://zolertia.io/z1>. Acessado em: 05-07-2016. Citado 2 vezes nas páginas 10 e 60.

ZURADA, J. M. **Introduction to artificial neural systems**. [S.l.]: West St. Paul, 1992. v. 8. Citado 3 vezes nas páginas 24, 25 e 26.