

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CARLOS OBERDAN ROLIM

**Uso de Sensibilidade à Situação em Redes
Oportunistas para Intensificar a
Comunicação de Dados em Aplicações de
Sensoriamento Urbano**

Tese apresentada como requisito parcial para
a obtenção do grau de Doutor em Ciência da
Computação

Orientador: Prof. Dr. Cláudio Fernando Resin
Geyer

Porto Alegre
2016

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Rolim, Carlos Oberdan

Uso de Sensibilidade à Situação em Redes Oportunistas para Intensificar a Comunicação de Dados em Aplicações de Sensoriamento Urbano / Carlos Oberdan Rolim. – Porto Alegre: PPGC da UFRGS, 2016.

186 f.: il.

Tese (doutorado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2016. Orientador: Cláudio Fernando Resin Geyer.

1. Cidades Inteligentes. 2. Sensoriamento Urbano. 3. Redes Oportunistas. 4. Sensibilidade à Situação. 5. Inteligência Computacional. I. Geyer, Cláudio Fernando Resin. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Rui Vicente Oppermann

Vice-Reitora: Prof^a. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Celso Giannetti Loureiro Chaves

Diretora do Instituto de Informática: Prof^a. Carla Maria Dal Sasso Freitas

Coordenador do PPGC: Prof. Luigi Carro

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

*“Nascer, morrer, renascer ainda,
progredir sempre, tal é a Lei.”*

AGRADECIMENTOS

Antes de mais nada agradeço a Deus pela dádiva da vida e por todas as oportunidades que me são dadas.

Agradeço aos meus pais Cecília e Carlos pela criação, conselhos e exemplos de vida. O amor incondicional de vocês faz eu me tornar a cada dia uma pessoa melhor. Obrigado por terem me recebido como filho. Amo vocês! Agradeço também a minha irmã Cássia pela sua firmeza de conduta e amorosidade em seus atos. Os laços que nos unem são eternos e nossas ligações fazem com que caminhemos de mãos dadas em auxílio mútuo. Te amo mana!

Agradeço a Tatiana, minha amada companheira de caminhada, que teve paciência e compreensão comigo, me dando suporte nos momentos que mais precisei. Que Deus sempre ilumine teus passos e eu possa estar perto para crescermos juntos. Te amo!

Agradeço ao meu orientador Prof. Dr. Cláudio Geyer pelos conselhos e ensinamentos passados ao longo desses anos. Eles me tornaram melhores, me mostrando a cada momento como melhorar ainda mais o meu trabalho e explorar meu potencial. São pessoas como o senhor que temos orgulho de nos espelharmos e chamar não somente de Orientador, mas também de amigo... Com certeza vou levar isso para o resto de minha vida. Obrigado por tudo.

Aos meus incansáveis colegas de jornada e amigos Anúbis e Valderi, os quais criamos um laço de empatia tão grande que permitiu o auxílio mútuo durante os últimos anos. Obrigado por estarem presentes em minha vida. Agradeço a todos os colegas da 205, em especial ao Guilherme, Júlio, Hélio, Paulo, Pedro e Wagner. Um agradecimento especial também ao João e o Rodrigo de Pelotas pelas trocas de ideias.

Agradeço a todos os meus familiares, em especial ao meu cunhado Lucas pelo companheirismo e a minha tia Lúcia que sempre me incentivou ao estudo. Não poderia deixar de agradecer a todos os meus amigos. Não vou nomear cada um, mas cada um sabe o quanto foi importante para essa conquista. :-)

RESUMO

Cidades Inteligentes são sistemas urbanos que usam as Tecnologias da Informação e Comunicação (TICs) para tornar a infraestrutura e os serviços públicos de uma cidade mais interativos, acessíveis e eficientes aos seus habitantes. Com isso, surge a necessidade de proporcionar novos tipos de serviços que busquem auxiliar na organização da cidade, no bem-estar das pessoas e auxiliem a melhorar a governança da cidade.

Nesse contexto, o Sensoriamento Urbano é um paradigma emergente, situado no escopo de Cidades Inteligentes, que combina a ubiquidade de smartphones e de diferentes tipos de sensores para coletar dados que retratam diferentes aspectos da cidade. Um aspecto importante nesse processo de sensoriamento é a transmissão dos dados coletados para serem processados por um sistema centralizado remoto. Isso demanda uma ampla e constante cobertura de infraestrutura de rede de comunicação, fato que nem sempre é possível. As Redes Oportunistas surgem como uma alternativa complementar e inovadora para situações como essa, onde as aplicações precisam transmitir dados porém a infraestrutura de rede é intermitente ou mesmo inexistente. Entretanto, com o uso de Redes Oportunistas, as aplicações além de herdarem os seus benefícios também herdam os desafios existentes na área relacionados à tomada de decisão de encaminhamento das mensagens.

Dessa forma, a presente tese busca responder ao questionamento de como intensificar a disseminação de conteúdo e o encaminhamento de mensagens em aplicações de Sensoriamento Urbano que fazem uso de Redes Oportunistas como paradigma complementar de comunicação. Para isso é proposto o Situs, um componente de software baseado em Redes Oportunistas que utiliza Sensibilidade à Situação com vistas à proatividade nas tomadas de decisões de roteamento para com isso intensificar a entrega de mensagens. Ele emprega Lógica Fuzzy para a compreensão da situação e uma rede neural chamada Echo State Network (ESN) para efetuar a projeção de situações.

Os resultados experimentais demonstraram que a sua performance supera algumas das principais iniciativas existentes na literatura. Por fim, pode-se concluir que ele é capaz de preencher as lacunas do estado da arte apresentadas durante o desenvolvimento da tese sendo capaz de proporcionar um comportamento proativo com o uso de Sensibilidade à Situação.

Palavras-chave: Cidades Inteligentes. Sensoriamento Urbano. Redes Oportunistas. Sensibilidade à Situação. Inteligência Computacional.

Use of Situation Awareness in Opportunistic Networks to Improve Data Communication of Social Sensing Applications

ABSTRACT

Smart cities are urban systems that use Information and Communication Technologies (ICTs) to make infrastructure and public services in a more interactive, accessible and efficient city to its inhabitants. With this comes the need to provide new types of services that seek to assist in the organization of the city, the well-being of people and assist to improve the governance of the city. In this context, urban sensing is an emerging paradigm, sited in the Smart Cities scope, combining the ubiquity of smartphones with the capability of measuring sensors to collect data that depict different aspects of the city.

This ecosystem consists of different types of mobile and fixed devices orchestrated by a computational architecture that encompasses the full sensing process. An important aspect of this process is the transmission of data collected for processing by a remote central system. This requires a broad and constant coverage of communication network infrastructure, a fact that is not always possible. Opportunistic Networks emerge as an innovative and complementary alternative for situations like this where the applications need to transmit data but the network infrastructure is intermittent or unavailable. However, using Opportunistic Networks, applications as well as inherit its benefits also inherit the existing challenges in the area related to decision-making of messages forwarding.

Thus, this thesis seeks to answer the question of how to improve the dissemination of content and message routing of urban sensing applications that makes use of Opportunistic networks as complementary communication paradigm. Therefore, it proposes *Situs*, a software component based on Opportunistic Networks that uses Situation Awareness towards a proactivity in making routing decisions. For such task, it applies fuzzy logic for situation comprehension and a kind of neural network called Echo State Network (ESN) for situation projection.

The results of the experiments showed that their performance outperforms some existent initiatives in literature. Finally, we argue it fulfills the gaps of state of art presented in this thesis and could provide a proactive behaviour with usage of situation awareness.

Keywords: Smart Cities, Urban Sensing, Opportunistic Networks, Situation Awareness, Computational Intelligence.

LISTA DE FIGURAS

Figura 2.1	Hierarquia de conceitos empregados na tese	26
Figura 2.2	Áreas de aplicações de sensoriamento usando smartphones	29
Figura 2.3	Paradigma Store-carry-and forward para transporte de mensagens	33
Figura 2.4	Níveis do modelo de Sensibilidade à Situação	39
Figura 2.5	Diferentes camadas de abstração de contexto	40
Figura 2.6	Relação entre os componentes de um sistema inteligente	41
Figura 2.7	Neurônio artificial	46
Figura 2.8	Rede Neural alimentada adiante	47
Figura 2.9	Rede Neural recorrente	48
Figura 2.10	Reservoir	49
Figura 2.11	Arquitetura de uma Echo State Network	50
Figura 3.1	Classificação das iniciativas de roteamento em OppNets	67
Figura 4.1	UrboSenti: Visão conceitual	86
Figura 4.2	Módulo Backend	87
Figura 4.3	Módulo Sensing	90
Figura 5.1	Cenário problema	93
Figura 5.2	Esquematização de funcionalidade do componente proposto	96
Figura 5.3	Mapa mental representando o modelo conceitual	97
Figura 5.4	Modelo de contexto	99
Figura 5.5	Modelo de Sensibilidade à Situação empregado	101
Figura 5.6	Processo de compreensão da situação	102
Figura 5.7	Processo de previsão usado para projeção	103
Figura 5.8	Modelo de tomada de decisão	106
Figura 5.9	Arquitetura do <i>Situs</i>	108
Figura 5.10	Processo de inferência utilizando encadeamento progressivo	112
Figura 5.11	Diagrama de atividades do tratamento de contexto	115
Figura 5.12	Diagrama de atividades da tomada de decisão	118
Figura 5.13	Seleção da melhor configuração de Rede Neural	129
Figura 5.14	Caminho analítico percorrido para a definição do <i>Situs</i>	131
Figura 6.1	Visão parcial das principais avenidas de Porto Alegre mapeadas no simulador	134
Figura 6.2	Número de mensagens entregues	136
Figura 6.3	Número de mensagens iniciadas	137
Figura 6.4	Métrica de custo para entrega	138
Figura 6.5	Mensagens entregues x custo para entrega	138
Figura 6.6	Tempo médio de latência para entrega	139
Figura 6.7	Média de nodos intermediários utilizados para entrega	140
Figura 6.8	Número de mensagens encaminhadas	140
Figura 6.9	Tempo médio de armazenamento em buffer	141
Figura 6.10	Comparativo da quantidade de mensagens entregues	143
Figura 6.11	Comparativo do custo para entrega	144
Figura 6.12	Comparativo do tempo médio de armazenamento em buffer	145
Figura 6.13	Comparativo do tempo médio de latência para entrega de mensagens	146
Figura A.1	Topologia de um MLP	178

Figura A.2	Aplicação de uma função de kernel.....	180
Figura A.3	Arquitetura NARX	181
Figura A.4	Arquitetura de um ANFIS implementando o modelo Sugeno.....	183
Figura A.5	Arquitetura de um MANFIS implementando o modelo Sugeno de primeira ordem com uma entrada, três regras e três saídas.....	184

LISTA DE TABELAS

Tabela 3.1 Comparativo entre as iniciativas de Sensoriamento Urbano	65
Tabela 3.2 Comparativo entre iniciativas em Redes Oportunistas	83

LISTA DE ABREVIATURAS E SIGLAS

ANFIS Adaptative Neuro-Fuzzy Inference Systems

ANN Artificial Neural Networks

API Application Programming Interface

CI Computational Intelligence

COG Centre of Gravity

COGS Centre of Gravity for Singletons

COA Centre of Area

DTN Delay and Disruption Tolerant Network

DTNRG Delay-Tolerant Networking Research Group

ESN Echo State Networks

IoT Internet of Things

IRTF Internet Research Task Force

LM Left Most Maximum

RM Right Most Maximum

ESN Echo State Network

FCL Fuzzy Control Language

FIS Fuzzy Inference System

IEC International Electrotechnical Commission

LSM Liquid State Machine

MANET Mobile Ad Hoc Network

M2M Machine to Machine

MANFIS Multioutput Adaptative Neural Fuzzy Inference System

MF Membership functions

MLP Multi Layer Perceptron

MSE Mean Squared Error

MVEL MVFLEX Expression Language

NN Neural Networks

RC Reservoir Computing

RMSE Root Mean Squared Error

SCF Store-Carry and Forward

UAV Unmanned Aerial Vehicles

URI Uniform Resource Identifier

WSN Wireless Sensor Networks

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Motivação	15
1.2 Definição do problema e proposta	17
1.3 Objetivos	22
1.3.1 Objetivo Geral.....	22
1.3.2 Objetivos Específicos	22
1.4 Contribuições da tese	23
1.5 Escopo da tese	24
1.6 Estrutura da tese	24
2 FUNDAMENTAÇÃO TEÓRICA	26
2.1 Computação ubíqua	27
2.2 Cidades Inteligentes	27
2.3 Sensoriamento Urbano	29
2.4 Redes Ad Hoc Móveis	31
2.4.1 Redes Tolerantes a Atrasos e Desconexões	32
2.4.2 Redes Oportunistas	34
2.5 Computação sensível ao contexto	35
2.5.1 Definição de contexto	36
2.5.2 Sensibilidade ao contexto	37
2.5.3 Sensibilidade à Situação	38
2.6 Inteligência Computacional	40
2.6.1 Lógica Fuzzy ou lógica nebulosa.....	42
2.6.2 Redes Neurais Artificiais	45
2.6.2.1 Reservoir Computing	49
2.7 Considerações finais do capítulo	56
3 ESTADO DA ARTE	58
3.1 Estado da arte em Sensoriamento Urbano	58
3.1.1 Análise das iniciativas de Sensoriamento Urbano	63
3.2 Estado da arte em Redes Oportunistas	67
3.2.1 Análise das iniciativas em Redes Oportunistas	81
3.3 Considerações finais do capítulo	84
4 UMA ARQUITETURA UBÍQUA ORIENTADA A SERVIÇOS PARA O SEN- SORIAMENTO URBANO	85
4.1 UrboSenti	85
4.2 Módulo Backend	86
4.3 Sensing	89
5 O COMPONENTE DE SOFTWARE SITUS	92
5.1 Cenário-problema	92
5.2 Modelagem conceitual	96
5.2.1 Modelo de contexto.....	99
5.2.2 Modelo de situação	100
5.2.3 Modelo de tomada de decisão.....	105
5.3 Arquitetura	107
5.4 Comportamento da arquitetura	114
5.5 Algoritmos	119
5.5.0.1 Context Store	119
5.5.0.2 Context Collector	120
5.5.0.3 Forecaster	120

5.5.0.4 Situation Manager	122
5.5.0.5 Decision Maker	123
5.6 Estratégia de implementação	124
5.7 Considerações finais do capítulo.....	129
6 AVALIAÇÃO DE DESEMPENHO	132
6.1 Descrição do ambiente.....	132
6.1.1 Configuração das simulações	132
6.1.2 Métricas de avaliação	134
6.2 Experimentos.....	135
6.2.1 Experimento 1 - Teste de performance e comportamento	136
6.2.2 Experimento 2 - Comparação com outras abordagens	142
6.3 Considerações finais do capítulo.....	146
7 CONSIDERAÇÕES FINAIS	149
7.1 Perspectivas futuras	152
7.2 Publicações.....	154
REFERÊNCIAS.....	159
APÊNDICE A — REDES NEURAIS	178
A.0.0.1 Perceptron Multicamadas - MLP	178
A.0.0.2 Máquinas de Vetor de Suporte - SVM	179
A.0.0.3 Modelo Autoregressivo não-linear com entrada exogena - NARX	181
A.0.0.4 Sistemas de Inferência Neuro-Fuzzy Adaptativos - ANFIS e MANFIS	182
A.0.0.5 Experimentos executados.....	185

1 INTRODUÇÃO

Sensoriamento Urbano é um paradigma emergente, situado no escopo de Cidades Inteligentes, que combina a ubiquidade de smartphones e de diferentes tipos de sensores para coletar dados que retratam diferentes aspectos da cidade. Os dados coletados são então transmitidos via Internet para serem processados por um sistema centralizado e posteriormente as informações resultantes são disponibilizadas aos habitantes ou para outros sistemas. Entretanto, para que os dados sejam transmitidos, é necessária uma ampla e constante cobertura de infraestrutura de rede de comunicação, fato que nem sempre é possível. Nesse cenário, as Redes Oportunistas surgem como uma alternativa inovadora para situações como essa, onde as aplicações precisam transmitir dados porém a infraestrutura de rede é intermitente ou mesmo inexistente. Dessa forma, elas são vistas como uma forma complementar de comunicação, capazes de serem usadas em locais onde as redes tradicionais, como wifi ou cabeadas, não estejam presentes ou não atendam de forma satisfatória os requisitos das aplicações.

Nas Redes Oportunistas, a mobilidade e os contatos entre os nodos são explorados como forma de proporcionar a comunicação dos dados. Assim, a cada contato efetuado, deve-se adotar uma estratégia para decidir se os dados devem ser encaminhados para o outro nodo ou devem permanecer armazenados localmente. Se for decidido pelo encaminhamento, o nodo receptor dos dados deve então armazená-los até o momento que seja capaz de repassá-los a outro nodo. O processo acontece sucessivamente até que o destinatário seja alcançado. Como se percebe a decisão de encaminhar ou não uma mensagem a um nodo é crucial para o correto funcionamento. Se forem tomadas decisões erradas podem ocorrer problemas como saturação da rede, atrasos ou até mesmo a perda de mensagens.

Dessa forma, essa tese propõe o *Situs*, um componente de software baseado em Redes Oportunistas que utiliza Sensibilidade à Situação com vistas à proatividade nas tomadas de decisões de roteamento para com isso intensificar a entrega de mensagens. Ele emprega Lógica Fuzzy para a compreensão da situação e uma rede neural chamada Echo State Network (ESN) para efetuar a projeção de situações. O *Situs* será embutido no módulo de comunicação de uma arquitetura de Sensoriamento Urbano chamada *UrboSenti* que também foi desenvolvida ao longo da tese e está em fase de implementação.

Como principais contribuições da tese está a demonstração de como o conceito de sensibilidade à situação pode ser aplicado em Redes Oportunistas para melhorar as decisões de encaminhamento e também a demonstração de como as Redes Oportunistas

podem ser utilizadas como forma de complementar à transmissão de dados em aplicações de sensoriamento urbano.

1.1 Motivação

A urbanização das cidades têm crescido drasticamente nos últimos anos e a previsão é de que essa migração de pessoas para áreas urbanas não irá parar (??). Com isso surgem desafios complexos ligados a como atender objetivos relacionados ao desenvolvimento socioeconômico e também à qualidade de vida dos habitantes. O conceito de Cidades Inteligentes é uma resposta a estes desafios (??).

Segundo ??), Cidades Inteligentes são sistemas urbanos que usam as Tecnologias da Informação e Comunicação (TICs) para tornar a infraestrutura e os serviços públicos de uma cidade mais interativos, acessíveis e eficientes. Ainda segundo o autor, o que motivou o surgimento do conceito de Cidades Inteligentes são as previsões de crescimento da população urbana em cerca de 70% até 2050. Com isso, surge a necessidade de proporcionar novos tipos de serviços que busquem auxiliar na organização da cidade, no bem-estar de seus habitantes e melhorar a governança da cidade.

Nesse contexto, a computação ubíqua vem sendo utilizada como base para a implementação de serviços inteligentes, que interagem de diferentes formas com os habitantes com vistas a proporcionar informações e conhecimento a respeito de diferentes aspectos da cidade (??). Para que tais serviços possam ser prestados com efetividade são necessários dados que representem diferentes visões da cidade. É exatamente este o papel da área de Sensoriamento Urbano, considerada por alguns pesquisadores como ??) e ??), como uma das áreas mais promissoras no âmbito da computação ubíqua e pervasiva.

O Sensoriamento Urbano é um paradigma emergente, situado no contexto de Cidades Inteligentes, que combina a ubiquidade dos smartphones com a capacidade de mensuração de sensores para coletar dados que retratem diferentes aspectos da cidade (??). Estes dados são utilizados na melhoria de vida dos cidadãos e no auxílio à tomada de decisão pelos responsáveis pela gestão da cidade.

Como exemplos de aplicação de Sensoriamento Urbano em Cidades Inteligentes podem ser utilizados os seguintes cenários:

Cenário 1: “Uma cidade emprega diferentes tipos de sensores como fonte de dados para o Sensoriamento Urbano. Tais sensores estão fixos em locais estratégicos (como por exemplo, semáforos, painéis, postes, lixeiras) ou em movimento (carregados

por ônibus, trens, carros, etc). Além do uso de sensores tradicionais, são disponibilizadas aos habitantes da cidade aplicações para smartphones e tablets, que permitem a coleta intencional e não intencional de dados. Dessa forma, o conjunto de sensores permite que seja efetuado o sensoriamento da qualidade do ar, poluição sonora, luminosidade, temperatura, humidade, quantidade de chuvas e até mesmo do nível do rio que atravessa a cidade. Também é possível que um morador indique que em tal localidade da cidade existe uma grande quantidade de lixo acumulado perto de um bueiro, que determinada via possui buracos ou ainda que um poste de luz da via pública está com a lâmpada queimada e precisa ser consertado. A grande massa de dados obtidos é uma importante fonte de informações sobre o “pulso” da cidade e pode ser utilizada para o desenvolvimento de serviços inteligentes aos habitantes e também auxiliar os responsáveis pela administração da cidade na tomada de decisões.”

Cenário 2: “Uma cidade deseja fazer o controle inteligente do consumo de energia, água e gás das residências. Para isso, cada residência foi instrumentalizada com medidores inteligentes que mensuram o consumo e então enviam os dados para um sistema central efetuar o processamento. O envio dos dados não é considerado crítico, o que proporciona que eles cheguem até a central de monitoramento com atraso, sem prejuízos aos interessados. Além disso, instalar uma conexão GSM ou via cabo em cada medidor inviabilizaria o projeto. A solução encontrada foi desenvolver uma aplicação que possibilite que dispositivos móveis usados pelos funcionários ou então instalados em veículos colem os dados dos medidores via wifi a medida que exista a proximidade com o local de coleta. Após a coleta, os dados são posteriormente enviados ao sistema central e então é proporcionado que os donos das residências acompanhem o consumo e acessem os demais serviços (reimpressão de fatura, débito em conta, abertura de chamados, etc.) usando um browser ou dispositivo móvel.”

Cenário 3: “A Secretaria Municipal do Meio Ambiente de uma cidade está implantando um projeto inovador na coleta do lixo. Foram instrumentalizadas 10.000 lixeiras ao longo da cidade que são capazes de indicar a uma central de monitoramento o seu nível de uso. Com isso, é possível que seja planejado o percurso que os caminhões do lixo deverão seguir com vistas a otimizar o tempo de coleta e o consumo de combustível. A cidade possui alguns pontos de acesso wireless (*hotspots*) públicos que podem ser utilizados para dar suporte ao envio dos dados das lixeiras até a central de monitoramento.

Porém, grande parte da cidade não é coberta pela rede wifi. Foi imaginado usar cabos para conectar as lixeiras, entretanto, isso tornaria o processo de coleta não operacional. Além disso, devido à grande quantidade de lixeiras, usar GSM torna o custo do projeto inviável.”

Os cenários apresentados acima são somente alguns exemplos de aplicações que constituem um ecossistema de Sensoriamento Urbano. Diversos tipos de dispositivos são empregados para a coleta de dados e também para o acesso aos serviços inteligentes prestados ao longo da cidade.

Em todos esses cenários existe uma arquitetura computacional responsável por todo o processo de sensoriamento (envolvendo as etapas de coleta de dados, processamento e geração de resultados). Um aspecto importante em tal processo é a transmissão dos dados coletados para serem processados por um sistema específico para tal fim. Isso demanda uma ampla e constante cobertura de infraestrutura de rede de comunicação, fato que nem sempre é possível.

1.2 Definição do problema e proposta

O mundo físico está se tornando cada vez mais saturado de dispositivos móveis com capacidades computacionais e de sensoriamento que proporcionam novas possibilidades de aplicações (????). Com isso, a integração de diferentes tecnologias está proporcionando que serviços ubíquos sejam desenvolvidos e prestados em diversas áreas (??). Nesse contexto, um exemplo de aplicação emergente é o Sensoriamento Urbano.

Tradicionalmente, atividades de sensoriamento são efetuadas por Redes de Sensores Sem Fios (do inglês Wireless Sensor Networks ou WSN). Segundo ??) o sucesso no uso das WSNs para controlar e monitorar um grande número de aplicações e fenômenos, tanto pela indústria quanto pela academia, está relacionado à crescente miniaturização dos componentes eletrônicos dos sensores. Assim, surgem interessantes aplicações das WSNs para diferentes fins, como por exemplo, um sistema de monitoramento doméstico acessível por dispositivos móveis (??), o monitoramento de saúde dos moradores de uma residência (??), a definição de uma arquitetura voltada para aplicações de emergência com alta mobilidade e monitoramento em tempo real (??), um sistema para resgate utilizando robôs móveis e WSN (??). Mais especificamente na área de Sensoriamento Urbano podem ser encontrados os trabalhos de ??) para o monitoramento de postes de luz em uma cidade, o sistema para monitoramento ambiental apresentado por ??), o sistema para mo-

monitoramento da qualidade do ar de ??) e a proposta de ??) para uso de uma WSN para monitoramento e evacuação de uma estação de metrô em caso de incêndio.

Porém, apesar de serem adequadas para determinados cenários, as WSNs possuem algumas limitações relacionadas, principalmente, ao consumo de energia e à pouca banda de rede disponível que as tornam inviáveis de serem adotadas para monitoramento urbano em larga escala (??). Para superar esses desafios alguns pesquisadores propõem a união de diferentes tecnologias para o monitoramento, como por exemplo, WSNs juntamente com MANETS (??), WSNs com RFID (??) ou ainda WSNs em conjunto com Veículos Aéreos não-tripulados (??????).

De forma alternativa, alguns autores como ??), ??) e ??) apontam que o sensoriamento está evoluindo do uso de redes de sensores tradicionais para o sensoriamento baseado em pessoas através de smartphones. Tal fato está sendo viável devido aos avanços tecnológicos que tornaram possível capturar situações na qual o usuário está envolvido (como localização, atividade, contatos sociais), permitindo assim a prestação de serviços que atendam as necessidades das pessoas (????). Para ??) o sensoriamento através destes dispositivos, em áreas urbanas com grande densidade de usuários, proporciona que diferentes cenários de aplicações sejam criados, como por exemplo, monitoramento ambiental de ruído, poluição, temperatura, níveis de intensidade de sinal de redes sem fios, ou outros nos quais os sensores embutidos nos aparelhos ou conectados a esses possam ser aplicados.

Além disso, quando a coleta de dados é executada por seres humanos, cuja mobilidade não é determinada, permitindo assim que sejam coletados dados de forma contínua e que poderiam não ser amostrados se uma WSN tradicional fosse aplicada (??). O benefício de ampla cobertura de sensoriamento com o uso de smartphones também é abordado por ??), ??) e por ??). Para ??) e ??), o emprego de smartphones para o sensoriamento possui uma série de vantagens em relação às WSNs tradicionais com relação ao custo para implantação, à densidade de dispositivos que podem ser empregados para cobertura de uma área, à cobertura espaço-temporal, ao consumo de energia, etc. Além disso, conforme apontado por ??), a omnipresença dos smartphones incrementa o surgimento de novas aplicações de sensoriamento usando tais dispositivos. Esse fato também é apresentado por ??) e por ??) que também indicam que a natureza pervasiva dos smartphones é uma fonte de recursos ainda não explorada, onde aos poucos começam a surgir aplicações de sensoriamento.

Entretanto, um aspecto importante que deve ser considerado nessas aplicações é a

necessidade de uma infraestrutura de rede sempre disponível para a transmissão dos dados de sensoriamento. Porém, a transmissão desses dados via redes celulares pode ter custo elevado ou ainda podem existir regiões em que a infraestrutura dessas redes é inexistente ou prestada de forma precária, resultando em conectividade nula ou intermitente (??).

Além disso, as características de mobilidade dos dispositivos que são carregados pelas pessoas torna impossível deles estarem conectados constantemente a uma infraestrutura de rede que forneça os níveis de qualidade de serviço (QoS) adequados para as aplicações (??). Como se percebe, em cenários urbanos de larga escala o uso da infraestrutura de redes sem fios (rede celular, WLAN ou redes WiMAX) nem sempre é uma alternativa satisfatória para prover suporte as aplicações ubíquas, como as de Sensoriamento Urbano (????). Segundo autores como (????), ??) e ??), isso é o resultado do crescente tráfego gerado pelo uso de smartphones e tablets no cotidiano das pessoas que acabam sobrecarregando as redes e também pela falta de investimento das operadoras de telefonia.

Nesse contexto, as Redes Oportunistas surgem como uma alternativa complementar e inovadora para situações como essa, onde as aplicações ubíquas precisam transmitir dados porém a infraestrutura de rede é intermitente ou mesmo inexistente (????). Elas são consideradas uma subcategoria das Redes Tolerantes a Atrasos (do inglês Delay Tolerant Networks ou DTNs) e constituem um conceito promissor na busca pelo paradigma da Internet de próxima geração. Autores como ??) apontam que elas estão sendo propostas como um possível componente chave para as redes móveis do futuro (no domínio das redes 5G), uma vez que elas são capazes de complementar as estruturas de redes sem fios, como redes Wifi e celulares, permitindo a disseminação direta de conteúdo entre os usuários próximos, contribuindo assim para diminuir o tráfego nas redes celulares.

As Redes Oportunistas estabelecem um cenário semelhante a uma Mobile Ad Hoc Network (MANET), no qual os nodos existentes são capazes de se comunicarem uns com os outros usando diferentes tecnologias de comunicação, como por exemplo, Wifi ou Bluetooth. Porém, conforme apontado por (????????????????) uma diferença crucial das Redes Oportunistas em relação às MANETs é o suporte à comunicação sem a necessidade de existência de um caminho fim-a-fim entre dois nodos¹. Dessa forma, dois nodos podem trocar dados sem jamais se encontrarem ou até mesmo não estarem conectados ao mesmo tempo na rede (????).

¹A necessidade de um caminho fim-a-fim entre dois nodos é uma característica do paradigma de comunicação orientada a conexão (referenciado por alguns autores como *IP centric* ou *host-centric*) que é a base da Internet, e também utilizado em redes MANET

Isso é possível através de uma estratégia de troca de dados chamada de Armazena-Carrega-Encaminha ou Store-Carry and Forward (SCF). Ela foi definida pela RFC 5050 (??) que descreve a forma com que mensagens são encaminhadas entre os nodos de uma DTN. O seu funcionamento consiste no constante encaminhamento de mensagens entre nodos explorando contatos oportunistas que surjam. Quando um nodo encontra outro, ele deve usar algum tipo de estratégia para decidir se este nodo encontrado é um bom candidato para ser usado como “mula de dados” (*data mule*) para transportar a mensagem até o seu destino, ou o mais próximo deste. Se for decidido pelo encaminhamento, o nodo receptor da mensagem deve então armazená-la até o momento que seja capaz de repassá-la a outro nodo. O processo acontece sucessivamente até que a mensagem alcance o destinatário. Como se percebe a decisão de encaminhar ou não uma mensagem a um nodo é crucial para o correto funcionamento. Se forem tomadas decisões erradas podem ocorrer problemas como saturação da rede, atrasos ou até mesmo a perda de mensagens.

Essa estratégia de comunicação utilizada nas Redes Oportunistas pode ser empregada em aplicações de Sensoriamento Urbano. Com o seu uso, os dispositivos móveis são capazes de transmitir os dados de sensoriamento em locais da cidade onde a conectividade à rede é nula ou intermitente. Porém, as aplicações além de herdarem os benefícios do uso de Redes Oportunistas também herdam os desafios existentes na área relacionados à tomada de decisão de encaminhamento das mensagens.

Assim, a presente tese busca responder a seguinte questão de pesquisa:

“Como intensificar a troca de dados em aplicações de Sensoriamento Urbano que fazem uso de Redes Oportunistas como paradigma complementar de comunicação?”

Diferentes propostas que buscam intensificar a troca de dados podem ser encontradas na literatura. Elas vão desde abordagens mais simplistas, baseadas em inundação de rede (*flooding*), até algumas consideradas mais inteligentes, que empregam dados de contexto ou aspectos sociais na tomada de decisão.

Segundo ??), o uso de dados de contexto, além de aspectos sociais, é uma alternativa interessante para a melhorar o uso de recursos e a taxa de mensagens entregues. Entretanto, as propostas existentes utilizam dados sociais e de contexto isoladamente ou mesmo agrupados de forma a caracterizar uma visão momentânea e específica do comportamento de um nodo ou aplicação. A visão instantânea, sem correlação alguma com objetivos futuros, pode resultar em insignificância, trivialidade e vulnerabilidade às pe-

quenas mudanças ou incertezas, impactando negativamente na tomada de decisão de encaminhamento de mensagens.

Uma forma de contornar tal problema é agrupar e derivar informações a partir de dados de baixo-nível, criando uma nova camada de alto-nível que permita capturar, representar e compreender a dinamicidade do ambiente no qual as entidades estão inseridas. Esta visão holística que caracteriza uma entidade é chamada de *Sensibilidade à Situação* (??).

A Sensibilidade à Situação envolve não somente ter a percepção dos atributos e estado atual de um nodo, mas permite compreender como as informações, eventos e ações irão impactar em seu comportamento de forma imediata e futura (??), possibilitando assim a projeção de situações com vistas a permitir a proatividade nas tomadas de decisões. Porém, segundo (?????????) a proatividade em Redes Oportunistas ainda não é bem explorada. Com isso, a Sensibilidade à Situação passa a ser uma opção interessante para melhorar a tomada de decisão de encaminhamento em Redes Oportunistas e assim intensificar a troca de dados em aplicações de sensoriamento urbano.

Assim, para contribuir com o estado da arte e responder a questão de pesquisa apresentada anteriormente, é defendida a seguinte hipótese:

“A Sensibilidade à Situação é capaz de fornecer os requisitos de proatividade necessários para intensificar a troca de dados em aplicações de Sensoriamento Urbano que fazem uso de Redes Oportunistas como paradigma complementar de comunicação.”

Baseado nessa hipótese, foram definidas as seguintes subquestões que visam guiar a investigação:

1. Quais são os elementos, interações e requisitos existentes em um cenário-problema que emprega Redes Oportunistas como paradigma de comunicação para o Sensoriamento Urbano em Cidades Inteligentes?
2. Como devem ser representados os dados de contexto para permitir a inferência e raciocínio através de uma visão de alto nível?
3. Como inferir a condição corrente do nodo e então fazer projeção de ações futuras que deverão ser tomadas?
4. Qual abordagem proporciona suporte à proatividade para ser aplicada no cenário e

é computacionalmente eficiente para ser executada em dispositivos móveis e embarcados?

5. Como deve ser projetada uma solução computacional que dê suporte aos requisitos de Sensibilidade à Situação?
6. Como deve ser implementado um componente de software que aplique Sensibilidade à Situação em Redes Oportunistas?
7. Qual o seu comportamento e performance quando aplicado em um cenário de Sensoriamento Urbano controlado?

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo geral desta tese é **propor um componente de software baseado em Redes Oportunistas que utiliza Sensibilidade à Situação para intensificar a comunicação de dados em aplicações de Sensoriamento Urbano.**

Esse componente será chamado de *Situs* e poderá ser embutido no módulo de comunicação de uma arquitetura de Sensoriamento Urbano, como a UrboSenti, que também foi desenvolvida ao longo da presente tese.

1.3.2 Objetivos Específicos

1. Efetuar a modelagem conceitual do cenário-problema (Resposta a subquestão 1);
2. Desenvolver modelos que forneçam suporte ao conceito de Sensibilidade à Situação (Resposta a subquestão 2);
3. Analisar a viabilidade de uso de técnicas de Inteligência Computacional para dar suporte à compreensão e projeção de situação (Resposta às subquestões 3 e 4);
4. Projetar a arquitetura de um componente de software para dar suporte aos elementos e interações definidos na modelagem conceitual (Resposta à subquestão 5);
5. Codificar o componente de software proposto (Resposta à subquestão 6);

6. Efetuar experimentos do componente desenvolvido em um cenário de Sensoriamento Urbano simulado (Resposta à subquestão 7).

1.4 Contribuições da tese

Como contribuições da tese pode-se destacar:

- Aplicação do modelo de Sensibilidade à Situação na área de Redes Oportunistas
 - O uso de Sensibilidade à Situação na área de Redes Oportunistas é uma das principais contribuições da tese. Na literatura existem indicativos dos benefícios de emprego de tal conceito para aplicações ubíquas, porém esse aspecto ainda não havia sido explorado nas decisões de encaminhamento em Redes Oportunistas.
- Demonstração de como as Redes Oportunistas podem ser utilizadas como forma de complementar à transmissão de dados em aplicações de sensoriamento urbano.
 - É apresentado o estado da arte que demonstra que as aplicações de Sensoriamento Urbano não exploram o paradigma de Redes Oportunistas como forma de complementar a troca de dados em cenários onde a infraestrutura de rede é intermitente ou indisponível.
- Definição da modelagem conceitual contendo elementos e interações necessários para dar suporte à Sensibilidade à Situação em Redes Oportunistas;
 - No desenvolvimento da tese são apresentados modelos conceituais que implementam os três níveis de Sensibilidade à Situação (percepção, compreensão e projeção). Na literatura não foram encontrados trabalhos que forneçam essas informações com foco em Redes Oportunistas.
- Apresentação de como e quais técnicas de Inteligência Computacional podem ser empregadas para possibilitar a compreensão e projeção de situação no contexto de Redes Oportunistas;
 - As pesquisas efetuadas pela tese demonstram quais técnicas de Inteligência Computacional podem ser aplicadas na compreensão e projeção de situações e como elas devem ser empregadas na prática. Foram efetuados experimentos que demonstram a viabilidade do uso de Lógica Fuzzy e Redes Neurais para tal finalidade.

- Proposição de uma arquitetura ubíqua orientada a serviços chamada UrboSenti que é capaz de envolver todo o processo de Sensoriamento Urbano.

→ Ao longo do desenvolvimento das pesquisas pode-se constatar a necessidade de uma arquitetura computacional capaz de proporcionar o suporte a todas as etapas do processo de Sensoriamento Urbano. Foi modelada então uma arquitetura ubíqua orientada a serviços para o Sensoriamento Urbano que atende os requisitos definidos e pode utilizar o componente de software que está sendo proposto nessa tese para a comunicação de dados.

1.5 Escopo da tese

Essa tese busca demonstrar como o conceito de Sensibilidade à Situação pode ser aplicada em Redes Oportunistas para incrementar a comunicação de dados de dispositivos de sensoriamento em uma Cidade Inteligente. Dessa forma, desde já define-se que o seu escopo será na apresentação de quais elementos, interações, requisitos e métodos são necessários para a modelagem de um componente de software baseado em Redes Oportunista para a troca de dados, bem como a apresentação de experimentos que demonstrem as suas funcionalidades básicas em cenários de Sensoriamento Urbano. Não são considerados em profundidade aspectos como segurança, privacidade, descobrimento, auto-organização, transmissão e qualidade de serviço que existem de forma intrínseca em redes móveis, nem mesmo aspectos relacionados às redes de sensores como gerenciamento de energia, segurança, protocolos, calibragem de sensores, tolerância a falhas e auto-organização.

1.6 Estrutura da tese

Esta tese está organizada da seguinte forma:

- **Capítulo 2 - Fundamentação Teórica:** apresenta os fundamentos que subsidiam a presente tese. Nesse capítulo são abordados os conceitos necessários para compreensão do problema e também aqueles que serão empregados na construção da proposta.
- **Capítulo 3 - Estado da arte:** são apresentados os principais trabalhos do estado

da arte nas diferentes áreas relacionadas à tese com vistas a demonstrar as lacunas preenchidas pelo componente que está sendo proposto;

- **Capítulo 4 - Uma arquitetura ubíqua orientada a serviços para o Sensoriamento Urbano:** efetua a proposta de uma arquitetura para o Sensoriamento Urbano chamada UrboSenti. São demonstrados os seus modelos conceituais juntamente com seus principais módulos e funcionalidades;
- **Capítulo 5 - O componente de software Situs:** é apresentado o componente de software que aplica Sensibilidade à Situação em Redes Oportunistas. Inicialmente é analisado o cenário problema, e então são definidos os modelos conceituais que serão empregados na definição da arquitetura computacional. Por fim, é proposto um algoritmo que define as funcionalidades internas do componente;
- **Capítulo 6 - Resultados:** são apresentados os resultados obtidos com a aplicação do componente proposto. Foram efetuadas simulações que demonstram o seu comportamento em diferentes cenários. Além disso, é feita uma análise dos resultados obtidos;
- **Capítulo 7 - Conclusões e perspectivas futuras:** efetua as considerações finais sobre a tese, apresenta as perspectivas futuras e por fim lista as publicações produzidas ao longo do seu desenvolvimento.

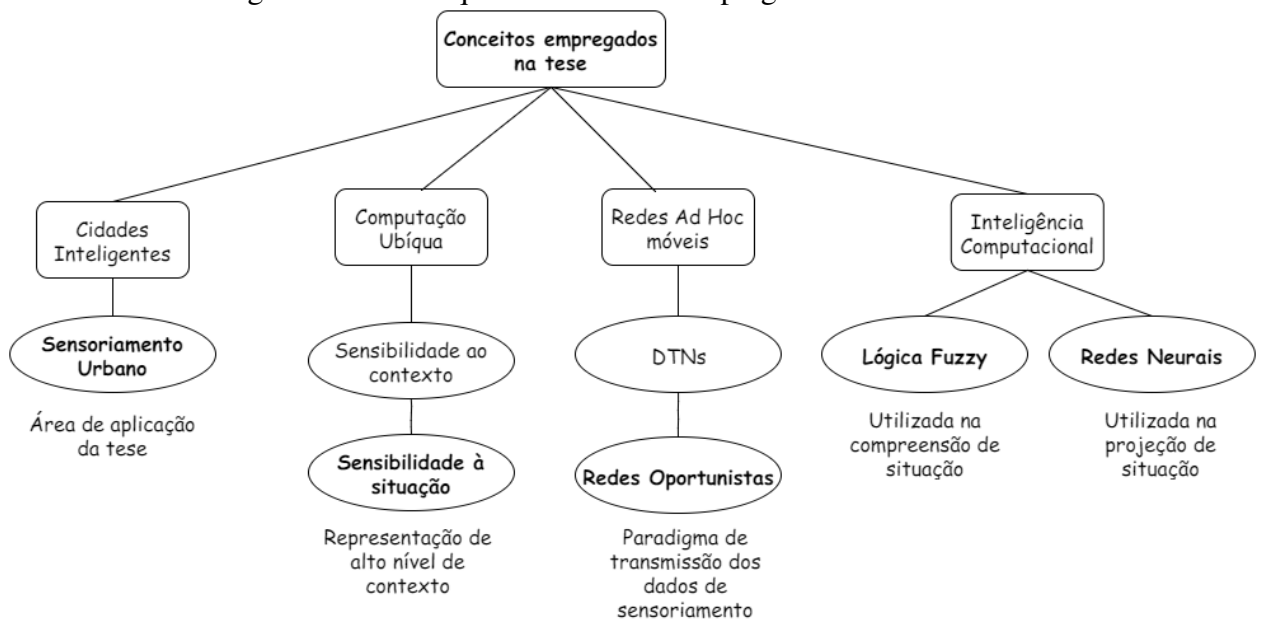
2 FUNDAMENTAÇÃO TEÓRICA

Esse capítulo visa sistematizar os conceitos utilizados no desenvolvimento do componente de software que está sendo proposto nessa tese. Inicialmente serão abordados os fundamentos da computação ubíqua com vistas a demonstrar as possibilidades de prestação de novos serviços quando se une a computação móvel com a computação pervasiva. Na sequência, será introduzido o conceito da macroárea de Cidades Inteligentes seguido de um exemplo de aplicação da computação ubíqua na subárea de Sensoriamento Urbano.

Em seguida, serão apresentadas as Redes Ad Hoc Móveis (MANETs) e também as Redes Tolerantes a Atrasos e Desconexões (DTNs) que servirão de base para a compreensão das Redes Oportunistas que é o paradigma de comunicação utilizado pelo componente que será proposto. Logo a seguir são expostos os conceitos de computação sensível ao contexto e sua representação de alto nível chamada de sensibilidade a situação. Por fim, é apresentado uma subárea da inteligência artificial chamada de inteligência computacional e duas de suas técnicas que serão empregadas na tese: Lógica Fuzzy e Redes Neurais.

A figura 2.1 demonstra hierarquicamente como estes conceitos estão organizados.

Figura 2.1: Hierarquia de conceitos empregados na tese



(Fonte: o autor)

2.1 Computação ubíqua

Na década de 1990 Mark Weiser, então pesquisador chefe da Xerox PARC, publica um artigo que prevê que os computadores pessoais desaparecerão do nosso olhar e passarão a fazer parte de todos os objetos, de forma integrada e onipresente (Ubiquitous Computing ou Ubicomp). Ele compara este fenômeno ao desaparecimento dos motores, que encolheram até passarem a fazer parte de pequenos objetos do dia a dia, porém sem tornarem-se visíveis aos olhos (??).

Diferentes autores tentam desenvolver formas inovadoras de como tornar o uso da computação ubíqua e pervasiva mais próxima da visão de Weiser segundo a qual “as tecnologias mais profundas são aquelas que desaparecem. Elas se integram à vida cotidiana até se tornarem indistinguíveis da mesma”. Nesse sentido, um dos desafios da UbiComp é a constante busca pela fusão da mobilidade da computação móvel com a capacidade de interação com o meio, agregada pela computação pervasiva.

Assim, a UbiComp possibilita o surgimento de cenários interessantes, que empregam o poder computacional em qualquer lugar, a qualquer momento (*always-on*), acessível com qualquer dispositivo. A completa integração de diversos (e diferentes) dispositivos, aplicações e plataformas com o ambiente físico possibilita a criação de ambientes dinâmicos e heterogêneos que visam a automatização e o aperfeiçoamento das soluções empregadas pelos usuários em problemas diários. Tais ambientes podem empregar a computação ubíqua e pervasiva em um único local, como por exemplo o quarto de uma casa, a casa toda, um prédio ou até mesmo cenários de larga escala que envolvem uma cidade inteira (????).

A presente tese situa-se nesse cenário de uma cidade instrumentalizada, com aplicações e serviços ubíquos dispersos ao longo da cidade. Como exemplos podem ser citados os trabalhos de ?????????).

2.2 Cidades Inteligentes

A urbanização das cidades vêm crescendo drasticamente e a expectativa é que a migração de pessoas para áreas urbanas aumente ainda mais (??). Por esse motivo, a questão de como proporcionar as condições socioeconômicas necessárias para dar suporte ao bem estar e a qualidade de vida dos habitantes é um assunto complexo com diversos desafios envolvidos. O conceito de Cidades Inteligentes é uma resposta para estes desafios

(??).

O conceito de Cidades Inteligentes foi utilizado pela primeira vez por ??) para expressar como o desenvolvimento urbano estava passando a utilizar a tecnologia, inovação e globalização em assuntos cotidianos. Com o passar do tempo o conceito passou a envolver também outras áreas relacionadas às cidades, como Economia, Infraestrutura e Governança. Com o amadurecimento gradual dos conceitos, foram definidas seis eixos que envolvem as iniciativas na área de Cidades Inteligentes: Smart Economy, Smart Governance, Smart Mobility, Smart Environment, Smart People e Smart Living. Todas esses eixos são permeados fortemente pelas Tecnologias de Informação e Comunicação (TICs) que proporcionam a infraestrutura básica necessária para o desenvolvimento de suas atividades.

Para autores como ??), objetos, serviços e pessoas em uma Cidade Inteligente estão interconectadas por uma infraestrutura de rede. Com isso, os autores afirmam que conceitos como Internet das Coisas (IoT), Internet de Serviços (IoS) e Internet de Pessoas (IoP) são subtópicos de Cidades Inteligentes. Para ??) essas áreas irão desempenhar um papel crucial nas interações entre a cidade e os cidadãos, dando a impressão que uma Cidade Inteligente não somente proporciona o gerenciamento de seus recursos de forma mais eficiente, mas também disponibiliza serviços de acordo com as necessidades dos usuários.

De uma forma mais completa, ??) define que Cidades Inteligentes são sistemas urbanos que empregam as TICs para tornar os serviços públicos de uma cidade mais interativos, acessíveis e eficientes. O autor ressalta que as previsões de crescimento da população urbana em torno de 70% em 2050 fizeram com que a busca por cidades mais inteligentes se tornasse uma conceito emergente. Como resultado, pesquisadores estão procurando alternativas para proporcionar aos habitantes da cidade novos serviços e aplicações que melhorem a qualidade de vida e ao mesmo tempo atendam os critérios de eficiência energética e sustentabilidade.

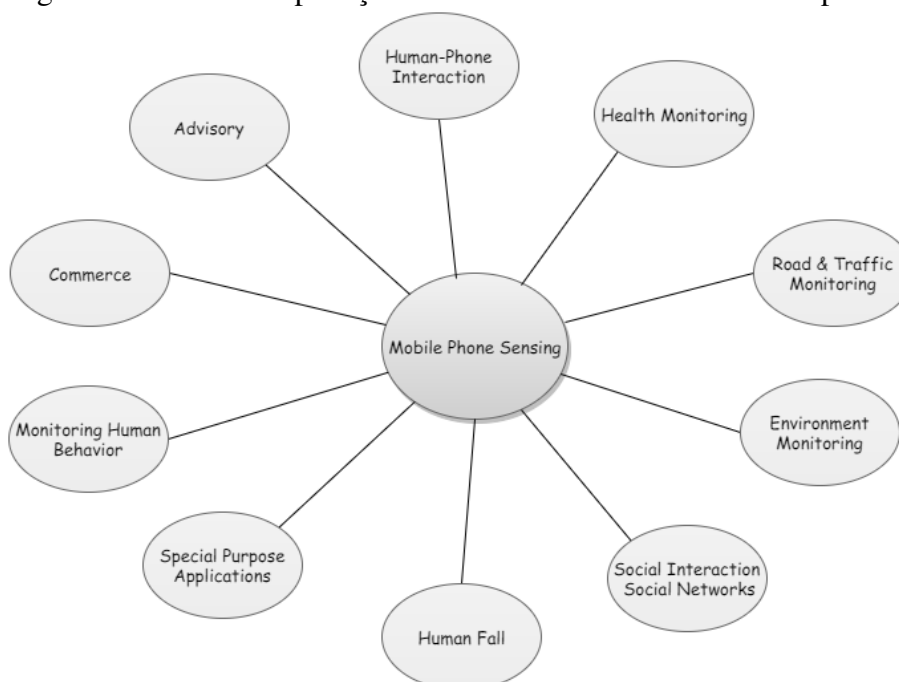
Uma das alternativas mais promissoras e que já possui aplicações reais é o uso de pessoas, dispositivos móveis e sensores para a coleta de informações a respeito de diferentes aspectos da cidade (????). Essa área é chamada de Sensoriamento Urbano e será apresentada na próxima seção.

2.3 Sensoriamento Urbano

Segundo [1] a incorporação e melhoria dos sensores em smartphones vêm permitindo que pesquisadores e engenheiros cada vez mais desenvolvam esforços para construir aplicações que explorem o potencial desses dispositivos. Ainda segundo os autores, as aplicações desenvolvidas não possuem foco somente no poder computacional e de comunicação desses dispositivos, mas também no uso dos sensores para a coleta de dados que podem ser utilizados com diferentes propósitos. A figura 2.2 representa as diferentes áreas onde as aplicações de sensoriamento com o uso de smartphones estão se tornando cada vez mais presentes.

Entre as áreas listadas, pode-se observar que existem aplicações desenvolvidas para coleta de dados ambientais (poluição, clima, ruído, luminosidade, etc), monitoramento de tráfego, interações entre pessoas, ambientes e também redes sociais. São esses tipos de aplicações que podem ser empregadas em uma área que segundo [2] busca envolver o humanos no “loop” de sensoriamento e está em expansão. Essa área que busca levar a computação ubíqua e pervasiva de dentro dos laboratórios para a coleta de dados em ambientes urbanos é chamada de Sensoriamento Urbano [3]. O Sensoriamento Urbano possui algumas variações de nomenclatura de acordo com os objetivos das aplicações e tipo de dispositivos que são empregados para o sensoriamento. Assim, surgem

Figura 2.2: Áreas de aplicações de sensoriamento usando smartphones



Fonte: Adaptado de [1] e [2]

termos como Sensoriamento Urbano centrado em pessoas (????), sensoriamento social (??) ou ainda sensoriamento com telefones móveis (??), porém todos eles objetivam a coleta de dados em determinada localidade para um propósito específico.

Segundo ??) o Sensoriamento Urbano efetuado através da interligação de pessoas, dispositivos móveis e sensores possibilita a observação, coleta, análise e disseminação de informações a respeito de eventos urbanos, sendo uma ferramenta crucial para o planejamento, gerenciamento e prestação de serviços aos habitantes de uma Cidades Inteligente.

Para ??) o Sensoriamento Urbano pode ser categorizado em duas classes maiores, de acordo com o envolvimento das pessoas que possuem os dispositivos: 1) Sensoriamento Participativo, onde o usuário é diretamente envolvido; e 2) Sensoriamento Oportunista, em que o usuário não é envolvido.

- **Sensoriamento Participativo:** A proposta participativa incorpora pessoas em etapas significantes de decisão no sistema de sensoriamento, tais como decidir que dados serão compartilhados e quais extensões dos mecanismos de privacidade devem ser autorizadas a afetar a fidelidade dos dados (??). Em outras palavras, a participação do usuário é diretamente envolvida com as ações de sensoriamento, por exemplo, fotografar determinados locais ou eventos e escolher se quer ou não compartilhar as imagens na rede.
- **Sensoriamento Oportunista:** No sensoriamento oportunista os usuários configuram seus dispositivos para permitir que os aplicativos sejam executados para a coleta de dados, contudo podem não estar cientes de quais aplicativos estão ativos em um determinado momento. O dispositivo é utilizado sempre que determinado estado (por exemplo um posição geográfica) ou determinada informação de contexto atender os requerimentos da aplicação. Este estado é detectado automaticamente pela aplicação sem que usuário precise ter conhecimento da alteração de estado para atender tais requisitos (??). Em outras palavras, o usuário não é diretamente envolvido nas decisões sobre o momento em que as aplicações estarão em funcionamento, pois a própria aplicação tomará as medidas necessárias para tanto, por exemplo: enviar os dados para um servidor informando as mudanças de contexto de localização geográfica ou temperatura corporal. Segundo ??), essa modalidade de sensoriamento explora sensores carregados por humanos (como smartphones) ou montados em veículos para coletar dados de forma ubíqua, sendo a forma mais indicada para aplicações de sensoriamento de larga escala.

Ainda de acordo com ??) embora soluções participativas e oportunista possam ser consideradas complementares, deve ser salientado que uma abordagem de projeto de sensoriamento oportunista produz um sistema que suporta mais facilmente implementações de grande escala e diversidade de aplicações.

Para ??) e ??) o sensoriamento participativo e o sensoriamento oportunista podem ser novamente divididos em subclasses levando em consideração o propósito de sua aplicação. Essa subclasses são assim descritas:

- **Sensoriamento pessoal:** o qual foca no monitoramento e armazenamento pessoal, tais como informações sobre o dia-a-dia, localização pessoal, monitoramento de saúde e de atividades físicas. As informações geradas, nesse caso, são utilizadas somente pelo próprio usuário;
- **Sensoriamento Social:** as informações são compartilhadas dentro de grupos sociais de interesse, amigos e comunidades; e
- **Sensoriamento Público:** os dados são compartilhados entre todos para o bem público, tais como informações sobre poluição, tráfego e engarrafamento de veículos.

2.4 Redes Ad Hoc Móveis

As Redes Ad Hoc Móveis ou MANET são redes compostas por um conjunto de dispositivos móveis que são capazes de se comunicar entre si através de difusão (*broad-casting*) ou então ponto-a-ponto (*point-to-point*), sem a necessidade de uma infraestrutura que coordene as suas atividades, como um Ponto de Acesso (*Access Point*).

Como exemplo de aplicações das redes MANETs podem ser citados: o monitoramento ambiental em regiões isoladas (ou sem infra-estrutura de comunicação), monitoramento civil em áreas urbanas; disseminação de conteúdo multimídia, dados topográficos, condições climáticas; rastreamento de objetos e/ou pessoas; coleta de dados de sensores; colaboração espontânea entre usuários; etc.

Como se percebe, todos os cenários de aplicação possuem nodos com alto grau de mobilidade. Tal fato ocasiona o surgimento de desafios relacionados a como lidar com a topologia de rede dinâmica, a largura de banda limitada, a conservação de energia, acesso ao meio compartilhado, segurança, descoberta de serviços/recursos e auto-configuração, as altas taxas de erros, e também ao roteamento de mensagens entre os nodos.

O roteamento de mensagens em redes ad-hoc tem tomado atenção da comunidade acadêmica e industrial que produziu centenas de artigos técnicos, sendo que alguns já foram padronizados pelo grupo de trabalho MANET da IETF.

Na literatura são encontrados propostas de diferentes classificações e taxonomias para os protocolos empregados em MANETs (mais informações em [1]). Entretanto, esses protocolos são utilizados em cenários onde existe conectividade fim-a-fim entre os nodos e baixa latência na troca de dados. Como o foco dessa tese está em cenários onde tais características não estão presentes está fora do escopo fazer uma análise detalhada dos protocolos MANETs existentes.

2.4.1 Redes Tolerantes a Atrasos e Desconexões

Na década de 90 um grupo de engenheiros, liderados por Vint Cerf, iniciou no Jet Propulsion Laboratory (JPL) da NASA o desenvolvimento do projeto Internet Inter Planetary (IPN). O objetivo da IPN é definir uma arquitetura de rede que permita que a Internet “convencional” possa ser capaz de se comunicar com uma Internet interplanetária, que envolva outros planetas e astronaves em movimento. O grande problema nesse cenário é que existe grande atraso nas comunicações e intermitência nas conexões.

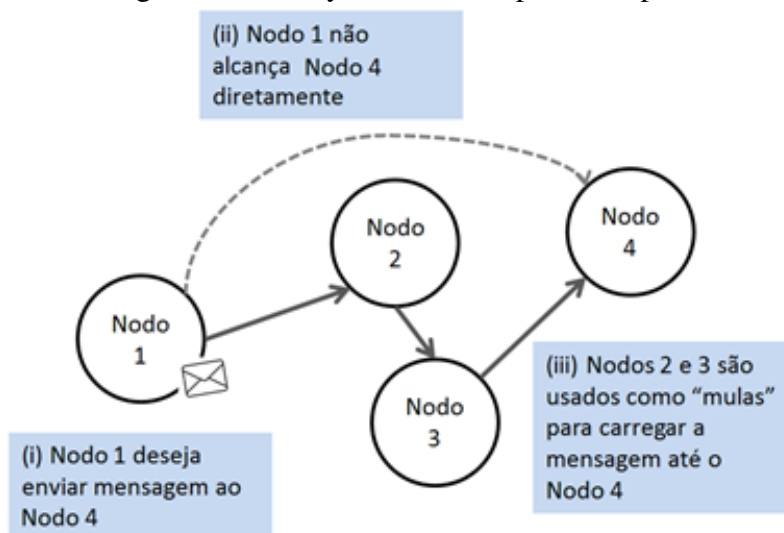
Com o tempo, observou-se que as soluções encontradas pelo projeto IPN também poderiam ser aplicadas em alguns cenários em que os protocolos Internet baseados na pilha TCP/IP não são adequados, como por exemplo: comunicações entre dispositivos móveis com restrições de energia, comunicações rurais, comunicações em campo de batalha, comunicações submarinas, etc. Conforme apontado por [2] estes ambientes são considerados “desafiadores” e possuem em comum a dificuldade de manter uma comunicação fim-a-fim com baixa latência e pequena perda de pacotes. Além disso, também existem desafios relacionados a frequentes desconexões, atrasos longos e/ou variáveis (da ordem de horas ou dias), conectividade intermitente, recursos limitados dos dispositivos de comunicação, alta taxa de erros etc.

Nesse contexto, o Internet Research Task Force (IRTF) criou em 2002 o Delay-Tolerant Networking Research Group (DTNRG), com o propósito definir protocolos e arquiteturas advindos do projeto IPN para lidar com questões de interconectividades em redes terrestres altamente heterogêneas, mesmo em situações onde a conectividade fim-a-fim nunca esteja presente.

A arquitetura de uma DTN foi definida na RFC 4838¹ e entre as suas características está o uso do paradigma de roteamento chamado de Armazena-Carrega-Encaminha ou *Store-Carry and Forward* (SCF) (??). O SCF foi criado como uma alternativa para lidar com os longos atrasos de uma DTN, que podem ser de horas ou até mesmo dias (??). Segundo os autores, isso deve-se as frequentes desconexões e altas taxas de erro – em decorrência de mudanças na topologia da rede devido a mobilidade dos nodos; por péssimas condições de comunicação; por economia de recursos; por interferências; zonas de sombra de sinal; etc.

O funcionamento do paradigma SCF consiste no constante encaminhamento de mensagens entre nodos explorando oportunidades de contatos que surjam. Quando um nodo encontra outro, ele pode se utilizar desse nodo como uma “mula de dados” (*data mule*) para transportar a mensagem até o seu destino, ou o mais próximo deste. Se for decidido pelo encaminhamento, o nodo receptor da mensagem deve então armazená-la localmente até o momento que seja capaz de repassá-la a outro nodo. O processo acontece sucessivamente até que a mensagem alcance o destinatário. Esse processo é ilustrado na figura 2.3.

Figura 2.3: Paradigma Store-carry-and forward para transporte de mensagens



(Fonte: o autor)

Outra característica da arquitetura DTN é com relação aos tipo de contatos entre os nodos para a troca de informações. Segundo ??) os contatos são classificados de acordo com a previsibilidade, e se alguma ação é necessária para que um contato possa ocorrer. Dessa forma, eles são caracterizados como:

¹<http://tools.ietf.org/html/rfc4838>

- **Persistentes:** são aqueles que estão sempre disponíveis;
- **Sob Demanda:** requerem alguma ação para que sejam instanciados, mas que, uma vez acionados, funcionam como contatos persistentes até serem encerrados;
- **Programados:** ocorrem em tempos determinados, que são conhecidos previamente;
- **Previsíveis:** são aqueles nos quais os nós podem fazer previsões sobre o horário e a duração dos contatos com base em históricos de contatos previamente realizados
- **Oportunistas:** correm sempre que dois nodos se encontram de forma não-programada (oportunista).

Os contatos oportunistas caracterizam as chamadas Redes Oportunistas que serão apresentadas na próxima seção.

2.4.2 Redes Oportunistas

As Redes Oportunistas (ou OppNets, do inglês *Opportunistic Networks*) são consideradas uma evolução das MANETs e um subcategoria de Redes Tolerantes a Atrasos e Desconexões (??). As Redes Oportunistas herdam todas as características das DTNs, porém elas exploram a mobilidade e os contatos entre os nodos como forma de proporcionar a entrega de dados (??). Assim, enquanto em DTNs há casos onde pontos de desconexão são conhecidos previamente e o roteamento pode ser feito da forma tradicional (*Internet-like fashion*), em Redes Oportunistas o roteamento é sempre efetuado de forma dinâmica (??). Dessa forma, mobilidade, desconexões, atrasos, particionamentos e demais aspectos considerados desafios em redes móveis são vistos como características das Redes Oportunistas.

Para ??), a mobilidade é explorada como uma forma de interligação entre as “nuvens” de nodos desconectadas entre si, possibilitando assim a comunicação. Nenhuma suposição é feita sobre a existência de caminhos completos entre os nodos que desejam se comunicar. A origem e o destino das mensagens podem não estar na mesma rede ou nem conectados ao mesmo tempo. Através do paradigma de Armazena-Carrega-Encaminha os nodos intermediários entre a origem e o destino armazenam a mensagem quando não existem oportunidades de encaminhamento e exploram qualquer oportunidade de contato futuro com outros dispositivos móveis para transportar a mensagem cada vez mais próxima do seu destino.

Essa abordagem de auto-organização da infraestrutura dos nodos é muito mais prática e próxima da realidade que o paradigma convencional das MANETs (??). Porém, segundo ??) as MANETs são orientadas à camada de rede enquanto as Redes Oportunistas são orientadas à camada de aplicação. Ainda segundo o autor, uma importante diferença entre ambos os tipos de rede consiste no roteamento: em uma MANETs todos os nodos se relacionam entre si com o objetivo de manter rotas fim-a-fim entre os nodos a medida que a topologia da rede evolui. Nas Redes Oportunistas, as vezes é impossível manter rotas fim-a-fim entre os nodos devido as constantes mudanças topológica da rede. Mesmo assim, o uso de nodos intermediários possibilita a troca de dados entre nodos mesmo não estando eles conectados ao mesmo tempo na rede. Essa característica que torna as Redes Oportunistas um conceito promissor para complementar a comunicação de dados em situações onde as redes tradicionais não conseguem atender plenamente os requisitos das aplicações.

Entretanto, devido a mobilidade e os contatos entre os nodos existe a necessidade de uso de uma estratégia para selecionar os nodos intermediários que serão usados como mola de dados. Esse é o principal desafio da área de Redes Oportunistas e as principais iniciativas desenvolvidas por pesquisadores serão apresentadas no próximo capítulo.

2.5 Computação sensível ao contexto

Aplicações ubíquas são empregadas em ambientes complexos e heterogêneos, que oferecem grande conectividade entre serviços computacionais e interações com o mundo físico. As interações que acontecem, principalmente com pessoas, aumentam ainda mais a complexidade e o dinamismo do ambiente.

É exatamente esse dinamismo que faz com que as aplicações precisem efetuar adaptações em seu comportamento de acordo com as mudanças que ocorram no ambiente que está a sua volta. As pesquisas relacionadas a quando e como efetuar essas adaptações são relacionadas a uma área chamada de computação sensível ao contexto que é considerada uma peça chave da computação ubíqua (??). A seguir será apresentada a definição de contexto com vistas a introduzir a computação sensível ao contexto.

2.5.1 Definição de contexto

Segundo o dicionário Aurélio a palavra contexto é definida como “um conjunto de circunstâncias que acompanham um acontecimento; julgar um fato em seu contexto histórico”. Isso é uma definição geral e não auxilia na compreensão do conceito na área computacional. O uso da palavra contexto tende a ser um tanto vago, pois todas as coisas no mundo acontecem em certo contexto.

Diversos pesquisadores procuram definir contexto através de exemplos. A primeira noção de contexto foi dada por ??), que descrevem contexto como sendo formado por locais, pessoas, objetos e mudanças desses objetos. Uma das definições mais aceitas por pesquisadores da área computacional é a sugerida por ??), que define contexto como “contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade é uma pessoa, um lugar ou um objeto que é considerado relevante para a interação entre o usuário e uma aplicação, incluindo o próprio usuário e a própria aplicação”.

De forma mais simplista ??) definiu contexto como “tudo aquilo que afeta a computação, exceto as entradas e saídas explícitas”. Para ??) contexto é o conjunto de estados do ambiente e configurações que determinam tanto o comportamento ou eventos de um aplicativo e são de interesse do usuário. Logo a seguir ??) conceituaram contexto como “propriedades que caracterizam um fenômeno, são sentidas (sensed) e potencialmente relevantes para as tarefas executadas por uma aplicação e/ou os meios pelos quais essas tarefas são desempenhadas”. Para ??) “contexto é toda informação relevante para a aplicação que pode ser obtida da infraestrutura computacional, cuja alteração em seu estado dispara um processo de adaptação na aplicação”.

Em seu trabalho mais atual a respeito de sensibilidade ao contexto, ??) remete à ideia que as informações de contexto referem-se a informações que caracterizam o ambiente físico onde se está inserido. Essas informações caracterizam um fenômeno que possui localização temporal e espacial e podem ser “sentidas” por seres humanos ou dispositivos. Segundo os autores, é importante que as aplicações sensíveis ao contexto sejam capazes de relacionar o acontecimento desse fenômeno em termos de tempo e de precisão bem como quais as entidades que esse fenômeno afetou.

Para ??) faz parte do contexto toda informação que pode descrever a situação das entidades (e suas relações) envolvidas em uma ação que é considerada importante pelo sistema. Essas entidades são todos os conceitos abstratos e objetos físicos presentes na

zona de observação do sistema num dado instante de observação.

Pela definição de contexto pode-se perceber que ele abrange todas as informações instantâneas que caracterizam uma entidade. A próxima seção busca apresentar informações a respeito de como as aplicações podem ser desenvolvidas de forma a serem sensíveis ao contexto.

2.5.2 Sensibilidade ao contexto

Os sistemas sensíveis ao contexto exploram as informações que descrevem a situação do usuário com o objetivo de adaptar seus serviços e trazer benefícios à usabilidade e ao desempenho do sistema (??). Segundo ??), um sistema é sensível ao contexto “se ele usa contexto para proporcionar informações ou serviços ao usuário de acordo as suas tarefas”.

Para ??), um sistema ubíquo sensível ao contexto é um sistema que utiliza o contexto (ou histórico de contextos) para garantir a realização da sua função, podendo para isso, alterar a sua interface de serviço ou modificar sua estrutura para gerar um novo comportamento. A construção do suporte à sensibilidade ao contexto para as aplicações apresenta inúmeros desafios, dentre eles: (i) a caracterização dos elementos de contexto para uso na aplicação; (ii) a aquisição do contexto a partir de fontes heterogêneas, tais como sensores físicos, base de dados, agentes e aplicações; (iii) a representação de um modelo semântico formal de contexto; (iv) o processamento e interpretação das informações de contexto adquiridas; (v) a disseminação do contexto a entidades interessadas de forma distribuída e no momento oportuno; (vi) o tratamento da qualidade da informação contextual (??); e (vii) a predição de contextos futuros visando a pró-atividade na execução das tarefas.

Segundo ??), o desenvolvimento e a implantação de sistemas sensíveis ao contexto ainda corresponde a uma operação complexa, que é resultante principalmente de três pontos críticos: i) a heterogeneidade dos dispositivos móveis de acesso; ii) a dificuldade em capturar/gerir as informações que descrevem o contexto de utilização; e iii) a complexidade relacionada à implementação dos mecanismos de adaptação e de filtragem de conteúdo baseados no contexto.

A sensibilidade ao contexto é uma característica intrínseca das aplicações da Ubi-comp. Quando informações de contexto computacional, contexto do usuário e contexto físico são agrupados dentro de um período de tempo se obtêm o contexto histórico desse

período o qual é útil para algumas aplicações.

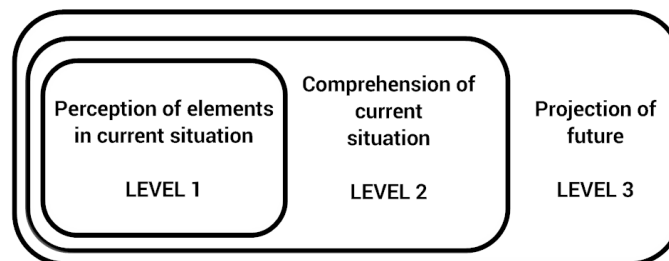
2.5.3 Sensibilidade à Situação

Informações de sensores físicos são chamadas de contexto de baixo nível e se obtidas sem maiores interpretações podem não terem significados, serem triviais ou vulneráveis a pequenas mudanças ou incertezas. As limitações dessas informações de baixo-nível acontecem quando torna-se necessário modelar o comportamento humano e seus relacionamentos. Uma forma de contornar esse problema é derivar informações de contexto a partir dos valores de baixo-nível criando uma nova camada de alto-nível no modelo de contexto que consiga capturar as percepções dos sensores como entradas para disparar ações no sistema (??). Na literatura diferentes noções têm sido empregadas para definir essa camada de alto-nível de representação do contexto. O termo mais usado é contexto situacional (??), sensibilidade a situação (??????) e situação (??). A noção de situação é usada como um conceito de alto-nível para representação de estado.

Para (??????), “situação é um conjunto de contexto relevantes para uma aplicação em um período de tempo que afetam o comportamento futuro do sistema”. De forma mais atual ??) define que situação é um conjunto de abstrações semânticas obtidas a partir de partes de informações de baixo-nível, conhecimento humano e interpretações do mundo. De forma mais simplista ??) define situação como uma abstração de eventos ocorrendo no mundo real derivadas de contextos e hipóteses sobre como os contextos observados são relacionados aos fatos de interesses do projetista e da aplicação. Computacionalmente falando, pode-se definir que uma situação é um estado particular que é abstraído dos dados dos sensores e é de interesse da aplicação, de forma que certas ações podem ser tomadas quando a situação está acontecendo.

As aplicações que utilizam técnicas de sensibilidade a situação são conhecidas como aplicações conscientes da situação ou Sensibilidade à Situação (*Situation Aware* ou SAW). Para essas aplicações, as situações são interpretações semânticas externas de contexto de baixo nível, permitindo uma especificação de alto-nível do comportamento humano e da sua interação com o sistema. Situações injetam significado às aplicações, são mais estáveis e mais fáceis de definir e manter do que informações contextuais básicas. Adaptações em aplicações sensíveis ao contexto são então causadas por mudanças nas situações (por exemplo, a mudança de um valor de contexto dispara uma adaptação se a atualização de contexto alterar a situação). O projeto dessas aplicações torna-se muito

Figura 2.4: Níveis do modelo de Sensibilidade à Situação



(Fonte: Adaptado de ??)

mais fácil, pois o projetista pode operar em alto nível de abstração (situação) e não em todas as pequenas informações de contexto que criam a situação.

Segundo ??) o modelo de Sensibilidade à Situação (figura 2.4) é composto por três níveis:

- **Nível 1 - Percepção dos elementos no ambiente:** é o primeiro passo para obtenção de sensibilidade à da situação. Ele envolve perceber estado, atributos e dinâmicas dos elementos relevantes no ambiente.
- **Nível 2 - Compreensão da situação corrente:** a compreensão da situação é baseada na síntese dos elementos obtidos no nível 1 de forma isolada. A ideia de compreensão vai além de simplesmente estar ciente dos elementos que estão presentes. Ela inclui o entendimento a significância deles em relação aos objetivos desejados.
- **Nível 3 - Projeção de estado futuro:** é a habilidade de projetar futuras ações dos elementos do ambiente, ao menos de forma aproximada. É o nível mais alto do modelo e é obtido a partir do conhecimento do estado dos elementos e também da compreensão da situação obtidos pelos níveis 1 e 2.

A vantagem de usar situações está na habilidade de proporcionar uma representação compreensível pelos humanos dos dados de sensores para as aplicações, enquanto abstrai delas a complexidade de leitura dos dados, de ruídos nestes dados e das atividades de inferência (?). O que distingue uma situação de uma atividade e reconhecimento de situação de reconhecimento de atividade é a inclusão de uma situação em aspectos temporais e estruturais, incluindo hora do dia, duração, frequência, etc. Uma situação pode ser simples, um estado abstrato de uma certa entidade (por exemplo, a sala está ocupada) ou uma ação humana que acontece em um local em um ambiente (exemplo, trabalhando ou cozinhando) ou ainda composta por outras situações mais granulares.

A figura 2.5 resume a relação de contexto ao nível de abstração, demonstrando que as informações de contexto de baixo nível são semanticamente interpretadas por camadas de contexto de mais alto-nível. As situações abstraídas dos dados de baixo-nível são reusadas em diferentes ambientes e aplicações. Os relacionamentos obtidos entre situações podem proporcionar uma maior abstração limitando a complexidade.

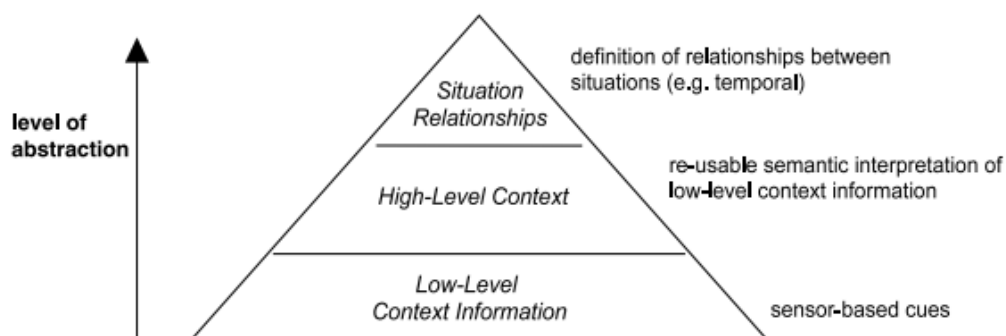
É justamente essa característica de visão de alto nível de dados de contexto que pretende-se utilizar no componente que está se propondo pois acredita-se que a melhor compreensão da situação dos nodos e a projeção de estados futuros possibilitará a proatividade nas tomadas de decisões de encaminhamento de mensagens. Não foram encontradas na literatura iniciativas que empreguem Sensibilidade à Situação em Redes Oportunistas.

A seguir será apresentado o conceito de Inteligência Computacional com vistas a demonstrar quais técnicas podem ser empregadas para a compreensão e a projeção de situações.

2.6 Inteligência Computacional

Embora utilizado em diferentes contextos, não existe uma definição amplamente aceita sobre o termo Inteligência Computacional (CI - do inglês *Computational Intelligence*) (??). O termo foi popularizado pelo pesquisador James Bezdek que o define como uma disciplina da área de Inteligencia Artificial composta por sistemas computacionais que utilizam dados numéricos, possuem capacidades de reconhecimento de padrões, exibem adaptatividade computacional e tolerância a falhas, além de possuírem taxa de erro próxima da performance humana (??). Nessa mesma linha, ??) conceitua CI como o estudo de mecanismos adaptativos para proporcionar ou facilitar o comportamento inte-

Figura 2.5: Diferentes camadas de abstração de contexto



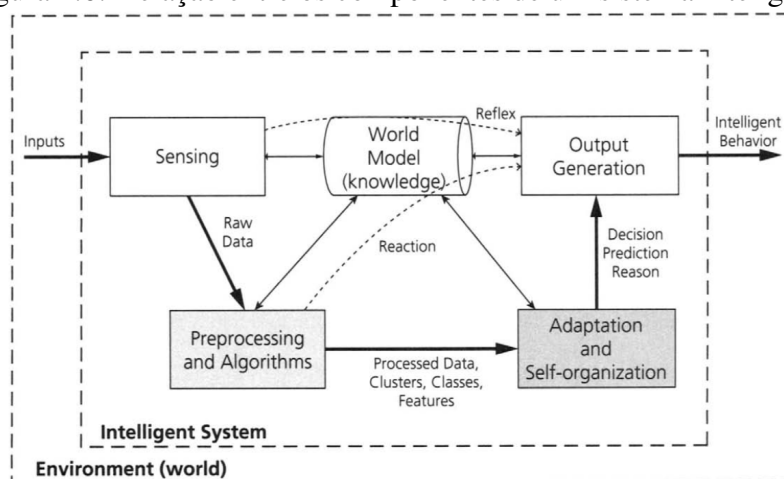
(Fonte: Adaptado de ??)

ligente em ambientes complexos e dinâmicos. Estes mecanismos incluem paradigmas de Inteligências Artificial que exibem a capacidade de aprender ou se adaptar à novas situações, de aprender, abstrair, descobrir e associar.

Para ??), CI é um ramo da ciência da computação que estuda problemas para os quais não existem algoritmos computacionais efetivos. Nessa linha de raciocínio ??), aponta que em contraste com os sistemas tradicionais baseados em IA, as técnicas de CI não requerem a construção de modelos precisos para lidar com imprecisão, incompletude e incerteza da informação. Por isso, as técnicas de CI têm sido muito utilizadas na modelagem e implementação de uma variedade de sistemas inteligentes para resolver problemas onde os modelos com maior formalismo são difíceis de serem estabelecidos ??).

Segundo ??) Inteligência Computacional é definida como uma metodologia que prove aos sistemas a habilidade de aprender e/ou lidar com novas situações. Assim, ela proporciona aos sistemas um ou mais atributos de raciocínio, como generalização, descoberta, associação e abstração. Como resultado, a saída de um sistema inteligente frequentemente inclui previsões e/ou decisões. De outra forma, CI compreende conceitos adaptações e auto-organização, paradigmas, algoritmos e implementações que habilitam ou facilitam ações apropriadas (comportamento inteligente) em ambientes complexos e dinâmicos. A visão do autor de como os componentes de um sistema inteligente se relacionam entre si e com o mundo está esquematizada na figura 2.6.

Figura 2.6: Relação entre os componentes de um sistema inteligente



Fonte: (??)

Como percebe-se a conceituação de CI é ampla e envolve diferentes tópicos de Inteligência Artificial. Conforme aponta a IEEE Computational Intelligence Society²,

²<http://cis.ieee.org/>

ela é composta pelas áreas de Redes Neurais, Sistemas Fuzzy, Algoritmos Genéticos e Computação evolutiva, incluindo Inteligência de Enxame.

Embora as aplicações de CI com frequência empreguem uma área em particular para operar efetivamente e atingir os objetivos desejados, as tendências apontam para o desenvolvimento de sistemas híbridos, que utilizam mais de uma abordagem, uma vez que uma em específico não é superior em todas as situações ??).

Assim, pode-se afirmar que o emprego de CI está diretamente relacionada à presente tese. Nesse contexto, o componente que está sendo proposto irá explorar duas abordagens em específico: Sistemas Fuzzy (para caracterização de situações a partir de dados de contexto e tomada de decisões) e Redes Neurais (para projeção de situações a partir da predição de dados de contexto). Algoritmos Genéticos e Computação Evolutiva não serão utilizados pois os mesmos são indicados para problemas de otimização e classificação (??).

2.6.1 Lógica Fuzzy ou lógica nebulosa

O conceito de Lógica Fuzzy foi introduzido, em 1965, por Lotfi A. Zadeh (Universidade da Califórnia, Berkeley) (?). Em meados da década de 60, Zadeh observou que os recursos tecnológicos disponíveis eram incapazes de automatizar as atividades relacionadas a problemas de natureza industrial, biológica ou química, que compreendessem situações ambíguas, não passíveis de processamento através da lógica computacional fundamentada na Lógica Booleana. Procurando solucionar esses problemas o Prof. Zadeh publicou em 1965 um artigo resumindo os conceitos dos conjuntos Fuzzy, revolucionando o assunto com a criação de sistemas Fuzzy. Em 1974, o Prof. Mamdani, do Queen Mary College, Universidade de Londres, após inúmeras tentativas frustradas em controlar uma máquina a vapor com tipos distintos de controladores, somente conseguiu fazê-lo através da aplicação do raciocínio Fuzzy. Esse sucesso serviu de alavanca para muitas outras aplicações, como em 1980, no controle Fuzzy de operação de um forno de cimento. Vieram em seguida, várias outras aplicações, destacando-se, por exemplo, os controladores Fuzzy de plantas nucleares, refinarias, processos biológicos e químicos, trocador de calor, máquina diesel, tratamento de água e sistema de operação automática de trens. Estimulados pelo desenvolvimento e pelas enormes possibilidades práticas de aplicações que se apresentaram, os estudos sobre sistemas Fuzzy e controle de processos avançaram rapidamente, culminando com a criação em 1984, da Sociedade Internacional de Sistemas

Fuzzy, constituída, principalmente, por pesquisadores dos países mais avançados tecnologicamente. A seguir serão apresentados os principais conceitos da Lógica Fuzzy com base no artigo original de ??).

- **Conjuntos Fuzzy:** Os conjuntos Fuzzy constituem uma “ponte” no caminho de aproximar o raciocínio humano ao da lógica executada pelo computador. Nas atividades usuais é comum o uso de um conjunto convencional ou tradicional (na literatura chamado de *crisp set*), com limites bruscos (por exemplo, o conjunto dos números maiores que 2), onde a transição dos membros para não-membros é abrupta e repentina. O grau de associação do conjunto é especificado, por exemplo, por um número “1” para os membros e “0” para os não-membros.

Em um conjunto fuzzy, ao contrário, as transições entre os membros e o não-membros estão numa faixa gradual, sendo associado um grau de pertinência aos elementos entre “0” (totalmente não-membro) e “1”(totalmente membro). Dessa forma, para a definição de um conjunto fuzzy existe o uso de uma função m_x que mapeia cada elemento de um conjunto X em um valor contínuo entre 0 e 1 de tal conjunto. Essa função é chamada de função de pertinência (na literatura citada como *membership function* ou MF). O conjunto X é referenciado como Universo de discurso ou simplesmente Universo.

- **Regras Fuzzy:** uma regra fuzzy, também chamada de implicação fuzzy, assume o formato definido pela equação (2.1)

$$if\ x\ is\ A\ then\ y\ is\ B \quad (2.1)$$

Onde A e B são valores linguísticos definidos por um conjunto fuzzy nos universos de discurso de A e B. A parte “x is A” é chamada de antecedente ou premissa e a parte “y is B” é chamada de conseqüente ou conclusão. São exemplos de regras “Se a pressão está alta então o volume é baixo” ou “Se o tomate está vermelho então ele está maduro”. Em essência uma regra fuzzy representa a relação entre duas variáveis x e y definida como relação fuzzy binária R no espaço de produto $X \times Y$. Essa relação binária R pode ser vista como uma extensão do produto Cartesiano clássico.

- **Inferência Fuzzy:** A inferência fuzzy representa o processo de tirar conclusões a partir de um conjunto de regras fuzzy. Como resultado obtém-se um conseqüente

com o mesmo grau de verdade ou não verdade que a premissa. Em lógica tradicional isso seria expresso como $p \rightarrow q$ (sendo que p e q são ambos verdadeiros ou falsos). Em Lógica Fuzzy isso é representado como $0.5p \rightarrow 0.5q$ (ambos possuem o mesmo grau de pertinência).

O processo de inferência fuzzy envolve as seguintes etapas: Inicialmente ocorre a etapa de **Fuzzificação** para mapear entradas precisas para um conjunto de valores fuzzy relevantes, sendo mais formal, essa etapa é utilizada para mapear o espaço de premissas de um espaço de entrada em formato fuzzy. Vários sistemas fuzzy assumem que todas as entradas são independentes e particionam o espaço de cada uma dessas variáveis independentemente. Lembrando que em fuzzy, a mudança de um valor para o outro não acontece abruptamente, mas sim de forma gradual. Logo a seguir é efetuada a **Inferência** sobre o conjunto de **Regras** (fornecidas por um especialista ou então extraídas de dados numéricos) usando alguma forma de raciocínio (geralmente modus ponens generalizado). Esse processo determina como as regras são ativadas e combinadas em conjuntos fuzzy de saída. Por fim a **Defuzzificação** efetua a interpretação dos conjuntos fuzzy de saída convertendo-os para saídas precisas. Esta última etapa necessita de uma estratégia de defuzzificação.

- **Modelos Fuzzy:** Existem diferentes modelos de construção de um sistema de inferência fuzzy sendo que estes diferem entre si nos processos de avaliação das regras e na defuzzificação, ou seja, diferem na forma como lidam com a determinação das consequentes. Os mais utilizados são o de Mamdani e o de Sugeno.

O **modelo Mamdani** foi desenvolvido na década de 70 sendo o mais utilizado por muitos anos. Segundo este modelo as regras de produção possuem relações difusas tanto em seus antecedentes como em seus consequentes. Tais regras são escritas no formato

$$\{Se (premissa_t) \text{ então } (consequente_t)\}_{t=1}^N$$

No **modelo Sugeno** as regras consequentes de um conjunto fuzzy, diferentemente do Mamdani, são implementadas através de uma função matemática. Assim, as regras são escritas na seguinte forma:

$$Se x \text{ é } A \text{ e } y \text{ é } B \text{ então } z \text{ é } f(x, y)$$

Onde x , y e z são variáveis linguísticas; A e B são conjuntos fuzzy no universo de discurso de X e Y e $f(x,y)$ é uma função matemática. Este modelo também é chamado de TSK (de Takagi, Sugeno e Kang) e foi originalmente proposto com o uso de um polinômio de primeira ordem. Um sistema de inferência implementado dessa forma é chamado de Sistema Sugeno de primeira ordem

O principal benefício do modelo Sugeno é a sua simplicidade computacional quando comparado com o modelo Mandani. Entretanto, não há um método direto de determinar os coeficientes para os polinômios quando implementa-se um Sistema Sugeno de primeira ordem. Para tal caso, assim como para saídas constantes no modelo de ordem zero, tais parâmetros devem ser aprendidos utilizando um algoritmo evolutivo. Outro ponto importante é que o modelo Sugeno não é tão “humanamente amigável” quanto o modelo Mandani – o conjunto fuzzy resultante e as variáveis linguísticas associadas são mais facilmente interpretadas e compreendidas por humanos que polinômios ou saídas constantes do modelo Sugeno.

Na área de computação ubíqua a Lógica Fuzzy têm sido utilizada para diferentes propósitos, como por exemplo predição de localização (??), inteligência ambiental (??) e inferência de contexto (??). Não foram encontrados trabalhos que a aplicam em sensoriamento ambiental. As iniciativas mais próximas na área de Redes Oportunistas foram o AFRON (??) que a empregou para descobrir os nodos com mais espaço em buffer e o Adaptive Fuzzy Spray and Wait (??) que a utilizou para otimizar o número de cópias de mensagens a ser disseminado na rede.

No componente que está se propondo a Lógica Fuzzy é utilizada para caracterização de situações que serão empregadas na tomada de decisões de roteamento e adaptação. Conforme aponta ??) o uso de Fuzzy proporciona suporte para decisões em tempo real em situações onde existe algum grau de incerteza ou incompletude de dados de contexto.

2.6.2 Redes Neurais Artificiais

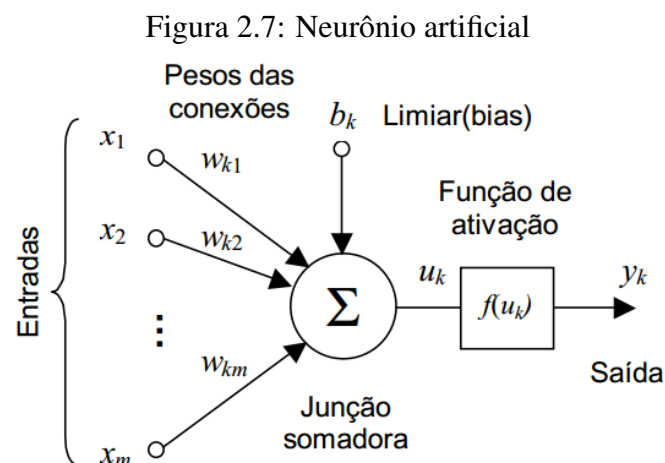
A seguir serão apresentados os conceitos que possibilitam que as Redes Neurais sejam compreendidas e utilizadas na predição de dados do contexto que provavelmente caracterizarão uma situação (projeção de estados de uma situação futura). Não está no escopo dessa tese fazer um estudo aprofundado do formalismo matemático sobre

As Redes Neurais computacionais são inspiradas no modelo de neurônio bioló-

gico. Um neurônio biológico é composto por dendritos, corpo celular, e um axônio. As conexões entre o dendrito e os axônios de outros neurônios são chamadas sinapses. Impulsos elétricos provenientes de outros neurônios são traduzidos em informações químicas em cada sinapse. Essas informações entram no corpo celular do neurônio e disparam um impulso elétrico que irá excitar outro neurônio se um certo limite de limiar for excedido. Isso desencadeia novas reações semelhantes em outros neurônios.

Rede Neurais Artificiais ou Artificial Neural Networks (ANN) são modelos do cérebro humano. O adjetivo “artificial” é empregado para diferenciar os modelos abstratos matemáticos dos modelos biológicos. Entretanto, na área computacional é comum nos referirmos somente como Rede Neurais ou Neural Networks (NN) aos modelos matemáticos implementados via software. Na presente tese será utilizado o termo Rede Neural ou simplesmente Rede para se referir a esse modelos computacionais.

Uma Rede Neural é formada por conjunto de neurônios que recebem entradas e fornecem saídas de acordo com funções de ativação internas e interconexões ponderadas com outros neurônios. Assim como no modelo biológico, cada neurônio recebe como entrada valores advindos de outros neurônios. Os valores fornecidos como entradas são então somados de forma ponderada pelos pesos das conexões juntamente com um peso de limiar (bias) que tem a função de limitar ou diminuir a influência do valor da entrada para a ativação do neurônio. A saída da soma é utilizada por uma função de ativação que têm a finalidade de introduzir a não-linearidade ao modelo e limitar a saída do neurônio. A figura 2.7 esquematiza o funcionamento de um neurônio artificial.



(Fonte: Adaptado de ??)

Matematicamente, a saída de um neurônio é dada por

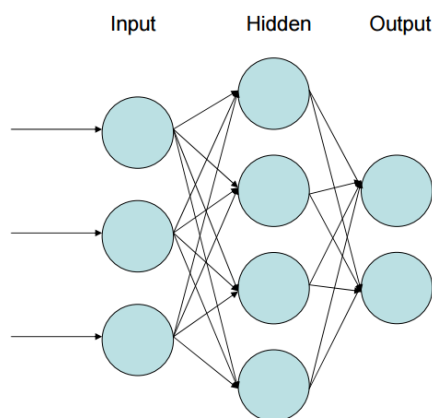
$$y(k) = f(u_k) = f\left(\sum_{j=1}^m w_{kj}x_j + b_k\right)$$

onde x_0 é um sinal de entrada de valor 1 e peso associado $w_{k0} = b_k$

Do ponto de vista estrutural as Redes Neurais são classificadas em alimentadas adiante (*feed-forward*) ou retroalimentadas (*feed-back*), também chamadas de recorrentes.

As **Redes Alimentadas Adiante** (figura 2.8) formam uma espécie de filtro que fornece a saída obtida pelo processamento de um sinal de entrada. Diferentes algoritmos são empregados para determinar os pesos sinápticos de forma que a saída obtida seja igual ou mais próximo possível da saída desejada. Um dos algoritmos mais populares de aprendizado supervisionado que interativamente define os valores dos pesos sinápticos em redes alimentadas adiante é o algoritmo *backpropagation*. O algoritmo *backpropagation* é uma extensão da regra delta³ que pode atualizar os pesos não somente entre as camadas de saída e oculta, mas também entre as camadas oculta e de entrada.

Figura 2.8: Rede Neural alimentada adiante



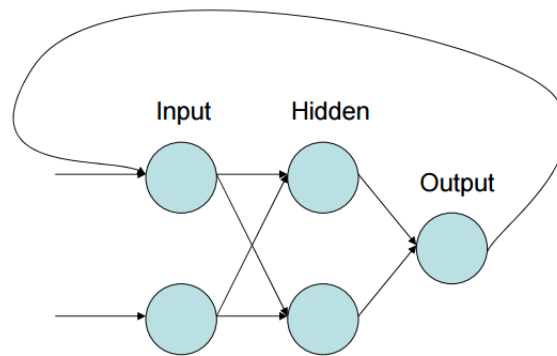
(Fonte: Adaptado de ??)

As **Redes Neurais Recorrentes** (figura 2.9) são empregadas em problemas que envolvam séries temporais, reconhecimento de fala, aproximação de uma média móvel ou outros problemas que necessitem combinar os dados ao longo do tempo. Como as Redes alimentadas adiante não são capazes de guardar estados (são consideradas *stateless*) foi necessária criar uma forma que permitisse que a rede mantivesse seus estados internos. A saída encontrada foi a criação de ligações das saídas às entradas, permitindo assim que o fluxo de dados se propaguem de forma bi-direcional, podendo ocorrer a partir dos neurônios localizados quase na saída da Rede em direção aos neurônios localizados na entrada.

Existem diferentes tipos de Redes Neurais. Cada uma foi desenvolvida com um propósito e se mostra mais adequada a determinado domínio de problema. Ao longo do desenvolvimento da tese foram efetuados experimentos com algumas Redes Neurais com

³regra delta é um algoritmo de aprendizado simples que atualiza seus pesos entre a camada de saída e a camada oculta

Figura 2.9: Rede Neural recorrente



(Fonte: Adaptado de ??)

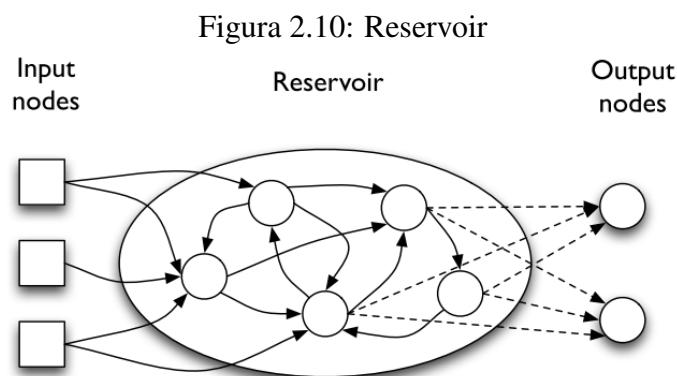
vistas a testar as suas capacidades preditivas para serem utilizadas como base para projeção de situação. Foram testados o Perceptron Multicamadas (MLP), Máquinas de Vetor de Suporte (SVM), Modelo Autoregressivo não-linear com entrada exógena (NARX) e também com uma Rede híbrida chamada de Sistemas de Inferência Neuro-Fuzzy Adaptativos (ANFIS e MANFIS). Explicações das Redes e a relação de publicações com os experimentos são apresentados no Apêndice A.

Pode-se constatar que os resultados dos experimentos com essas Redes na predição não foram satisfatórios. Além delas não terem uma acurácia suficiente, o custo computacional para execução foi considerado alto, inviabilizando o seu uso em dispositivos móveis. Outro fato negativo é apontado pela literatura: alguns autores como ??????) indicam que as Redes Alimentadas Adiante, como as que foram testadas, possuem a sua natureza de “caixa preta” que as torna difíceis de serem compreendidas e exigem muito esforço manual para encontrar a melhor configuração da rede. Também é apontado por ??) que o processo de treinamento utilizado, através dos algoritmos de descida de gradiente não convergem todas as vezes, são limitados em aprender dependências, demandam memória de maior capacidade e são computacionalmente custosos. Isso tudo devido à necessidade dos algoritmos de desempenharem duas tarefas altamente complexas: alterar os parâmetros da Rede para que ela opere de maneira correta e fazer com que a Rede tenha o comportamento de entrada e saída desejados. As Rede Neurais Recorrentes, em especial a abordagem conhecida como *Reservoir Computing* fornecem uma solução para esses problemas (??).

Reservoir Computing

Reservoir Computing (RC) é um novo paradigma de Redes Neurais Recorrentes, proposto por [?]), como uma forma de unificar as pesquisas sobre Liquid State Machine (LSM)[(?) e Echo State Network (ESN) [?]). As arquiteturas da RC se destacam dois componentes principais: um conjunto de neurônios interconectados de forma aleatória em uma topologia recorrente não-linear, chamada de reservatório (reservoir), e uma camada de saída linear, chamada readout, que é treinada por exemplos utilizando mecanismos de treinamento simples, como regressão linear. Os estados do reservatório são chamados de *echo states* e são eles que mantêm as informações sobre o histórico dos padrões de entrada. Um diferencial do Reservoir Computing com relação às Redes Neurais “tradicionais” é que somente a camada de saída (readout) é treinada enquanto que o reservatório tem pesos aleatórios. Como resultado, obtêm-se uma Rede Neural Recorrente com baixo custo computacional. Além disso, [????] apontam que as Redes Neurais Recorrentes possuem boa capacidade de previsão de dados. Tais conclusões motivam o uso desse tipo de Rede na presente tese.

A figura 2.10 demonstra a arquitetura da Reservoir Computing.



(Fonte: Adaptado de [?])

Para ilustrar o ideia de funcionamento da RC pode-se fazer a seguinte analogia: Imagine um balde com água onde são jogadas pequenas pedras. Quando uma pedra bate na água causa pequenas ondulações na sua superfície. As ondulações causadas pelas pedras criam padrões de perturbações que diminuem com o tempo e acabam sumindo. A partir dos padrões das ondulações na superfície seria possível fazer algum tipo de inferência sobre o momento e local onde as pedras foram jogadas na água. Ou seja, utiliza-se um reservatório como uma espécie de memória que aos poucos vai se extinguindo e então pode-se olhar o estado desse reservatório para tirar conclusões a respeito das entradas.

A seguir serão apresentados dois tipos de RC: Echo States Networks (a qual irá prever dados utilizados na projeção de situação) e Liquid State Machines.

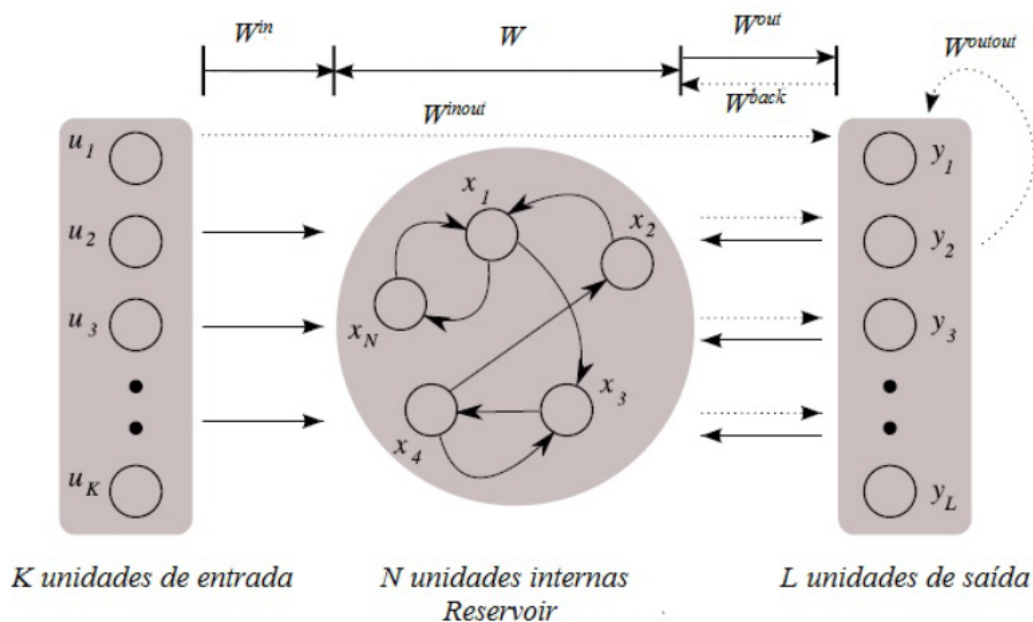
→ Echo State Networks

Segundo (??), as Echo State Networks (ESNs) representam um dos métodos pioneiros de Reservoir Computing e se baseiam na observação de que uma Rede Neural Recorrente aleatória combinada com o treinamento apenas da camada de saída (*readout*) é frequentemente suficiente para alcançar bons resultados em aplicações práticas. A parte fixa (não treinada) é chamada de reservatório dinâmico (*dynamic reservoir*), e os estados resultantes $x(n)$ são estados de eco (*echo states*) do histórico das entradas.

O nome “Echo State Networks” é baseado em uma propriedade definida como *Echo State Property* (ESP). Segundo essa propriedade, ao contrário das Rede que “esquecem” seu estado inicial, uma Rede com a propriedade ESP deve “esquecer” as informações de forma gradual. Assim ela contém em suas variáveis internas um conjunto de dados não lineares do sinal de entrada corrente e de instantes passados que “ecoam” na Rede. Na prática, (??) aponta que a existência de “echo states” é obtida através da ligação esparsa e randômica dos neurônios do reservatório e escalonando as conexões recorrentes para terem um valor de raio do espectro (que será apresentado mais abaixo) para um valor menor que 1.

O baixo custo computacional das ESNs é obtido através de uma estratégia simples e elegante: uma rede neural recorrente com uma topologia formada por neurônios sim-

Figura 2.11: Arquitetura de uma Echo State Network



Fonte: (??)

ples que utilizam a função sigmóide como função de ativação e possui conexões e pesos definidos dinamicamente é construída. Então um sinal externo é fornecido na entrada da rede e a saída obtida é utilizada para utilizada para treinar uma regressão linear simples ou uma função de classificação.

O criador das ESN, (??), as define como Redes Neurais Recorrentes com K entradas, N unidades internas na camada intermediária (reservoir) e uma camada de saída linear com L unidades (chamada de *readout*). De forma mais específica a arquitetura de uma ESN é apresentada na figura 2.11 e assim caracterizada:

- K unidades internas de entrada $u(n) = (u_1(n), \dots, u_k(n))^T$
- L unidades de saída $y(n) = (y_1(n), \dots, y_L(n))^T$
- N unidades internas do reservatório $x(n) = (x_1(n), \dots, x_N(n))^T$
- uma matriz $N \times K$ representando os pesos entre a camada de entrada e o reservatório W^{in} . Estes pesos podem ser gerados através de uma distribuição uniforme entre $[-a, a]$. Quanto maior o valor de a , mais os estados internos vão ser influenciados pela entrada. Usualmente $a = 1$.
- uma matriz $N \times N$ representando os pesos do reservatório W
- uma matriz $N \times L$ representando os pesos das conexões recorrentes entre a camada de saída e o reservatório W^{back} . Também são gerados de forma similar à W^{in} . Valores maiores de a criam forte dependência da saída com relação à entrada.
- uma matriz $L \times (K + N + L)$ representando os pesos entre o reservatório e camada de saída W^{out} . É a única matriz que é treinada pelos algoritmos de treinamento da ESN.
- uma matriz opcional W^{inout} representando as conexões entre camada de entrada e camada de saída
- uma matriz opcional W^{outout} representando as conexões recorrentes da camada de saída;
- uma função de ativação $f()$ das unidades do reservatório (usualmente \tanh ou outra função sigmóidal)
- uma função de ativação f^{out} das unidades da camada de saída (usualmente uma função linear)

Além das matrizes e funções de ativação uma ESN utiliza os seguintes parâmetros globais:

- Tamanho do reservatório (*Size of the Reservoir*): indica o número de neurônios que serão utilizados no reservatório;
- Espaçamento do reservatório (*Sparsity of the Reservoir*): representa as conexões entre as unidades;
- Raio do espectro (*Spectral Radius*): é representado por ρ e considerado um dos principais parâmetros de uma ESN. Ele determina a persistência da memória do reservatório e também o quão estáveis as ativações do reservatório são. Em outras palavras, valores de $\rho < 1$ garantem que a “Echo State Property” seja satisfeita. Na prática quando mais próximo de 1, maior será o tamanho do histórico que determina o estado corrente. Em tarefas que dependem do histórico mais recente, ρ deve ser definido com valores menores.
- Taxa de vazamento (*Leaking Rate*): é representada por α e indica a velocidade das dinâmicas de atualização do reservatório discretizadas no tempo. Valores pequenos de α geram dinâmicas de aprendizagem mais rápidas, valores mais altos geram dinâmicas mais lentas.

Segundo ??), seja uma sequência de treinamento composta por amostras de entrada/saída esperada $u(1), d(1), \dots, (u(T), d(T))$. Uma ESN $(W^{in}, W, W^{back}, W^{out})$ cuja saída y_n se aproxime da saída esperada d_n quando treinada por uma entrada u_n é obtida através dos seguintes passos:

Passo 1: Obter um reservatório dinâmico (*dynamic reservoir*) caracterizado pela tupla (W^{in}, W, W^{back}) que atende a *Echo State Property* e cujos neurônios internos possuam determinado comportamento quando excitados. Para isso as seguintes heurísticas devem ser executadas:

- Gerar uma matriz de pesos W^{in} completamente conectada com valores definidos de forma aleatória;
- Gerar uma matriz de pesos W com conexões esparsas;
- Definir o valor do raio de espectro de W da seguinte forma:
 - Calcular o valor do raio de espectro de W em α_0 .

- Normalizar a matriz W dividindo ela por seu raio de espectro ($W = W / \alpha_0$)
- Definir o raio de espectro de W para um valor α_0 desejado ($W = W \alpha$)

Passo 2: Atualizar as dinâmicas da rede. Para isso é apresentado um sinal de entrada u_n e o sinal de saída esperada d_n . Os estados do reservatório são ajustados através da equação

$$x(n+1) = f(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{d}(n))$$

e coletar os resultados dessas ativações ($x(1), x(2), \dots, x(n)$) em uma matriz S ($n \times N$, onde n corresponde ao número de entradas e N corresponde ao número de neurônios no reservatório). O valor de $x(0)$ pode ser qualquer valor arbitrário.

Além disso, utilizar as linhas de S para gerar uma matriz contendo as saídas desejadas D ($n \times L$, onde L é o número de neurônios na saída da rede).

Deve-se então descartar as k primeiras entradas das matrizes S e D para “limpar” os estados iniciais da Rede. A constante k pode até ser definida para 0.

Passo 3: Atualizar os pesos de saída W^{out} minimizando o Mean Squared Error (MSE) entre y_n e d_n . Segundo ??), para a atualização dos pesos podem ser utilizados os seguintes algoritmos:

- multiplicação da pseudo-inversa de S por D .

$$(W^{out}) = (S^{-1}D)^T$$

- método de Wiener-Hopf.

$$(W^{out}) = ((S^T S)^{-1} S^T D)^T$$

O uso da pseudo-inversa é mais lento que o método de Wiener-Hopf, porém é numericamente mais estável.

- Ridge regression (Tikhonov regularization)

$$(W^{out}) = ((S^T S + \alpha^2 I)^{-1} S^T D)^T$$

Com um parâmetro $\alpha > 0$ esse método resulta em uma rede com melhores habilidades de generalização que o método de Wiener-Hopf, mas possui a desvantagem de necessitar escolher α de uma maneira arbitrária.

Passo 4: Jaeger se refere a esse passo como “fase de exploração” (exploitation phase). Nesse passo, a ESN ($W^{in}, W, W^{back}, W^{out}$) está pronta para o uso. Um novo sinal $u(n)$ é fornecido para a rede e a saída $y(n)$ é obtida utilizando a equação

$$y(n + 1) = f^{out}(\mathbf{W}^{inout}\mathbf{u}(n + 1) + \mathbf{W}^{out}x(n + 1) + \mathbf{W}^{outout}\mathbf{y}(n))$$

A essência desse algoritmo offline é que a topologia da rede não é alterada a cada nova referencia de valores como acontece nos algoritmos online. Ao contrário, a topologia é alterada somente após cada sinal a ser aprendido é propagado pela Rede. Isso significa que os valores das matrizes S contendo os estados do reservatório e D contendo as saídas desejadas são coletados enquanto o sinal de aprendizagem é apresentado à Rede e somente então após isso o novo valor da matriz de pesos W^{out} é calculado.

As ESN têm sido aplicadas para resolver problemas práticos em diferentes domínios. Na presente tese uma ESN será utilizada como base para efetuar a projeção de estados que caracterizam uma situação futura. Em outras palavras, uma ESN será utilizada como técnica base para predição de estados. Com relação ao seu uso para predição as seguintes iniciativas podem ser citadas: na área energética, ??) usaram uma ESN para a predição de períodos de manutenção preventiva em equipamentos utilizados na geração de energia; ??) aplica uma ESN para a predição de carga em uma rede elétrica; ??) e ??) empregam ESNs para a predição de falhas em turbinas de combustão a gás para geração de energia; ??) utiliza uma ESN para predição de geração de energia eólica com 5 dias de antecedência. Essa predição é utilizada para auxiliar no planejamento do sistema energético. O interessante desse trabalho é que os autores obtiveram resultados satisfatórios com o valor de Raio do espectro (*Spectral Radius*) maiores que 1.

Em ??) os autores utilizam uma ESN para predição da taxa de câmbio. Além disso os autores realizam experimentos comparativos da performance da ESN com relação à outras técnicas de predição como fazem um comparativo de uma ESN em relação a outras técnicas de predição como naïve bayes, modelos de Médias Móveis (ARMA) e uma MLP. A ESN apresentou melhores resultados.

No trabalho de ??) é demonstrada a aplicação de uma ESN para auxiliar no processo de fornecimento de créditos em um banco. Segundo os autores, os métodos estatísticos de regressão utilizados pelos bancos na predição de falências demandam o processamento de grande quantidade de dados. Uma ESN foi proposta para prever a probabilidade de solvência de uma empresa, auxiliando nas tomadas de decisões.

Em ??) é empregada uma ESN combinada com RBM (*Restricted Boltzmann Ma-*

chines) e Lógica Fuzzy para efetuar a predição de falhas em componentes de uma rede ferroviária.

??) aplicaram uma ESN para predição de eventos de segurança em uma rede de computadores. Os autores configuraram os parâmetros globais da ESN utilizando conceitos de Redes de Mundo Pequeno (*Small World Networks*) e obtiveram bons resultados na predição da frequência e intensidade de ataques maliciosos.

Também pode ser citado ??) que aplicou ESN para predizer o fluxo de gás necessário para otimizar fornos utilizados em uma siderúrgica.

Por fim, o trabalho que mais se aproxima do escopo dessa tese é o de ??) que utiliza uma extensão de uma ESN para predição de mobilidade de nodos em uma rede celular. Os autores propuseram uma variação na construção do reservatório da ESN chamada de “Prior knowledge based Clustered Complex Echo State network (PCCESN)”. Essa variação emprega conceitos de Redes de Mundo Pequeno e Redes sem escala (Scale-Free Networks) na criação de aglomerados (*clusters*) que irão influenciar na conectividade dos neurônios utilizados no reservatório. Os resultados demonstraram que essa estratégia de construção de reservatório pode ser empregada somente em séries de tráfego multiperíodicos.

→ Liquid State Machines

Liquid State Machines (LSM) é outro método pioneiro de RC, proposto por ??), e desenvolvida de forma independente e simultânea às ESNs. Assim como as ESNs, uma LSM é composta por uma grande coleção neurônios interligados aleatoriamente uns aos outros formando um reservatório (aqui chamado de líquido) e uma função de readout sem memória (geralmente um perceptron simples) que é usado para extrair informações do líquido.

Segundo ??) as LSMs implementadas com neurônios pulsantes, utilizam modelos mais sofisticados de sinapse e são mais difíceis de implementar, de inicializar e configurar corretamente e tipicamente são computacionalmente mais caras do que as ESNs. Assim elas são menos utilizadas em aplicações da engenharia. Entretanto, os neurônios pulsantes são capazes de processar informações mais complexas pois mantém codificadas informações do tempo de disparo gerados pelos sinais.

Com base nessa afirmação e devido a necessidade de utilizar uma técnica de predição com baixo custo computacional, capaz de ser executada em dispositivos móveis, foi descartado o uso de uma LSM na presente tese. Entretanto ela parece ser uma proposta interessante que pode ser melhor avaliada futuramente, quando surgirem dispositivos mó-

veis com maior poder de processamento e menor consumo de energia.

2.7 Considerações finais do capítulo

O presente capítulo teve por objetivo apresentar os conceitos básicos que são empregados na tese. Foi abordado o conceito de computação ubíqua e então introduzida a definição da macroárea de Cidades Inteligentes e um exemplo de uso da computação ubíqua nessa área que é o Sensoriamento Urbano, área de aplicação dessa tese.

Em seguida foi apresentado o conceito de Redes Ad Hoc Móveis (MANETs) que são utilizadas para a comunicação *ad hoc* entre os nodos. Constatou-se que os protocolos das MANETs usam o paradigma de comunicação orientada a conexão (referenciado por alguns autores como *IP centric*) que necessita de um caminho fim-a-fim entre dois nodos para que a comunicação ocorra. Como em cenários de Sensoriamento Urbano tal fato nem sempre é possível, o emprego de tais protocolos foi descartado.

Passou-se então ao estudo dos conceitos das Redes Tolerantes a Atrasos e Desconexões (DTNs) como forma de introduzir a sua subcategoria que utiliza contatos entre os nodos para troca de dados chamada de Redes Oportunistas. Tal paradigma de comunicação é utilizado na tese como forma de complementar a troca de dados entre os nodos em locais onde outras redes não atendam de forma satisfatória os requisitos das aplicações de sensoriamento.

Em seguida foram abordados os conceitos de computação sensível ao contexto e então foi apresentado o conceito de Sensibilidade à Situação. O uso desse conceito buscou demonstrar que a derivação de dados de contexto para uma visão de nível mais alto fornece os meios necessários para alcançar o comportamento proativo que até então não é explorado em Redes Oportunistas. Um aspecto importante da Sensibilidade à Situação é a sua capacidade de compreensão e projeção de ações futuras. Porém, para que isso aconteça são necessárias técnicas que sejam capazes de lidar de forma “inteligente” com os dados de contexto. Por tal motivo foi introduzido o conceito de Inteligencia Computacional.

Entre as técnicas de Inteligencia Computacional duas foram selecionadas para serem utilizadas na tese: Redes Neurais e Lógica Fuzzy. A escolha de Redes Neurais deveu-se principalmente a sua capacidade de efetuar predição de valores, que é uma necessidade para projeção de situações. Entre os diferentes tipos de Redes analisados uma chamada Echo State Networks (ESN) se destacou pela sua acurácia nas predições com baixo custo

computacional, sendo propícias para o uso em dispositivos móveis com restrições de processamento e energia. A outra técnica, a Lógica Fuzzy, foi escolhida como base para a compreensão de situação devido a sua capacidade de proporcionar o raciocínio sobre informações complexas, imperfeitas e incertas que são características dos dados de contexto de um cenário de sensoriamento.

Conforme explicado anteriormente, esse capítulo abordou somente os conceitos relacionados à tese. A apresentação e análise do estado da arte será efetuada no próximo capítulo.

3 ESTADO DA ARTE

Esse capítulo irá apresentar o estado da arte nas duas principais áreas que compõem a tese: sensoriamento urbano e redes oportunistas.

Na seção do estado da arte em Sensoriamento Urbano serão analisadas as principais iniciativas com vistas a demonstrar que o uso de sensores móveis e fixos para atividades de sensoriamento permite a melhor compreensão das dinâmicas de uma cidade. Também será demonstrada a viabilidade das aplicações de sensoriamento utilizarem Redes Oportunistas como uma forma complementar de comunicação, que possibilite a troca de dados quando uma infraestrutura de rede é intermitente ou indisponível. Entretanto, com o uso de Redes Oportunistas, herda-se os desafios dessa área, principalmente, os relacionados às decisões de encaminhamento.

Para superar tais desafios, variadas iniciativas foram concebidas com foco em diferentes cenários de aplicação. Assim, na sequência será feita uma análise do estado da arte em Redes Oportunistas com a apresentação das principais iniciativas existentes. Essa análise visa demonstrar que se tais iniciativas forem aplicadas em cenários de Sensoriamento Urbano, elas não serão capazes de atender plenamente aos requisitos apresentados, resultando assim, na lacuna do estado da arte que é explorada pela presente tese.

3.1 Estado da arte em Sensoriamento Urbano

Na literatura, podem ser encontradas diferentes aplicações para Sensoriamento Urbano. Pode-se citar o BeCity (??) que explora bicicletas em uma cidade para auxiliar na melhoria do tráfego da cidade, o Mosaic (??) que embarca nodos de sensoriamento em ônibus para mensurar a qualidade do ar, o EcoSensor (??) que também analisa a qualidade do ar, porém utiliza sensores móveis baseados em Raspberry e Waspote juntamente com smartphones Android, o SenSpeed (??) que emprega o acelerômetro dos smartphones para auxiliar na melhoria da estimativa de velocidade de aplicações de trânsito, o Cruisers(??) e o CarRank (??) que empregam veículos para a coleta de dados de sensoriamento, o P^3 (??) que usa smartphones posicionados dentro de carros para localizar buracos nas ruas, o ParkGauge (??) que emprega o sensoriamento coletivo usando smartphones para determinar de forma autônoma o estado do veículo e então auxiliar na localização de vagas em garagens de estacionamento, a solução proposta por (??) do MIT que emprega uma plataforma baseada em Arduíno montada no teto de veículos para

monitorar a infraestrutura de iluminação da cidade.

Cada uma dessas aplicações buscam solucionar um problema pontual usando Sensoriamento Urbano. No que se refere, de modo específico, à plataformas e arquiteturas para o suporte à execução e construção de aplicações de Sensoriamento Urbano pode-se destacar os seguintes trabalhos:

- **MetroSense** é uma arquitetura desenvolvida por ??) que oferece um paradigma para Sensoriamento Urbano em larga escala utilizando redes de sensores oportunistas. Na proposta é considerado o uso de nodos sem fio para suporte ao sensoriamento, distribuição de tarefas e coleta de dados. Para validação utiliza um caso de estudo simulado usando o NS-2. Simularam utilizando 10k e 15k nodos por 20 minutos.
- **AnonySense** é um *framework* proposto por ?????). Ele têm foco em aplicações de sensoriamento oportunistas e participativas que coloca uma forte ênfase em privacidade. Ele adota um modelo *Polling* para a distribuição das tarefas a fim de não revelar a localização dos dispositivos móveis para a infraestrutura. Isso torna mais garantido o processo de anonimato de dados publicados, antes mesmo de serem utilizados pelos serviços ou aplicações. Além disso, as tarefas são escritas utilizando uma linguagem de domínio específico chamada AnonyTL, na qual são usados predicados baseados na contexto do dispositivo de telefonia móvel, tais como sua localização e se a dispositivo está em movimento ou não. O AnonySense possui alto custo computacional (*overhead*) oriundo do uso do modelo *pull*. Tal modelo proporciona mais privacidade que o modelo *push*, porém não escala bem com aplicações de larga distribuídas que exijam escalabilidade. O cenário de validação do AnonySense foi desenvolvido com um servidor e outros elementos para anonimização feitos em linux. A mensuração de desempenho da rede feito usando WireShark. Além disso, foi empregado somente um Nokia N800 com Linux PDA para analisar o consumo de energia, CPU, uso de banda e memória e o download de tarefas. Além disso, os relatos foram feitos em SML e a comunicação via HTTPS.
- **Medusa** foi proposto por ?????) como um framework que provê uma linguagem de programação e um sistema de execução distribuído para sensoriamento móvel participativo tendo como foco multidão (*crowd-sensing*). A proposta busca prover uma plataforma comum para realizar qualquer tipo de tarefa suportada pelos sensores de smartphones. Para este fim, Medusa utiliza uma linguagem de programação de

alto nível baseada em XML (*eXtensible Markup Language*), chamada Med-Script. Essa linguagem especifica a sequência de tarefas de sensoriamento, incentivos e processamento de tarefas que serão executadas nos smartphones (*workers*), sendo coordenadas por serviços em nuvem. Para validação da iniciativa foram desenvolvidas e testadas 10 aplicações em um dispositivo com Android v2.3.6.

- **PRISM** é uma plataforma para sensoriamento móvel desenvolvida por ??) que têm por objetivo tornar mais fácil o desenvolvimento e implantação de aplicações participativas e oportunistas. Tais aplicações podem ser executadas em um grande número de telefones móveis. Além disso, PRISM procura tratar alguns desafios, relacionados à segurança, privacidade, escalabilidade e ao controle de acesso aos recursos dos smartphones. Para validação foi utilizado um cenário com 15 smartphones reais e uma base de dados com 260 mil entradas de veículos monitorados que foram utilizados para simular smartphones para testes de escalabilidade.
- **MobiSens** é uma plataforma de sensoriamento móvel proposta por ??) que busca ser versátil para ser suportada por diversos tipos de aplicações oportunistas e participativas. Os autores buscaram dar flexibilidade ao desenvolvimento de aplicações tanto em escala individual (e.g. Monitoramento de queda de idosos) como social e pública (e.g. coleta de dados de participantes para inferência de comportamentos coletivos). O trabalho também, busca atender alguns requisitos comuns nesses tipos de aplicações em questões de otimização de energia, interação entre servidor e cliente móvel, reconhecimento, segmentação e anotação de atividades além de questões de privacidade. A avaliação da iniciativa foi efetuada com voluntários que baixaram a aplicação. Foram 310 usuários monitorados por 5 meses totalizando 13.993 horas de dados coletados.
- **Pogo** é um middleware proposto por ??) para sensoriamento de telefonia móvel visando facilitar a construção e experimentação de aplicações em larga escala, tanto oportunistas como participativas. Ele permite o controle em nível de usuário de recursos em granularidade fina com o objetivo de proteger a privacidade dos smartphones dos usuários voluntários, uma vez que realiza o monitoramento de forma contínua do comportamento da aplicação. Pogo utiliza o protocolo XMPP para disseminar os conjuntos de dados. Para validação foram desenvolvidas duas aplicações, 24 dias de monitoramento, 8 participantes, dentre eles 6 usaram Samsung Galaxy Nexus, outros dois usaram Sony Ericsson Xperia x10 mini e Samsung

Nexus S.

- **CenceMe**: é uma plataforma para Sensoriamento Urbano que possui como maior diferencial o compartilhamento de informações classificadas em redes sociais como iGoogle e Facebook. O CenceMe foi proposto por ??). Por ser uma das primeira plataformas de Sensoriamento Urbano carece de diversos mecanismos para garantir maior segurança, privacidade, customização e o equilíbrio custo desempenho. A validação foi feita com 22 voluntários durante 3 semanas efetuando tarefas de sensoriamento através de telefones Nokia N95.
- **SenseMyCity**: é um framework proposto por ??) para o desenvolvimento de aplicações de sensoriamento coletivo em larga escala. O framework explora os sensores dos smartphones para a coleta de dados e a posterior transmissão para um servidor central efetuar o processamento. Os dados podem ser transmitidos sob-demanda ou em tempo real, de acordo com o tipo de conexão configurada conforme o cenário. A característica modular do framework permite que novos sensores sejam acrescentados de acordo com o tipo de aplicação. Os autores realizaram experimentos para verificação do consumo de energia com diferentes tipos de sensores e variações de taxas de coleta de dados. Foram realizados testes reais com 115 voluntários com diferentes casos: sensoriamento de dados ao longo cidade para melhoria de conectividade de veículos e pessoas, acompanhamento de stress cardíaco em motoristas de ônibus público, coleta de dados de veículos para melhoria do consumo de combustível. Os resultados demonstraram uma estimativa do consumo de energia e o espaço para armazenamento necessários para cada tarefa de sensoriamento. O framework também está sendo usado por pesquisadores de universidades como ferramenta para coleta de dados diversos para pesquisas.
- **MCS IoT framework** é um framework para sensoriamento coletivo em uma Cidade Inteligente proposto por ??). O framework possibilita a sua integração com uma arquitetura oneM2M¹. Além disso, ele permite a utilizar diferentes tipos de sensores e atuadores que possam ser conectados à Internet, gateways Machine to Machine (M2M) e também ambientes de nuvem computacional para oferecer novos serviços aos usuários. Os autores demonstram o potencial do framework no desenvolvimento de uma aplicação de sensoriamento sensível ao contexto, que ajusta seu

¹ oneM2M é uma proposta de padronização para comunicação M2M e Internet das Coisas - <http://www.onem2m.org/>

comportamento com vistas a poupar a bateria do smartphone do usuário. Apesar de citar o desenvolvimento dessa aplicação o trabalho não apresenta qualquer tipo de análise do framework que demonstre resultados.

- **SD-IoT** proposto por ??) propõe um arquitetura IoT baseada em software para o Sensoriamento Urbano. O interessante dessa arquitetura é que ela desacopla as aplicações da infraestrutura física com o uso de controladores centralizados que gerenciam os dispositivos através de uma interface padrão. Com isso, diversas aplicações de sensoriamento podem ser executadas ao mesmo tempo, compartilhando a infraestrutura. Outro diferencial da arquitetura é o uso massivo de APIs para ocultar a complexidade e a heterogeneidade dos dispositivos usados, reduzindo assim o tempo para desenvolver novas aplicações e disponibiliza-las para o uso. A arquitetura é composta de três camadas (Physical Infrastructure Layer, Control Layer e Application Layer) que possibilitam a flexibilidade usar diferentes dispositivos, redes de comunicação e ambientes de processamento baseados em nuvem. Os autores efetuaram testes com 5 aplicações de sensoriamento (rua, tempo, ruído, ambiente e poeira) que demonstraram que a proposta é funcional, consegue compartilhar sensores entre aplicações e otimiza o uso de recursos.
- **SmarterSantander**: É uma infraestrutura proposta por ??) para implantação de sensoriamento em cidades de forma participativa. Os autores chamam essa abordagem de *Participatory Sensor Networks*. Para isso, a proposta faz o uso do framework Dynamix ??), que é executado em segundo plano nos dispositivos móveis dos usuários, para proporcionar o sensoriamento e a comunicação. O foco da infraestrutura é possibilitar um auto deploy de experimentos em larga escalas entre nós móveis e dentro da infraestrutura de Internet das Coisas (*Internet of Things (IoT)*). Devido a esse caráter genérico não considera questões de otimizações do uso dos recursos como apresentados por outros trabalhos de Sensoriamento Urbano. Para avaliação da plataforma, foram utilizadas duas aplicações durante 7 dias por 30 voluntários. Uma das aplicações tinha por finalidade detectar a quantidade de redes sem fios disponíveis ao longo da cidade e a segunda fazia a coleta do nível de ruído em locais movimentados da cidade.

3.1.1 Análise das iniciativas de Sensoriamento Urbano

Com base nas iniciativas de Sensoriamento Urbano apresentadas anteriormente, foi definido um conjunto de características que as definem:

- **Classe:** indica se a iniciativa é para sensoriamento participativo ou oportunista;
- **SubClasse:** indica se ela é utilizada para sensoriamento pessoal, social ou público;
- **Arquitetura:** apresenta qual o tipo de arquitetura empregada na modelagem da iniciativa (exemplo: cliente/servidor, SOA, ...);
- **Dispositivos:** demonstra quais tipos de dispositivos são empregados no sensoriamento, como por exemplo smartphones, sensores fixos ou móveis;
- **Cenário de validação:** indica em qual tipo de cenário a iniciativa foi avaliada. É considerado como *real* quando foram efetuados testes em ambientes reais, com usuários voluntários; *controlado* quando foi aplicado em um ambiente real, porém com usuários conhecidos e que mantinham contato com os autores da iniciativa e; *simulado* quando foram efetuadas análises com resultados de simulações;
- **Paradigma de comunicação:** define o paradigma de comunicação utilizado entre os dispositivos. Ela pode ser *padrão* quando se utiliza a comunicação baseada na pilha TCP/IP; ou *alternativa* quando emprega comunicação não baseada na pilha TCP/IP como por exemplo Redes Oportunistas;
- **Emprega contexto:** define se a iniciativa utiliza dados de contexto para alguma finalidade, como adaptação;
- **Usa redes sociais:** indica se o sensoriamento faz uso de algum tipo de dado de redes sociais;
- **Segurança:** indica se aspectos de segurança foram considerados na implementação da solução;
- **Proteção a privacidade:** indica se foram considerados conceitos de privacidade, como por exemplo, anonimato dos dados coletados;
- **Tolerância a falha:** indica se algum conceito de tolerância a falhas foi considerado pela iniciativa;

- **Tarefas de sensoriamento:** indica se as tarefas de sensoriamento são predefinidas em tempo de desenvolvimento ou se a iniciativa é flexível para permitir que elas sejam descritas dinamicamente, em tempo de execução, através de regras, códigos ou descritores de tarefas;
- **Adaptação:** indica se a iniciativa executa algum tipo de adaptação de forma autônoma.

A partir de tais características, foi estruturada a tabela 3.1 que resume cada iniciativa:

A partir das informações da tabela pode-se perceber que a maior parte das iniciativas apresenta suporte tanto ao sensoriamento participativo quanto ao oportunista. As exceções são o *Medusa* que somente proporciona sensoriamento participativo e o *MetroSense* e o *CenceMe* com somente sensoriamento oportunista. Com exceção do *CenceMe* que é voltado para o sensoriamento pessoal, todas as iniciativas podem ser empregadas no sensoriamento social.

Também pode ser observado que algumas propostas empregam somente smartphones, não executando as atividades de sensoriamento através de sensores fixos e móveis (como por exemplo, carregados por automóveis). Os únicos trabalhos que consideram tal possibilidade são o *MetroSense* e o *SmarterSantander*. Nem todas as iniciativas consideram dados de redes sociais para melhorar o sensoriamento.

Quanto ao cenário de validação, pode-se perceber que o *MobiSens*, *SenseMyCity* e o *SD-IoT* foram validados em cenário reais, com resultados interessantes. Com exceção do *MetroSense* que rodou em um simulador, as demais iniciativas rodaram em ambientes reais controlados, onde usuários conhecidos executaram as aplicações por alguns dias e então foram coletados os resultados para análises.

A maior parte das iniciativas de sensoriamento permite que as tarefas de sensoriamento sejam definidas de forma remota e sejam encaminhadas para os dispositivos a realizar em tempo de execução. As estratégias para isso são variadas, mas basicamente se resumem a aplicação baixar códigos remotos para o nó local ou, as vezes, mais complexas como a do *MetroSense* que efetua a migração de máquinas virtuais específicas da aplicação (ASVMs) para os nós de sensoriamento. Somente o *MobiSens* e o *CenceMe* que executam as tarefas definidas em tempo de desenvolvimento. Isso acaba por tornar tais iniciativas pouco flexíveis para execução de tarefas de sensoriamento.

No que se refere ao paradigma de comunicação utilizado, pode-se constatar que *Medusa*, *PRISM*, *MobiSens*, *Pogo*, *SenseMyCity* e o *MCS IoT Framework* transmitem os

Tabela 3.1: Comparativo entre as iniciativas de Sensoriamento Urbano

	AnonySense	Medusa	PRISM	MobiSens	Pogo	MetroSense	SmarterSantar	CenceMe	SenseMyCity	MCS IoT framework	SD-IoT
Classe	Oportunista e Participativa	Participativo	Oportunista e Participativa Individual, Comunitária/Social e Pública	Oportunista e Participativa	Oportunista e Participativa	Oportunista	Oportunista e Participativa	Oportunista	Oportunista e Participativa	Oportunista e Participativa	Oportunista e Participativa
Sub-Classe	Social/Comunitária e Pública	Social e Público	Hierárquica (Cliente/servidor e Mestre/escravo)	Pessoal, social e público	Pessoal, social e público	Pessoal, Social e Pública	Pessoal, Social e Pública	Pessoal	Pessoal, Social e Pública	Pessoal, Social e Pública	Pessoal, Social e Pública
Arquitetura	Mestre/escravo	Mestre/escravo	Hierárquica (Cliente/servidor e Mestre/escravo)	Cliente/Servidor	Publish/subscribe	Hierárquico (Mestre/escravo)	Cliente/Servidor (com Web-Services) e Mestre/escravo	Cliente/servidor	Cliente/servidor	Cliente/Servidor com uso massivo de APIs	Cliente/Servidor com uso massivo de APIs
Dispositivos	Sensores móveis (smartphones)	Sensores móveis (smartphones)	Sensores móveis (com windows Mobile)	Sensores móveis (smartphones)	Sensores móveis (smartphones)	Sensores fixos e móveis	Sensores fixos e móveis	Smartphones com Symbian	Sensores fixos e móveis	Sensores fixos e móveis	Sensores fixos e móveis
Cenário de validação	Controlado	Controlado	Controlado	Real	Controlado	Simulado	Controlado	Controlado	Real	Protótipo de uma aplicação	Protótipo de 5 aplicações
Tarefas de sensoriamento	Dinamicamente de forma remota	Dinamicamente de forma remota (linguagem Medusa-ML)	Dinamicamente de forma remota	Predefinidas em tempo de desenvolvimento (com framework proposto)	Dinamicamente de forma remota (com JavaScript)	Dinamicamente de forma remota	Dinamicamente de forma remota (binários em Java)	Predefinidas em tempo de desenvolvimento	Predefinidas em tempo de desenvolvimento	Predefinidas em tempo de desenvolvimento	Predefinidas em tempo de desenvolvimento
Segurança Privacidade	✓	✓	✓	×	×	✓	✓	✓	✓	×	×
Tolerância a falha	×	✓	×	⊗	⊗	⊗	×	✓	×	⊗	×
Paradigma de comunicação	DTN	TCP/IP via HTTPS e push states via SMS	TCP/IP via GPRS/EDGE/3g e Wi-Fi	TCP/IP via Internet	TCP/IP via 2G e 3G	Adaptive Multihop approach to communication (DTN)	TCP/IP (DTN futuramente)	TCP/IP via Internet	TCP e UDP	TCP/IP e "non-IP" com o uso de gateways	Independe da aplicação
Emprega contexto	×	⊗	✓	✓	✓	✓	✓	⊗	×	⊗	×
Usa redes Sociais	×	×	×	✓	×	×	×	✓	×	✓	×
Adaptação	×	⊗	✓	✓	×	⊗	⊗	⊗	×	✓	×

*Legenda: ✓= sim ✗= não ⊗= parcial

(Fonte: o autor)

dados de sensoriamento através da Internet ou utilizando a pilha TCP/IP. O *AnonySense* e o *MetroSense* utilizam DTN para transmitir dados mesmo em situações de inexistência de infraestrutura. O *SmarterSantander* coloca o suporte a DTN como trabalho futuro. O *MCS IoT Framework* possui o diferencial de utilizar gateways para permitir a comunicação da infraestrutura TCP/IP com equipamentos que utilizam protocolos para M2M. Já o *SD-IoT* têm uma interessante proposta de permitir que diferentes tipos de arquiteturas de redes sejam utilizadas sem que a aplicação tome conhecimento de tal fato. Isso é possível devido ao uso massivo de APIs para troca de informações. Essa proposta é interessante e pode ser melhor explorada por outras iniciativas.

Com exceção da *AnonySense* todas as iniciativas utilizam dados de contexto. O impacto disso pode ser notado no critério de adaptação: as iniciativas que atendem mesmo que parcialmente esse critério empregam algum dado de contexto. Com exceção do *PRISM* e *MetroSense* que atendem perfeitamente o critério, as demais como no caso a *AnonySense*, *Pogo*, *SenseMyCity* e *SD-IoT*, executam apenas adaptações parciais ou nenhuma adaptação.

Assim, a partir dessa análise, pode-se constatar que o uso de sensores móveis (como smartphones ou carregados por veículos) em conjunto com sensores fixos é uma forma viável de efetuar o sensoriamento social e proporcionar a melhor compreensão das dinâmicas de uma cidade e por consequência fornecer serviços ubíquos que melhor atendam as necessidades dos cidadãos (??). Também pode-se observar que as iniciativas que possuem algum tipo de adaptação utilizam dados de contexto. Acredita-se que se esses dados fossem explorados para garantir a proatividade nas tomadas de decisões ocorreria uma melhora nas funcionalidades da aplicação.

Além disso, nota-se que as aplicações de sensoriamento são capazes de enviar dados através de rede celulares (GPRS/Edge,3G/4G) e redes WiFi. Entretanto, essa abordagem não pode ser aplicada em cenários onde o envio de dados através dessas redes tem um valor elevado ou ainda onde a cobertura da rede é intermitente ou inexistente (??). Algumas iniciativas estudadas já suportam esse tipo de cenário, empregando o paradigma de comunicação baseado em DTNs.

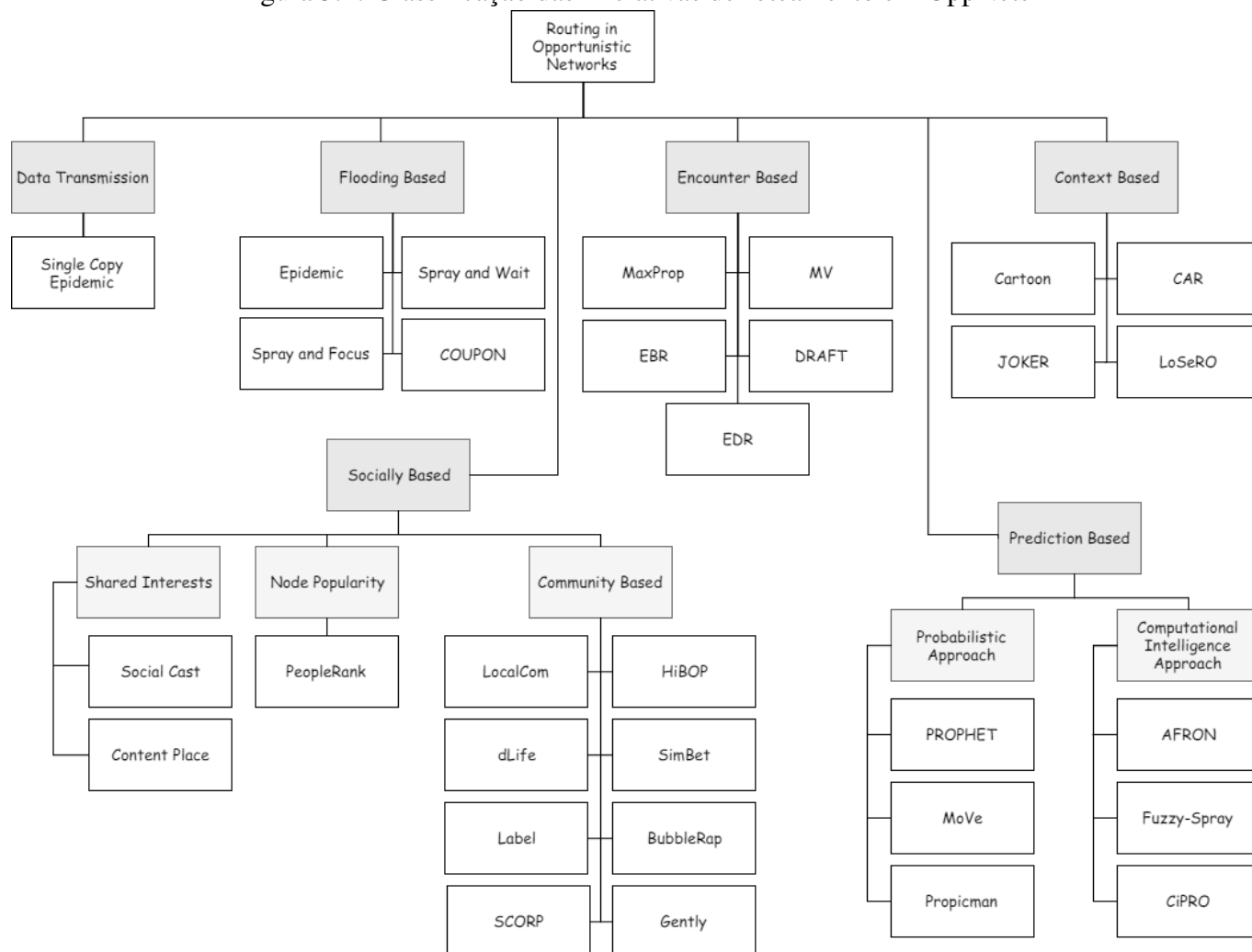
Dessa forma, pode-se concluir que: (i) existe a necessidade de arquitetura de sensoriamento que envolva todo o processo de sensoriamento urbano e seja capaz de contemplar de forma satisfatória as características apresentadas; (ii) o uso de dados de contexto para adaptação e também de DTN são viáveis de serem empregados em aplicações de sensoriamento. Essa conclusão serve de base para demonstrar que o componente de software

que está sendo proposto na presente tese têm aplicabilidade.

3.2 Estado da arte em Redes Oportunistas

Para tornar mais fácil a compreensão, as iniciativas em Redes Oportunistas foram agrupadas conforme a figura 3.1. Essa organização é baseada nas classificações propostas por ?????????????):

Figura 3.1: Classificação das iniciativas de roteamento em OppNets



(Fonte: o autor)

- **Direct Transmission:** Nessa categoria de protocolos, o nodo origem gera a mensagem e a armazena em buffer até que encontre o destino. Como exemplo pode ser citado:

→ **Single Copy:** é uma estratégia onde uma única cópia da mensagem é encaminhada da origem ao destino (??). Dessa forma, uma mensagem somente é enca-

minhada ao destinatário quando o mesmo é diretamente encontrado. Os autores apontaram em estudos posteriores que essa estratégia não garante boa taxa de entrega e possui atraso muito alto (??).

- **Flooding Based:** nessa categoria, os nodos geram múltiplas cópias da mensagem na rede. A replicação da mensagem ocorre até que o destino receba uma dessas cópias.

→ **Epidemic:** proposto por ??) é considerado o primeiro protocolo de roteamento para redes com frequentes desconexões e conectividade intermitente. Ele assume que o nodo origem não possui conhecimento sobre a localização do destino e nem mesmo sabe o melhor caminho para alcançá-lo. A mobilidade dos nodos é utilizada como uma forma de possibilitar que os nodos se encontrem aleatoriamente e possam efetuar a troca de mensagens. Quando dois nodos se encontram eles trocam uma lista de mensagens que cada um possui em seu buffer local. Depois que cada nodo identifica quais mensagens que o nodo vizinho possui e ele não, é solicitado o recebimento dessas mensagens que são então armazenadas no buffer local. Esse processo se repete cada vez que dois nodos se encontram de forma que rapidamente várias cópias de uma mesma mensagem existem na rede e maior é a probabilidade da mensagem ser entregue ao destino com pouco atraso.

O protocolo Epidemic é capaz de entregar praticamente toda as mensagens transmitidas quando for utilizado um buffer ilimitado. Como pode-se perceber o maior problema dessa abordagem se refere ao grande número de réplicas de mensagem que trafegam na rede, consumindo recursos de comunicação e também de armazenamento (buffer). Dessa forma, ele não é escalável quando o número de mensagens trocadas entre os nodos cresce.

→ **Spray and Wait:** foi proposto por ??) como uma forma de controlar a quantidade de múltiplas mensagens na rede. Ele faz uma mescla das características do Epidemic e da categoria de entrega direta (Direct Transmission). Ele é constituído por duas fases: na fase de *Spray*, L cópias de uma mensagem são espalhadas de um nodo origem para L nodos distintos. Na fase de *Wait* cada nodo que possui uma cópia da mensagem faz a entrega da mesma. Os autores propuseram algumas variações dessa estratégia, que diferem entre si em como as L cópias de mensagens devem ser espalhadas. A mais eficiente é a chamada de *Binary Spray*, onde um nodo A com $L > 1$ cópias de uma mensagem fornecem $L / 2$ cópias da mensagem a cada nodo que encontra.

Experimentos conduzidos pelos autores demonstraram que o Spray and Wait possui menor custo computacional que o Epidemic uma vez que o número de mensagens encaminhadas é limitado a L , porém em alguns cenários a latência de entrega das mensagens é muito grande.

→ **Spray and Focus:** é uma variante do Spray and Wait proposta por ??). Ele também possui duas fases: a fase de Spray ocorre quando uma mensagem é criada em um nodo. Nesse momento também são criados *tokens* de encaminhamento (forwarding tokens) para essa mensagem. Um token de encaminhamento implica que um nodo pode gerar e encaminhar uma cópia adicional da mensagem. Quando o nodo que recebeu a mensagem possui somente um token de encaminhamento da mesma ele muda para a fase de Focus. Na fase de Focus uma mensagem pode ser encaminhada para outros nodos de acordo com critérios de encaminhamento. No caso, o critério adotado é baseado em temporizadores que indicam o tempo decorrido desde o último encontro entre os dois nodos.

O Spray and Focus demonstrou ser uma alternativa melhor que o Spray and Wait e algumas outras alternativas baseadas em flooding. Porém, como sua performance depende dos tempos de encontro, ele pode ter seus resultados afetados negativamente em cenários onde os nodos possuem alta mobilidade, uma vez que os temporizadores de encontros podem se tornar obsoletos (??).

→ **COUPON:** o Cooperative Framework for Building Sensing Maps in Mobile Opportunistic Networks (????) é uma iniciativa proposta para a construção de mapas de sensoriamento através de encaminhamento cooperativo. O COUPON explora o paradigma de DTN para o encaminhamento de mensagens com o uso de variantes cooperativas de protocolos de inundação (flood based). Os autores propõem variantes do protocolo Epidemic e do Spray and Wait com o uso de fusão de dados. Dessa forma, conseguiu-se reduzir em 93% o número de amostras de dados coletados. Além disso, conseguiu-se reduzir o overhead de transmissão em 78% quando comparado com o algoritmo de roteamento epidêmico tradicional. A característica de fusão de dados mensura a “qualidade” do dado, agrupando aqueles que sejam correlacionados nos domínios do tempo e espaço, pode ser uma boa alternativa para ser empregada por outras iniciativas. Uma carência dessa abordagem de fusão cooperativa irá ocorrer quando nodos não efetuarem coleta de dados em locais/tempos próximos. Nesse caso, os dados não serão agregados e o protocolo passa a ter seu comportamento normal, baseado em inundação exaurindo

recursos dos nodos. Para isso, como trabalhos futuros, os autores propõem que novos experimentos devem ser efetuados para considerar outros tipos de protocolos que utilizem aspectos sociais e dados de contexto.

- **Encounter Based:** em tal categoria, os nodos escolhem a quem encaminhar a mensagem com base na frequência de contato entre os nodos ou no tempo transcorrido desde o último encontro.

→ **MaxProp:** foi proposto por (??). É baseado na priorização da mensagens que devem ser transmitidas aos outros nodos e de quais devem ser removidas quando o buffer está praticamente cheio. O agendamento de transmissão e deleção são baseados nos prováveis caminhos entre os nodos de acordo com dados históricos. Além disso, reconhecimento de recepção da mensagem e lista de intermediários anteriores são alguns dos mecanismos utilizados pela abordagem. Os estudos dos autores em um ambiente de uma frota de ônibus chamado de UMassDieselNet, demonstraram que o MaxProp possui performance superior a alguns outros protocolos baseados em oráculo. Segundo (??) o MaProp possui alto custo computacional devido a quantidade de informações de estado que devem ser armazenadas e processadas.

→ **RAPID:** O Resource Allocation Protocol for Intentional DTN routing (??) é um protocolo que trata o roteamento de pacotes como um problema de alocação de recursos onde as métricas de roteamento são determinadas de acordo com funções de utilidade que buscam intencionalmente minimizar métricas de rede, como atraso médio/máximo e perda de pacotes. No RAPID, cada mensagem carrega metadados que são trocados entre os encontros para auxiliar no cálculo da função de utilidade e também o resultado da função de utilidade que é calculada de acordo com a métrica que se quer minimizar. As mensagens são replicadas em uma ordem descendente de acordo com resultado dessa função de utilidade de forma que aquelas que contribuem mais para minimizar a métrica em questão são replicadas mais vezes na rede. De acordo com os experimentos de (??) a taxa de entrega do RAPID é muito boa, porém em determinadas situações o atraso na entrega é alto.

→ **MV:** o protocolo Meeting and Visits (??) utiliza a frequência de encontro entre os nodos para decisões de roteamento. Os nodos lembram de seus caminhos (visitas) e dos outros nodos que eles encontram durante o percurso. Cada nodo armazena

- variáveis para todos os nodos que foram encontrados descrevendo a probabilidade de sucesso na entrega de mensagem a esses nodos. Quando dois nodos se encontram diretamente, eles definem mutuamente a probabilidade de entrega para 1 e sincronizam as suas variáveis. Essa probabilidade vai sendo degradada a medida que nodos com probabilidades mais altas são encontrados. A decisão de encaminhar uma mensagem ao nodo encontrado depende da probabilidade deste de encontrar novamente o nodo destinatário.
- **EBR:** o Encounter Based Routing (??) é um protocolo que controla de forma fixa a quantidade de cópias de uma mensagens que podem ser propaganda na rede no momento da criação da mensagem. Para as decisões de encaminhamento, o protocolo assume que os nodos que possuem altas taxas de encontros possuem maior potencial de encaminhar uma mensagem ao seu destino. Com base no histórico de encontros passados o nodo pode estimar encontros futuros e então escolher o melhor momento para encaminhar as mensagens. O protocolo simplesmente conta a quantidade de encontros com outros nodos (chamado de *Current Window Counter*) para calcular a taxa de encontros passados (*Encounter Value* - EV). Quanto maior o EV, maior a probabilidade de entregar uma mensagens.
 - **3R:** o trabalho de ??) é a primeira iniciativa que analisa os dados de um cenário de larga-escala baseado em Bluetooth em um campus universitário. Os resultados demonstraram que os padrões de encontro são regulares e podem ser preditos. Com isso foi proposto o protocolo 3R que utiliza a regularidade de contatos granulares entre os nodos para caracterizar os hábitos dos usuários e assim maximizar a probabilidade de entrega das mensagens.
 - **DRAFT:** o Distributed Rise and Fall spatio-Temporal (??) é apontado pelos autores como uma forma de resolver os problemas de clusterização baseados em encontros sociais do BubbleRap. O DRAFT utiliza um método de clusterização de nodos espaço-temporal baseado na dinamicidade de encontros. Para isso, ele combina a clusterização espacial com uma função de decaimento que proporciona a representação de contatos recentes, excluindo nodos que não foram vistos por longo tempo. Os autores demonstraram que o protocolo conseguiu boa performance na entrega de dados, sendo superior ao Prophet e ao BubbleRap.
 - **EDR:** o Encounter and Distance based Routing (??) é um protocolo que utiliza um parâmetro de encaminhamento, chamado *forward parameter*, para determinar a seleção do próximo nodo. Esse parâmetro é calculado de acordo com o número

de encontros e a distância de cada nodo com relação a determinado destino. Para as decisões de encaminhamento é feita uma comparação do parâmetro de encaminhamento com um limiar calculado de forma dinâmica. O cálculo do limiar é feito a partir da média do parâmetro de encaminhamento e objetiva maximizar o número de encontros e a minimizar a distância do destino para um nodo. Quando o limiar for atingido as mensagens são encaminhadas ao nodo encontrado. O protocolo não considera questões energéticas e assume que todos os nodos possuem energia ilimitada.

- **Prediction Based:** os protocolos dessa categoria que fazem algum tipo de predição do comportamento dos nodos para auxiliar na tomada de decisão de encaminhamento. De acordo com o tipo de previsão aplicado eles são subdivididos em *Probabilistic Approach* e *Computational Intelligence Approach*:

- **Probabilistic Approach:** utilizam técnicas baseadas na teoria da probabilidade para predição. Os principais protocolos são:

→ **PROPHET:** no Probabilistic Routing Protocol using History of Encounters and Transitivity (PROPHET) uma métrica chamada de probabilidade de entrega (delivery predictability) é calculada no nodo A para cada destino conhecido B. Essa métrica indica qual a probabilidade do nodo A encontrar um destinatário B. Quando dois nodos se encontram, eles trocam um vetor de informações contendo a probabilidade de entrega para cada um dos nodos com os quais já se encontraram. Esta probabilidade de entrega é determinada pela frequência de contatos que cada possível nodo intermediário têm com o destino final ou com outros nodos que possuem melhores condições de entrega. Essa informação é mutualmente trocada, sendo utilizada para atualizar as probabilidades dos nodos e também determinar se uma mensagem em particular deve ou não ser encaminhada ao nodo encontrado. A decisão de encaminhamento é bastante simples: quando um nodo encontra outro, uma mensagem somente é transferida ao outro se a probabilidade de entrega do nodo encontrado for maior que a do nodo corrente.

No PROPHET, se dois nodos se encontram com frequência, a probabilidade de entrega vai crescer rapidamente. Se dois nodos deixam de se encontrar com frequência a probabilidade cai abruptamente. Se eles voltarem a se encontrar essa probabilidade é restaurada ao valor anterior. Durante esse tempo que os nodos ficaram sem se encontrar o encaminhamento de mensagens não ocorrerá devido

- ao baixo valor da probabilidade de entrega ocasionando atrasos na rede. Para contornar esse fato, ??) propuseram uma versão melhorada do protocolo chamada de Advanced PROPHET que utiliza a média das probabilidades de entrega ao invés da probabilidade de entrega para tomadas de decisão de encaminhamento.
- **MoVe:** o Motion Vector é uma abordagem proposta por ??) que utiliza informações de mobilidade (no caso, a velocidade e a trajetória) de um nodo e seus vizinhos para prever a localização futura de um nodo. Assim como o Epidemic, o MoVe possui considerável custo computacional de processamento e de uso de buffer.
 - **Propicman:** o Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Networks (Propicman) (??) utiliza o perfil para prever a mobilidade dos nodos e descrever o ambiente social do usuário e seus relacionamentos. O protocolo considera que os usuários possuem padrões de comportamento que se repetem com frequência (dia, semana e mês). Dessa forma, assume-se a premissa que se um nodo visitou um local diversas vezes, provavelmente ele vai visitar tal local futuramente. Com base nisso e nos perfis do próprio nodo e de seus vizinhos, informações relevantes podem ser deduzidas, como por exemplo a probabilidade de entrega de cada nodo ou do mesmo se mover em direção ao destinatário. No processo de encaminhamento, uma mensagem contendo informações de contexto conhecidas do destinatário é trocada entre os nodos. Com base nessas informações e nos dados de perfil de cada nodo a probabilidade de entrega de cada um é calculada. O nodo que possui maior probabilidade de entrega será selecionado para o encaminhamento. Apesar de possuir baixo custo computacional para ser implementado, os autores perceberam que algumas aplicações que utilizam dispositivos móveis, como sensores, sofriam com a quantidade de mensagens trocadas entre os nodos. Foi então proposta uma variação do Propicman chamada de PROSAN (??). As principais diferenças do Propicman em relação ao PROSAN é que enquanto no primeiro os nodos intermediários que foram utilizados para o encaminhamento guardam cópias das mensagens recebidas e continuam as repassando para outros nodos que encontrem, enquanto no segundo os nodos intermediários armazenam as mensagens, porém somente as encaminham quando encontram os seus destinatários. Outra diferença refere-se às informações de perfil trocadas entre os nodos. No PROSAN são enviadas informações referentes ao tipo de aplicação que gerou a mensagem e ela é calculada para o cálculo da probabilidade. Assim,

somente serão escolhidos nodos que pertençam a mesma “comunidade” (mesmo tipo de aplicação).

- **SPRINT:** O Social PRedIction-based routing in opportunistic NeTworks (SPRINT) foi proposto por ??) como uma iniciativa que combina predição de encontro entre os nodos e informações sociais para tomar decisões de encaminhamento de mensagens. A ideia explorada pelo trabalho é que os nodos interagem com aqueles nodos que já foram encontrados antes ou ao qual possuem maior conexão social. Isso proporciona que o histórico de encontros entre os nodos possa ser utilizado para prever encontros futuros. Os encontros preditos são utilizadas juntamente com funções de utilidade calculadas com base em informações de redes sociais, como Facebook, e irão determinar se determinado nodo possui potencial de encontrar determinada comunidade e então entregar a mensagem ao destino. Segundo os autores, a taxa de entrega de mensagens do SPRINT é superior ao BubbleRap em 3% em determinados cenários. A abordagem utilizada pelo SPRINT para entregar mensagens não somente para os nodos com alta centralidade juntamente com a priorização de quais mensagens devem ser entregue a cada encontro é interessante. Porém, não é feito nenhum controle ou adaptação no gerenciamento do buffer dos nodos. Como resultado, os autores apontam que em cenários com muitas mensagens de pequeno tamanho ocorre descarte indevido de mensagens e a taxa de entrega não é satisfatória.

- **Computational Intelligence Approach:** essa categoria é composta por protocolos que empregam alguma técnica de inteligência computacional para a predição. Principais representantes:

- **AFRON:** o Adaptive Fuzzy Routing in Opportunistic Network ??) é um protocolo que utiliza Lógica Fuzzy para priorizar as mensagens que estão armazenadas no buffer local. Os autores propoem que o AFRON pode reduzir o consumo energético e de recursos computacionais. Para definição das funções de pertinência são utilizados como entrada os parâmetros das mensagens como contador de transmissão, tamanho da mensagem e TTL. A saída do processo de inferência fuzzy é a prioridade que a mensagem possui para ser transmitida. Os esquema de funcionamento é bastante simples e os resultados demonstraram que o AFRON possui boa performance em cenários onde nos nodos possuem movimentação aleatória. Apesar disso, em tais cenários, o seu comportamento é similar ao Epide-

mic.

- **Fuzzy-Spray:** foi proposto por ??) como uma abordagem que utiliza replicação de mensagens em conjunto com Lógica Fuzzy para selecionar quais mensagens devem deixar o buffer local. Ele utiliza dois parâmetros, chamados de contador de encaminhamentos da mensagem (forward transmission count) e o tamanho da mensagem, com entrada para funções de pertinência usadas no processo de inferência fuzzy que irá priorizar a ordem que as mensagens devem sair do buffer. Quando um nodo encontra o outro, o Fuzzy-Spray envia o máximo possível de mensagens que possuam mais alta prioridade. Os resultados dos experimentos demonstraram que o Fuzzy-Spray consegue entregar com pouco atraso boa quantidade de mensagens, porém apesar de controlar a quantidade de réplicas de uma mensagem espalhada na rede, ele sofre de problemas de saturação.
- **CiPRO:** o Context Information Prediction for Routing in OppNets (??) é uma proposta construída com base no Propicman, apresentado anteriormente. Ele utiliza as interações sociais diárias para classificar os tipos de contato e uma rede neural MLP para prever a probabilidade de encontros futuros entre emissor e destinatário. Quando dois nodos se encontram com frequência é utilizada a abordagem do CiPRO. Nesse caso, são armazenadas informações sobre a hora e local do encontro bem como informações que contexto dos nodos (nome da operadora, nacionalidade, endereço, nível da bateria, memória, etc). Tais dados são agrupados em perfis que são utilizados para o cálculo da probabilidade de encontro em períodos de tempo específico. Com isso, o emissor possui conhecimento de quando e onde ele possui alta probabilidade de entrega devendo assim, iniciar o processo de roteamento com vistas a minimizar o atraso e não sobrecarregar a rede. Se dois nodos não se encontram com frequência é utilizada a abordagem original do Propicman.
- **Context Based:** essa categoria é formada pelos protocolos que exploram os diferentes tipos de informações de contexto do ambiente ou dos nodos para tomar decisões sobre o melhor nodo a carregar as mensagens.
 - **CARTOON:** o Context-Aware Routing Protocol for Opportunistic Networks ??) possui a capacidade de detectar e classificar o contexto no qual o nodo está inserido baseado em um conhecimento prévio e então determina qual a ação a ser tomada com vistas a melhorar o desempenho de alguma métrica como taxa de

- entrega, a latência, a quantidade de transmissões, etc. O CARTOON utiliza diversas variáveis que são atualizadas dinamicamente a cada contato ou transmissão. Com base em seus componentes adaptativos, tais como o cálculo dinâmico de densidade, percentual de ocupação, envelhecimento de rotas, etc., o CARTOON determina qual o melhor modo de disseminação. Inicialmente as mensagens são entregues em modo probabilístico, se os seus componentes adaptativos determinarem como viável, o encaminhamento de mensagens passa a ocorrer de modo epidêmico. Os resultados demonstraram que o CARTOON possui a sua performance prejudicada em cenário com poucos nodos e onde devido a movimentação dos nodos são formadas de regiões desconexas.
- **CAR:** o Context-Aware Routing protocol (??) faz o uso de filtros de Kalman para prever se um destinatário pertence ao mesmo segmento de rede do remetente (ou do nodo intermediário). Ele utiliza a quantidade de vizinhos e o nível de energia como informação de contexto para a predição e a tomada de decisão de roteamento. Se o destinatário está conectado no mesmo segmento de rede, a mensagem é encaminhada usando DSDV (*ad hoc routing*); senão, ele emprega uma função de utilidade baseada na taxa de conectividade do nodo e na probabilidade do destinatário estar no mesmo *cluster* do nodo que será utilizado como intermediário para o encaminhamento.
 - **JOKER:** o auto-adJustable Opportunistic acKnowledgegment/timEr-based Routing (??) é um protocolo que considera a seleção de bons nodos intermediários e a coordenação dos nodos para maximizar a entrega de dados multimídia em tempo real e ao mesmo tempo reduzir o consumo energético. Como base de estratégia de roteamento, o JOKER utiliza uma métrica que considera as condições do link para entrega das mensagens e também a distância percorrida até o destino final. O JOKER utiliza um cabeçalho específico em cada mensagem que contém o endereço MAC dos potenciais nodos capazes de carregar a mensagem de forma adequada entre um total de candidatos configurados como um parâmetro do protocolo. Além disso, o destinatário final da mensagem é incluído em um campo específico do cabeçalho com o intuito de agilizar o processamento, reduzindo assim atrasos de transmissão e evitando descarte de pacotes válidos. Os resultados dos experimentos demonstraram que o JOKER possui boa capacidade de entrega para dados multimídia em tempo real como por exemplo IPTV. Apesar dos autores considerarem a iniciativa como oportunista, a solução é baseada no protocolo

IP, sendo necessário pelo menos um caminho fim-a-fim entre a origem e o destino.

→ **LoSeRO:** O Locality Sensitive Routing Protocol in Opportunistic Networks (??) é um protocolo que utiliza a trajetória dos nodos para gerar conhecimento a respeito dos usuários. Ele emprega os locais mais visitados como base para efetuar a entrega de mensagens utilizando multicast entre os nodos próximos. O protocolo cria um perfil de mobilidade dos nodos com base em zonas pré-definidas. É criado então um vetor chamado MobyZone contendo as n zonas mais visitadas, se uma mensagem pertence a uma determinada zona e essa zona está no vetor de um nodo, ele é definido como um candidato para carregar a mensagem até o destino. Os autores efetuaram experimentos com um número limitado e ilimitado de nodos utilizados como intermediários. Os resultados demonstraram que quando utilizado um número ilimitado de intermediários a iniciativa escala bem e possui bons resultados. Entretanto, apesar do protocolo necessitar que os nodos frequentem constantemente uma localidade para aumentar a entrega, não foram efetuados experimentos que demonstrem o que acontece quando os nodos não repetem o padrão de mobilidade, fato esse que pode impactar negativamente nos resultados.

- **Socially Based:** nessa categoria, os protocolos utilizam conceitos de redes sociais para decisões de encaminhamento. As conexões sociais podem ser organizadas em comunidades (Community Based), por interesses dos membros (Shared Interests) ou por popularidade dos nodos (Node Popularity).

- **Community Based:** partem do pressuposto de as pessoas que carregam os dispositivos se organizam em algum tipo de comunidade e isso pode ser explorado para o encaminhamento de mensagens. Segundo ??) esta abordagem sofre de overhead na detecção das comunidades.

→ **HiBOp:** a ideia principal do HiBOp (??) é procurar por nodos que possuam similaridade com atributos de contexto do destinatário. O protocolo assume que uma alta similaridade significa alta probabilidade do nodo entregar a mensagem na comunidade onde encontra-se o destinatário. Um nodo que utilize o HiBOp assinala no cabeçalho da mensagem atributos que o destinatário deve possuir. Qualquer nodo no caminho entre a origem e o destino têm seus atributos verificados. Se o nodo encontrado possui maior similaridade então a mensagem é encaminhada. Esse processo se repete até o destinatário ser alcançado. Os resul-

tados demonstraram que o protocolo reduziu o consumo de recursos e melhorou significativamente a perda de mensagens.

- **SimBet**: o SimBet (??) é um protocolo baseado em interações sociais e utiliza métricas de centralidade e similaridade para definir a probabilidade de um nodo encontrar o destinatário de uma mensagem. Ele assume que nodos com alta centralidade transitam entre comunidades e estes, quando encontram nodos com alta similaridade têm maior probabilidade de possuir um vizinho em comum com o destino. O conceito de centralidade é utilizado para troca de mensagens entre comunidade e o de similaridade para entregar mensagens dentro de uma comunidade. Segundo os autores, ele possui performance similar ao Epidemic com overhead reduzido. Porém, (??) aponta que essa abordagem sobrecarrega os nodos com maior centralidade.
- **Label**: proposto por (??), o Label [29] foi a primeira proposta de uso de conceitos de redes sociais em roteamento oportunista, indicando que a exploração de similaridades sociais possui potencial de melhorar a taxa de entrega e otimizar o custo (quantidade de mensagens total e duplicadas). Experimentos foram desenvolvidos etiquetando os nodos de acordo com a sua filiação e grupo de interesses. Com isso foi possível entregar as mensagens diretamente ao destino, ou próximo deste, através do encaminhamento dentro do mesmo grupo (com a mesma etiqueta). A taxa de entrega do Label é diretamente relacionada com o TTL das mensagens: com TTL curto a taxa é baixa e degrada se não for encontrado um valor adequado. A razão para isso é que o protocolo encaminha uma mensagens somente para nodos que pertençam a uma mesma comunidade do destino. Dessa forma, se nunca for encontrado um nodo da mesma comunidade a entrega irá falhar.
- **BubbleRap**: o BubbleRap (??) é um protocolo que utiliza métricas sociais para decidir a qual nodo encaminhar uma mensagem. Ele utiliza um método distribuído para detectar comunidades e então, uma vez que cada nodo sabe a que comunidade pertence, são utilizadas métricas de centralidade (como ranqueamento) para encaminhar as mensagens àqueles que possuam maior ranking.
- **LocalCom**: o LocalCom (??) é um protocolo epidêmico baseado em comunidades. Ele utiliza um algoritmo distribuído que, baseado somente em informações locais como frequência de encontros, duração do encontro e período entre encontros, constrói um grafo para detecção de comunidades fortemente ligadas. A replicação de mensagens acontece dentro da comunidade se o nodo origem e destino

- estão dentro desta comunidade. Nós considerados “pontes” inter-comunidades (*gateway nodes*) são definidos estática ou dinamicamente com vistas reduzir a redundância e facilitar o encaminhamento de pacotes entre comunidades diferentes.
- **Gently:** o Gently (??) é baseado nos conceitos contexto e redes sociais dos protocolos CAR e LABEL. Ele suporta unicast e multicast de mensagens marcadas com etiquetas para identificar comunidades, de forma que o destino de uma mensagem pode ser um único nó ou um grupo (uma comunidade). O roteamento baseado no protocolo CAR é utilizado para melhorar a eficiência quando a comunidade destino não é detectada. Quando uma mensagem chega a comunidade destino, a estratégia do LABEL é utilizada.
 - **dLife:** O dLife (??) utiliza métricas de similaridade social e considera o dinamismo do comportamento do usuário em atividades cotidianas para construir grafos sociais. Ele acompanha as diferentes interações sociais do usuário em suas atividades diárias com o objetivo de inferir sobre as suas relações sociais em diferentes períodos do dia, identificando assim comunidades. O encaminhamento de mensagens acontece em períodos de tempo considerados com maior conectividade social e, conseqüentemente, maior potencial de entrega de mensagens. Segundo ??), com grande quantidade de contatos o dLife pode demorar para ter uma visão estável da rede em termos de conexões sociais, resultando em replicações inúteis que podem exaurir o buffer de armazenamento e proporcionar que mensagens parem de ser encaminhadas.
 - **SCORP:** O Social-aware Content-based Opportunistic Routing Protocol, (??) é um protocolo que utiliza as interações sociais para medir a proximidade entre os nós e compartilhar dados de interesse comum. Ele trabalha de forma similar ao dLife e ao BubbleRap empregando conceitos de redes sociais e métricas de similaridade, bem como utiliza o comportamento do usuário em atividades cotidianas. Porém, ele emprega as relações sociais entre os usuário para calcular a importância do nó e definir quais tipos de dados são de interesse comum, enquanto o dLife e o BubbleRap utilizam as relações sociais para identificar comunidades e os nós mais populares (com maior centralidade). No SCORP, o encaminhamento de mensagens acontece quando é encontrado um nó que possui maior importância social e compartilhe o interesse pelos mesmos tipos de dados. Uma apontamento importante dos autores é que a escalabilidade do SCORP depende diretamente das configurações de interesse do nó. Em redes direcionadas à

conteúdo (*content-driven networks*), encontros com nodos com muitos interesses podem exaurir os recursos de memória de dispositivos móveis.

- **Shared Interests:** é uma categoria composta por protocolos que assumem que nodos com o mesmo interesse possuem maior potencial de encontro entre si e isso maximiza a entrega de mensagens dentro desses grupos. Ainda conforme Moreira, isso nem sempre é verdade uma vez que um nodo com interesse similar a determinado grupo pode nunca entrar em contato com os nodos pertencentes a tal grupo.

→ **SocialCast:** proposto por ??), o SocialCast utiliza dados de interesse compartilhado entre os nodos para a tomada de decisão de encaminhamento. Foi a primeira iniciativa a utilizar o paradigma Publish/Subscribe, para permitir que os usuários difundam os seus interesses e então o conteúdo é disseminado para os interessados ou então encaminhado para nodos intermediários que possuem maior potencial calculado de acordo com uma função de utilidade que considera determinado interesse. O roteamento ocorre em três etapas: disseminação de interesses, seleção de nodos intermediários e disseminação de mensagens. Na fase de seleção de nodos intermediários, as mesmas métricas de utilidade empregadas no CAR são empregadas através de filtros de Kalman para calcular a similaridade de mobilidade social dos nodos. Segundo ??), ele funciona bem quando todos os membros de cada comunidade estão interessados no mesmo tipo de conteúdo, mas não fica claro como ele funciona em cenários gerais.

→ **ContentPlace:** O ContentPlace (??) considera as informações a respeito das relações sociais para melhorar a disseminação de conteúdo. Ele calcula uma função de utilidade para cada objeto considerando: (i) a probabilidade de acesso a cada objeto e o custo envolvido no seu acesso e; (ii) a conectividade social do usuário em diferentes comunidades que ele interage. A ideia é que sejam disseminados os objetos que maximizem a função de utilidade, escolhendo os objetos que sejam do tipo e do interesse do usuário e que possam ser disseminados nas comunidades que o usuário possui maior relacionamento social. No ContentPlace, o buffer dos nodos são preenchidos com objetos que podem não ser de interesse do usuário, mas são de interesse dos outros nodos da comunidade que o usuário está inserido. A seleção dos objetos é efetuada através de políticas de seleção. Segundo ??) as métricas empregadas nos experimentos do artigo não refletem o dinamismo do comportamento dos usuários.

- **Node Popularity:** os protocolos dessa categoria utilizam a popularidade dos nodos através de ranqueamento para a tomada de decisões de encaminhamento.

→ **PeopleRank:** foi proposto por ??) com base na ideia de ranqueamento de páginas do Google. As informações a respeito das interações sociais dos usuários a partir de redes sociais são usadas para ranquear os nodos em um grafo social (assim como faz o algoritmo de PageRank da Google). As relações sociais que definem as ligações do grafo podem ser baseadas em amizades explícitas, comunicações pessoais ou mesmo interesses comuns. O nodo com maior valor de ranqueamento será o mais central na rede social e é considerado um melhor nodo intermediário. O algoritmo proposto pelo PeopleRank pode ser implementado de uma forma centralizada ou então completamente distribuído.

→ **SREP:** o Social Relationship Enhanced Predictable routing protocol (??) assume que os usuários possuem comportamento determinísticos, visitando com frequência locais e comunidades preferenciais. É utilizado um processo de Markov semi-determinístico para modelar o comportamento de um nodo. Também é utilizado um algoritmo de ranqueamento para especificar o grau de conectividade social dos nodos.

3.2.1 Análise das iniciativas em Redes Oportunistas

A partir das iniciativas em Redes Oportunistas apresentadas foi montada a tabela 3.2. Ela busca sintetizar as classes das iniciativas, o ano de suas propostas, as suas principais características, suas deficiências e cenários de aplicação. Além disso, foram incluídas na tabela critérios que estão diretamente relacionados à hipótese da presente tese, como uso de predição, gerenciamento de situação e proatividade nas tomadas de decisão. Com isso, busca-se demonstrar se alguma das iniciativas analisadas poderia ser utilizada como base para a transmissão de dados do componente que está sendo proposto. Ressalta-se que devido ao tamanho da tabela, as iniciativas anteriores a 2005 não foram incluídas.

As características e critérios definidos são os seguintes:

- **Principal característica:** Define qual o principal diferencial explorado pela iniciativa;

- **Principal deficiência:** Apresenta qual a principal deficiência que a iniciativa possui;
- **Cenário:** Aponta o tipo de cenário para o qual a iniciativa foi concebida;
- **Utiliza predição:** Critério que indica se é empregada alguma técnica para predição;
- **Gerencia buffer:** Critério que indica se a iniciativa efetua o gerenciamento de buffer para evitar descarte de mensagens;
- **Usa contexto:** Critério que indica se dados de contexto são considerados nas tomadas de decisão;
- **Usa situação:** Critério que indica se o conceito de Sensibilidade à Situação é utilizado para alguma finalidade pela iniciativa.
- **Proatividade:** Critério que indica se a iniciativa efetua tomadas de decisão de adaptações de forma proativa.

Com base nas características apresentadas nota-se que somente o COUPON foi pensado para um cenário de sensoriamento em uma Cidade Inteligente. Entretanto, conforme apontado por ??), tais cenários possuem grande heterogeneidade e densidade de dispositivos além de ocorrerem mudanças constantes de topologia devido à não existência de padrões regulares de mobilidade dos nodos. Com isso, existe a necessidade de constantes adaptações e proatividade nas tomadas de decisões. A abordagem de fusão de dados cooperativa do COUPON é interessante, mas ela acaba necessitando que exista correlação espaço-temporal na coleta de dados. Em Cidades Inteligentes essa correlação nem sempre existe. Isso acaba impactando no comportamento da iniciativa, que passa a efetuar o encaminhamento da mesma forma que os protocolos de inundação, podendo assim, exaurir recursos dos nodos.

Além disso, pode-se constatar que nenhuma das iniciativas faz uso de Sensibilidade à Situação conforme está sendo proposto nesse tese. Assim, será proposto um componente de software que aplica Sensibilidade à Situação em Redes Oportunistas como forma de proporcionar proatividade nas tomadas de decisão de encaminhamento e assim maximizar a troca de dados em cenários de Sensoriamento Urbano.

Tabela 3.2: Comparativo entre iniciativas em Redes Oportunistas

Protocolo	Classe	Ano	Principal característica	Principal deficiência	Cenário	Utiliza predição	Gerencia buffer	Usa contexto	Usa situação	Produtividade
Spray and Focus	Flooding Based	2008	Uso de temporizadores na fase de Focus	Depende dos tempos de encontro entre nodos	Cenários com pouca mobilidade	×	×	×	×	×
COUPON	Flooding Based	2014	Fusão de dados usando correlação espaço-temporal	Se não existir correlação pode levar ao uso excessivo de recursos dos nodos	Cenários com correlação espaço-temporal dos nodos	×	×	×	×	×
MaxProp	Encounter Based	2006	Priorização de pacotes no buffer	Custo computacional elevado	Redes veiculares	×	✓	✳	×	×
RAPID	Encounter Based	2009	Uso de uma função de utilidade para controlar a replicação das mensagens	Atraso na entrega em determinadas situações	Redes com alta mobilidade	×	×	×	×	×
3R	Encounter Based	2011	Caracteriza os hábitos dos usuários	Exige longo tempo até iniciar o funcionamento (warm up time)	Redes Sociais com padrões regulares	×	×	×	×	×
EBR	Encounter Based	2009	Utiliza histórico de encontros para estimar encontros futuros	Alta taxa de atraso	Redes com alta mobilidade	✓	×	×	×	×
DRAFT	Encounter Based	2013	Utiliza um algoritmo que considera o tempo de contato para formação de clusters	Sofre quando não existem contatos frequentes entre os mesmos nodos	Redes com padrões similares	×	×	×	×	×
EDR	Encounter Based	2016	Utiliza o número de encontros e a distância de cada nodo com relação a um destino para a tomada de decisões	Não considera questões de energia ou outros aspectos que possam interferir no desempenho	Redes com alta mobilidade	×	×	×	×	×
Propicman	Probabilistic Approach	2007	Utiliza histórico de locais visitados	Grande quantidade de mensagens trocada entre os nodos	Mobilidade regular e padrões de visitas	✓	×	✳	×	×
SPRINT	Probabilistic Approach	2013	Combina predição de encontro e informações sociais para tomar decisões	Baixa performance em cenário com grande quantidade de pequenas mensagens	Redes sociais ou com contatos repetitivos	✓	×	✳	×	×
AFRON	Computational Intelligence	2012	Uso de Lógica Fuzzy	Uso de buffer semelhante ao Epidemic	Movimentação aleatória	×	✓	✳	×	×
Fuzzy-Spray	Computational Intelligence	2009	Priorização de mensagens no buffer	Sofre com saturação da rede	Movimentação aleatória	×	✓	✳	×	×
CiPRO	Computational Intelligence	2012	Utiliza perfil dos nodos como base para predições de encontros	Pode sobrecarregar redes densas	Cenários com padrões de mobilidade	✓	×	✓	×	×
CARTOON	Context Based	2012	Mistura disseminação probabilística e epidêmica	Performance prejudicada em cenários com poucos nodos	Cenários com padrões de mobilidade	×	✳	✓	×	×
CAR	Context Based	2007	Combina dados de contexto e probabilidade para encaminhamentos	Demanda muito tempo para aprendizado do contexto	Redes Sociais	✓	×	✓	×	×
JOKER	Context Based	2016	Entrega dados multimídia em tempo real atendendo critérios de QoS/QoE e baixo consumo energético	Necessita de pelo menos um caminho fim-a-fim entre origem e destino	Redes com conectividade entre os nodos	×	×	×	×	×
LoSeRO	Context Based	2016	Utiliza visitas frequentes para definição de um perfil dos nodos que são bons candidatos a mula de dados	Necessita que os nodos visitem as localidades com frequência	Redes com padrões de mobilidade repetíveis	×	×	×	×	×
HiBOp	Community Based	2007	Utiliza similaridade entre usuários	Demora em aprender os interesses dos usuários	Redes sociais	×	×	✳	×	×
SimBet	Community Based	2007	Utiliza métricas de centralidade e similaridade	Sobrecarga dos nodos com maior centralidade	Redes sociais	×	×	×	×	×
Label	Community Based	2007	Primeira abordagem a utilizar redes sociais	Taxa de entrega relacionada com o TTL das mensagens	Redes sociais	×	×	×	×	×
BubbleRap	Community Based	2011	Agrupa e ranqueia os nodos em comunidades	Pode não funcionar em redes com conexões aleatórias	Redes sociais	×	×	×	×	×
LocalCom	Community Based	2009	Emprego de roteamento inter e intra-comunidades	Pode ocorrer sobrecarga dos nodos gateways	Redes sociais	×	×	×	×	×
Gently	Community Based	2008	Suporte a multicast e unicast	Sofre das mesmas carencias do CAR e LABEL	Redes sociais	✓	×	✳	×	×
dLife	Community Based	2012	Explora as rotinas diárias dos usuários e suas interações	Em ambientes com muitos contatos pode exaurir o buffer e interromper o encaminhamento	Redes sociais	×	×	✳	×	×
SCORP	Community Based	2014	Explora as interações sociais do usuários e seus interesses	Configuração com muitos interesses pode exaurir a memória dos nodos	Redes sociais	×	×	✳	×	×
SocialCast	Shared Interests	2008	Primeira iniciativa a explorar paradigma publish/subscribe	Funciona bem quando todos os membros de cada comunidade estão interessados no mesmo tipo de conteúdo	Redes sociais com interesses comuns	✓	×	✳	×	×
ContentPlace	Shared Interests	2008	Explora informações a respeito de relações sociais dos usuários	As métricas empregadas podem não refletir o dinamismo do comportamento dos usuários	Redes sociais	✳	×	✳	×	×
PeopleRank	Node Popularity	2010	Ranqueamento dos nodos centrais	Pode sobrecarregar os nodos mais relações sociais (mais centrais)	Redes sociais	×	×	×	×	×
SRE	Node Popularity	2011	Rankeia os nodos centrais dentro de uma comunidade	Pode sobrecarregar os nodos centrais das comunidades Redes Sociais	×	×	×	×	×	×

*Legenda: ✓= sim ×= não ✳= parcial

(Fonte: o autor)

3.3 Considerações finais do capítulo

O presente capítulo teve por objetivo apresentar o estado da arte nas principais áreas relacionadas à tese. A respeito de Sensoriamento Urbano foram analisadas algumas iniciativas existentes. Com base nessa análise pode-se constatar que elas não envolvem todo o processo de sensoriamento (da coleta de dados a disponibilização de informações aos habitantes) e também não contemplam de forma satisfatória as características apresentadas. Além disso, verificou-se que o uso de Redes Oportunistas ainda não é plenamente explorado pelas aplicações de sensoriamento e isso pode ser utilizado como uma forma complementar à comunicação tradicional via pilha TCP/IP, permitindo a troca de dados mesmo em locais onde a infraestrutura de rede é intermitente ou inexistente.

A partir dessa constatação, as pesquisas foram direcionadas à análise do estado da arte em Redes Oportunistas. Foi apresentada uma taxonomia dos protocolos da área e então cada um destes protocolos foi descrito com mais detalhes. O resultado desse estudo foi sintetizado em uma tabela que demonstrou que nenhum dos protocolos existentes foi concebido de forma a atender às necessidades de cenários de Sensoriamento Urbano.

Dessa forma, pode-se concluir que o estudo desenvolvido foi amplo e serviu para demonstrar o potencial de uso de Redes Oportunistas no Sensoriamento Urbano e também apontou inexistência de uma iniciativa que aplique Sensibilidade à Situação em Redes Oportunistas. Com isso, justifica-se a proposta de um componente de software para suprir tais lacunas.

O próximo capítulo irá apresentar a UrboSenti, um arquitetura computacional que foi desenvolvida ao longo da tese para suprir as principais lacunas encontradas no estado da arte de Sensoriamento Urbano. O componente que está sendo proposto irá ser empregado na UrboSenti para melhorar a comunicação de dados via Redes Oportunistas.

4 UMA ARQUITETURA UBÍQUA ORIENTADA A SERVIÇOS PARA O SENSORIAMENTO URBANO

O presente capítulo apresenta de forma conceitual a arquitetura ubíqua orientada a serviços para o Sensoriamento Urbano chamada de UrboSenti. Desde já ressalta-se que o foco da presente tese é o componente para melhorar a comunicação de dados dessa arquitetura e não a arquitetura em si. Dessa forma, não serão apresentados detalhes de implementação ou mesmo experimentos que objetivem a sua validação.

Informações mais detalhadas a respeito de suas funcionalidades podem ser encontradas no Relatório Técnico disponibilizado em <http://gppd-wiki.inf.ufrgs.br/index.php/UrboSenti>.

4.1 UrboSenti

O nome UrboSenti é formado pela união de duas palavras em Esperanto: *Urbo* = “urbano, cidade” e *Senti* = “sentir, ter a sensação de”. Esse nome foi escolhido pois a ideia principal da arquitetura está em uma solução computacional capaz de “sentir o pulso” da cidade.

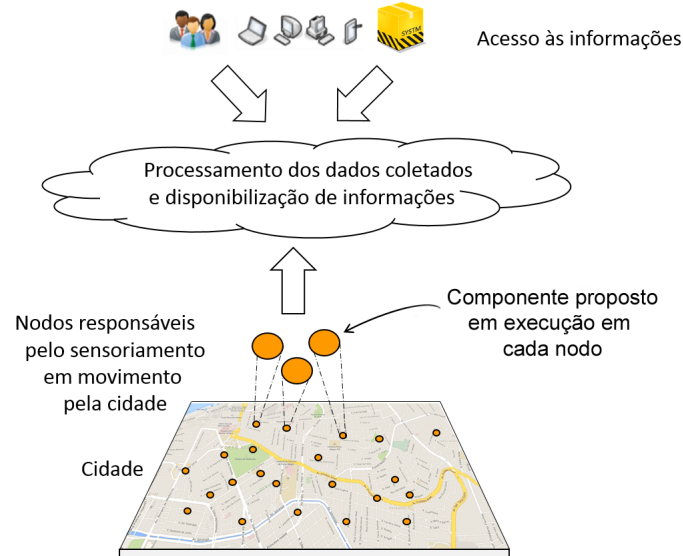
Entre os principais diferenciais de tal arquitetura com relação as outras soluções de Sensoriamento Urbano pode-se apontar:

- (i) a disponibilização de serviços que envolvem todo o processo de Sensoriamento Urbano, indo da coleta de dados até a disponibilização de relatórios que fornecem uma visão de alto-nível dos eventos que acontecem ou poderão acontecer em uma cidade, através de técnicas de predição e correlação;
- (ii) o suporte ao sensoriamento tradicional conjuntamente com sensoriamento sócia através de diferentes fontes de dados (como redes de sensores, redes veiculares e redes sociais), e;
- (iii) a disponibilização de um conjunto de serviços que buscam facilitar o desenvolvimento de novas aplicações de Sensoriamento Urbano.

A figura 4.1 fornece uma visão conceitual de alto-nível da arquitetura UrboSenti juntamente com a indicação de onde o componente que está sendo proposto será executado. Nela podem ser observados diferentes nodos (fixos e móveis) dispersos ao longo

da cidade efetuando a coleta de dados. Os dados coletados são então processados para posterior fornecimento de informações aos cidadãos da cidade e outros sistemas.

Figura 4.1: UrboSenti: Visão conceitual



(Fonte: o autor)

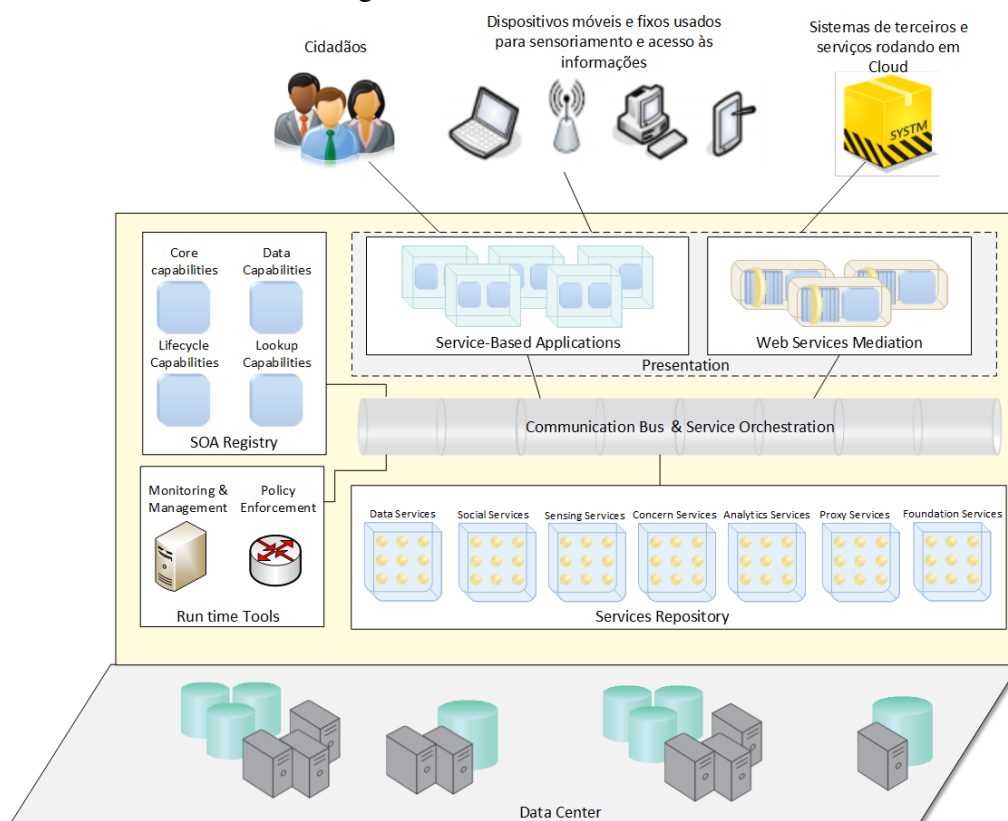
A arquitetura proposta faz o uso da abordagem de Arquiteturas Orientadas a Serviços (*SOA - Services Oriented Architecture*) para guiar a sua modelagem. Por isso, não foram utilizados os conceitos de camadas (*layers*). Essa forma de modelagem tradicional iria restringir a flexibilidade das funcionalidades dos módulos, resultando na interdependência dos componentes. Ao contrário, foi utilizado o conceito de serviço. Serviço é principal modo com que *SOA* expõe as funcionalidades dos componentes utilizadas por outros componentes ou módulos, proporcionando flexibilidade e re-usabilidade na composição de novos serviços. A arquitetura é dividida em dois módulos principais: módulo Sensing e módulo Backend.

4.2 Módulo Backend

O coração e cérebro da UrboSenti é o módulo **Backend**. Ele é executado em um data center e, de forma sucinta, é responsável por receber os dados de sensoriamento, processar eles e fornecer informações obtidas de tais dados para os cidadãos e outros sistemas. Seus componentes internos são descritos na figura 4.2 e seu comportamento descrito abaixo:

- **Services Repository:** é o repositório de todos os serviços disponibilizados. Os serviços

Figura 4.2: Módulo Backend



(Fonte: o autor)

são agrupados em categorias de acordo com seus propósitos. Os serviços disponibilizados são:

- (i) *Data services*: são utilizados para gerenciar os dados manipulados pela arquitetura. São disponibilizados serviços para armazenar / recuperar dados, e também para tratar inconsistência e “ruídos” dos dados coletados das diferentes fontes através de técnicas de fusão de dados;
- (ii) *Social Services*: são os serviços responsáveis pela manipulação de dados advindos de redes sociais;
- (iii) *Sensing Services*: são serviços utilizados para interagir com os diferentes tipos de sensores existentes e também para agregar os dados coletados. Ele faz uso de padrões públicos e abertos para facilitar a troca de dados e dar suporte a novos dispositivos;
- (iv) *Concern Services*: são serviços utilizados para tratar questões de segurança, criptografia e privacidade dos dados e dos usuários;
- (v) *Analytics Services*: é a “parte pensante” da UrboSenti. É composto pelos serviços

de classificação e mineração dos dados e também pelo fornecimento de relatórios. Tais tarefas são efetuadas por funcionalidades internas que conseguem fazer o processamento dos fluxos de dados entrantes em tempo real. Além disso, emprega técnicas de estatística e de aprendizagem de máquina para proporcionar a correlação dos dados e a predição de eventos futuros na cidade. Tal característica é de grande auxílio para a detecção de situações que podem ocasionar grande impacto na cidade de forma proativa;

- (vi) *Proxy Services*: são os serviços que fazem interações com sistemas externos como redes veiculares ou redes de sensores e também efetuam a troca de dados com sistemas de terceiros;
 - (vii) *Foudation Services*: envolvem todos os serviços básicos que são usados pelos outros serviços para execução de suas tarefas.
- **Communication Bus & Services Orchestration**: é o barramento utilizado para a comunicação entre os componentes. Também faz a coordenação e orquestração das chamadas a múltiplos serviços para proporcionar que as funcionalidades sejam expostas como um único serviço agregado (característica típica de uma SOA).
 - **Web Services Mediation**: é o intermediário entre as entidades externas (como os dispositivos de sensoriamento e serviços de terceiros) e o *Services Repository* (invocado através do *Communication Bus & Service Orchestration*)
 - **Service-Based Applications**: são todas as aplicações desenvolvidas a partir dos serviços disponibilizados pela arquitetura. Em conjunto com o *Web Services Mediation* forma o componente mais externo da arquitetura chamado de *Presentation* cuja função é interagir com os usuários e outros sistemas.
 - **SOA Registry**: é um componente responsável pelo gerenciamento dos serviços existentes. Seus serviços internos manipulam os meta-dados dos serviços disponibilizados visando fornecer a cada serviço uma identidade única. Além disso, armazena as informações necessárias para a identificação do responsável por um serviço e especificações de como o mesmo se comporta em tempo de execução (etapas do ciclo de vida de um serviço). Também proporciona artefatos para manipular dados de UDDI (*Universal Description, Discovery and Integration*) para registro e descoberta dos serviços.
 - **Run Time Tools**: envolve as ferramentas de gerenciamento e monitoramento dos serviços. Contém os artefatos gerados em tempo de execução, como *logs* de mensagens,

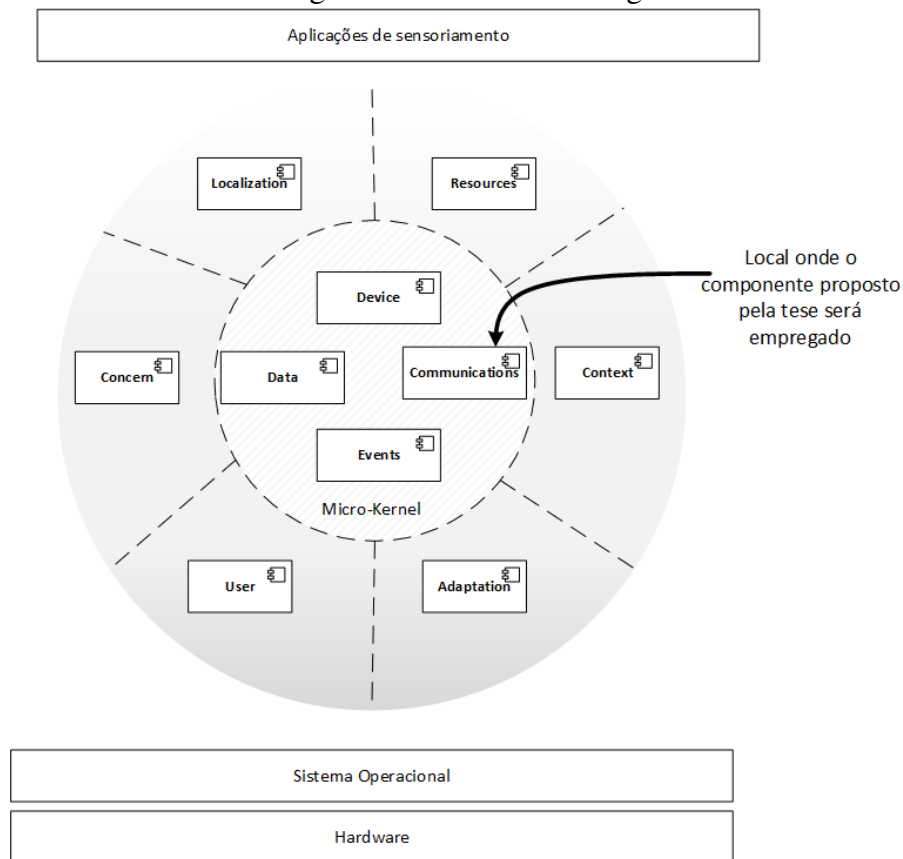
dados de execução e saúde do sistema, *heartbeat* dos principais componentes e também indicadores de performance (*KPIs - Key Performance Indicators*) para serem usados em painéis de gerenciamento (*dashboards*) do ambiente. Faz uso de políticas (*Policy Enforcement*) que visam garantir que as mensagens sejam geradas corretamente e os serviços executados dentro dos padrões definidos pelo Acordo de Nível de Serviços (*SLA - Service-Level Agreement*).

4.3 Sensing

O outro módulo principal da UrboSenti é chamado de **Sensing**. Ele é executado em dispositivos móveis ou em sensores fixos dispersos ao longo da cidade que são utilizados nas atividades de coleta de dados de forma intencional (com o usuário interagindo diretamente com a aplicação no fornecimento de dados) ou não intencional (com a aplicação coletando dados em segundo plano, sem interação direta com o usuário). Além do sensoriamento tradicional (onde sensores são empregados para coletar dados), é fornecido suporte ao sensoriamento social (segundo ??) nesse tipo de sensoriamento, os dados de sensores são integrados com dados provenientes de redes sociais visando maior compreensão do ambiente externo). Os componentes internos do módulo são apresentados na figura 4.3 e seu funcionamento descrito a seguir.

- **Micro-kernel:** é o núcleo do módulo. Sua principal função é proporcionar os serviços básicos para os componentes mais externos. Internamente, o *Micro-kernel* é estruturado em:
 - (i) *Device:* proporciona informações básicas a respeito do dispositivo (nome, endereço de rede, interfaces existentes, GPS, sensores internos, etc.);
 - (ii) *Communications:* proporciona métodos para enviar e receber dados utilizando a estrutura de rede disponível (IEEE 802.11b/g/n - estruturado e ad-hoc, GPRS/EDGE/3G e Ethernet - 802.3 - como base para a troca de dados através de TCP ou UDP. Quando uma destas infra-estrutura de rede estiver intermitente ou não disponível, o componente fornece suporte para a troca de dados utilizando o paradigma de Redes Oportunistas. É aqui que será executado o componente que está sendo proposto nessa tese com vistas a intensificar a comunicação de dados quanto Rede Oportunistas estiverem em uso;
 - (iii) *Data:* manipula as operações de armazenar e recuperar dados;

Figura 4.3: Módulo Sensing



(Fonte: o autor)

(iv) *Events*: captura eventos externos de interesse (mudança de posição devido a movimentação do usuário, alteração no *status* de uma interface do dispositivo, etc.). Os eventos detectados são disponibilizados para serem utilizados pelos outros componentes.

- **Localization**: controla questões relacionadas a informações de geolocalização, Pontos de Interesse (*POI - Points of Interest*) e Serviços Baseados em Localização (*LBS - Location Based Services*).
- **Resources**: conjunto de componentes e serviços utilizados para monitorar os recursos locais e descobrir recursos existentes em outros dispositivos.
- **Concern**: lida com questões de segurança, criptografia e privacidade das entidades envolvidas no sensoriamento.
- **Context**: disponibiliza às aplicações um conjunto de componentes e métodos para dar suporte à sensibilidade ao contexto (*context reasoning, context knowledge, context discovery*)

- **User:** manipula os dados pessoais básico do usuário, como preferencias e perfis de redes sociais.
- **Adaptation:** são os componentes e serviços disponibilizados para proporcionar a adaptação no comportamento do dispositivo e das aplicações. Utiliza um conjunto de políticas e regras para descrever como as aplicações devem se adaptar e monitora dados básicos sobre a qualidade dos serviços para inferir quando o processo de adaptação é necessário.
- **Aplicações:** representa as aplicações de sensoriamento rodando no topo do módulo.
- **Sistema Operacional e Hardware:** representam a parte mais baixa do dispositivo no qual o módulo é executado.

Novamente ressalta-se que a UrboSenti não é o foco da presente tese. A sua implementação está em andamento. Atualmente o grupo de pesquisa vinculado ao seu desenvolvimento está trabalhando no módulo *Sensing*, mais especificamente no *Micro-kernel*. As funcionalidades básicas dos componentes internos estão concluídas, exceto o componente *Communications*. Tal componente é o responsável pela manipulação de diferentes tipos de interfaces de rede para enviar e receber dados, de acordo com a infraestrutura de rede disponível. O suporte para comunicação TCP e UDP via rede sem fios e rede cabeada está pronto. O mecanismo de predição utilizado como base para as comunicações via DTNs também foi concluído e as simulações efetuadas foram apresentadas em [1].

Acredita-se que a partir do momento que a UrboSenti estiver plenamente codificada e funcional ela irá suprir as lacunas dos trabalhos relacionados e poderá ser aplicada com sucesso no cenário-problema inicialmente descrito. Além disso, o andamento dos trabalhos abriu um “guarda-chuvas” onde diferentes pesquisas estão sendo desenvolvidas pelo grupo de Computação Ubíqua do GPPD/UFRGS na área multidisciplinar de Cidades Inteligentes.

Por fim, destaca-se que as definições da arquitetura da UrboSenti apresentadas nesse capítulo foram publicadas no seguinte periódico:

- ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GERONIMO, G. A. ; SANTOS, T. F. M. ; SOUZA, A. M. ; GEYER, C.F.R. Urbosenti: An Ubiquitous Service-Oriented Architecture for Urban Sensing – In: Agent Technology for Intelligent Mobile Services and Smart Societies. [S.l.]: Springer Berlin Heidelberg, 2015. p.1–10. (Communications in Computer and Information Science, v.498). 28

5 O COMPONENTE DE SOFTWARE SITUS

Este capítulo apresenta o *Situs*, um componente de software baseado em Redes Oportunistas que utiliza Sensibilidade à Situação com vistas à proatividade nas tomadas de decisões de roteamento para com isso intensificar a entrega de mensagens em aplicações de Sensoriamento Urbano. Conforme apresentado na seção 2.3, tais aplicações se caracterizam por envolverem humanos, dispositivos móveis e sensores para a coleta de informações a respeito de diferentes aspectos da cidade.

O nome *Situs* é formado pelas palavras *Situation* e *Urban Sensing*, em alusão a sua capacidade de lidar com situações e pela sua aplicação no Sensoriamento Urbano. Para desempenhar suas funções, ele implementa os 3 níveis do modelo de situação, sendo capaz de perceber, compreender e projetar situações futuras que auxiliarão nas tomadas de decisões.

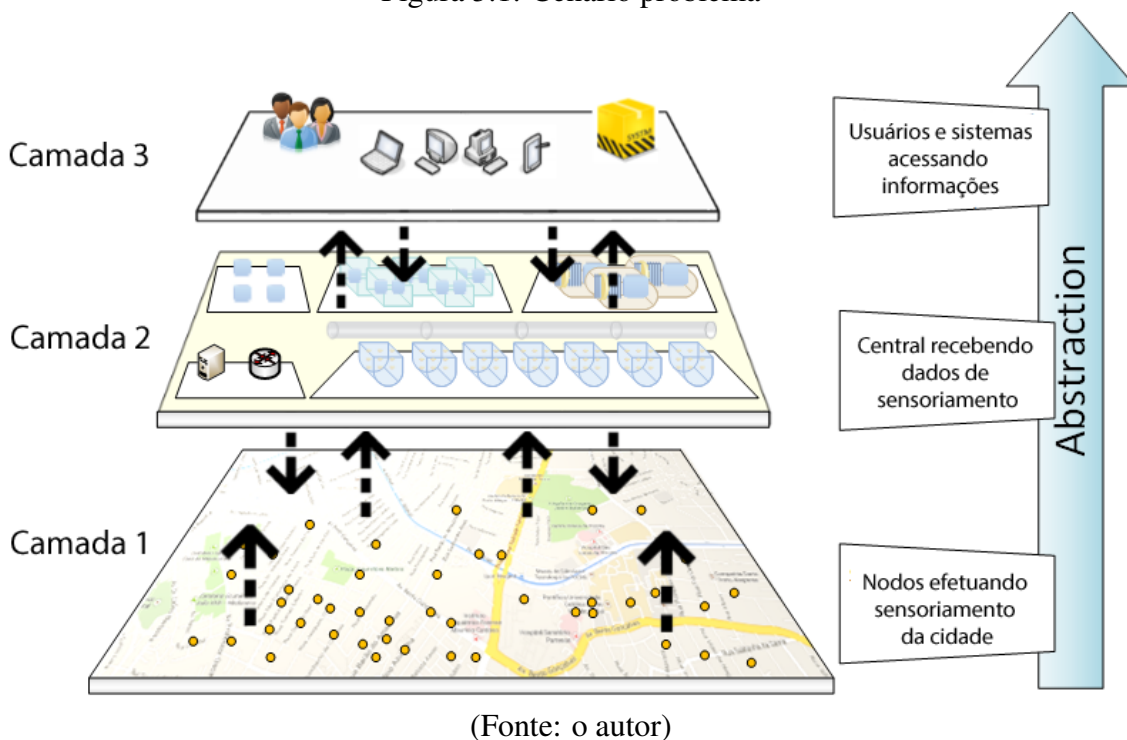
Para sua definição é utilizada uma abordagem *top-down*. Inicialmente será apresentado um cenário-problema e então definido o modelo conceitual que representa os seus principais componentes e interações. São então demonstrados os modelos de contexto, situação, e tomada de decisão empregados na definição da arquitetura computacional. É apresentado então o comportamento da arquitetura e por fim, são descritos os algoritmos resultantes da integração de todos os conceitos utilizados.

5.1 Cenário-problema

Seja considerado o cenário-problema apresentado abaixo e representado graficamente na figura 5.1:

“Em uma cidade, diferentes tipos de dispositivos e sensores dispersos ao longo da cidade são utilizados como fonte de dados para o Sensoriamento Urbano. Veículos equipados com sensores proporcionam que sejam coletados dados referentes à situação das ruas e avenidas da cidade, permitindo, dessa forma, a identificação de pontos de congestionamento ou que apresentem algum tipo de situação que acarrete impactos na mobilidade urbana. Aliados aos sensores dos veículos existem sensores fixos que são posicionados em locais estratégicos da cidade e permitem o sensoriamento da poluição do ar e sonora, luminosidade, temperatura, humidade, quantidade de chuvas e até mesmo, o nível do rio que atravessa a cidade. Além disso, são disponibilizadas aos habitantes dessa cidade aplicações para dispositivos móveis, como celulares e tablets, que permitem

Figura 5.1: Cenário problema



a coleta intencional e não intencional de dados. Assim, um morador pode indicar que em determinada localidade da cidade existe uma grande quantidade de lixo acumulada perto de um bueiro, que determinada via possui buracos ou ainda que um poste de luz da via pública está com a lâmpada queimada e precisa ser concertado. Aliado a esse conjunto de dispositivos e aplicações, a cidade também faz o uso de redes sociais, como Twitter, Facebook e G+ como forma de coletar dados que retratam diferentes visões e acontecimento da cidade. Essa grande massa de dados obtidos através do sensoriamento é uma importante fonte de informações sobre o “pulso” da cidade. A partir do processamento, classificação e análise dos dados são gerados relatórios para a tomada de decisão pelos responsáveis pela condução da cidade ou ainda disparados alertas de eventos críticos que não poderiam ser detectados a partir de somente uma única fonte de dados (por exemplo, pode-se imaginar que correlacionando a quantidade de chuva de determinado ponto da cidade com a quantidade de lixo existente nos bueiros naquela região consiga-se identificar ou até mesmo prever a possibilidade de um transbordamento do rio e que isto irá afetar a população ribeirinha). Além disso, no sentido da extensibilidade, alguns dados foram disponibilizados publicamente para que sejam utilizados por outros sistemas ou para que novas aplicações de terceiros sejam desenvolvidas.”

Esse cenário é um exemplo de ecossistema de Sensoriamento Urbano. Conceitual-

mente, ele pode ser dividido em 3 camadas. A *Camada 1* representa a Cidade Inteligente com os dispositivos e aplicações utilizados para a coleta de dados de sensoriamento. Esses dados são enviados para um *Centro de Serviços Inteligentes* representado pela *Camada 2*. Nessa camada, ocorre o processamento e análise dos dados. As informações geradas a partir do processamento permitem que diferentes visões da cidade sejam criadas. Com isso, as informações podem ser classificadas e correlacionadas com vistas a simular cenários “e se?” (*what-if?*) e até mesmo prever tendências ou situações que seriam difíceis de serem percebidas somente pela observação de fatos isolados. A *Camada 3* representa os usuários e aplicações que utilizam essa grande massa de conhecimento a respeito da cidade como auxílio na tomada de decisões ou no fornecimento de serviços de sensoriamento aos usuários da cidade.

Com base nessa estruturação em camadas e também nos apontamentos de autores como (????????????????) foram definidos os seguintes requisitos computacionais que precisam ser tratados para a efetiva prestação de serviços de sensoriamento aos habitantes de uma Cidade Inteligente:

- Camada 1

- (i) Mecanismos de sensibilidade ao contexto e adaptação são necessários para que as aplicações de sensoriamento rodando nos dispositivos tenham ciência do ambiente que estão executando e se adaptem proativamente as necessidades;
- (ii) Necessidade de uso de técnicas capazes de lidar com dados de contexto ruidosos ou incertos;
- (iii) Heterogeneidade de dispositivos, tanto móveis quanto fixos, utilizados para coleta dos dados podem ter restrições de energia e baixo poder de processamento demandam por algoritmos com baixo custo computacional;
- (iv) Infraestrutura de rede nem sempre disponível ou intermitente demanda que os dispositivos sejam capazes de se comunicarem usando diferentes técnicas de comunicação;

- Camada 2

- (v) Incerteza sobre os dados coletados demanda por técnicas capazes de gerar informações pertinentes para a tomada de decisão ou para serem utilizados por outros sistemas;

- (vi) Grande quantidade de dados dispersos ao longo da cidade são gerados continuamente, necessitam de alto poder de processamento, preferencialmente processados em tempo real;
- (vii) Suporte à correlação dos dados coletados e uso de técnicas estatísticas e de aprendizagem de máquina que possibilitem a predição de eventos futuros com vistas ao gerenciamento pró-ativo da cidade;
- Camada 3
 - (viii) Segurança sobre os dados armazenados e privacidade sobre quem os está coletando e acessando;
 - (ix) Acesso ubíquo às informações resultantes das análises dos dados processados;
 - (x) Possibilidade de reaproveitamento dos componentes de software desenvolvidos para facilitar o desenvolvimento de novas aplicações.

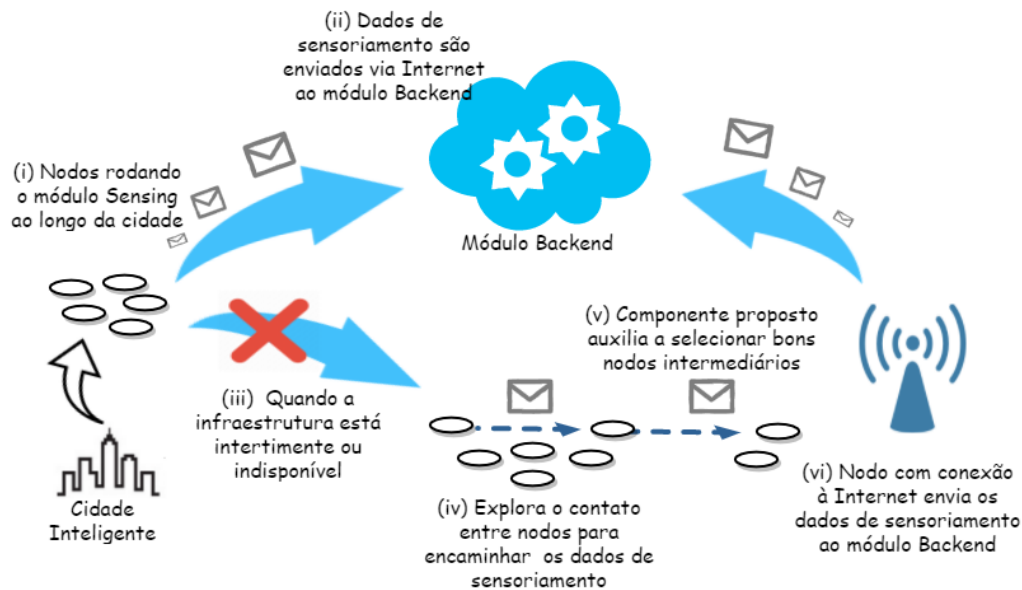
O *Situs* utiliza Sensibilidade à Situação como representação de alto nível de contexto para perceber, compreender e projetar situações futuras dos nodos. Ele será ser empregado pelo módulo *Sensing* da *UrboSenti* para intensificar a comunicação de dados de aplicações de sensoriamento urbano.

Destaca-se que ele irá atender os requisitos da Camada 1 do cenário apresentado acima. Para isso, as seguintes abordagens serão empregadas:

- Proatividade nas necessidades do nodo através da projeção de situações futuras com o uso de predições por uma Rede Neural \implies atendimento ao requisito (i);
- Inferência sobre dados de contexto ruidosos e imprecisos que caracterizam uma situação com o uso de Lógica Fuzzy \implies atendimento ao requisito (ii);
- O tipo de Rede Neural empregada possui o diferencial de baixo custo computacional \implies atendimento ao requisito (iii);
- Para permitir que os dispositivos se comuniquem mesmo em situações onde a infraestrutura de rede estiver intermitente ou indisponível será utilizado o paradigma de Redes Oportunistas \implies atendimento ao requisito (iv);

As funcionalidades do *Situs* estão esquematizadas na figura 5.2. Ele é executado em todos os nodos que fazem o uso do módulo *Sensing* da *UrboSenti* para efetuar o sensoriamento ao longo da cidade (i). Em cenários onde existe conectividade, esses dados

Figura 5.2: Esquemática de funcionalidade do componente proposto



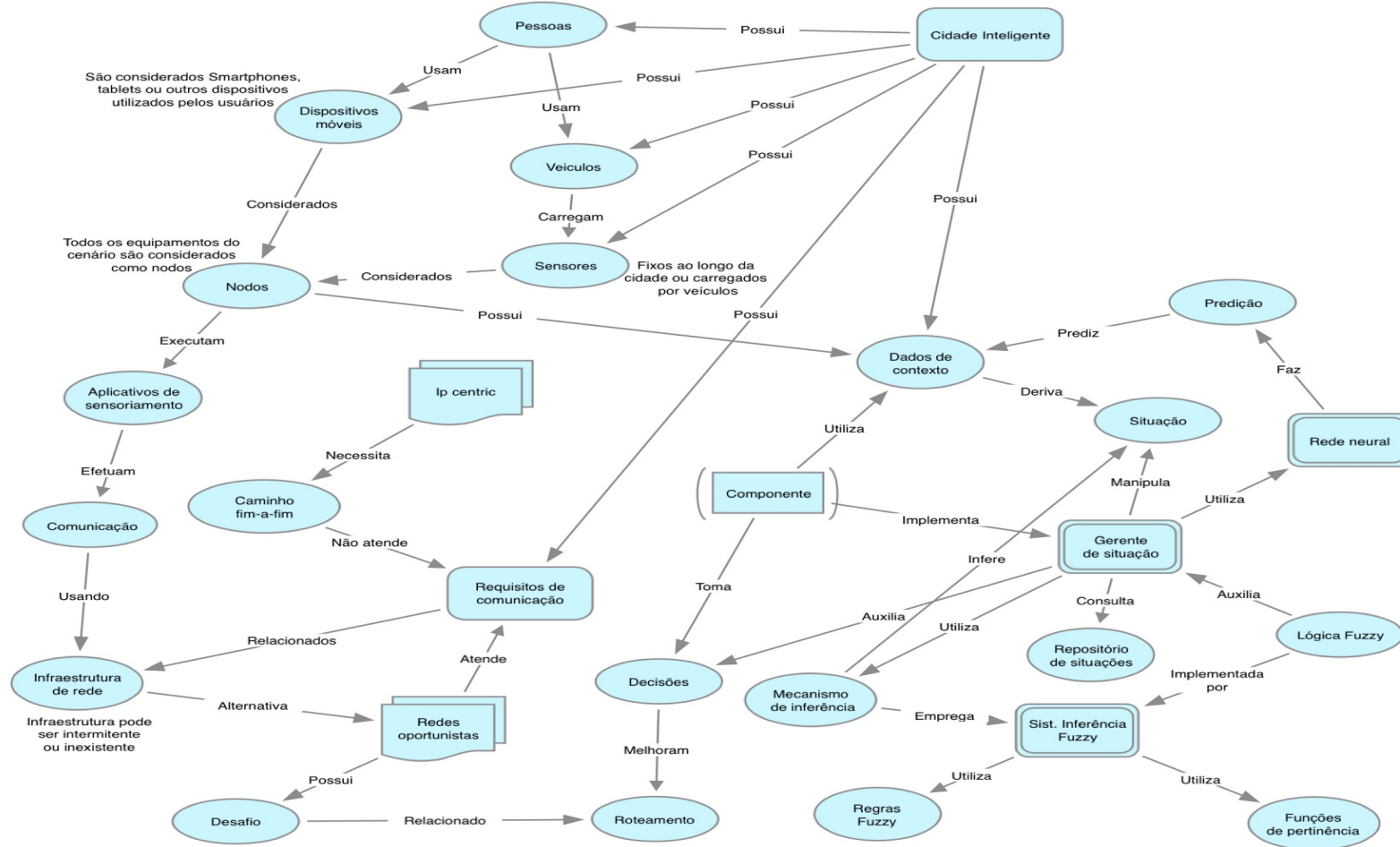
(Fonte: o autor)

são enviados via Internet para serem processados pelo módulo Backend da UrboSenti (ii). Quando a infraestrutura da rede estiver intermitente ou indisponível (iii), é explorado o contato entre os nós para encaminhamento dos dados de sensoriamento (iv). O encaminhamento das mensagens utiliza o paradigma “store, carry and forward” e para isso o componente irá atuar na seleção de bons nós intermediários para carregarem os dados (v) até que um nó com conectividade seja encontrado e possa enviá-los para serem processados remotamente (v).

5.2 Modelagem conceitual

Modelagem conceitual é a atividade de descrever formalmente alguns aspectos do mundo físico com o objetivo de compreensão e comunicação (?). O modelo conceitual dos componentes e interações da *Camada 1* do cenário é representado através de um *mapa mental* representado graficamente na figura 5.3 e detalhado a seguir:

Figura 5.3: Mapa mental representando o modelo conceitual



(Fonte: o autor)

As atividades de sensoriamento são desenvolvidas em um *cenário*, como por exemplo uma Cidade Inteligente. Esse cenário possui diferentes tipos de entidades envolvidas no sensoriamento, como *pessoas*, *veículos*, *dispositivos móveis* (tablets e smartphones) utilizados por pessoas e *sensores* que podem ser carregados por veículos ou instalados em locais fixos ao longo da cidade. Tanto dispositivos móveis quanto sensores são considerados como *nodos*.

Cada *nodo* executa *aplicativos de sensoriamento* utilizados nas atividades de coleta de dados de forma intencional ou não-intencional. Os aplicativos são encarregados de coletar os dados de sensoriamento e os transmitir para o Centro de Serviços Inteligentes onde serão processados e analisados. Para enviar tais dados os *nodos* precisam fazer a *comunicação* utilizando a *infraestrutura de rede*, a qual nem sempre existe ou pode ser intermitente (devido a alguns pontos da cidade não possuírem qualquer tipo de acesso a rede, a rede pode estar saturada, pontos de sombra de sinal, interferências, etc.).

A infraestrutura de rede nem sempre disponível ou intermitente para proporcionar a *comunicação* é considerado um *requisito de comunicação* do *cenário*. O uso de um paradigma de comunicação *IP Centric* é inviável devido à necessidade da *infraestrutura de rede* estar sempre disponível para proporcionar um *caminho fim-a-fim* entre os nodos (????????????). Dessa forma, o emprego do paradigma *IP Centric* não atende aos *requisitos de comunicação*.

As *Redes Oportunistas* surgem como uma alternativa inovadora para complementar a comunicação em situações como essa, onde as aplicações ubíquas precisam trocar dados porém a infraestrutura de rede é intermitente ou mesmo inexistente (????). Conforme apresentado anteriormente na seção 1.2, um *desafio* existente nesta área está relacionado à disseminação e ao *roteamento* eficiente de mensagens entre os nodos através da técnica de *store-carry-and forward*.

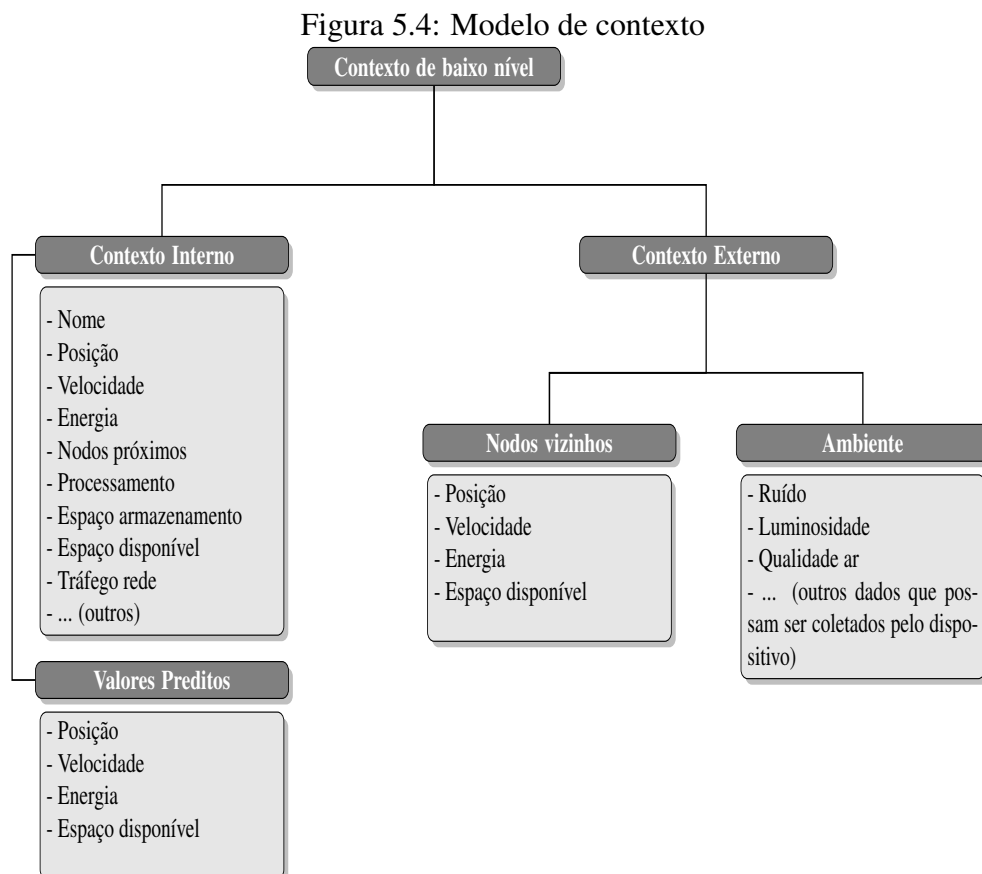
O *componente* de software que está sendo proposto nessa tese é executado em cada um dos nodos envolvidos na tarefa de sensoriamento. Ele utiliza os *dados de contexto* existentes no *ambiente* e também os dados internos do próprio *nodo* para tomada de *decisões* quem buscam melhorar o *roteamento*. O conjunto de dados de contexto é agrupado em uma visão de mais alto nível chamada de situação. O *gerente de situação* é o responsável pela identificação da situação atual do nodo e também por projetar a sua situação futura. Para a inferência da situação atual é consultado um repositório de regras e também utilizado o *mecanismo de inferência*. Esse mecanismo emprega um *sistema de inferência fuzzy* que a partir da aplicação de *regras fuzzy* e *funções de pertinência* efetua a

compreensão da situação do nodo. Para efetuar a projeção de situação é empregada uma *Rede Neural*. A função da *Rede Neural* é efetuar a *predição de dados de contexto* que serão utilizadas para a projeção de uma situação futura do nodo.

Com base nessa representação de alto nível do cenário foram propostos modelos que são empregados no *Situs* e serão descritos nas próximas seções.

5.2.1 Modelo de contexto

O modelo de contexto utilizado no *Situs* é apresentado na figura 5.4. Ele foi definido com base nos trabalhos de ??), ??) e ??) e objetiva estruturar as informações que representam as propriedades temporais e espaciais do ambiente e do fenômeno observado com vistas a caracterizar a situação e as relações das entidades envolvidas.



(Fonte: o autor)

Embora ontologias sejam adotadas por outros trabalhos devido a sua expressividade e capacidade de representação, optou-se por não utilizá-las devido as deficiências apresentadas por ??) no que se refere ao custo computacional e problemas de perfor-

mance, a limitações de raciocínio semântico em tempo real (??) e também devido a limitações dos motores de inferência para tratar eventos e fatos lógicos em tempo real(??).

Assim, segundo o modelo proposto, cada nodo do cenário possui um conjunto de dados de contexto chamado de **contexto de baixo nível**. Ele é representado pelos dados que serão utilizados para a definição da visão de nível mais alto chamada de situação. Ele é definido pelo conjunto de dados de contexto interno e de contexto externo.

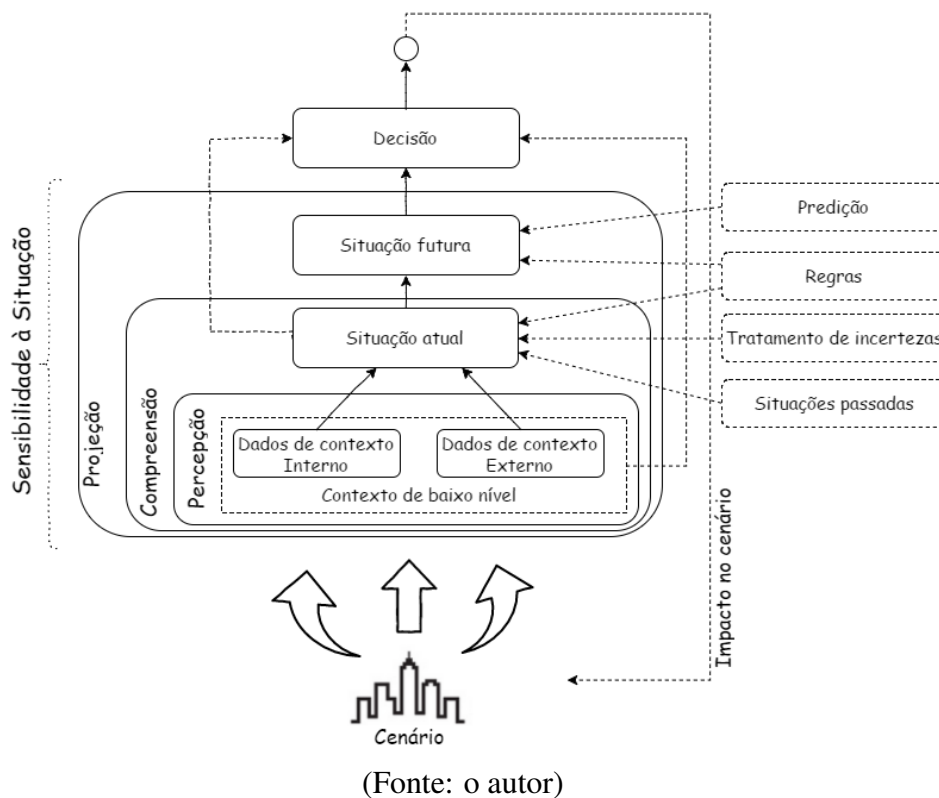
O **contexto interno** envolve os dados que caracterizam o nodo em um determinado instante corrente ou de seu histórico de contextos. Como exemplos de atributos de contexto interno podem ser considerados: nome, posição, velocidade, energia, nome e endereço dos nodos próximos, distância percorrida total, distância percorrida desde o último encontro, capacidade de processamento, número de mensagens no buffer, espaço disponível no buffer para armazenamento, tráfego na rede e também outros atributos suportados pelo dispositivo. O contexto interno também é formado por valores que provavelmente irão caracterizar a situação do nodo em um momento futuro. Eles são chamados de **valores preditos** e são gerados a partir do uso de técnicas de predição por uma Rede Neural. A ideia é que em um tempo t sejam gerados $t+n$ passos adiante através de predições. Isso acontece de forma cíclica em intervalos de tempo.

Além do contexto interno, cada nodo é capaz de obter informações dos nodos vizinhos e também do ambiente que o circunda. Esses dados são chamados de **contexto externo**. Como contexto externo são considerados os atributos dos nodos vizinhos como: velocidade, capacidade de processamento, espaço em buffer, energia disponível, etc. O ambiente possui atributos que podem ser coletados pelos sensores disponíveis no nodo, como por exemplo: ruído, luminosidade, qualidade do ar e também outros atributos que possam ser coletados pelos dispositivos.

5.2.2 Modelo de situação

O contexto de baixo nível do nodo (descrito na seção anterior) é utilizado para auxiliar na tomada de decisão de encaminhamento de mensagens e também nas adaptações das funcionalidades internas do *Situs*. Porém, conforme apontado por ??), utilizar partes isoladas de dados de contexto pode não apresentar significado útil para a tomada de decisões. Assim, com a finalidade de agrupar os dados de contexto de baixo nível e então derivar uma nova visão de alto-nível que facilite a tomada de decisão é empregado o conceito de **Sensibilidade à Situação** (??).

Figura 5.5: Modelo de Sensibilidade à Situação empregado

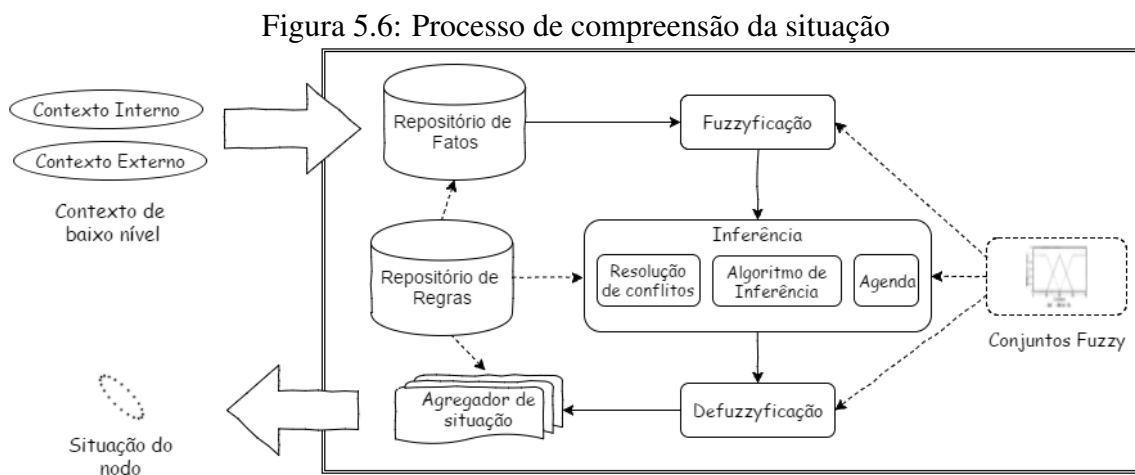


O modelo de Sensibilidade à Situação é baseado nos 3 níveis definidos por ??) que foram apresentados anteriormente na seção 2.5.3. Ele é graficamente representado na figura 5.5 e assim descrito:

- Nível 1 - Percepção:** é considerado o nível mais baixo da Sensibilidade à Situação. Na tese ele têm por finalidade perceber o status e valores dos atributos dos elementos e interações do contexto de baixo nível, e então agrupa-os de forma estruturada para serem utilizados pelas camadas superiores. Em intervalos cíclicos de tempo, os valores coletados e estruturados são armazenados no histórico de contexto. É importante notar que alterações nos valores do contexto de baixo nível poderão caracterizar uma situação atual diferente da anterior, tornando assim o sistema dinâmico. ;
- Nível 2 - Compreensão:** utiliza os dados de contexto de baixo nível advindos do Nível 1 para derivar a visão de alto nível chamada de situação. Em outras palavras, a principal função do Nível 2 é de caracterizar a situação dos nodos. O grande desafio para essa tarefa está relacionado a caracterização de uma situação de forma precisa uma vez que os dados de contexto podem ser incompletos ou ruidosos. Ou seja, esse nível precisa lidar com incertezas geradas a partir dos dados de contexto de

baixo e mesmo assim ser capaz de fornecer a situação o mais próximo possível da realidade. Para isso, é utilizada a Lógica Fuzzy. O uso de Lógica Fuzzy é adequado para representar o conhecimento em casos como esse, onde um sistema especialista teria dificuldades na definição (ou precisaria de uma grande quantidade) de regras (??). Segundo ??), a Lógica Fuzzy é uma alternativa viável quando é necessário raciocinar e tomar decisões certas com base em informações complexas, imperfeitas e incertas.

O processo de compreensão da situação é representado na figura 5.6. Os dados de contexto de baixo nível são fornecido como entrada e então armazenados em um *Repositório de fatos* que também funciona como um histórico de dados. A seguir é efetuada a *fuzzyficação* dos dados utilizando regras fuzzy armazenadas no *repositório de regras* juntamente com um conjunto de *funções de pertinência* (Membership functions (MF)). Os dados fuzzificados passam então pelo processo de *inferência* que faz uso de um *Sistema de Inferência Fuzzy* (Fuzzy Inference System (FIS)) que implementa um sistema Mandani. A escolha de Mandani é justificada por este ser mais intuitivo e proporcionar maior capacidade na captura de conhecimento através da definição de regras mais amigáveis que um sistema Sugeno (????).



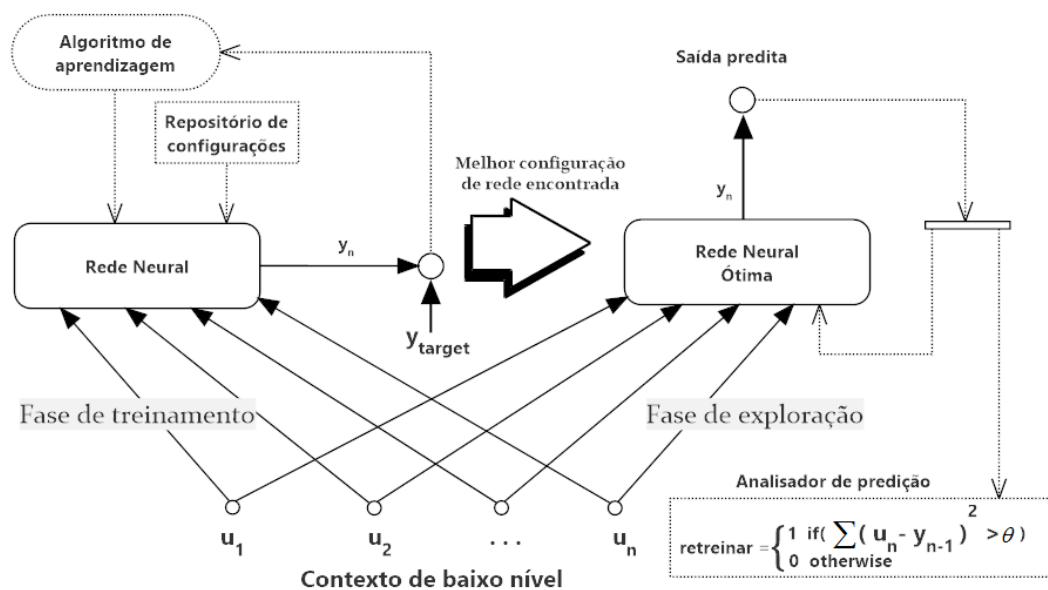
(Fonte: o autor)

Na inferência, são aplicadas regras fuzzy (do repositório de regras) e algoritmos para *resolução de conflitos* quando são disparadas regras que gerem algum tipo de conflito na saída (por exemplo uma variável de contexto indicando que não existe espaço em buffer e a bateria está com 100% de carga enquanto na verdade deveria estar com pouca carga). Para tornar a inferência mais eficiente é utilizado um mecanismo de *agenda* que agrupa regras semelhantes em grupos que evitam que todas

regras sejam avaliadas a cada interação. Como saída do processo de inferência é caracterizado o grau de pertinência dos dados de contexto para diferentes situações (por exemplo, 0.98 para a situação de bateria com carga alta e 0 para situação de bateria sem carga; 0.6 para situação de buffer cheio e 0.4 para situação de buffer vazio; etc.). Os graus de pertinência de cada uma das variáveis são defuzzificados para uma saída precisa e então o *agregador de situação* o agrupa e aplicada regra que resultam na caracterização da situação do nodo.

- Nível 3 - Projeção:** tem por finalidade projetar situações futuras dos nodos que permitam a proatividade na execução de ações de roteamento e adaptação. O suporte para essa projeção ocorre através do uso de dados de situações passadas que permitirão a predição de dados futuros. Para isso, a técnica base para a predição é uma Rede Neural com aprendizagem supervisionada (representado na figura 5.7). Ela segue o modelo básico de uma Rede Recorrente conforme proposto por Jager (????) e também de outros trabalhos apresentados anteriormente na seção 2.6.2. Porém, as diferenças do modelo proposto com relação aos demais trabalhos é que o modelo aqui proposto têm as suas variáveis de configuração como tamanho e espaçamento do reservatório, raio de espectro e taxa de vazamento definidos dinamicamente de acordo com os dados de contexto de cada nodo da rede. Ou seja, ao invés de definir a topologia da Rede de forma global e então aplica-la exatamente igual a todos os nodos, cada instância constrói a sua própria topologia. Com isso,

Figura 5.7: Processo de predição usado para projeção



(Fonte: o autor)

a Rede Neural é construída dinamicamente, de acordo com o contexto do nodo e com isso existe a possibilidade de aumento da acurácia das predições. Essa ideia é semelhante a adotada no trabalho de ??), porém ao invés de selecionar o melhor algoritmo para predição são selecionados de forma dinâmica os parâmetros utilizados na construção da rede que efetuará as predições. Além disso, foi criado um analisador de predição que basicamente consiste em um mecanismo de automonitoramento do valor de limiar de erro para determinar quando a Rede está errando demais e precisa ser retreinada, implicando assim em alterações na sua topologia. A motivação de uso de uma Rede Neural deve-se a sua capacidade de solucionar problemas de natureza não-lineares, como por exemplo a classificação e predição de valores (??).

Para a construção da Rede Neural a primeira etapa necessária é o treinamento. Inicialmente as configurações da Rede são lidas do repositório de configurações que contém dados básicos para a construção da Rede. Feito isso, as últimas N entradas do histórico de dados de contexto de baixo nível é fornecido como sinal de exemplo para a aprendizagem. São esses dados que irão “ensinar” à Rede quais os padrões que devem ser aprendidos.

Através das sinapses internas da Rede é obtido y_n que representa o valor predito. Esse valor é então comparado a uma saída esperada y_{target} indicando o quando a Rede está errando nas suas predições. O algoritmo de aprendizagem então é executado buscando minimizar o erro entre a saída obtida e a saída esperada através de ajustes nos pesos sinápticos. Esse processo de ajuste dos pesos de acordo com os erros é o que se chama de processo de aprendizagem e é efetuado diversas vezes até que o erro se mantenha estável. Nesse momento diz-se que a Rede convergiu e é capaz de reproduzir o comportamento das entradas de acordo com os padrões apresentados. Esse processo de testar diferentes configurações de Rede é executado diversas vezes até que seja encontrada a Rede com o menor MSE. Essa Rede é chamada de de *Rede Neural Ótima*, cujas configurações são salvas para serem utilizadas na próxima etapa, chamada de exploração.

Na etapa de exploração a Rede já está adequada aos dados do nodo e será empregada na predição. Algumas amostras do histórico de dados de contexto de baixo nível são fornecidos como entrada para que a Rede inicie suas atividades sinápticas. Na sequência os dados de contexto atuais são fornecidos como entrada para a Rede e então são fornecidos na saída valores preditos que caracterizarão a situação futura

do nodo. Além disso, os valores preditos serão utilizados como novas entrada para o processo de retroalimentação da Rede.

Como são preditos $t+n$ passos adiante, quando o limite de passos preditos é atingido o analisador de predição é invocado para verificar a acurácia da Rede. Para isso, os valores que foram preditos no passado ($t-n$ passos) são comparados com os valores de contexto de baixo nível atuais e então é calculado o quanto a Rede está errando nas suas predições. Se o acumulado de erro da Rede tiver atingido determinado limiar θ significa que os dados de contexto variaram muito e as configurações da topologia da Redes estão defasadas. Nesse caso, é executado novamente o procedimento de treinamento para novamente encontrar a Rede Neural Ótima para o nodo.

É importante ressaltar que devido às restrições de energia e processamento dos nodos existentes no ambiente é importante que as atividades dos níveis 2 e 3 sejam desenvolvidas com custo computacional que não comprometa a precisão das ações.

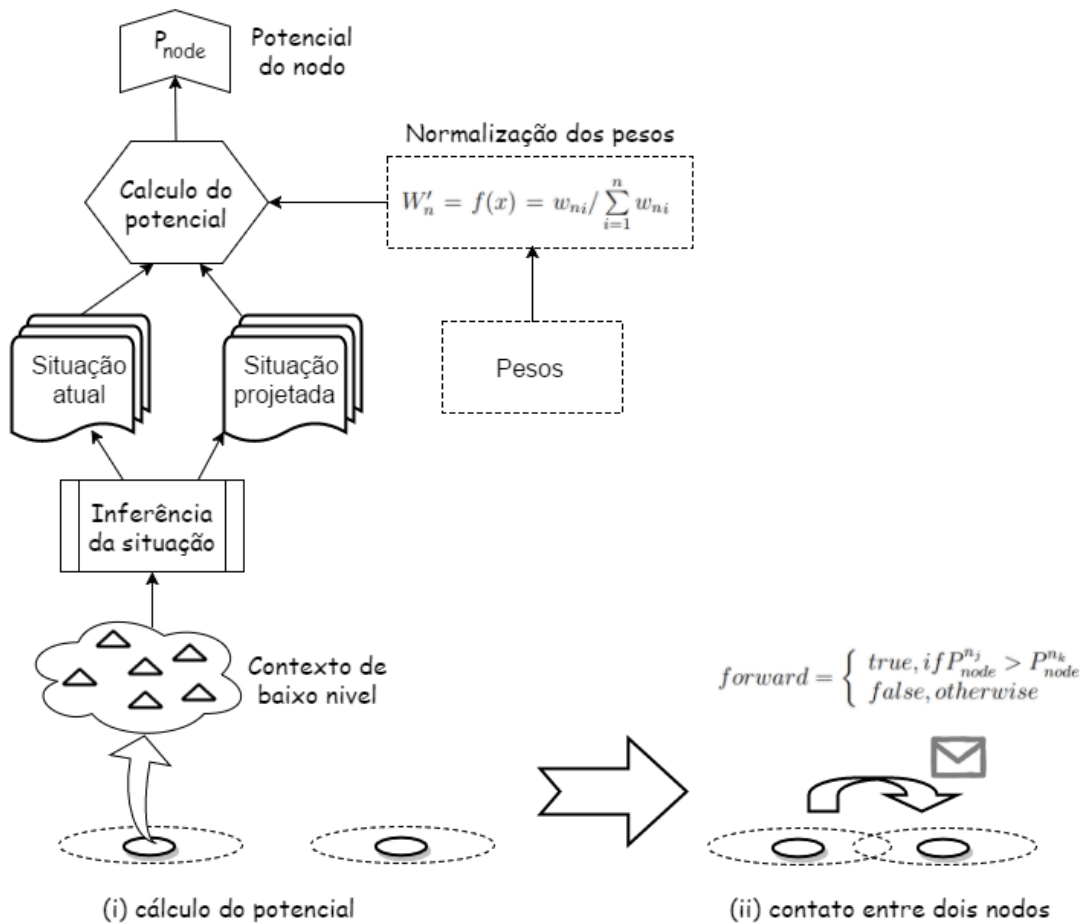
5.2.3 Modelo de tomada de decisão

O modelo de tomada de decisão apresentado na figura 5.8 é utilizado para permitir que as decisões de encaminhamento de mensagens sejam executadas. Conforme apresentado na seção 2.4.1, os nodos intermediários utilizados pelo paradigma Armazena-Carrega-Encaminha (SCF), empregado nas Redes Oportunistas, são chamados de “mula de dados” (*data mule*). O desafio existente nas decisões de encaminhamento está na estratégia utilizada para decidir se o nodo encontrado é um bom candidato para ser usado como “mula de dados”. A estratégia empregada no *Situs* é o uso de um valor chamado de *potencial* que representa a capacidade do nodo em ser uma boa “mula de dados”.

Duas etapas são utilizadas na estratégia de tomada de decisão de encaminhamento. Na etapa (i), ocorre cálculo do potencial de cada nodo. Para isso, os dados de *contexto de baixo nível* denotados por C_n onde $n = \{power, buffer, node_type, \#of_messages, \dots\}$ são utilizados para a *inferência da situação*. Como saída do processo de inferência são obtidos os valores da situação atual Vsa_n e da situação projetada Vsp_n de cada uma das variáveis de contexto do nodo conforme modelo descrito na seção anterior.

Algumas variáveis de contexto possuem maior importância no processo de tomada de decisão. Para permitir essa diferenciação é atribuído um peso W_n para cada C_n de

Figura 5.8: Modelo de tomada de decisão



(Fonte: o autor)

acordo com sua importância. Com a finalidade de uniformizar os valores dos pesos de forma que estes tenham média 0 e desvio padrão 1 é utilizada a normalização através da equação (5.1).

$$W'_n = f(x) = w_{ni} / \sum_{i=1}^n w_{ni} \quad (5.1)$$

Uma vez que os pesos estão normalizados, é calculado o *Potencial do Nodo* P_{node} com a equação (5.2). Todo esse processo é executado ciclicamente em determinados períodos de tempo.

$$P_{node} = (Vsa_{n1} * W'_{n1} + Vsa_{n2} * W'_{n2} + \dots + Vsa_{nN} * W'_{nN}) + (Vsp_{n1} * W'_{n1} + Vsp_{n2} * W'_{n2} + \dots + Vsp_{nN} * W'_{nN}) \quad (5.2)$$

A etapa (ii) ocorre quando um nodo n_j encontra outro nodo n_k e precisa ser de-

cididor se as mensagens que estão no buffer local devem ser encaminhadas para o nodo encontrado ou não. O potencial P_{node} calculado na etapa anterior é então utilizado através da aplicação da equação (5.3).

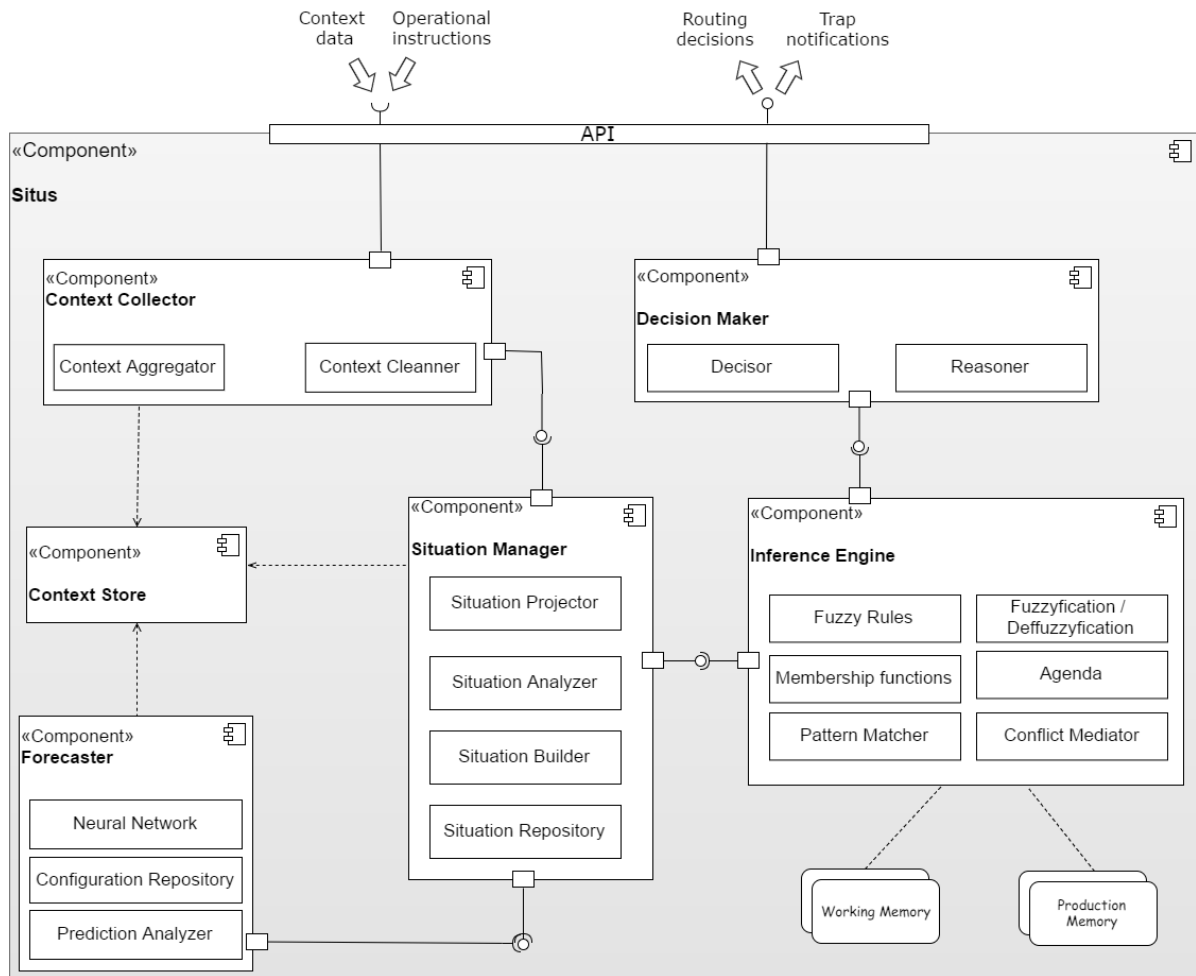
$$forward = \begin{cases} true, & \text{if } P_{node}^{n_j} > P_{node}^{n_k} \\ false, & \text{otherwise} \end{cases} \quad (5.3)$$

Com isso, se o potencial P_{node} de n_j for maior que o potencial de n_k as mensagens são encaminhadas, caso contrário elas permanecem no buffer local.

5.3 Arquitetura

Componentes de software, ou simplesmente componentes, são unidades de software independentes que oferecem serviços por meio de interfaces bem definidas. Eles são flexíveis e capazes de serem “plugados” a um sistema maior ou mesmo reusado de acordo com as necessidades. Para isso, as funcionalidades internas dos componentes são encapsuladas, de forma que um componente aparece para o seu usuário como uma espécie de “caixa-preta” que tem suas funcionalidades expostas pelas interfaces. Geralmente, os componentes possuem dois tipos de interfaces: fornecidas e requeridas. Interfaces fornecidas especificam quais os serviços que o componente disponibiliza ao ambiente. As interfaces requeridas, por outro lado, especificam quais os serviços que o componente requer do ambiente.

Os diagramas de componentes auxiliam na visualização de alto nível da estrutura de um componente e o comportamento de serviço que essas partes fornecem e consomem através de interfaces. Dessa forma, a arquitetura do *Situs* é apresentada na figura 5.9 através de um diagrama de componentes da UML e a descrição da sua estrutura interna é fornecida a seguir:

Figura 5.9: Arquitetura do *Situs*

(Fonte: o autor)

→ Context data

Representa todos os dados de contexto do nodo (contexto interno) e do ambiente (contexto externo) que são utilizados como entrada para o *Situs*.

→ Operational instructions

São instruções operacionais que são utilizadas para indicar o que *Situs* deve fazer (por exemplo, uma instrução indicando que um conjunto de dados de contexto foi carregado).

→ Routing decisions

Representa a saída do *Situs*. A saída indica qual a decisão de roteamento deve ser tomada com as mensagens que estão no buffer.

→ Trap notifications

São sinalizações (*trap*) enviadas à aplicação para indicar que uma adaptação externa é necessária.

→ API

A Application Programming Interface (API) é a interface de comunicação do *Situs* com o ambiente externo. Ela disponibiliza um conjunto de métodos que encapsulam as funcionalidades internas do componente.

→ Context Collector

É o subcomponente responsável por efetuar a coleta dos dados de contexto interno e externo em intervalos de tempo. Além de efetuar a coleta desses dados ele assegura que os dados estejam coerentes e corretamente estruturados para serem armazenados. Para desempenhar suas funções são utilizados as seguintes classes:

- **Context Cleaner:** faz a verificação de consistência dos dados coletados. O pré-processamento dos dados efetuados por esse subcomponente proporciona que eventuais dados duplicados ou que sejam considerados “ruidosos” sejam removidos das amostras coletadas;
- **Context Aggregator:** efetua o agrupamento dos dados de contexto interno e externo em uma única amostra coletada pelo nodo. O Context Agregator também é o responsável por assegurar que a amostra gerada possua a estrutura adequada para ser manipulada pelo componente de armazenamento da arquitetura.

→ Context Store

É o subcomponente responsável pelo armazenamento e gerenciamento de todos os dados de contexto. A sua estrutura interna é organizada para armazenar os dados em camadas, de acordo com sua finalidade:

- **Camada 1 - Dados do nodo:** armazena as informações básicas que caracterizam cada nodo do ambiente. Como por exemplo, nome do nodo, endereço de rede, interfaces de rede, etc;
- **Camada 2 - Dados de contexto:** são armazenados os dados de contexto interno e contexto externo do nodo (conforme descrito na seção 5.2.1). O conjunto de dados dessa camada representa o histórico de dados de contexto do nodo;
- **Camada 3 - Dados preditos:** armazena os dados de contexto preditos pelo componente de predição da arquitetura.

→ Situation Manager

É o subcomponente responsável pelo gerenciamento de situação. Ele implementa o modelo de situação descrito anteriormente na seção 5.2.2. Para o desenvolvimento de suas tarefas as seguintes classes são utilizadas:

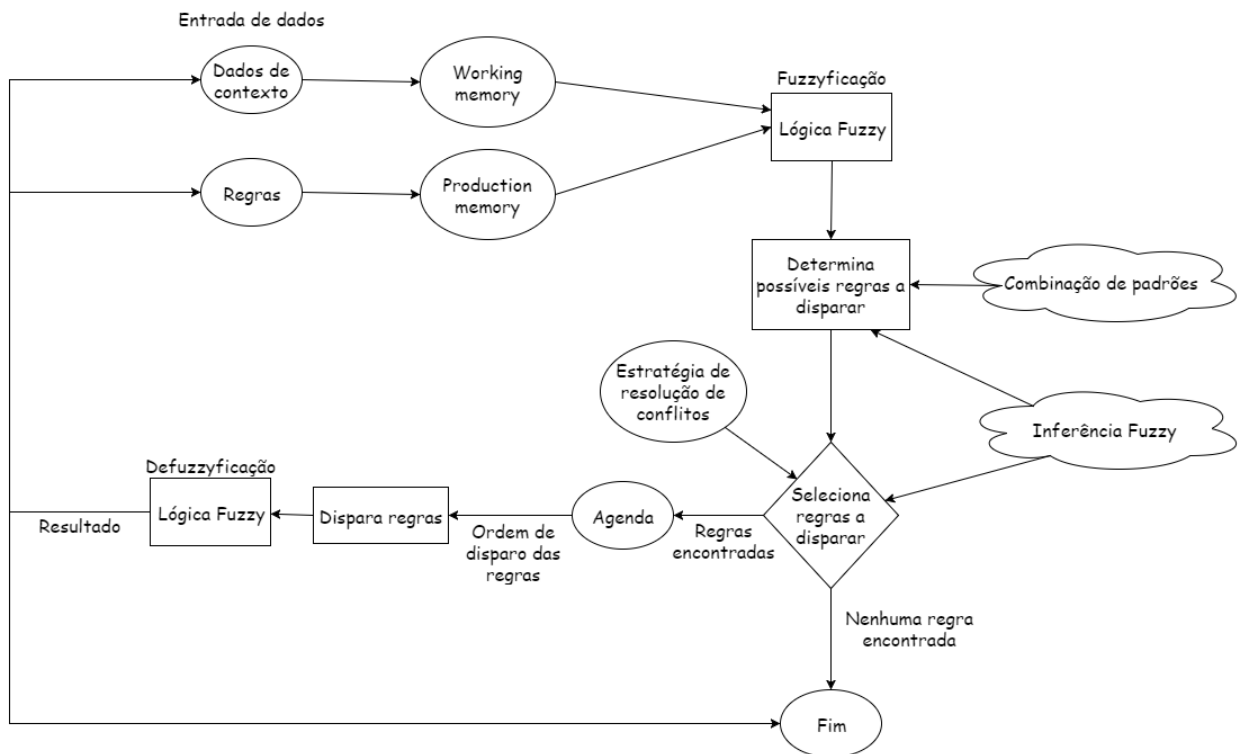
- **Situation Repository:** manipula o repositório onde são armazenadas as situações passadas e o conjunto de regras utilizadas para a definição das diferentes situações e ações que devem ser tomadas de acordo com as situações geradas. As regras são definidas inicialmente de forma estática, utilizando uma linguagem de domínio específico (DSL, do inglês Domain Specific Language) com vistas a facilitar o processo de criação e avaliação das regras. As situações passadas são utilizadas como base para a criação das regras. Quando uma situação diferente das conhecidas é encontrada, ocorre a incorporação de forma dinâmica da mesma ao repositório através de um processo de aprendizagem.
- **Situation Builder:** implementa o Nível 1 - Percepção do modelo de situação. Ele analisa os dados de contexto interno e externo do nodo e os agrupa em estruturas que serão utilizadas para a definição de uma situação;
- **Situation Analyzer:** implementa o Nível 2 - Compreensão, do modelo de situação. Esse subcomponente tem a finalidade de definir a situação do nodo. Ele utiliza os dados estruturados fornecidos pelo *Situation Builder* para buscar no *Situation Repository* regras compatíveis. Conforme apresentado na descrição do modelo, para a inferência de regras é utilizada Lógica Fuzzy implementada através do *Inference Engine*. Se não for encontrada uma regra compatível com os dados é sinalizado ao *Situation Repository* que uma situação não vista anteriormente foi encontrada e que a mesma precisa ser incorporada ao repositório de regras. Com isso, uma nova regra é criada dinamicamente para uso posterior;
- **Situation Projector:** implementa o Nível 3 - Projeção, do modelo de situação. A responsabilidade desse subcomponente é de gerenciar a inovação dos métodos de treinamento e predição desempenhados pelo *Forecaster*. É ele quem indica quais dados do histórico de contexto serão utilizados para treinamento da rede e quais dados preditos serão armazenados no *Context Store* para serem utilizados no processo de tomada de decisão desempenhado pelo *Decision Maker*.

→ Forecaster

É o subcomponente responsável pelas previsões necessárias para a projeção de situação (Nível 3 do modelo de situação). Ele é composto pelas seguintes classes:

- **Configuration Repository:** implementa o repositório que armazena as diferentes configurações que serão utilizadas para encontrar a Rede Neural mais adequada aos dados de contexto do nodo (treinamento da rede). Quando solicitado, ele busca no *Context Store* dados históricos do contexto interno do nodo e então executa um loop onde a cada interação, diferentes parâmetros de configurações globais da Rede (como por exemplo, *size of reservoir*, *sparsity of the reservoir*, *spectral radius* e *leaking rate*) são utilizados no treinamento e teste. O loop é encerrado quando todas as configurações disponíveis forem aplicadas. Quando isso acontece, a configuração de Rede que teve o menor MSE é salva para ser utilizada para a previsão. Essa estratégia foi adotada para que cada nodo possua uma configuração adequada aos seus dados de contexto;
- **Neural Network:** é empregada na fase de exploração da Rede Neural, após a mesma já estar treinada e adequada aos dados de entrada. Ele é o responsável por prever os valores de contexto que irão caracterizar a situação futura do nodo. Inicialmente é carregado do *Configuration Repository* a configuração da melhor Rede obtida na fase de treinamento. A Rede é então alimentada com o histórico de dados de contexto de baixo nível, indicados pelo *Situation Projector*, que estão armazenados na Camada 2 do *Context Store*. A Rede então efetua previsões de n passos adiante e as saídas previstas são repassadas ao *Situation Projector* que as armazena como dados de Camada 3 do *Context Store*. Além disso, as saídas previstas também são utilizadas para retroalimentação da Rede;
- **Prediction Analyzer:** verifica a acurácia das previsões realizadas. É invocado para calcular o erro das previsões efetuadas com relação aos dados de contexto coletados. Caso o erro esteja acima de um limiar definido no repositório de configurações o *Prediction Analyzer* indica que as configurações da Rede Neural estão defasadas. Nesse caso, é executado novamente o procedimento de encontrar a melhor configuração de Rede.

Figura 5.10: Processo de inferência utilizando encadeamento progressivo



(Fonte: o autor)

→ Inference Engine

O *Inference Engine* é baseado em um mecanismo de regras (rule engine). A escolha por um mecanismo de regras deve-se a maior flexibilidade na representação do conhecimento e também na possibilidade de separar a lógica de inferência do código fonte. O processo de inferência é representado na figura 5.10 e basicamente se caracteriza pela aplicação de regras carregadas na *Production Memory* sobre um conjunto de fatos que estão na *Working memory* para inferir conclusões que resultam em ações. Esse processo é chamado de combinação de padrões (pattern matching). Assim, o *Inference Engine* utiliza o processo de combinação de padrões para caracterizar situações utilizando dados de contexto que podem conter incertezas ou imprecisão. Por tal motivo, o *Inference Engine* emprega a teoria de conjuntos fuzzy apresentada na seção 2.6.1 para efetuar suas inferências.

Para o desenvolvimento das funções as seguintes classes são empregadas:

- **Fuzzy Rules Base:** disponibiliza o conjunto de regras do tipo “when condição then ação” que são empregadas no processo de inferência. Essas regras são descritas segundo o dialeto *MVFLEX Expression Language (MVEL)*. O MVEL é uma linguagem de expressão (expression language) dinâmica que originalmente foi con-

cebida como utilitária para páginas web dinâmicas em Java e acabou tornando-se um projeto desenvolvido de forma independente. Uma regra possui basicamente a seguinte estrutura:

```
rule "NAME"
when
    << condition >>
then
    << action >>
end
```

As regras do *Fuzzy Rules Base* são empregadas na fuzzyficação/defuzzyficação e também pelo *Pattern Matcher*.

- **Fuzzyfication / Defuzzification:** no processo de fuzzyficação, essa classe é responsável por receber valores precisos e determinar o grau de pertinência destes valores a determinados conjuntos fuzzy. Como saída obtêm-se um valor entre 0 e 1 que representa o grau de pertinência do valor a um conjunto fuzzy. A fuzzyficação e defuzzyficação ocorre usando por base as regras definidas no *Fuzzy Rules Base*.
- **Membership Function Base:** é a base onde fica armazenado o conjunto de funções de pertinência (MF) utilizadas pelas regras.
- **Conflict Mediator:** no processo de combinação da padrões pode acontecer de mais de uma regra ser selecionada para ser disparada. O *Conflict Mediator* é o responsável pela estratégia de qual regra é disparada primeiro. Para isso, primeiro é disparada a regra mais específica, em seguida a regra que usa um fato mais recente, depois a regra com maior prioridade e por fim a primeira regra da lista de regras que combinam com o padrão.
- **Agenda:** gerencia a ordem de execução das regras utilizando a estratégia definida pelo *Conflict Mediator*.
- **Pattern Macher:** é a classe responsável por gerar respostas a partir da base de conhecimento expressa em regras. A inferência sobre as regras utiliza encadeamento progressivo (forward chaining) através de uma implementação otimizada do algoritmo Rete(??) chamado de ReteOO. Para dar suporte ao tratamento de incertezas, o *Pattern Macher* faz uso da Lógica Fuzzy.

→ Decision Maker

É o subcomponente responsável pela tomada de decisões de encaminhamento e adaptações das funcionalidades internas.

- **Decisor:** decide quais mensagens devem ser encaminhadas para o nodo encontrado e quais ficam no buffer local além de indicar quais adaptações das funcionalidades internas devem ser efetuadas. Para isso, ele efetua a orquestração dos demais componentes para o cálculo do potencial do nodo (lembrando que o termo potencial é utilizado para representar a capacidade do nodo em ser uma boa “mula de dados”) e também controla as projeções de ações futuras executadas pelo *Situation Manager*;
- **Helper:** auxilia nas tomadas de decisão. Além disso, ele também desenvolve algumas funções de suporte: executa o cálculo do potencial do nodo com base nos dados do *Inference Engine*; analisa o impacto das decisões tomadas através de um conjunto de métricas internas que são utilizadas para melhorar o acerto das decisões de encaminhamento e adaptações efetuadas e também solicita a remoção de dados de contexto antigos para não exaurir recursos do nodo.

Uma vez definida a estrutura do *Situs*, a seguir será apresentado o seu comportamento interno através de diagramas de atividades da UML.

5.4 Comportamento da arquitetura

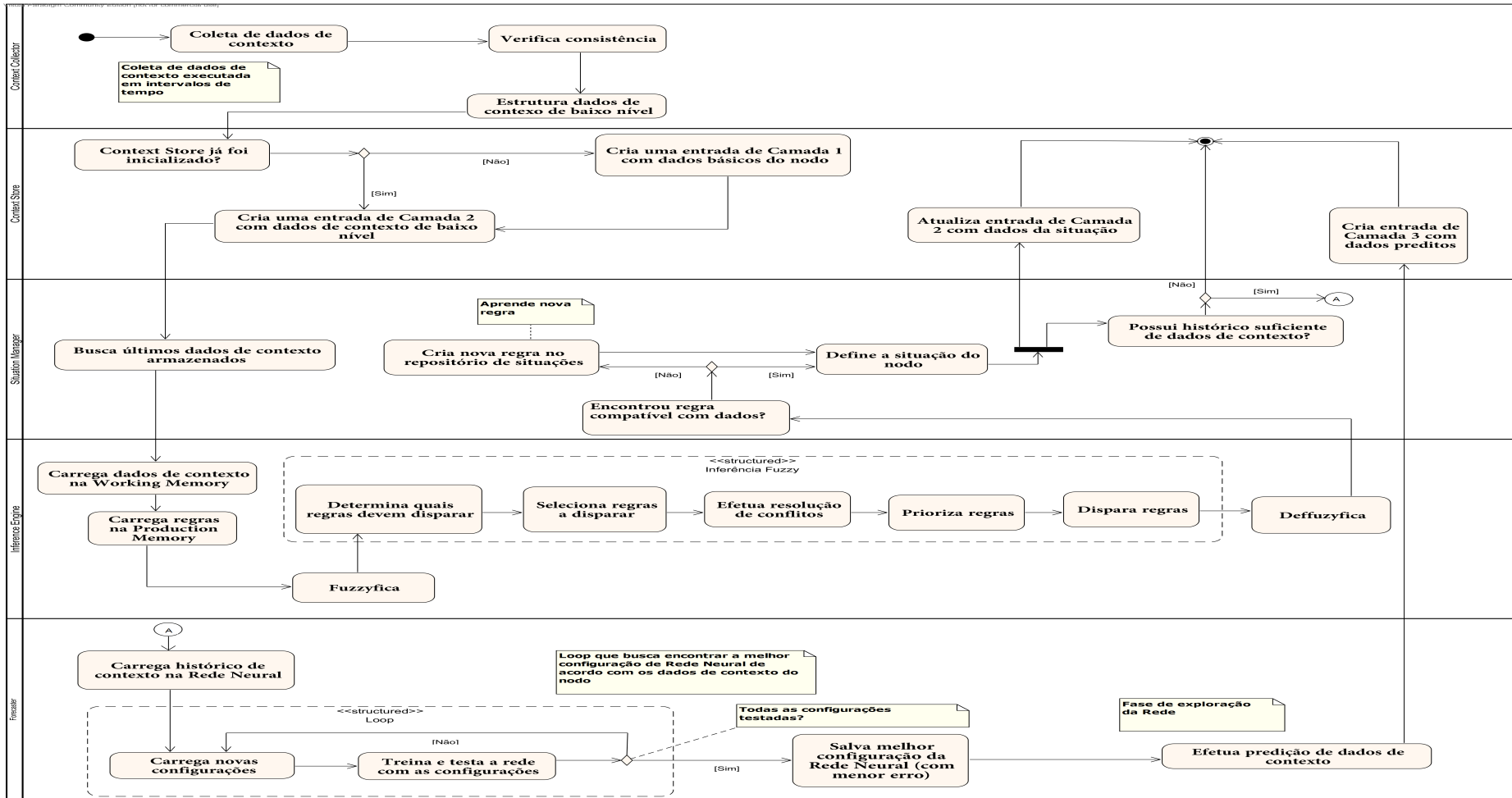
Para demonstrar o dinamismo da arquitetura, a seguir serão apresentados os diagramas de tratamento de contexto e tomada de decisões da arquitetura utilizando o diagrama de atividades da UML.

→ Tratamento de Contexto

O tratamento de contexto envolve as atividades necessárias para a coleta, inferência, tratamento de situação, predição e armazenamento dos dados de contexto que serão utilizados para tomada de decisão sobre o encaminhamento de mensagens. As atividades são apresentadas no diagrama da figura 5.11 e descritas textualmente da seguinte forma:

Em determinados intervalos de tempo o *Context Collector* efetua a coleta dos dados de contexto de baixo nível, verifica a consistência deles e então os agrupa em uma estrutura adequada para o armazenamento no *Context Store*. Antes de armazenar os dados é verificado se as variáveis internas do *Context Store* já foram inicializadas. Caso não

Figura 5.11: Diagrama de atividades do tratamento de contexto



(Fonte: o autor)

tenham sido, elas são inicializadas e então é criada uma entrada de camada 1 contendo as informações básicas do nodo. Em seguida os dados de contexto de baixo nível coletados são armazenados em uma entrada de camada 2.

Para iniciar o tratamento de situação, o *Situation Manager* busca os dados de contexto interno armazenados no *Context Store*. O *Inference Engine* então agrupa esses dados e os carrega na *Working Memory*. Em seguida as regras de inferência e funções de pertinência são carregadas na *Production Memory*. Os dados são então fuzzyficados para dar início ao processo de inferência fuzzy. Para isso são determinadas quais regras devem ser disparadas e então são selecionadas aquelas que combinam com os padrões de entrada fuzzyficados (pattern match). Caso mais de uma regra combine com os dados e sejam selecionadas para disparar é efetuada a resolução de conflito e então definida a prioridade de disparo de cada uma. A seguir, cada uma das regras é disparada e a saída é defuzzyficada para um valor entre 0 e 1 que representa o grau de pertinência a um conjunto fuzzy.

Se a saída da Inferência Fuzzy for nula significa que não foi encontrada uma regra compatível para os dados de entrada. Nesse caso, é criada uma nova regra através de um processo de aprendizagem que incorpora os parâmetros da regra ao repositório de regras. Caso a saída da inferência não seja nula é definida a situação do nodo e uma nova entrada de camada 2 com a situação definida é gravada no *Context Store*. Além disso, o *Situation Manager* verifica se existem dados de contexto suficiente para fazer a projeção de situação. Se não houver quantidade suficiente o processo encerra-se até o próximo ciclo de tempo. Caso contrário inicia-se a projeção.

Para a projeção, o *Situation Manager* indica que devem ser buscadas no *Context Store* por N entradas do histórico de contexto do nodo. Esses dados são fornecidos como entrada à Rede Neural. Porém, antes de fazer a predição é necessário o treinamento da Rede. Para o processo de encontrar a melhor Rede, é executado um loop que a cada interação carrega diferentes configurações globais da Rede, a treina e então efetua testes da mesma. Ao final do processo, todas as configurações disponíveis no repositório de configurações foram testadas e as configurações da Rede que apresentou a melhor acurácia (menor MSE) é salva. Essas configurações são utilizadas na fase de exploração da Rede para fazer a predição dos dados de contexto que serão utilizados para projetar uma situação futura. Os dados preditos são salvo como uma entrada de nível 3 no *Context Store* para serem utilizados para tomada de decisão.

→Tomada de Decisão

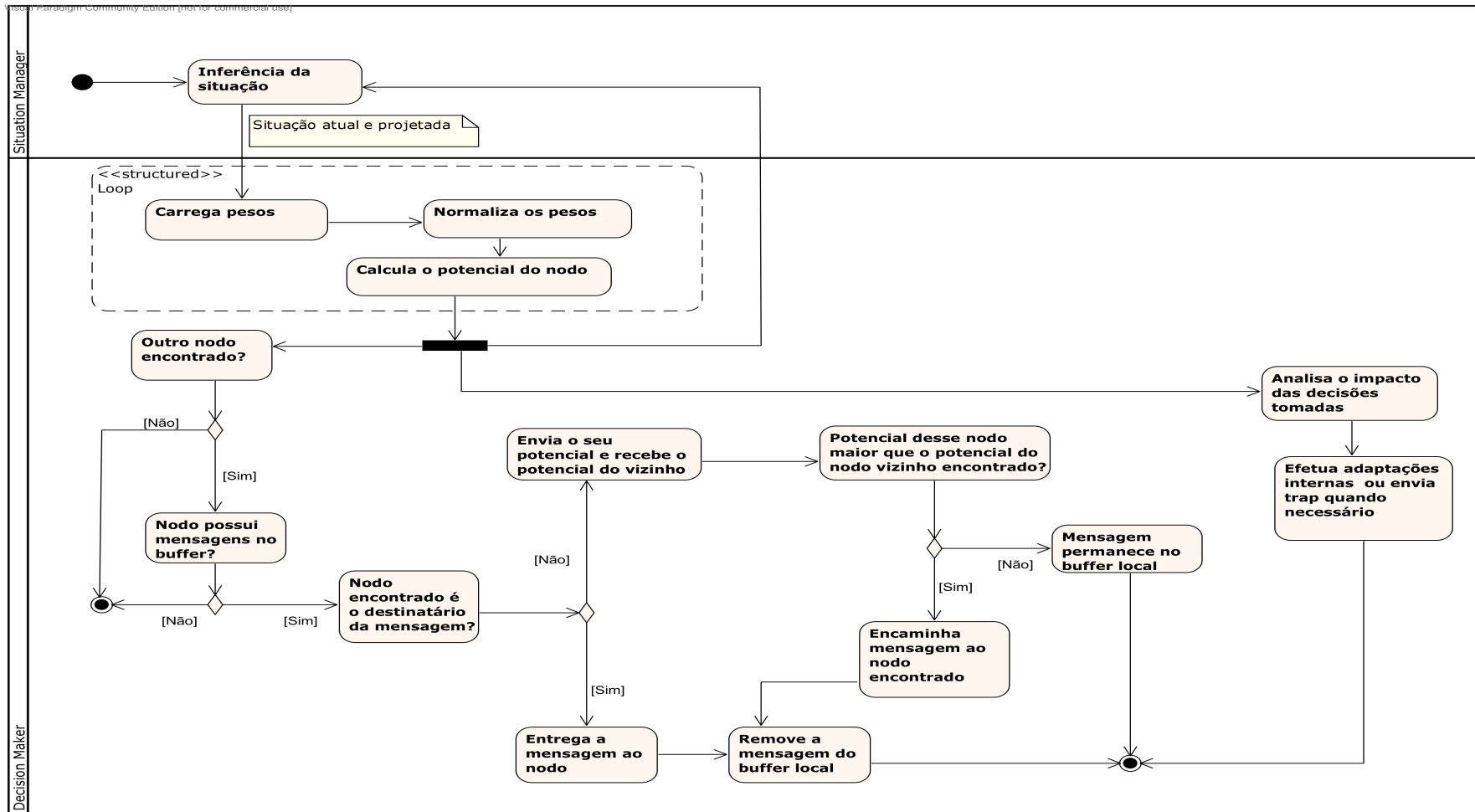
A tomada de decisão ocorre sempre que dois nodos se encontram e precisa ser decidido se alguma mensagem do buffer deve ser encaminhada ao nodo encontrado ou deve permanecer no buffer local. Lembrando que essa forma de encaminhamento segue o paradigma “store, carry and forward” e para isso é necessário decidir se o nodo encontrado é uma boa “mula de dados”. Conforme apresentado anteriormente, para essa decisão é empregado o termo “potencial” que representa a capacidade de um nodo ser um bom candidato a carregar as mensagens. As atividades necessárias para a tomada de decisão são apresentadas no diagrama da figura 5.12 e descritas textualmente da seguinte forma:

A tomada de decisão começa com o cálculo do potencial do nodo. Para isso são utilizados os dados da situação atual e projetada obtidos a partir do processo de inferência da situação. A seguir os pesos que representam a importância e impacto de cada uma das variáveis de contexto são carregados e normalizados. Os pesos normalizados e os dados de situação são utilizados então para o cálculo do potencial. Esse processo ocorre ciclicamente em determinados períodos de tempo de forma que o potencial do nodo sempre está atualizado de acordo com seus dados de contexto.

Quando outro nodo é encontrado verifica-se se existem mensagens no buffer que podem ser encaminhadas. Caso exista alguma mensagem, é analisado se o nodo encontrado é o destinatário de alguma, caso seja, a mensagem é entregue e removida do buffer local. Caso não seja, existe a possibilidade de encaminhamento e deve-se verificar se o outro nodo é uma boa “mula de dados” através da comparação do potencial de ambos. Para isso, é efetuada a troca mútua de potencial e comparado quem possui o maior valor. Se o potencial do nodo encontrado for maior, as mensagens são encaminhadas e removidas do buffer local. Caso contrário, as mensagens permanecem no buffer local até o próximo encontro.

Em paralelo ao processo de decisão de encaminhamento é efetuada uma análise das decisões tomadas. Essa análise busca verificar por exemplo se existem mensagens que estão ficando muito tempo no buffer e precisam ser encaminhadas mesmo que o potencial do nodo encontrado seja menor, se algumas mensagens não estão constantemente retornando ao nodo mesmo após já terem sido encaminhadas ou se o potencial calculado está variando muito bruscamente e o intervalo de tempo entre os cálculos precisa ser diminuído. Essas adaptações de variáveis internas visam tornar o comportamento do *Situs* mais adequado ao contexto no qual o nodo está inserido, melhorando assim as tomadas de decisões.

Figura 5.12: Diagrama de atividades da tomada de decisão



(Fonte: o autor)

5.5 Algoritmos

Com base nos modelos de comportamento apresentados anteriormente foram definidos algoritmos que representam as funcionalidades do textitSitus. A sua rotina inicial (algoritmo 1) é executada ciclicamente em intervalos de tempo, invocando os demais procedimentos. Eles serão descritos algorítmicamente a seguir.

Algoritmo 1 Component execution

```

1: Given: every node  $n \in N$ 
2: CONTEXTSTORE( $n$ )
3: for  $t \leftarrow 1; \dots; T$  do
4:   CONTEXTCOLLECTOR( $n$ )
5:   SITUATIONMANAGER( $n$ )
6:   if node  $n$  find a node  $k$  then                                ▷ encounter of two nodes
7:     DECISIONMAKER( $n$ )
8:   end if
9:   PRUNE OLD AND UNUSED DATA ROUTINE( $n$ )
10: end for

```

Context Store

Conforme apresentado anteriormente, o Context Store é responsável por armazenar e recuperar dados de contexto armazenados em cada nodo. Os principais procedimentos vinculados a ele são descritos no algoritmo 2. Primeiramente é verificado se as estruturas internas de armazenamento do nodo n_n foram inicializadas. Caso ainda não tenham sido, a mesma é inicializada.

Para efetuar armazenamento de dados é utilizado o procedimento $Store_Context(c, l)$ onde c são os dados de contexto que deseja-se armazenar e l indica em qual camada os dados devem ser armazenados (camada 1 - dados do nodo; camada 2 - dados de contexto; camada 3 - dados preditos). Uma vez armazenados, os dados podem ser recuperados utilizando $Get_Context(c)$ onde c indica o dado de contexto que deve ser retornado.

Algoritmo 2 Context Store

```

1: procedure CONTEXTSTORE( $n$ )
2:   if  $\neg$ IS_SET( $n_n$ ) then
3:     INITIALIZE( $n_n$ )
4:   end if
5: end procedure

6: procedure Store_Context( $c, l$ ) ▷ store data
7:   store  $c$  in layer  $l$  of  $G_n$ 
8: end procedure

9: procedure Get_Context( $c, v$ ) ▷ return stored data
10:   $d \leftarrow$  retrieve  $c$  from layer  $v$  of  $G_n$ 
11:  return  $d$ 
12: end procedure

```

Context Collector

A coleta dos dados de contexto é executada segundo o algoritmo 3. Inicialmente são inicializadas todas as variáveis internas do *Situs*. A seguir são coletadas informações do nodo fornecidas pelo sistema operacional I_n que serão armazenadas como uma entrada de camada 1 do Context Store. Então é executado um loop que coleta dados de sensoriamento C_e e os armazena como uma entrada de camada 2 do Context Store.

Algoritmo 3 Context Collector

```

1: initialize internal variables
2:  $C_i \leftarrow$  basic information about node
3: STORE_CONTEXT( $C_i, 1$ )
4: while true do
5:    $C_e \leftarrow$  get external sensed data
6:   STORE_CONTEXT( $C_e, 2$ )
7: end while

```

Forecaster

As atividades de predição de dados de contexto que serão utilizados para a projeção de uma situação futura do nodo são realizadas pelo algoritmo 4. Primeiramente é

verificado se as configurações da Rede não estão defasadas de acordo com o erro reportado pelo *Prediction Analyzer*. Caso positivo, é invocado o procedimento de treinamento da Rede. Nesse procedimento, diferentes configurações são testadas. A configuração que apresentar o menor erro de predição é armazenada e retornada e então criada uma nova Rede N com essas configurações. Se as configurações não estiverem defasadas, é criada uma nova Rede N com essas configurações. A a predição utilizando C como entrada é executada e retornada.

Algoritmo 4 Forecaster

```

1: procedure FORECASTER( $C$ )
2:    $error \leftarrow$  last error reported from Prediction Analyzer
3:   if  $error \geq \theta$  then                                     ▷ Configuration outdated
4:      $S \leftarrow$  TRAIN( $C$ )
5:      $N \leftarrow$  create new neural network with configuration  $S$ 
6:   else
7:      $S \leftarrow$  load last stored network configuration
8:      $N \leftarrow$  create new neural network with configuration  $S$ 
9:   end if
10:  return  $predict(N, C)$ 
11: end procedure

12: procedure TRAIN( $C$ )
13:  repeat                                                       ▷ find optimal neural network
14:     $S \leftarrow$  load new configuration
15:     $N_t \leftarrow$  create new neural network with configuration  $S$ 
16:     $e \leftarrow$  train and test  $N_t$  with configuration  $S$  and input  $C$ 
17:    if  $e < e_{(t-1)}$  then                                       ▷ compare error with previous interaction
18:       $S_b = S$ 
19:      STORE( $S_b$ )
20:    end if
21:  until test all configurations
22:  return  $S_b$ 
23: end procedure

24: procedure PREDICT( $N, C$ )                                     ▷ predict future context data
25:   $P \leftarrow$  run  $N$  with input  $C$ 
26:  return  $P$ 
27: end procedure

```

Situation Manager

O *Situation Manager* está descrito no algoritmo 5. Inicialmente são carregados os dados de contexto interno C_i e externo C_e do nodo n a partir do Context Store e também são carregadas as regras do Situation Repository e armazenadas em R . Na sequencia C_i e C_e são agrupados e armazenados em estruturas internas que representam somente os dados de contexto atuais C . Com isso está concluída a fase de percepção da situação

Na fase de compreensão da situação as regras fuzzy R são carregadas do repositório e utilizadas para fuzzyficar os dados de contexto C resultando no dados fuzzyficados C_f . A partir da aplicação das regras R sobre os dados C_f é efetuado o processo de inferência da situação resultando S_f . Se as regras conseguiram caracterizar a situação então é calculado o grau de pertiência de cada uma das variáveis de contexto que descrevem a situação e a situação S é definida. Se as regras não definiram a situação então uma situação não vista anteriormente foi encontrada. Nesse caso, é criada um novo conjunto de regras que são armazenados no repositório para uso na próxima vez que o *Situation Manager* for executado. Com isso, a fase de compreensão da situação é finalizada.

Para a fase de projeção é verificado se o tamanho de dados históricos de contexto é maior que um total τ . Caso positivo, significa que o *Situation Manager* possui capacidade de projetar situações. Então os dados de contexto são passados para *Forecaster* e o retorno é predições é armazenado como uma entrada de nível 3 no *Context Store*. Esse dados serão utilizados depois pelo processo de tomada de decisão executado pelo *Decision Maker*. Por fim, ocorre a defuzzyficação e a situação definida é retornada.

Algoritmo 5 Situation Manager

```

1: procedure SITUATIONMANAGER(n)
2:    $C_i \leftarrow \text{GET\_CONTEXT}(n, 1)$       ▷ load internal context data from Context Store
3:    $C_e \leftarrow \text{GET\_CONTEXT}(n, 2)$       ▷ load external context data from Context Store
4:    $C \leftarrow C_i \cup C_e$                   ▷ situation perception
5:    $R \leftarrow$  load fuzzy rules from repository      ▷ start situation comprehension
6:    $C_f \leftarrow \text{FUZZIFY}(C)$               ▷ Fuzzyfication of C
7:    $S_f \leftarrow \text{FUZZY\_INFERENCE}(C_f, R)$     ▷ Inference of situation using fuzzy rules
8:   if  $S_f \supseteq R = \text{true}$  then              ▷ current situation is found
9:      $S \leftarrow \text{CALCULATE\_DEGREE}(R)$       ▷ calculate degree of situation variables
10:  else                                       ▷ new/unknown situation
11:     $R_n \leftarrow \text{CREATE\_RULE}(C)$           ▷ create new rule
12:     $\text{STORE\_RULE}(R_n)$                     ▷ store new rule in repository
13:  end if                                     ▷ end situation comprehension
14:  if  $\text{sizeof}(C) > \tau$  then                ▷ situation projection
15:     $S_p \leftarrow \text{FORECASTER}(C)$ 
16:     $\text{STORE\_CONTEXT}(S_p, 3)$ 
17:  end if
18:   $S \leftarrow \text{DEFUZZIFY}(S_f)$               ▷ Defuzzyfication
19:  return S
20: end procedure

```

Decision Maker

O procedimento de tomada de decisão efetuado pelo *Decision Maker* do nodo n é representado no algoritmo 6 e descrito da seguinte forma: Em intervalos de tempo a situação inferida pelo *Situation Manager* é lida e carregada em S juntamente com os pesos W que serão normalizados em W' . O potencial do nodo P é calculado utilizando os dados da situação S juntamente com os pesos normalizados W' .

Quando o nodo corrente encontra outro nodo é verificado se ele é o destinatário de alguma das mensagens que estão no buffer local. Caso positivo, as mensagens são entregues diretamente ao destino. Caso contrário, o potencial P dos nodos é trocado mutuamente.

Então é verificado se existem mensagens no buffer. Para cada uma das mensagens existentes é verificado se o potencial do nodo encontrado é maior que o potencial do nodo corrente. Caso positivo a mensagem é encaminhada para o outro nodo e apagada do buffer

local. Caso contrário, ela permanece no buffer.

Por fim, são analisados os impactos das decisões tomadas e caso seja atingido determinado limiar, são executadas adaptações internas do *Situs* ou enviada uma notificação indicando que uma adaptação externa é necessária.

Algoritmo 6 Decision Maker

```

1: procedure DECISIONMAKER(n)
2:   At interval of time calculate potential of node {
3:      $S \leftarrow \text{LOAD\_SITUATION}$ 
4:      $W \leftarrow \text{LOAD\_WEIGHTS}$ 
5:      $W' \leftarrow \text{NORMALIZE}(W)$ 
6:      $P \leftarrow \text{CALCULATE\_POTENTIAL}(S,W')$ 
7:   }
8:   if other node found then
9:     if this node have a message m to other node then  $\triangleright$  found final destination
10:      delivery message m to other node
11:     else
12:       exchange  $P$  with other node  $\triangleright$  mutual exchange of potential
13:       for all messages m in buffer do
14:         if P of other node  $>$ P of this node then  $\triangleright$  good data mule
15:           forward m to other node
16:           remove m from local buffer
17:         else
18:           not forward m
19:         end if
20:       end for
21:        $L \leftarrow$  analyze impact of decisions
22:       if  $L > \theta$  then
23:         adjust internal parameters or send trap for external adaptation
24:       end if
25:     end if
26:   end if
27: end procedure

```

5.6 Estratégia de implementação

A partir da definição de funcionalidade de cada um dos componentes, foi efetuada a implementação. A seguir será demonstrado como foram implementados cada um dos níveis do modelo de Sensibilidade à Situação:

→ Nível 1 - Percepção

A percepção de situação envolve a coleta dos valores dos atributos de contexto de baixo nível e a estruturação desses valores para serem armazenados para uso posterior. Os seguintes dados de cada nodo foram coletados: nome do nodo, endereço, tipo de nodo, timestamp, velocidade das interfaces de rede, energia da bateria, velocidade atual, velocidade média, distância percorrida desde o último encontro, total da distância percorrida, últimas coordenadas, espaço disponível no buffer, número de mensagens armazenadas localmente, total de mensagens encaminhadas, número de nodos vizinhos e número total de conexões.

O exemplo de implementação desse nível é demonstrado no código fonte 5.1. São utilizados métodos específicos para a leitura do valor cada um desses atributos. Após a leitura é criado um objeto *context* com métodos para definição dos valores. Por fim, o objeto *context* é salvo para uso posterior.

```
Context context = new Context(nodeName, nodeType, nodeAddress);
context.setTimestamp(time);
context.setPower(power);
context.setPowerPercent(power_percent);
context.setDistance ( distance );
context.setTotaDistance ( total_distance );
context.setNumConnections(numConnections);
context.setTotalMessages ( totalMessages );
context.setNumMessages(numMessages);
context.setBuffer (bufferOccupacy);
context.setSpeed(speed);
storeContext ( context );
```

Código fonte 5.1: Implementação da percepção de situação

→ Nível 2 - Compreensão

A compreensão da situação envolve a manipulação dos dados de contexto que foram coletados pelo nível 1. Para as atividades de compreensão da situação foi utilizado como base o JBoss Drools¹. O Drools é uma de sistema de inferência híbrido baseado em regras. Seu motor de inferência permite a construção de regras que serão avaliadas através de algoritmos baseados na combinação de encadeamento progressivo e regressivo, como RETE networks (??) (no Drools chamado de ReteOO para indicar suporte à orientação a objetos) e também uma evolução desse algoritmo chamado de PHREAK.

Para definição de regras, o Drools utiliza uma linguagem própria chamada DRL (Drools Rule Language) que é compatível com o dialeto MVEL (MVFLX Expression

¹<http://www.drools.org/>

Language). Um arquivo DRL, basicamente, contém um conjunto de regras do tipo “ when LHS then RHS” onde LHS (Left Hand Side) ou antecedente corresponde aos elementos condicionais da regra e RHS (Right Hand Side), também chamado consequente, corresponde as ações que serão executadas em caso de disparo da regra. Tanto o LHS e o RHS podem ser formados por conjuntos de zero, um ou mais itens.

Como o Drools nativamente somente é capaz de utilizar regras sem tratamento de incertezas, foi utilizado uma extensão do sistema chamada de Drools Chance ou simplesmente, Chance(????). O Chance permite que as regras sejam definidas e avaliadas utilizando Lógica Fuzzy. Dessa forma, a partir das definições de funções de pertinência e da aplicação dos algoritmos de inferência Fuzzy consegue-se efetuar o tratamento de incertezas conforme requerido pelo modelo de situação apresentado na seção 5.2. O Chance segue a mesma lógica de declaração de variáveis e regras de um arquivo DRL normal. Entretanto, deve-se declarar a forma como os valores serão manipulados pela Lógica Fuzzy. Para isso, as variáveis linguísticas são definidas através de funções de pertinência para então serem utilizadas nas regras.

Foram criadas regras onde cada uma dos atributos de contexto de baixo nível teve uma variável linguística definida com três funções triangulares com valores variando em escala de “pouco”, “médio” e “alto” de acordo com sua magnitude. Um exemplo de declaração dessas variáveis linguísticas pode ser visto no código fonte 5.2 abaixo, onde foi declarada a variável para permitir a inferência da situação da bateria do nodo.

```
declare enum POWER
  @LinguisticPartition
  LOW("low", new de.lab4inf.fuzzy.polygons.FuzzyTriangle( -0.01, 4000, 7000 ) ),
  MEDIUM("medium", new de.lab4inf.fuzzy.polygons.FuzzyTriangle( 6000, 10500, 15000
  ) ),
  HIGH("high", new de.lab4inf.fuzzy.polygons.FuzzyTriangle( 14000, 17000, 20000 ) );
end
```

Código fonte 5.2: Variável linguística representando a quantidade de bateria do nodo

Após isso foi criada uma variável que armazena o valor de contexto coletado e então é efetuado o vínculo com a variável linguística correspondente para permitir a manipulação via Lógica Fuzzy (código fonte 5.3).

```

declare _Situation
...
_power      : POWER
    @Imperfect( kind=ImpKind.FUZZINESS, type=ImpType.LINGUISTIC,
    degree=DegreeType.SIMPLE, support="power" )
...
power       : Double
...
end

```

Código fonte 5.3: Vinculação com a variável linguística

Essas variáveis foram empregadas em regras que são definidas segundo uma variante do padrão JSR94² utilizado pelo Drools (código fonte 5.4).

```

rule " situation_power "
dialect "mvel"
no-loop
lock-on-active
when
    s : _Situation (power ~is _power)
    @Imperfect( label =power )
then
    map.put( " situation_power ",    chance.getDegree( "power" ).value );
end

```

Código fonte 5.4: Regra para inferência da situação da energia do nodo

As regras definidas são utilizadas pelo sistema de inferência fuzzy que como saída produz valores que representam o grau de pertinência de cada atributo de contexto de baixo nível a determinada situação. Esses valores serão utilizados para o cálculo do potencial do nodo.

→ Nível 3 - Projeção

Uma vez que foi definida a situação de cada atributo de contexto é efetuada a projeção através da predição de valores futuros. Para a implementação da predição é utilizada uma Rede Neural ESN (os detalhes dessa Rede foram apresentados na seção 2.6.2.1). Para isso, foi empregado o software ESNJava³. A escolha do ESNJava deve-se ao fato dele ser a única biblioteca encontrada que possui uma API para embutir uma ESN em aplicações Java.

O código fonte 5.5 apresenta de forma resumida como ocorre a predição de valores

²JSR94 Rule Engine API. <https://www.jcp.org/en/jsr/detail?id=94>

³<http://www.wsi.uni-tuebingen.de/lehrstuehle/cognitive-modeling/code/overview.html>

utilizando a ESN. Inicialmente é criado um objeto *forecaster* que é responsável por todas as atividades de treinamento e predição de dados utilizando a Rede Neural. O histórico de dados de contexto de baixo nível é fornecido como entrada e saída esperada na fase de treinamento da ESN. Para fins de ensinar a rede que os dados atuais dependem do contexto anterior, a saída esperada é definida como a entrada defasada de um ciclo de tempo. Em seguida são definidas as variáveis de configuração da Rede de acordo com o tamanho do histórico dos dados. Então é invocado o método responsável pelo treinamento da ESN. Nessa fase de treinamento, conforme apresentado na seção 5.2.2, são testadas diferentes configurações da Rede até ser encontrada a Rede Neural ótima cujas configurações são salvas para serem utilizadas na predição.

```

...
Forecaster_ESN forecaster = new Forecaster_ESN();

forecaster .setData(inputData, outputData);

// internal configuration of forecaster
forecaster .setTestSteps ( (inputData .size () / 2 ) / 2);
forecaster .setExploitationSteps (inputData .size () / 2);
forecaster .setTrainingSteps ( (inputData .size () / 2 ) / 2);

// train
forecaster .trainNetworks ();

// if best network was found
if ( forecaster .isTrained () {

    // steps to predict
    forecaster .setStepsAhead(5);

    // predict
    forecaster .forecast ();

    // get predicted values
    double predicted [][] = forecaster .getNetworkOutput();

}
...

```

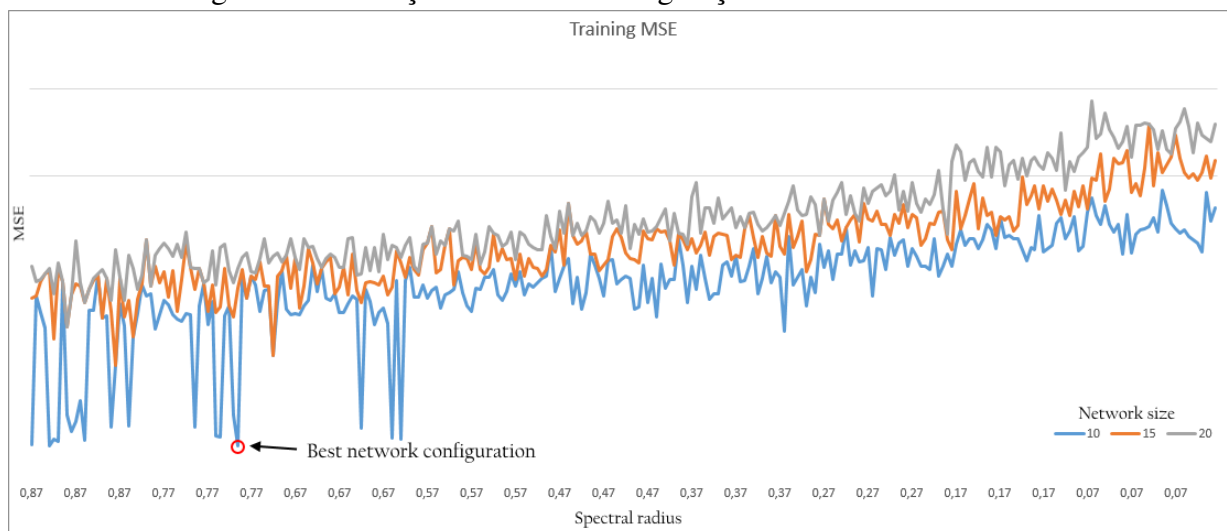
Código fonte 5.5: Predição utilizando a ESN

Como cada nodo das simulações possui a sua própria Rede, construída de forma autônoma, especificamente de acordo com seus dados, é inviável mostrar todas as configurações utilizadas. Os tamanhos dos reservatórios da Rede de cada nodo variaram entre 10 e 20 neurônios com raio de espectro (*spectral radius*) setado entre 0.77 a 0.85. Os

únicos valores que ficaram fixos para todos os nodos foi o espaçamento do reservatório (*Sparsity of the Reservoir*) definido com valor 1 e a taxa de vazamento (*Leaking Rate*) com valor 0.

A figura 5.13 demonstra graficamente o processo de seleção da melhor configuração de Rede em um nodo específico da simulação (lembrando que cada nodo possui sua própria Rede, treinada de acordo com seus dados). Pode-se observar que o tamanho da Rede foi definido com 10 neurônios e o raio de espectro com valor 0.77.

Figura 5.13: Seleção da melhor configuração de Rede Neural



(Fonte: o autor)

Em seguida, é verificado se foi encontrada a melhor configuração de Rede capaz de efetuar previsões. Caso tenha sido encontrada, são definidos quantos passos adiante deseja-se prever para então invocar o método que executa as previsões. Por fim, são obtidos os valores preditos que serão salvos para uso no cálculo do potencial do nodo.

Com essas configurações, em média 104 passos foram utilizados na fase de treinamento e 54 na fase de predição (exploração). O MSE⁴ na fase de treinamento ficou entre $4.01e^{-8}$ e $4.55e^{-10}$ e na de exploração variou entre $1.21e^{-7}$ e $7.67e^{-8}$. Esses valores demonstram que a Rede conseguiu aprender e prever com uma taxa de acerto aceitável.

5.7 Considerações finais do capítulo

Esse capítulo apresentou a proposta do *Situs*, um componente de software que aplica Sensibilidade à Situação em Redes Oportunistas para melhorar a troca de dados

⁴MSE ou Erro Quadrático Médio, mensura a diferença entre a saída atual obtida e a saída esperada. Quanto mais distante de 0 o valor da diferença, mais imperfeita a simulação (??)

em aplicações de Sensoriamento Urbano. Ele será embutido no módulo de comunicação da arquitetura de Sensoriamento Urbano chamada UrboSenti. Para a sua definição foi adotada uma abordagem *top-down*. Inicialmente foi descrito um cenário de aplicação de Sensoriamento Urbano em Cidades Inteligentes e então demonstrado que esse cenário pode ser estruturado em 3 camadas. A Camada 1 representando os dispositivos e aplicações utilizadas no sensoriamento; a Camada 2 representando a arquitetura computacional responsável pela análise dos dados coletados e; a Camada 3 representando os usuários e demais sistemas que acessam as informações. A partir da estruturação dessas camadas e com base em diferentes autores foram definidos um conjunto de requisitos. Foi explicitado então que o *Situs* irá contribuir para o atendimentos dos requisitos da Camada 1.

Para aprofundar um pouco o nível de abstração, foi utilizado um mapa mental com vistas a esquematizar o modelo da Camada 1 do cenário-problema. O objetivo dessa modelagem foi de proporcionar uma maior compreensão das entidades existentes na Camada e também as relações existentes. Além disso, foi possível demonstrar onde o componente irá atuar e como as abordagens são aplicadas.

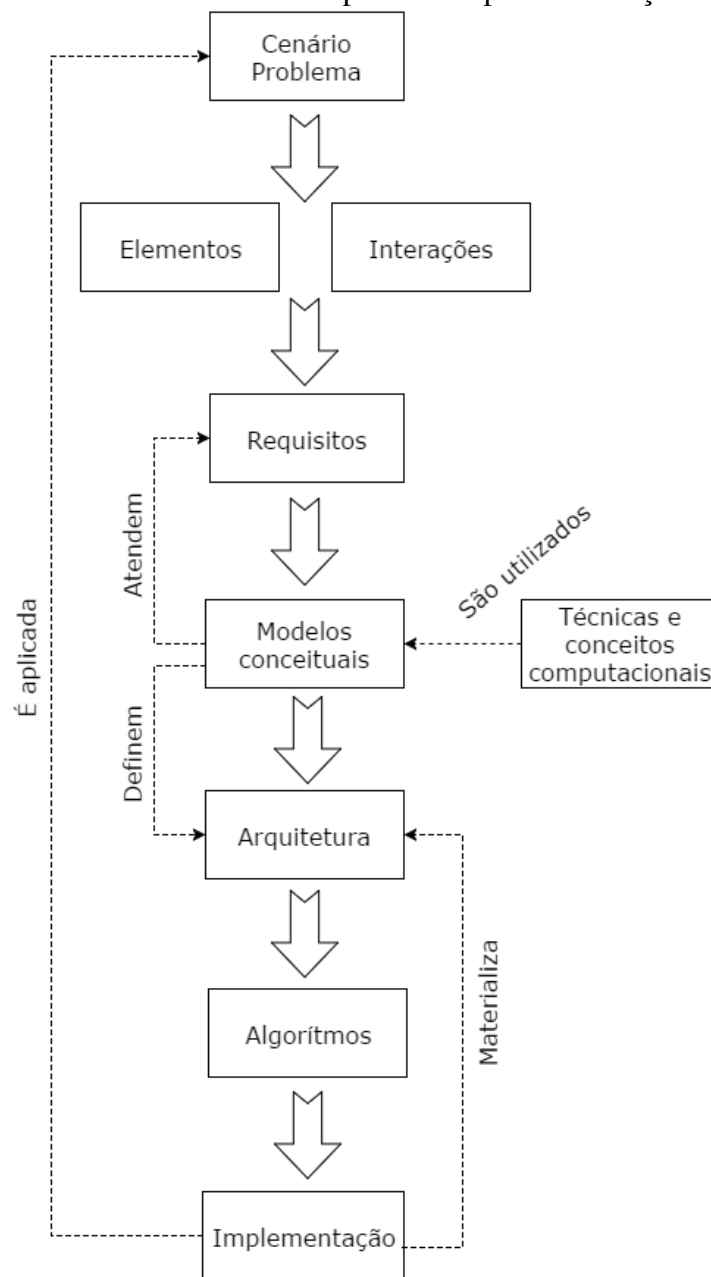
Na sequência, o foco de abstração passou a ser a definição dos modelos que compõem o *Situs*. Foi definido o modelo de contexto empregado juntamente com os atributos que serão utilizado pelos componentes internos da arquitetura. A seguir foi definido o modelo de situação baseado nos 3 níveis: percepção, compreensão e projeção. Cada nível foi implementado por um submodelo que emprega técnicas de Inteligência Computacional (Lógica Fuzzy e Redes Neurais) para o desenvolvimento de suas funções.

Aprofundando o nível de abstração, na próxima etapa foi definida a arquitetura do *Situs*. Foram descritos cada um dos seus componentes, o motivo de sua existência, as suas funcionalidades internas e também detalhados os seus subcomponentes. Ainda nessa etapa, foi apresentado o comportamento destes componentes através de diagramas de atividades utilizando a UML. Após isso, se aproximando do nível de implementação, foram apresentados os algoritmos que contemplam as funcionalidades e por fim demonstrado como foi efetuada a codificação dos componentes.

Destaca-se que a materialização do *Situs* é o resultado de todo um caminho percorrido desde a definição do cenário-problema até a apresentação de uma proposta que é capaz de atender os requisitos apresentados. O caminho analítico percorrido para o desenvolvimento do *Situs* é apresentado na figura 5.14.

Por fim, pode-se apontar que os principais diferenciais do componente referem-se

Figura 5.14: Caminho analítico percorrido para a definição do *Situs*



(Fonte: o autor)

ao uso de Sensibilidade à Situação e da aplicação de técnicas de Inteligência Computacional (mais especificamente Redes Neurais e Lógica Fuzzy) para proporcionar a proatividade nas decisões de roteamento e adaptação e assim melhorar a disseminação das mensagens das aplicações de Sensoriamento Urbano.

No próximo capítulo serão apresentados experimentos que buscam verificar o comportamento e a performance do *Situs* através de experimentos em ambientes simulados.

6 AVALIAÇÃO DE DESEMPENHO

Esse capítulo irá apresentar os resultados de avaliações de performance e comportamento do *Situs* em cenários que simulam o comportamento de uma Cidade Inteligente.

A ideia usada nas simulações é de que existam nodos fixos e móveis efetuando o sensoriamento ao longo da cidade. Através do paradigma de Redes Oportunistas os dados de sensoriamento são encaminhados a cada contato entre os nodos até um *Access Point* (AP). Os APs foram posicionados em locais estratégicos na cidade e são os responsáveis por receberem os dados e os transmitirem via Internet para serem processados por um sistema central.

As configurações desses cenários empregados não simulações são descritas a seguir juntamente com as métricas de avaliação que serão utilizadas. São então apresentados os experimentos e analisados os resultados obtidos. Por fim, são efetuadas considerações sobre o seu desempenho.

6.1 Descrição do ambiente

6.1.1 Configuração das simulações

Todas as simulações foram executadas utilizando o ONE (Opportunistic Network Environment) Simulator¹. A sua escolha deve-se a ser um simulador específico para Redes Oportunistas, com código aberto, extensível e já utilizado pela comunidade acadêmica da área de Redes Oportunistas em suas simulações (??).

Segundo ??) o uso de sensores fixos e móveis, carregados por humanos (como por exemplo smartphones) ou instalados em veículos (carros e ônibus) fornecem um ecossistema de sensoriamento propício para serem aplicados em ambientes de larga escala, como uma cidade inteligente. Dessa forma, para simular a troca de dados de sensoriamento foram utilizados tanto nodos fixos quanto nodos móveis dispersos ao longo da cidade.

Os nodos móveis foram definidos em 2 grupos: pessoas e veículos. Em cada uma das simulações foi adotada uma quantidade crescente de nodos (10 nodos de cada grupo na primeira simulação, 25 de cada grupo a segunda, 50, 75, 100, 500 e 1000 nodos de cada grupo na última simulação. Ou seja, no último cenário existiam 1000 veículos e 1000 pessoas ao longo da cidade efetuando o sensoriamento. Ambos os grupos se

¹<http://www.netlab.tkk.fi/tutkimus/dtn/theone/>

movimentavam no mapa utilizando o modelo de mobilidade *Shortest Path Map Based Movement* (nesse modelo, os nodos utilizam caminhos aleatórios disponíveis no mapa e utilizam o algoritmo de Dijkstra para encontrar o caminho mais curto entre os nodos e Pontos de Interesse).

Foram aplicadas restrições de energia baseadas no trabalho de ??): todos os nodos iniciaram com uma carga completa de bateria de 19080 Joules (que corresponde a uma bateria comumente encontrada em celulares de 5.3W/h com 3.7V) e uma recarga sendo efetuada randomicamente em intervalos entre 4 horas e 4 horas e meia. A energia gasta para a busca de outros nodos foi definida em 0.092 mW/s e o gasto para enviar e receber dados definido em 0.08 mW/s.

O grupo formado por pessoas se movimentava com velocidade entre 0.5 e 1.5 Km/h e tinham um celular equipado com uma interface Wifi com alcance de 20 metros e 10Mbps de velocidade de transmissão.

O grupo de carros se movimentava com velocidades entre 10-50 km/h e possuía uma interface WiFi com as mesmas características do grupo de pessoas. Tanto o grupo de carros quanto o de pessoas foi definido com um buffer para armazenamento de dados com 50M de tamanho.

Os nodos fixos foram definidos em 2 grupos: um grupo de sensores fixos e outro de APs. No grupo de sensores fixos, foram criados 10 nodos posicionados estaticamente em locais variados ao longo da cidade. Esses sensores são os responsáveis pela coleta de dados como níveis de intensidade sonora, qualidade do ar, etc. Cada um deles foi equipado com uma interface WiFi com alcance de 20 metros e 10Mbps de velocidade de transmissão. Para armazenamento dos dados, cada nodo foi definido com um buffer de 256M de tamanho.

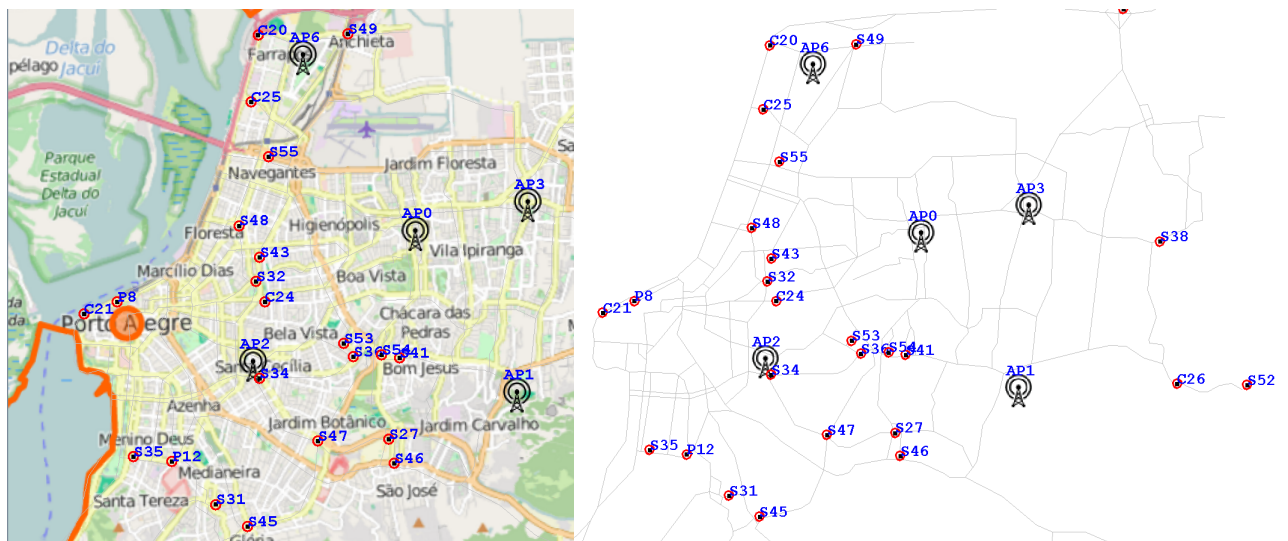
No outro grupo de nodos fixos, foram criados 7 APs. Eles foram colocados em locais estratégicos da cidade com a finalidade de receber os dados de sensoriamento coletados pelos outros nodos (estáticos e móveis) e então os encaminhá-los via rede cabeada para serem processados por um sistema centralizado. Para isso, eles foram equipados com uma interface WiFi com alcance de 30 metros e 10Mbps de velocidade de transmissão e uma interface cabeada com 1Gb de velocidade. Por serem equipamentos dedicados, esses APs estavam conectados permanentemente na energia, não precisando de recarga de elétrica, e possuíam o buffer para armazenamento dos dados com 512M de tamanho.

Tanto os sensores fixos quanto os sensores móveis foram equipados com uma aplicação responsável por simular o sensoriamento de eventos ao longo da cidade. Os

eventos de sensoriamento ocorreram em intervalos de 200 segundos, sendo que os dados foram gerados com tamanho variando segundo uma distribuição uniforme de 100k a 2M. Cada mensagem com os dados coletados foi definida com um TTL de 300 segundos (5 horas). Esse é o tempo máximo para que uma mensagem seja roteada no cenário e então entregue ao primeiro APs que o nodo que carrega tal mensagem tiver contato.

As simulações foram executadas durante um período de 12 horas em um cenário da cidade de Porto Alegre/RS. Para isso, foi obtido o mapa da cidade no formato OSM disponibilizado pelo projeto Open Street Map² e então com o aplicativo OpenJump³ a cidade teve as suas principais avenidas mapeadas e então suas coordenadas foram exportadas no formato WKT (Well-known text) que é reconhecido pelo simulador. Para dar uma ideia do ambiente de simulação, a figura 6.1 apresenta parcialmente o mapa com e sem overlay da cidade para o cenário com 10 nodos em cada grupo.

Figura 6.1: Visão parcial das principais avenidas de Porto Alegre mapeadas no simulador
(a) Mapa com overlay (b) Mapa sem overlay



6.1.2 Métricas de avaliação

Para as avaliações foram utilizadas as seguintes métricas de performance propostas por ??) e suportadas pelo ONE Simulator:

- Created: Quantidade de mensagens criadas durante as simulações – elas não incluem mensagens replicadas. O ideal é que o número de mensagens criadas seja o

²<http://www.openstreetmap.org/>

³<http://www.openjump.org/>

mesmo de mensagens entregues. Porém, devido ao tipo de cenário algumas mensagens serão perdidas;

- **Started:** Quantidade de transmissões iniciadas entre nodos da rede, independentemente se a transmissão foi bem sucedida ou não. O valor de mensagens iniciadas tende a crescer a medida que a quantidade de nodos no cenário aumente e com isso mais contatos ocorrerão;
- **Relayed:** Quantidade de mensagens encaminhadas entre nodos. Essa métrica impacta diretamente no custo para entrega das mensagens. Se ocorrerem muitos encaminhamentos e poucas entregas o custo será alto. O ideal é que exista um equilíbrio entre a quantidade de entregar e de encaminhamentos;
- **Delivered:** Quantidade de mensagens entregues durante a simulação. É a principal métrica. Ela indica o quanto a solução está conseguindo tomar boas decisões que impactarão na quantidade de mensagens entregues;
- **Latency average:** Média de tempo de entrega da mensagem desde a sua criação. Quanto menor esse valor, mais rápido está conseguindo-se entregar mensagens aos destinos;
- **Buffer average:** Média de tempo que as mensagens ficaram armazenada em buffer em cada nodo. Um valor muito alto indica que a solução está demorando muito tempo para encaminhar as mensagens e isso pode impactar em atrasos na entrega;
- **Hop count average:** Média de nodos utilizados como intermediários entre a origem e destino das mensagens;
- **Overhead ratio:** Uma medida de eficiência do uso da banda. Utiliza a fórmula $\frac{R-D}{D}$, sendo R = Relayed messages e D = Delivered messages. Quanto maior for o valor dessa métrica, mais alto o custo para entregar uma mensagem e conseqüentemente, pior é a performance da solução.

6.2 Experimentos

Para analisar o comportamento do *Situs* foram efetuados os experimentos que serão apresentados a seguir.

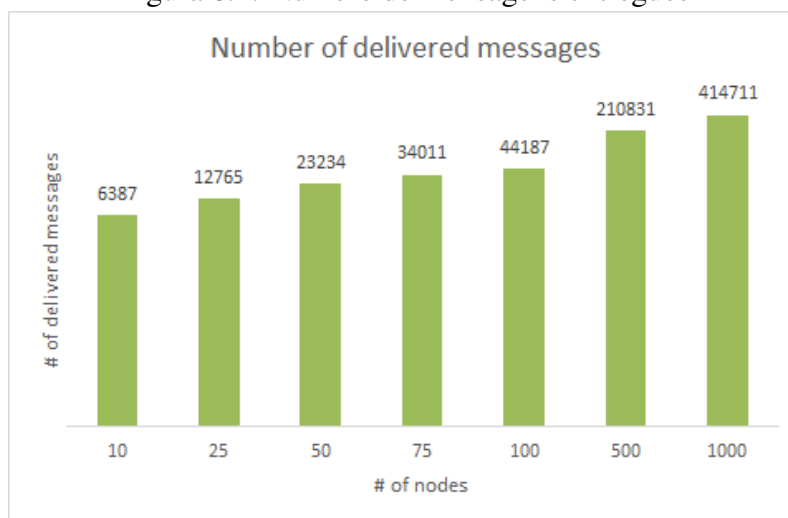
6.2.1 Experimento 1 - Teste de performance e comportamento

O primeiro experimento teve por objetivo verificar a performance do *Situs* e o impacto da quantidade de nodos no total de mensagens entregues ao destinatário. Para isso, foram utilizados cenários com um número crescente de nodos. A quantidade de sensores fixos e APs permaneceram as mesmas em todos os cenários (10 sensores fixos e 7 APs). O primeiro cenário foi definido com 10 nodos móveis de cada grupo (veículos e pedestres), o segundo com 25, e então respectivamente 50, 75, 100, 500 e 1000 nodos.

→ Mensagens entregues

A partir dos resultados, pode-se notar que a quantidade de nodos existente no cenário impacta diretamente na quantidade de mensagens entregues (figura 6.2). Aumentando 10 vezes a quantidade de nodos, de 10 para 100, ocorreu um incremento de cerca de 690% na taxa de mensagens entregues (*delivered*). Se for feito um comparativo entre os cenários de 500 e 1000 nodos houve um incremento de cerca de 96% nessa métrica. Entre os cenários de 50 e 100 nodos ocorreu um incremento de 90% de mensagens entregues. Ou seja, observa-se que por mais que ocorra um aumento de 100% na da quantidade de nodos nos cenários essa não será a taxa de aumento de mensagens entregues.

Figura 6.2: Número de mensagens entregues



(Fonte: o autor)

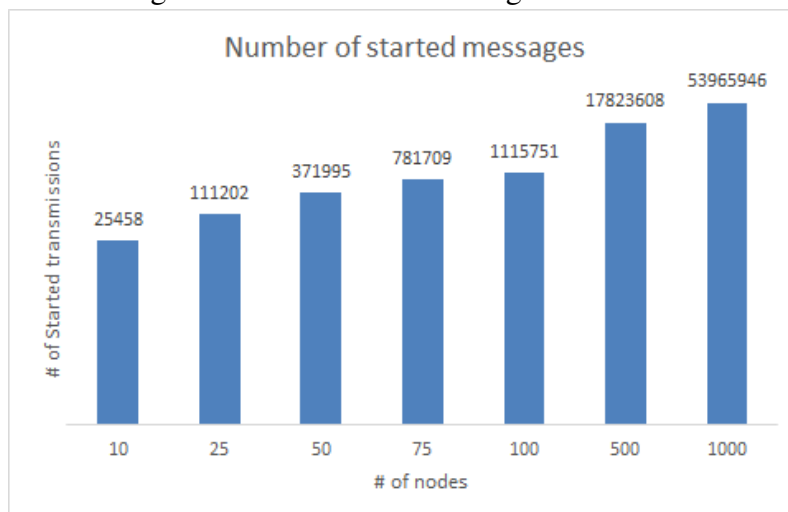
Tais taxas são esperadas e aceitáveis em Redes Oportunistas pois a inexistência de padrões de mobilidade dos nodos, as constantes desconexões e desligamentos dos nodos por falta de bateria ocasionam que algumas mensagens acabem não sendo entregues aos destinatários.

→ Mensagens iniciadas

Com relação a quantidade de transmissões iniciadas (*started*) pode-se notar que existe um incremento nessa métrica a medida que o número de nodos na rede aumenta (figura 6.3). Conforme definido anteriormente, essa métrica indica a quantidade de transmissões iniciadas entre os nodos da rede e isso inclui aquelas que foram encaminhadas com sucesso (*relayed*) e as que tiveram alguma falha e foram abortadas. Esse valor não está diretamente relacionado ao percentual de aumento dos nodos da rede, porém acredita-se que ele é decorrente da inexistência de um padrão de mobilidade dos nodos.

Dessa forma, quanto maior a quantidade de nodos, mais contatos existirão e maior será o valor de transmissões iniciadas. Esses contatos existentes são utilizados como oportunidades para envio de mensagens e não impactam diretamente nas taxas de entrega. Entretanto, deve-se observar que os constantes contatos podem exaurir mais rapidamente a bateria dos nodos.

Figura 6.3: Número de mensagens iniciadas



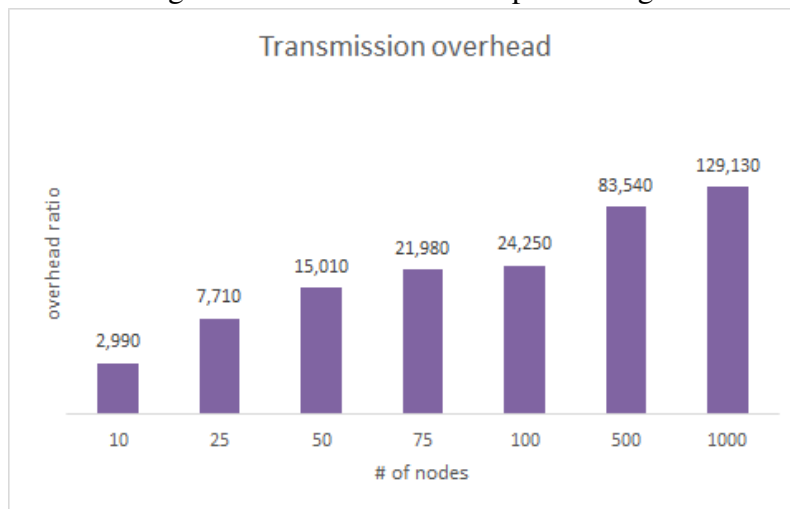
(Fonte: o autor)

→ Custo para entrega

O acerto na estratégia de tomadas de decisão de encaminhamento é demonstrado pela métrica de custo para entrega, *overhead* (figura 6.4). Nessa métrica, o cenário com 1000 nodos teve um aumento de 96% na quantidade de mensagens entregues com relação ao cenário com 500 nodos, mas com somente 54% a mais na taxa de *overhead*. No comparativo entre os cenários com 50 e 100 nodos o *overhead* ocorreu um aumento de 61%.

A relação de aumento do custo para entrega com a quantidade de mensagens entregues é representada na figura 6.5 onde pode-se observar que o percentual de crescimento

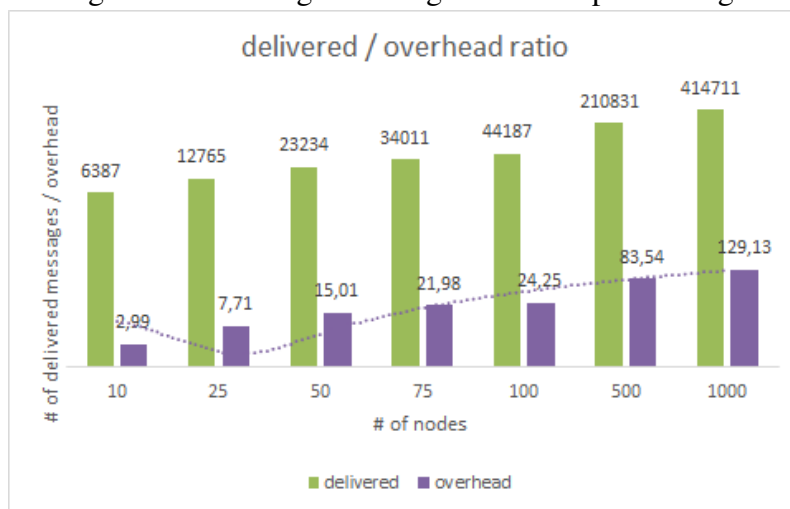
Figura 6.4: Métrica de custo para entrega



(Fonte: o autor)

de custo sempre é menor que o percentual de entrega de mensagens, além disso, por mais que a quantidade de nodos tenha aumentado 100%, o *overhead* não seguiu essa proporção.

Figura 6.5: Mensagens entregues x custo para entrega



(Fonte: o autor)

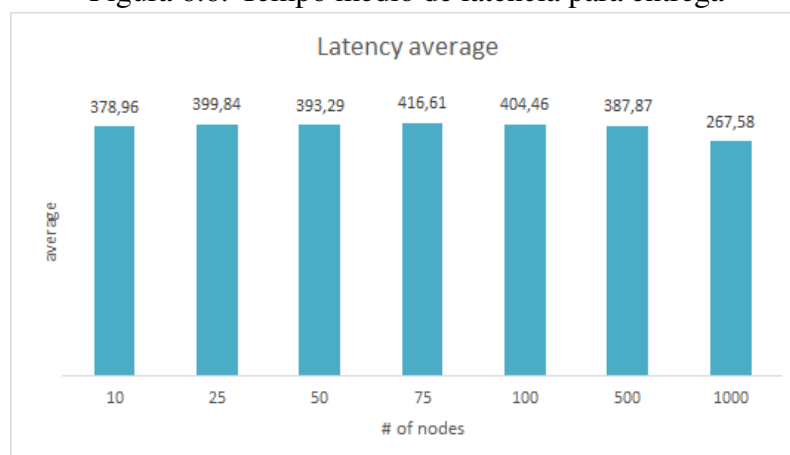
Novamente esses resultados reforçam que a aplicação de Sensibilidade à Situação para tomadas de decisão de roteamento está funcionando de maneira adequada.

→Latência para entrega

O tempo médio de latência (figura 6.6) teve um aumento de 2% no comparativo entre os cenários com 50 e 100 nodos e um aumento de cerca de 24% entre os cenários com 500 e 1000 nodos. Isso demonstra que mesmo aumentando em 100% a quantidade de nodos, o atraso no tempo de entrega não cresceu na mesma proporção.

Tal fato demonstra que os contatos entre os nodos estão sendo explorados de forma satisfatória na estratégia de encaminhamento, aumentando assim a taxa de entrega sem impactos consideráveis no tempo de entrega.

Figura 6.6: Tempo médio de latência para entrega



(Fonte: o autor)

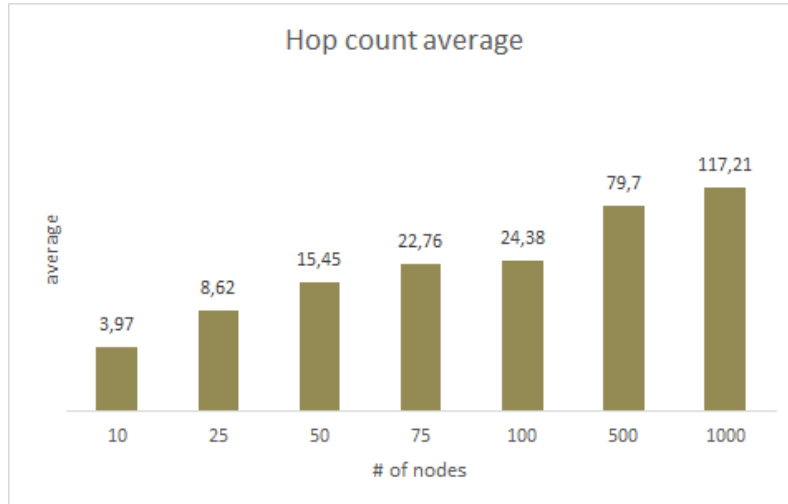
→Nodos intermediários

A média de nodos intermediários utilizados para a entrega de mensagens (*hop count average*) é representada na figura 6.7. No comparativo dos cenários com 50 e 100 nodos ocorreu um aumento de 57% na quantidade de nodos intermediários, enquanto na comparação dos cenários com 500 e 1000 nodos esse aumento foi de 47%.

O aumento na quantidade de novos intermediários é justificado pelo aumento da quantidade de nodos. Entretanto, por mais que esse percentual não tenha aumentado na mesma proporção do crescimento da quantidade de nodos do cenário, a medida que mais nodos vão sendo utilizados como intermediários, maior é a quantidade de mensagens encaminhadas entre os nodos e isso impacta nos custos para entrega.

Porém, com base nos gráficos de custo apresentados anteriormente, pode-se constatar que essas taxas de encaminhamento não estão impactando diretamente no custo e sim maximizando a quantidade de entregas efetuadas.

Figura 6.7: Média de nodos intermediários utilizados para entrega

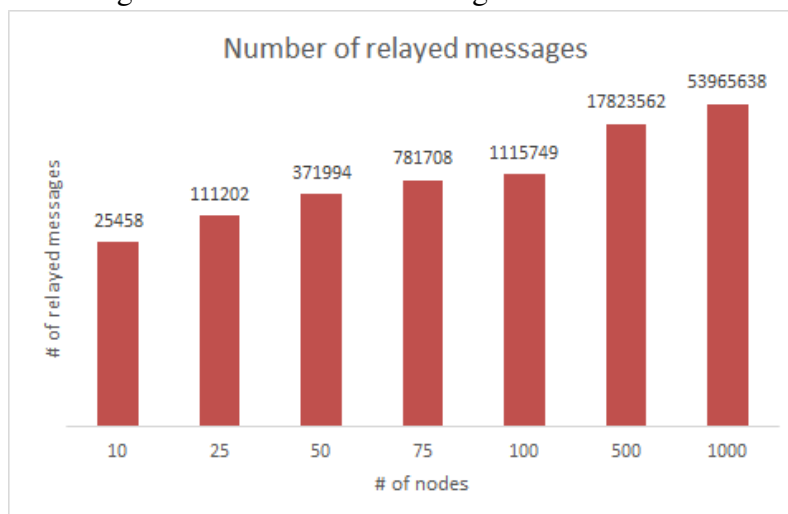


(Fonte: o autor)

→ Mensagens encaminhadas

A figura 6.8 demonstra a quantidade de mensagens encaminhadas entre os nodos dos experimentos. No comparativo dos cenários com 50 e 100 nodos e também com 500 e 1000 nodos foi observado um aumento em torno de 200% na quantidade de mensagens encaminhadas. Por mais que essa taxa pareça alta ela é justificada pela política do *Situs* de forçar que as mensagens saiam do buffer o mais rápido possível, para minimizar o tempo de entrega. Em cada contato entre os nodos, quando o *Situs* infere que o outro nodo é uma boa mula de dados, todas as mensagens que estão no buffer local são encaminhadas para o nodo encontrado. Isso explica as taxas obtidas.

Figura 6.8: Número de mensagens encaminhadas



(Fonte: o autor)

Por um lado essa estratégia acaba aumentando a probabilidade e a diminuição do

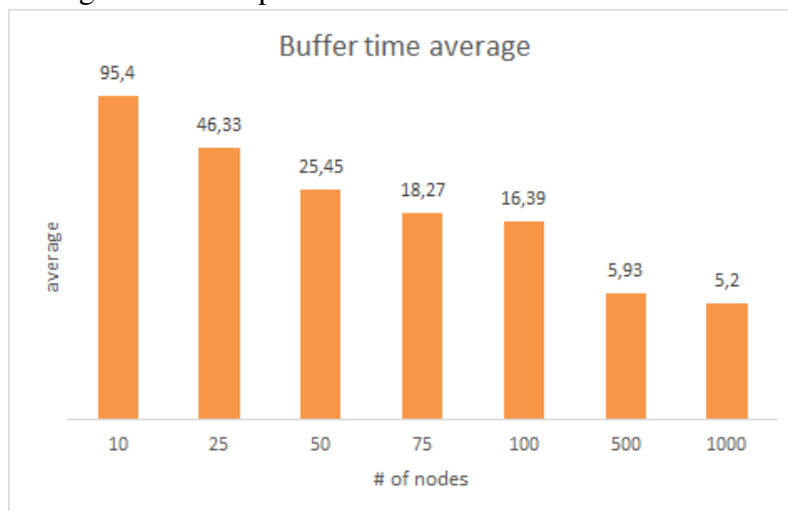
tempo de entrega das mensagens. Por outro, quanto maior for a quantidade de encaminhamentos, mais recursos de bateria podem ser consumidos para a transmissão dos dados. Para evitar que isso aconteça, uma alternativa seria utilizar alguma política de priorização de buffer para selecionar quais mensagens devem ser encaminhadas a cada momento.

→Tempo de armazenamento

O impacto da política de controle do buffer pode ser observado nos resultados da média de tempo que uma mensagem permanece no buffer dos nodos (figura 6.9). No comparativo entre os cenários de 50 e 100 nodos o tempo médio diminuiu em cerca de 35%. O valor médio absoluto foi ainda menor no comparativo dos cenários de 500 e 1000 nodos, passando de 5,93 para 5,2. Uma redução de 12% no tempo. Esse valor de decaimento é ainda maior se for considerado os cenários com 10 e 1000 nodos: 16,39 para 5,2 representando uma queda de 403% no tempo.

Dessa forma, quanto mais nodos existirem no cenário pode-se perceber que mais rápido consegue-se encontrar um bom nodo intermediário para encaminhamento. Esses valores são justificados pela existência de mais nodos trocando dados de contexto entre si e isso auxilia na melhora de inferência da situação e conseqüentemente na tomada de decisão de encaminhamento.

Figura 6.9: Tempo médio de armazenamento em buffer



(Fonte: o autor)

Por fim, com base nos resultados apresentados, pode-se concluir que existe um incremento na quantidade de mensagens entregues à medida que o número de nodos aumenta, porém o custo computacional (*overhead*) não segue as mesmas taxas de crescimento. Com mais nodos no cenário, maior será o contato entre eles e com isso existe

o aumento do histórico de dados de contexto que podem ser utilizados para a inferência e predição de situações. Em outras palavras, o *Situs* torna-se “mais inteligente” a medida que mais nodos são encontrados. O resultado disso é a descoberta crescente de boas mulas de dados, fato esse que impacta positivamente no aumento da quantidade de mensagens entregues com a constatare redução de tempo. Porém, um cuidado que deve ser tomado é que com o crescimento do histórico de contexto as rotinas de eliminação de dados antigos devem ser invocadas em intervalos menores de tempo ou isso além de impactar negativamente nas predições também pode usar muito espaço de armazenamento do nodo consumindo recursos desnecessários.

6.2.2 Experimento 2 - Comparação com outras abordagens

Nesse experimento foi comparada a performance do *Situs* com algumas das principais abordagens de roteamento e disseminação de conteúdo apontadas pelo estado da arte em Redes Oportunistas.

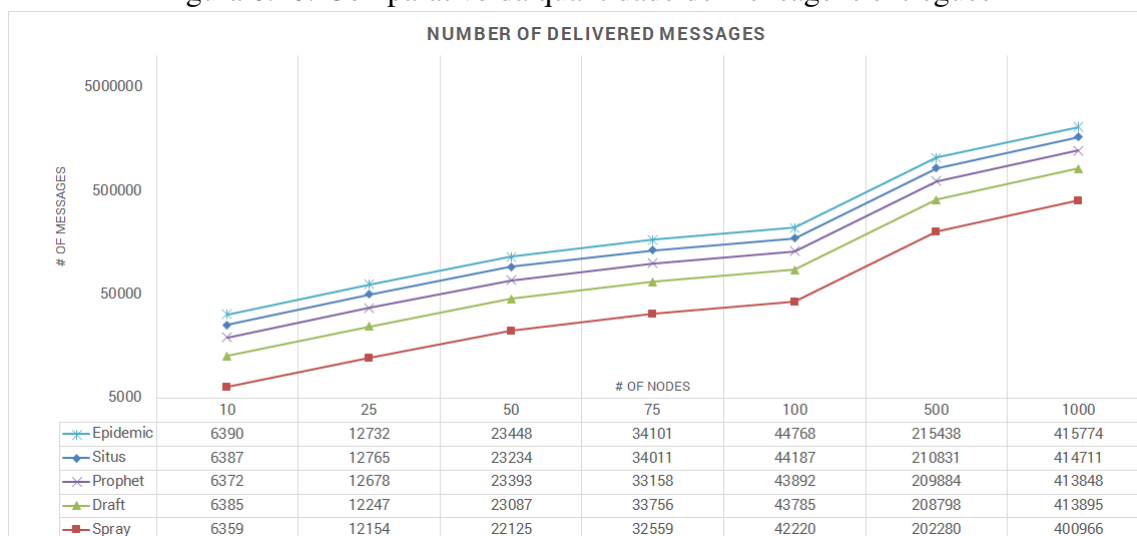
As iniciativas usadas para comparação foram o Epidemic (??), Prophet (??), DRAFT (??) e o Spray&Wait (??). Eles foram selecionados pois representam diferentes classes de protocolos e foram exaustivamente estudadas por outros pesquisadores (??????).

Foram utilizados os seguintes parâmetros de configuração: para o Prophet o tamanho em segundos de uma unidade de tempo que é usada para calcular a idade das predições de entrega de mensagens (*secondsInTimeUnit*) foi definida em 30 segundos. O Spray&Wait teve o número de cópias definido em 6 e o seu comportamento de encaminhamento binário habilitado. Para o DRAFT a duração em que encontros cumulativos passam a ser incluídos no processo clusterização (*familiarThreshold*) foi definida em 120 segundos, as unidades de tempo em que encontros cumulativos são deteriorados (*frame size*) foi definido em 3600 segundos e a taxa em que encontros cumulativos são deteriorados a cada unidade de tempo (*degrade*) foi definida em 0.5.

Sobre a quantidade de mensagens que foram entregues com sucesso ao destino (figura 6.10), o Epidemic teve as melhores taxas em todos os cenários. Entretanto, mesmo sendo um protocolo que inunda a rede com cópias das mensagens seus resultados foram apenas 2.2% melhor que o *Situs* no cenário com 500 nodos e 1.3% no cenário com 1000 nodos. O *Situs* teve taxas de entrega variando de 0.5% a 1% melhores que o Prophet e de 1 a 1.5% melhores que o Draft em todos os cenários. Como percebe-se o *Situs* teve a segunda melhor taxa de entrega, com valores maiores que o Prophet que é considerado um

dos melhores protocolos na área de Redes Oportunistas. Talvez ajustando-se os valores da idade das predições de entrega de mensagens do Prophet as suas taxas podem ser melhoradas. O Draft resultados abaixo do esperado, ficando em penúltimo lugar nas avaliações. Essa baixa taxa provavelmente deve-se a não formação de clusters com base em contatos recentes ao longo da simulação. Esse fato é algo importante de ser destacado, pois em cenários de larga escala como os utilizados em Sensoriamento Urbano nem sempre haverá nodos se encontrando repetidamente. Com isso, podem-se criar “ilhas” de nodos isoladas umas das outras e isso impacta negativamente nas taxas de entrega. Uma possibilidade para melhoria de suas taxas pode ser o aumento da unidade de tempo de decaimento e a redução do grau de encontros cumulativos. O Spray&Wait teve as piores taxas de entrega em todos os cenários. Seus resultados estão associados a configuração da quantidade de cópias de mensagens que devem ser replicadas. Entretanto, deve-se tomar o cuidado para não utilizar valores muito grandes pois isso tornaria o comportamento epidêmico.

Figura 6.10: Comparativo da quantidade de mensagens entregues

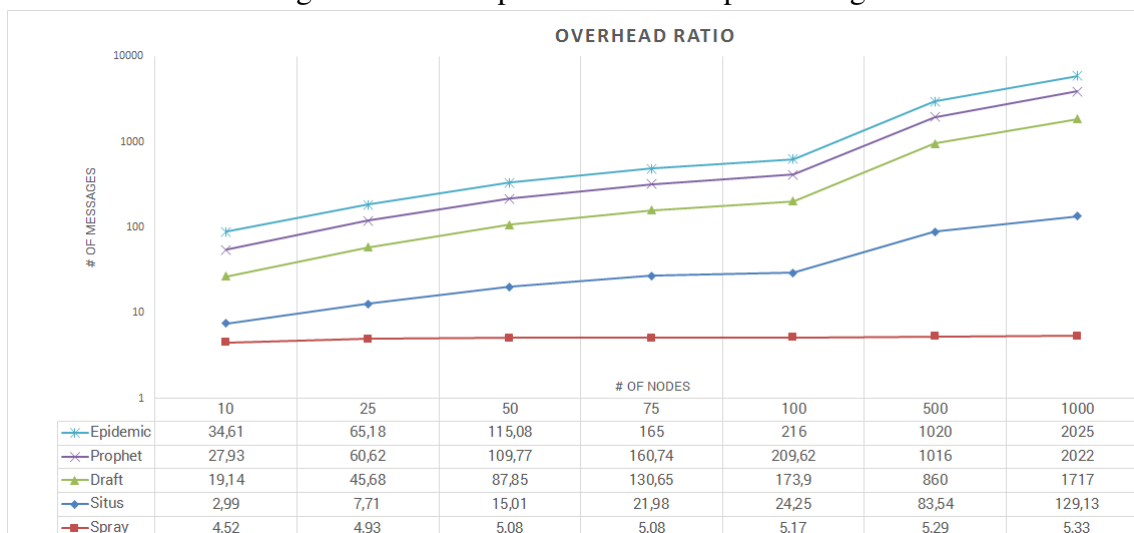


(Fonte: o autor)

Outra métrica analisada foi o custo para entrega (figura 6.11). Como era esperado o Epidemic teve as maiores taxas devido ao seu comportamento de espriar diversas cópias das mensagens para maximizar a entrega. Porém pode-se perceber que o Prophet teve valores em torno de 1% menores em todos os cenários. Ou seja, novamente percebe-se que o Prophet não consegue lidar bem com um cenário de Sensoriamento Urbano composto por diferentes tipos de nodos. O Draft teve a terceira maior taxa de custo computacional, ficando em torno de 15% menor que o Prophet em todos os cenários. Apesar desse percentual ser menor que o Epidemic e o Prophet eles podem ser considerados altos e novamente indicam a dificuldade do Draft em conseguir formar clusters de nodos.

Devido a falta de contato recente entre os mesmos nodos, o DRAFT cria vários clusters com poucos nodos, isso resulta na disseminação duplicada de mensagens dentro desses clusters que acaba impactando no custo. O *Situs* teve a segunda melhor taxa de custo de entrega, ficando acima somente do Spray&Wait. Os valores foram cerca de 1300% melhores que o Draft e 1500% melhores que o Epidemic no cenário com 1000 nodos. Esses valores confirmam que a estratégia de descoberta e encaminhamento de mensagens para boas mulas de dados está tendo efetividade. O Spray&Wait teve as menores taxas de custo e pode-se observar que ela decai a medida que a quantidade de nodos aumenta. Esse comportamento deve-se ao fato da quantidade de cópias de mensagens não ter sido alterada a cada cenário e também ao seu comportamento binário que faz com que a quantidade de mensagens replicadas seja dividida ao meio a cada retransmissão.

Figura 6.11: Comparativo do custo para entrega

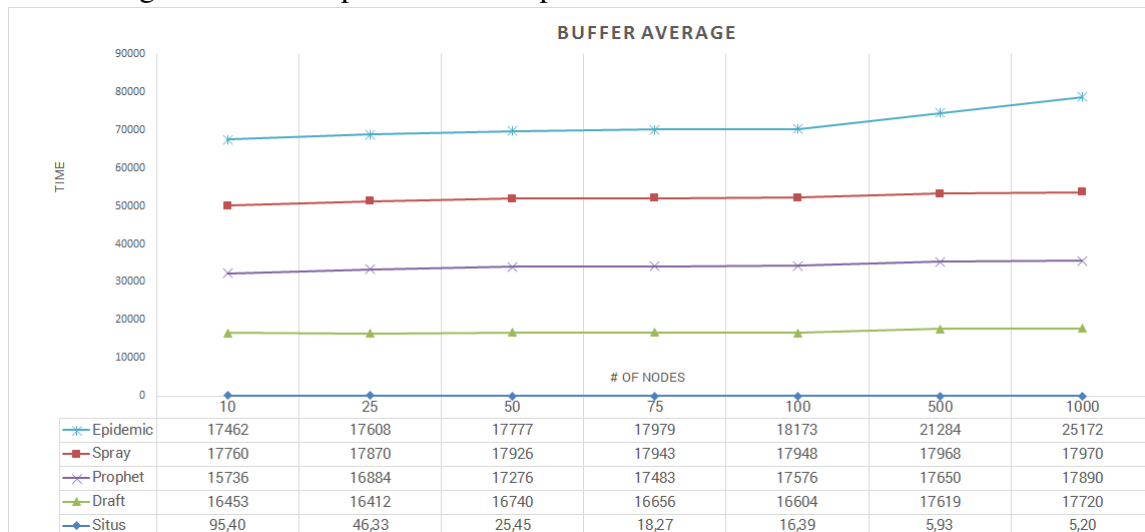


(Fonte: o autor)

Com relação ao tempo médio de armazenamento em buffer (figura 6.12), pode-se notar que o tanto o Epidemic quanto o Spray&Wait possuem comportamento bastante próximo em tempo de buffer. Nos cenários com 500 e 1000 nodos eles utilizam em média 17% a 27% mais de tempo que o Draft e de 17% a 28% mais de tempo que o Prophet. Esse comportamento justifica-se pelo fato deles fazerem cópias das mensagens para a maior quantidade possível de nodos, removendo-as do buffer local somente quando a mesma for entregue ao destinatário ou o tempo de vida da mensagem (TTL) esgotar. Esse comportamento pode esgotar rapidamente o buffer para armazenamento de novas mensagens, implicando que os nodos que utilizam essas iniciativas podem começar a recorrer armazenar novas mensagens por falta de espaço. Por sua vez, o *Situs* apresentou alta dinamicidade de uso do buffer, com resultado extremamente menores que os demais. Pode-se

perceber que em comparação com os demais, o *Situs* tende a fazer com que as mensagens fiquem menos tempo no buffer local. Isso deve-se a sua estratégia de dinamizar ao máximo a seleção de boas mulas de dados na tentativa de maximizar a entrega das mensagens.

Figura 6.12: Comparativo do tempo médio de armazenamento em buffer

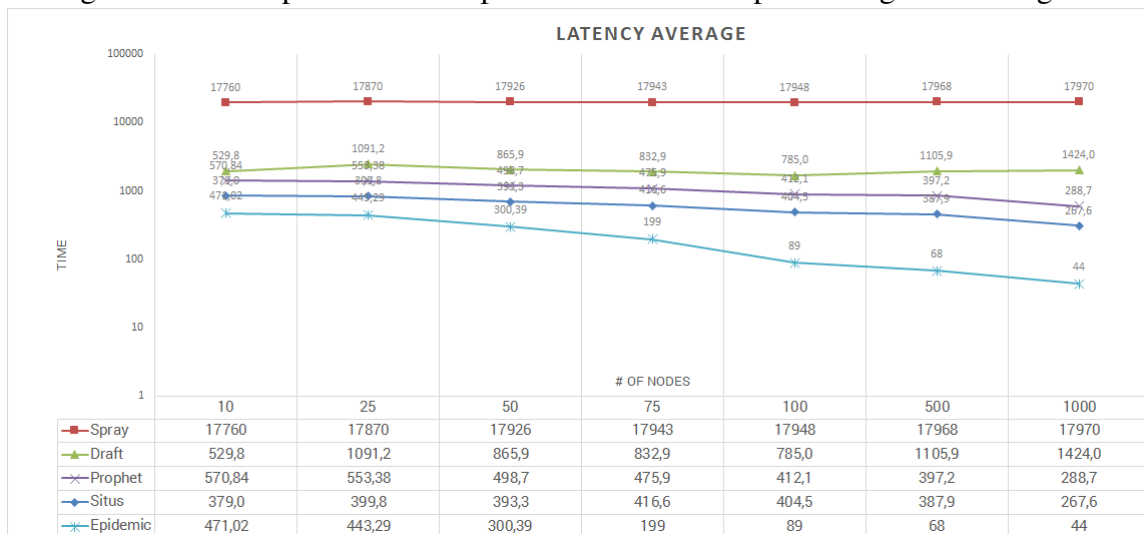


(Fonte: o autor)

Em outro experimento realizado foi analisado o tempo de latência para entrega das mensagens desde a sua criação (figura 6.13). O Epidemic possui a menor taxa de latência entre todos os protocolos nos diferentes cenários. Porém, conforme demonstrado anteriormente, para conseguir isso, ele inunda a rede com cópias das mensagens de forma epidêmica, o que resulta no maior custo para entrega (overhead). O *Situs* teve o segundo melhor tempo em todos os cenários. No cenário com 1000 nodos ele foi 7% melhor que o Prophet e 432% melhor que o Draft. A grande diferença percentual em relação ao Draft novamente deve-se ao fato de tentativa de clusterização dos nodos. Os indicadores do *Situs* em relação ao Prophet demonstram que a estratégia de encaminhamento de mensagens está tendo efetividade, porém sem todo o custo computacional gerado pelo Prophet. Por fim, precisa ser melhor analisado o motivo que levou ao Spray&Wait ter resultado tão distantes das demais iniciativas.

Para finalizar, os indicadores do *Situs* podem ser considerados satisfatórios quando comparado com as demais iniciativas. Ele teve praticamente a mesma quantidade de mensagens entregues que o Epidemic com um tempo de entrega um pouco maior (foi o segundo mais rápido). Entretanto, o seu custo para entrega foi bem melhor que o Draft e o Prophet, ficando atrás apenas do Spray&Wait. Ou seja, o *Situs* está conseguindo entregar grande quantidade de mensagens, em um tempo aceitável, com baixo custo computacio-

Figura 6.13: Comparativo do tempo médio de latência para entrega de mensagens



(Fonte: o autor)

nal. Isso novamente reforça que a estratégia de proatividade nas tomadas de decisões a partir da aplicação de sensibilidade à situação é uma alternativa viável de ser utilizada em Redes Oportunistas.

6.3 Considerações finais do capítulo

Esse capítulo apresentou os resultados de performance e comportamento do *Situs* obtidos a partir de experimentos em diferentes cenários através de simulações. As simulações buscaram reproduzir o comportamento de um ecossistema de monitoramento urbano composto por sensores fixos e móveis, carregados por pessoas e veículos, juntamente com pontos de acesso a Internet (APs) responsáveis por enviar os dados de sensoriamento para serem processados por sistemas especializados. O *Situs* foi codificado em Java e então em cada nodo do simulador foi executada uma instância dele juntamente com uma aplicação responsável por coletar dados de sensoriamento ao longo da cidade.

Em relação a sua codificação, pode-se destacar as diferentes bibliotecas de software que foram utilizadas para facilitar o desenvolvimento. Para implementar a compreensão de situações foi utilizado o Drools Chance que é um sistema de inferência híbrido baseado em regras. Com o uso do Drools foi possível definir regras para a caracterização das situações de uma forma simplificada e sem a necessidade de recompilar o código a cada alteração das definições. O tratamento de incertezas proporcionado pelo pacote Chance com o uso de Lógica Fuzzy permitiu flexibilidade no processo de inferência dos

dados para definir a situação e proporcionar o cálculo do potencial do nodo.

Para a possibilitar a projeção de situação foi utilizada uma ESN (Rede Neural) implementada utilizando a API do software ESNJava. Entretanto, por padrão esse software somente é capaz de manipular situações onde uma sequência de valores que devem ser aprendidos (sinal de aprendizagem) é recebido na entrada da rede e então a Rede aprende a como reproduzir o comportamento dinâmico desejado do sinal original. Em outras palavras, o ESNJava somente busca reproduzir o sinal aprendido e não proporciona suporte para predição, conforme requerido pelo *Forecaster*. Tal restrição obrigou que fossem feitas alterações no código original do software para possibilitar que a Rede fosse capaz de utilizar a sua saída como um novo sinal de entrada, retroalimentando a Rede, e possibilitando assim, que as predições usadas para a projeção de situação fossem efetuadas. A fase de treinamento da Rede se manteve conforme o formato original, com os pesos sendo ajustados através de regressão linear com base no histórico de dados de contexto de baixo nível que foi utilizado como sinal de exemplo para a aprendizagem. A fase de exploração sofreu alterações nas suas funcionalidades. Ao invés de usar todo o histórico de dados de contexto como entrada da Rede já treinada, somente algumas amostras passaram a ser apresentadas para ativar o reservatório. Em determinado momento essa entrada foi interrompida e então as saídas do passo anterior foram retroalimentadas à Rede como novas entradas, possibilitando assim que as predições acontecessem. Essas alterações no código fonte foram enviadas ao autor do software para o mesmo ser incorporado em versões futuras da API. O uso da ESN foi um dos grandes achados da tese. A sua precisão foi melhor que outras Redes apontadas pela literatura e testadas no *Forecaster*. O seu baixo custo computacional permitiu que cada nodo tivesse a sua própria instância de Rede, construída de forma autônoma de acordo com seus dados de contexto.

Com relação ao experimento 1, realizado para verificar o comportamento e a performance do *Situs*, pode-se notar que a sua capacidade de entrega aumenta com o crescimento da quantidade de nodos no cenário sem impactos significativos no custo computacional e no atraso na entrega das mensagens. Essas características indicam que a estratégia de uso de Sensibilidade à Situação para escolha de boas mulas de dados e encaminhamento de mensagens está funcionando adequadamente. Também pode-se verificar que o *Situs* conseguiu lidar com a grande quantidade de dados de contexto gerados a medida que a quantidade de nodos existentes no cenário aumentou. Tais dados contribuíram para melhorar a compreensão e projeção de situações e isso teve impacto positivos nas tomadas de decisões. Porém um aspecto que ficou evidente nas simulações é a necessidade

de uma rotina interna para eliminar os dados de contexto antigos de forma que eles não impactem negativamente nas projeções de situações via predição. Pode-se perceber que quanto mais nodos o cenário possui os dados gerados ficam defasados rapidamente e isso precisa ser melhor tratado.

A respeito do experimento 2 pode-se concluir que as principais iniciativas de Redes Oportunistas apontadas no referencial teórico possuem bons resultados em cenários compostas por nodos homogêneos, pela formação de comunidades ou com determinados padrões de mobilidade. Porém, pode-se constatar que quando aplicados em cenários de Sensoriamento Urbano, formado por nodos heterogêneos, móveis e fixos, sem padrões de mobilidade, elas não são adequadas. Prova disso, são os baixos resultados do Prophet e do Draft. Talvez até consiga-se resultados um pouco melhores com o ajuste de seus parâmetros de configuração. Porém, devem ser executadas diversas simulações e suas configurações precisam ser alteradas todas as vezes até conseguir-se valores considerados aceitáveis. O *Situs* por sua vez, não precisou de nenhum ajuste, sendo que ele conseguiu adaptar-se perfeitamente aos diferentes cenários encontrados. Outro aspecto não demonstrado nos gráficos mas que pode ser percebido foi o grande tempo consumido para as simulações. Enquanto o *Situs* simulou todos os cenários em alguns minutos, o Prophet e o Epidemic levaram vários dias até as concluírem. O simulador utilizado impactou nesse tempo gasto, mas esse fato já demonstra o quanto essas iniciativas possuem alto custo computacional e são inviáveis de serem empregadas em dispositivos móveis e sensores com baixa capacidade energética e pouco poder de processamento.

Por fim, esses experimentos demonstraram que o *Situs* é viável de ser aplicado para intensificar a transmissão de dados na arquitetura de Sensoriamento Urbano UrboSenti quando for utilizado o paradigma de Redes Oportunistas para complementar a comunicação.

7 CONSIDERAÇÕES FINAIS

Essa tese buscou responder ao questionamento de como intensificar a troca de dados em aplicações de Sensoriamento Urbano que fazem uso de Redes Oportunistas como paradigma complementar de comunicação. Na busca de respostas foi definida a hipótese de que um comportamento proativo obtido pelo emprego de Sensibilidade à Situação pode fornecer os requisitos necessários para tais aplicações. Tal hipótese norteou as pesquisas desenvolvidas que serviram de base para a proposta de um componente de software baseado em Redes Oportunistas que utiliza o modelo de situação para tomadas de decisão de encaminhamento.

Para atingir esse objetivo, inicialmente foram efetuados estudos do estado da arte nas áreas que permeiam a tese. Entre as principais descobertas, primeiramente pode-se apontar que as plataformas de Sensoriamento Urbano existentes não são capazes de atender todo o processo de sensoriamento, envolvendo desde a coleta dos dados, o processamento, a disponibilização das informações geradas e também a possibilidade de cenários “what-if”, onde são simulados possíveis cenários se ocorrerem determinados eventos. Para suprir essa lacuna foi desenvolvida uma arquitetura ubíqua orientada a serviços para dar suporte às atividades de Sensoriamento Urbano chamada UrboSenti. A arquitetura da UrboSenti foi modelada seguindo uma abordagem de Arquiteturas Orientadas a Serviços (*SOA - Services Oriented Architecture*) e é organizada em dois módulos principais: módulo *Backend* e módulo *Sensing*. O *Backend* é responsável por receber todos os dados de sensoriamento obtidos pelos sensores dispersos ao longo da cidade. Ele é executado em um data center e possui seus componentes internos organizados de forma a prover escalabilidade para armazenar os dados e efetuar o processamento dos mesmos. A partir disso são geradas informações a respeito da cidade que podem ser utilizadas pelos habitantes ou administradores das cidades ou ainda por outros sistemas. O módulo *Sensing* é executado nos dispositivos móveis dos usuários como smartphones, em veículos ou então em sensores fixos dispersos ao longo da cidade. Ele é o responsável por coletar dados de forma intencional (com o usuário interagindo diretamente com a aplicação no fornecimento de dados) ou não intencional (com a aplicação coletando dados em segundo plano, sem interação direta com o usuário). Devido à diversidade de dispositivos que podem ser envolvidos no sensoriamento, esse módulo foi concebido para usar poucos recursos computacionais e com capacidade de transmitir dados em cenários com conectividade de rede ou até mesmo onde a conectividade é nula ou intermitente. Para isso, o componente

responsável pelas comunicações da UrboSenti foi modelado de forma a dar suporte ao emprego do paradigma de Redes Oportunistas como forma complementar para a comunicação de dados.

É justamente essa a outra descoberta encontrada a partir dos estudos: as aplicações e plataformas de Sensoriamento Urbano existentes ainda não empregam efetivamente o conceito de Redes Oportunistas, ficando assim dependentes da disponibilidade de uma infraestrutura de rede para que a troca de dados seja efetuada. Por outro lado, a partir do estado da arte pode-se constatar que as iniciativas de roteamento e disseminação de conteúdo em Redes Oportunistas não foram concebidas para serem aplicadas em cenários de Sensoriamento Urbano. Em outras palavras, os estudos demonstraram que existe uma lacuna na literatura por uma solução que viabilize a aplicação de Redes Oportunistas em Sensoriamento Urbano.

Assim, foi proposto o *Situs*, um componente de software baseado em Redes Oportunistas, capaz de agir de forma proativa nas tomadas de decisões de encaminhamento utilizando uma visão holística de contexto chamada de situação e com isso intensificar a entrega de mensagens em aplicações de Sensoriamento Urbano. Em suma, o diferencial do *Situs* está na aplicação de Sensibilidade à Situação em Redes Oportunistas. Além disso, ele atende os requisitos necessários para ser utilizado no componente de comunicação da UrboSenti.

Para a concepção do *Situs*, foi executada uma sequência de passos segundo uma abordagem *top-down*. Inicialmente, foi efetuada uma análise de um cenário-problema de sensoriamento urbano onde foram definidos os elementos, interações e requisitos que forneceram subsídios para a definição de modelos conceituais (objetivo específico 1 definido). Em seguida, foi definida a modelagem de contexto capaz de fornecer a estrutura necessária para a caracterização dos estados dos nodos do cenário seguido da proposta do modelo de situação baseado em três níveis, sendo capaz de perceber, compreender e projetar situações (objetivo específico 2). Para cada um desses níveis foi proposto um sub-modelo que empregou técnicas de Inteligência Computacional apontadas pela literatura como mais adequadas para desempenhar suas funções: Lógica Fuzzy para a compreensão de situação e Redes Neurais para a projeção de situação (objetivo específico 3). Posteriormente, foram projetados os componentes que compõem a arquitetura do *Situs*, seu comportamento e os algoritmos utilizados para desempenhar suas funcionalidades (objetivo específico 4).

Após a definição da arquitetura foi efetuada a codificação das suas principais fun-

cionalidades (objetivo específico 5). Nessa etapa, o uso de um sistema de inferência híbrido baseado em regras, proporcionou facilidade e flexibilidade na definição de como caracterizar as situações. Além disso, a Lógica Fuzzy possibilitou que o tratamento de incerteza dos dados de contexto pudesse ser adequadamente tratado para a compreensão de situações e posterior cálculo do potencial de cada nodo. Esse aspecto evidencia que abordagens híbridas para a inferência de contexto são alternativas interessantes de serem empregadas. Outro aspecto que deve ser destacado relaciona-se a rede neural ESN utilizada como base para a projeção de situações. A literatura apontou que uma ESN possui baixo custo computacional devido a estratégia de treinamento da Rede somente na camada de saída. Essa característica pode ser comprovada durante os experimentos onde constatou-se que além do baixo custo computacional a Rede conseguiu lidar com boa acurácia nas previsões efetuadas. Com isso, ao invés de treinar uma Rede Neural de forma externa e então depois utilizá-la de forma idêntica em todos os nodos do cenário, foi desenvolvido uma solução que possibilitou que cada nodo do ambiente treinasse a sua própria Rede de forma autônoma, de acordo com seus próprios dados de contexto, melhorando consideravelmente os resultados com relação às outras Redes testadas.

Na sequência, foram efetuados testes através de simulações para verificar o comportamento e performance do *Situs* em um típico ecossistema de monitoramento urbano (objetivo específico 6). Com base nos experimentos pode-se comprovar que o uso de Sensibilidade à Situação como base para a estratégia de decisões de encaminhamento foi acertada. A proatividade proporcionada pela projeção de situações possibilitou que boas mulas de dados fossem selecionadas antes que determinadas situações como falta de bateria ou enchimento de buffer ocorressem. Com isso, o *Situs* conseguiu lidar de forma adequada com os diferentes tipos de nodos, fixos e móveis, sem haver degradação na sua performance a medida que o número de nodos no cenário aumentava.

Ao contrário dele, os experimentos demonstraram que as principais iniciativas de Redes Oportunistas apontadas pelo estado da arte, não conseguiram lidar adequadamente com o cenário de Sensoriamento Urbano. Segundo a literatura, elas possuem bons resultados em cenários compostas por nodos homogêneos, pela formação de comunidades ou com determinados padrões de mobilidade. Porém quando aplicadas em cenários com nodos heterogêneos, carregados por pessoas ou veículos, sem padrões de mobilidade ou então fixos em locais estratégicos da cidade sua performance foi baixa. Até tentou-se fazer ajustes nos parâmetros de configurações das iniciativas testadas para verificar se não eram problemas de configuração. Porém, mesmo após várias tentativas, os seus melho-

res resultados tiveram performance inferior comparada ao *Situs*. Por sua vez, o *Situs* não necessitou que nenhuma configuração fosse ajustada para lidar adequadamente com as diferentes situações encontradas ao longo das simulações. Isso deve-se a sua capacidade de adaptação e manipulação de contexto proporcionadas pela Sensibilidade à Situação. Assim, pode-se apontar que as iniciativas existentes não podem ser empregadas sem adaptações ou ajustes em cenários de Sensoriamento Urbano.

Por fim, pode-se concluir que o comportamento proativo do *Situs* obtido pela aplicação de Sensibilidade à Situação em Redes Oportunistas intensificou a troca de dados com relação às outras iniciativas existentes e com isso comprovou a validade da hipótese defendida por essa tese.

7.1 Perspectivas futuras

Como perspectivas de trabalhos futuros pode-se apontar:

- Encontrar uma forma de otimizar os valores dos conjuntos fuzzy e regras em tempo de execução (“*on the fly*”) de acordo com dados do nodo
 - As regras utilizadas para a definição de situações implicam no uso de valores pré-definidos para determinar a ocorrência de uma situação ou de outra. Uma possibilidade interessante seria empregar alguma técnica estatística ou mesmo de Inteligência Computacional que possibilite que esses valores sejam adaptados com o passar do tempo ou quando determinado limiar (*threshold*) de erro seja atingido. Com isso, poderia-se obter maior aderência das inferências com as situações encontradas;
- Efetuar melhorias nos mecanismos de gerenciamento de recursos e de adaptação do *Situs*
 - Apesar da concepção do componente ter levado em consideração restrições energéticas e de processamento, os experimentos demonstraram que o *Situs* precisa gerenciar melhor os recursos computacionais do nodo que o está executando. A grande quantidade de dados de contexto coletadas pode defasar rapidamente as previsões e exaurir o espaço de armazenamento (buffer) e a bateria do nodo devido às tomadas de decisões erradas. Deve-se implementar melhorias nas rotinas que fazem análise dos recursos que está sendo utilizados e então proporcionar que sejam feitas melhores adaptações para otimizar o uso dos recursos.

- Explorar o uso de Redes Neurais Profundas – Deep Belief Network (DBN) ou Restricted Boltzmann machines (RBMs) – como técnica para predição de dados utilizados na projeção de situação
 - Ao longo do desenvolvimento da tese foram efetuados diversos experimentos para testar diferentes tipos de Redes Neurais. Com base nesses experimentos, na literatura e nos requerimentos do cenário-problema optou-se por utilizar Echo State Networks como base para a projeção de situações. Entretanto, devido ao seu potencial, as Redes Neurais Profundas podem ser uma solução interessante de serem empregadas para predição de valores. Um aspecto que deve ser considerado é que elas exigem recursos vastos computacionais para o treinamento. Como alternativa poderia-se fazer o treinamento dessas Redes de forma externa aos nodos e aplicações (em ambientes de nuvem computacional), e então efetuado o “deploy” da Rede já treinada para o nodo. Esse processo deveria ser efetuado constantemente, sempre que houvesse conectividade com a Internet, devido a defasagem dos dados de contexto, de forma a não impactar negativamente na acurácia das predições.
- Investigar a possibilidade de computação cognitiva em Redes Oportunistas.
 - Ao longo do desenvolvimento da tese uma das estratégias testadas foi o uso agentes de software para disseminação de conteúdo em Redes Oportunistas. Apesar de resultados promissores, as pesquisas nessa linha não evoluíram devido a restrições de escopo da tese. Uma abordagem que merece ser investigada é o uso de um conceito que está sendo difundido por pesquisadores da área de inteligência artificial chamada de “Multi-Modal Cognitive Computing”. O uso de computação cognitiva na área de Redes Oportunistas ainda não é explorado. Poderia-se desenvolver agentes que fossem capazes de perceber, compreender, agir e aprender sobre as decisões tomadas com base no comportamento dos humanos. As informações geradas e trocadas por esses agentes poderia ser usada para permitir a organização autônoma de uma infraestrutura oportunista empregada para diferentes tipos de aplicações.
- Verificar o uso de Computação Oportunista no Sensoriamento Urbano
 - Alguns autores como ??) apontam que as Redes Oportunistas podem futuramente evoluir para um conceito chamado de Computação Oportunista. Com isso, seria possível que os recursos computacionais dos dispositivos fossem compartilhados sempre que ocorressem contatos oportunistas. Seguindo essa linha de raciocínio, um equipamento que não seja equipado com determinado tipo de sensor ou de

capacidade de processamento exigida para determinada tarefa poderia fazer o uso dos recursos computacionais que estivessem presentes em outro dispositivo com o qual teve-se contato. Com isso poderia-se melhorar a qualidade dos dados de sensoriamento e também proporcionar melhor prestação de serviços aos habitantes de uma cidade inteligente.

- Concluir a codificação da UrboSenti e efetuar testes de performance em ambiente controlado.

→ A UrboSenti foi concebida durante o desenvolvimento da tese. Apesar das suas funcionalidades internas terem sido modeladas e documentadas, somente seus componentes básicos foram implementados para possibilitar que o trabalho de alguns mestrandos fossem validados. Espera-se que a sua codificação seja concluída para que então possam ser efetuados testes em ambientes controlados, como por exemplo, um “living campus”, com vistas a definir aspectos que precisam ser ajustados. Além disso, devido a sua proposta modular, os componentes da UrboSenti podem ser empregados sem grandes dificuldades em outras arquiteturas computacionais para Cidades Inteligentes que vierem a ser desenvolvidas com os desdobramentos das pesquisas que estão em andamento pelo grupo.

7.2 Publicações

A seguir estão listadas as publicações como primeiro autor que foram efetuadas ao longo do desenvolvimento da tese.

► Artigos completos publicados em periódicos

- * ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GERONIMO, G. A. ; SANTOS, T. F. M. ; SOUZA, A. M. ; GEYER, C.F.R. Situation Awareness and Computational Intelligence in Opportunistic Networks to Support the Data Transmission of Urban Sensing Applications. *Computer Networks*, v. 0, p. 1–16, Ago. 2016. – **Qualis: A2**

→ Apresentação dos resultados da tese. Foi apresentada a penúltima versão do componente de software, sua arquitetura, modelos e parte dos experimentos realizados para a sua validação.

- * ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GERONIMO, G. A. ; SANTOS, T. F. M. ; SOUZA, A. M. ; GEYER, C.F.R. Urbosenti: An Ubiquitous Service-Oriented Architecture for Urban Sensing – In: Agent Technology for Intelligent Mobile Services and Smart Societies. [S.l.]: Springer Berlin Heidelberg, 2015. p.1–10. (Communications in Computer and Information Science, v.498). 28 – **Qualis: B3**

→ Apresentação de uma arquitetura utilizada para o Sensoriamento Urbano chamada de UrboSenti. O componente que está se propondo na tese será utilizado nessa arquitetura para melhorar a comunicação de dados de sensoriamento utilizando o paradigma de Redes Oportunistas. Esse trabalho abriu uma espécie de “guarda-chuvas” no grupo de computação ubíqua do GPPD onde hoje estão envolvidos um mestrando e dois graduandos.

- * ROLIM, C. O. ; LEITHARDT, V. R. Q. ; ROSSETO, A. G ; SANTOS, T. F. M. ; SOUZA, A. M. ; GEYER, C. . Six Degrees of Separation to Improve Routing in Opportunistic Networks. International Journal of Ubicomp, v. 4, p. 11-22, 2013.

→ Definição preliminar de disseminação de conteúdo utilizando paradigma de “small-world networks”. Demonstrou que o paradigma de redes de mundo pequeno é uma abordagem interessante, porém a consideração de somente aspectos sociais para tomada de decisão de encaminhamento pode não funcionar adequadamente em cenários de Sensoriamento Urbano.

- * ROLIM, C. O. ; Rossetto, Anubis G.M. ; Leithardt, Valderi R.Q. ; GEYER, C. F. R. Analysis of a hybrid neural network as underlying mechanism for a situation prediction engine. Journal of Applied Computing Research, v. 2, p. 22-31, 2012. – **Qualis: B5**

→ Análise de aplicação uma rede MANFIS na predição de dados de situação. Os resultados obtidos demonstraram que uma rede MANFIS possui alto poder de predição, porém sofre com alto custo computacional de aprendizagem sendo inviável de ser executada em dispositivos móveis.

► Artigos publicados em anais de congressos

- * ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GERONIMO, G. A. ; SANTOS, T. F. M. ; SOUZA, A. M. ; Geyer, C.F.R. . A Novel En-

gine to Underlie the Data Transmission of Social Urban Sensing Applications. In: 20th IEEE Symposium on Computers and Communication (ISCC). Proceedings of the 20th IEEE Symposium on Computers and Communication (ISCC). Larcana, Cyprus. 2015 – **Qualis: A2**

→ Apresentação de uma versão preliminar do componente de software e dos seus resultados.

- * ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GERONIMO, G. A. ; SANTOS, T. F. M. ; SOUZA, A. M. ; Geyer, C.F.R. . Towards a Novel Engine to Underlie the Data Transmission of Social Urban Sensing Applications. In: 18th International Conference on Enterprise Information Systems (ICEIS). Proceedings of the 18th International Conference on Enterprise Information Systems (ICEIS). Barcelona, Spain. 2015 – **Qualis: B1**

→ Apresentação da análise do cenário, argumentos e modelos preliminares da definição do componente de software. Esse trabalho ainda não possuía resultados preliminares das simulações.

- * ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GERONIMO, G. A. ; SANTOS, T. F. M. ; SOUZA, A. M. ; Geyer, C.F.R. . Putting Opportunistic, Situational and Smart Approaches to Underlie the Data Transmission of Social Urban Sensing Applications. In 7th Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP). Recife, PE, 2015. – **Qualis: B5**

→ Apresentação de forma resumida das principais funcionalidades do componente com alguns experimentos. Esse artigo foi selecionado entre os melhores do evento e convidado para publicação da versão estendida em um journal.

- * ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GERONIMO, G. A. ; SANTOS, T. F. M. ; SOUZA, A. M. ; Geyer, C.F.R. . Towards Predictive Routing Agents in Opportunistic Networks. In: Fifth International Workshop on Collaborative Agents Research & Development, CARE for Intelligent Mobile Services, 2014, Paris. Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS). New York: Springer-Verlag Berlin Heidelberg, 2014. – **Qualis: A1**

→ Demonstra a tentativa de uso de agentes para o roteamento em Redes Oportunistas. Os resultados foram interessantes e demonstraram que poderia ser adotada uma abordagem multi-agente no cenário de Sensoriamento Urbano.

Essa abordagem não foi adotada na tese por questões de delimitação de escopo. Porém, o seu estudo será retomado futuramente para maiores investigações.

- * ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; BORGES, G. ; SANTOS, T. F. M. ; SOUZA, A. M. ; Geyer, C.F.R. . Towards a Ubiquitous Service-Oriented Architecture for Urban Sensing. In: The First International Workshop on Social Computing and Urban Intelligence and Smarter Lives (SMARTLIFE), 2014, Stanford. The Sixth ASE International Conference on Social Computing (SocialCom), 2014. – **Qualis: B4**

→ Apresentação dos esboços iniciais da arquitetura da UrboSenti. Esse paper foi convidado para uma versão estendida no periódico *Communications in Computer and Information Science*.

- * ROLIM, C. O. ; ROSSETO, A. ; LEITHARDT, V. R. Q. ; BORGES, G. A. ; SANTOS, T. F. M. ; SOUZA, A. M. ; GEYER, CLAUDIO F. R. . UrboSenti: Uma arquitetura ubíqua orientada a serviços para o Sensoriamento Urbano. In: VI Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SB-CUP), 2014, Brasília. Anais do XXXIV Congresso da Sociedade Brasileira da Computação, 2014. – **Qualis: B5**

→ Apresentação dos estados iniciais da UrboSenti em importante simpósio nacional na área de computação ubíqua. Os revisores apontaram que a arquitetura proposta tem potencial de aplicação porém, por ser um projeto novo não possui resultados de aplicação ainda.

- * ROLIM, C.O.; GEYER, CLAUDIO F. R. . Reasoning and prediction on opportunistic networks to improve data dissemination. In: the ACM International Conference, 2013, Ischia. Proceedings of the ACM International Conference on Computing Frontiers - CF '13. New York: ACM Press. p. 1. – **Qualis: B1**

→ Proposta inicial de uso de raciocínio e predição em Redes Oportunistas. Não foram apresentados resultados, somente argumentações sobre o potencial de melhoria de decisões de encaminhamento se fossem utilizadas técnicas de Inteligência Computacional para predição de dados de contexto. O trabalho motivou o desenvolvimento de maiores estudos.

- * ROLIM, C. O. ; SCHUBERT, F. ; ROSSETO, A. G ; LEITHARDT, V. R. Q.

; GEYER, C. F. R. ; WESTPHALL, C. B. . Comparison of a Multi output Adaptative Neuro-Fuzzy Inference System (MANFIS) and Multi Layer Perceptron (MLP) in Cloud Computing Provisioning. In: XXX Simposio Brasileiro de Redes de Computadores - X Workshop em Clouds, Grids e Aplicações WCGA, 2012, Ouro Preto - MG. Anais do X Workshop de Computação em Grade e Aplicações - WCGA, 2012. – **Qualis: B4**

→ Estudo comparativo da performance de um MLP e um MANFIS para predição. Apresentou os primeiros resultados comparativos de Redes Neurais para predição em ambiente de nuvem computacional. As redes MANFIS demonstraram ser uma alternativa com maior poder preditivo que uma MLP, além de ser mais simples de ser modelada.

- * ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GEYER, C. F. R. . Análise de uma Rede Neural Híbrida como base para um Mecanismo de Predição de Situação. In: XXXII Congresso da Sociedade Brasileira de Computação - CSBC, 2012, Curitiba. Anais do IV Simpósio Brasileiro de Computação Ubíqua e Pervasiva - SBCUP 2012, 2012. – **Qualis: B5**

→ Demonstração da performance de um MANFIS na predição de dados de situação. Uma vez que foi constatado que uma MANFIS possuía boa aplicabilidade na área de nuvem computacional, a mesma foi aplicada na predição de dados de contexto de situações. Os resultados comprovaram que ela é inviável de ser utilizada por dispositivos móveis devido ao alto custo computacional. Esse artigo foi escolhido entre os 5 melhores do evento.

* Relatório técnico

- * ROLIM, C. O. ; BORGES, G. ; GEYER, C. F. R. UrboSenti – Uma arquitetura ubíqua orientada a serviços para o Sensoriamento Urbano, RP 369, Nov 2015 Disponível em <http://gppd-wiki.inf.ufrgs.br/index.php/UrboSenti>
- Apresenta os detalhes da modelagem da arquitetura da UrboSenti.

REFERÊNCIAS

AGGARWAL, C. C.; ABDELZAHER, T. Social Sensing. In: AGGARWAL, C. C. (Ed.). **Managing and Mining Sensor Data**. [S.l.]: Springer US, 2013. p.237–297.

ALAIAD, A.; ZHOU, L. Patients' Behavioral Intentions toward Using WSN Based Smart Home Healthcare Systems: an empirical investigation. **2015 48th Hawaii International Conference on System Sciences**, [S.l.], p.824–833, 2015.

ALEGRE, U.; Carlos Augusto, J.; CLARK, T. Engineering Context-Aware Systems and Applications: a survey. **Journal of Systems and Software**, [S.l.], v.117, p.55–83, 2016.

ALI, S.; KHUSRO, S. Mobile Phone Sensing: a new application paradigm. **Indian Journal of Science and Technology**, [S.l.], v.9, n.19, 2016.

ALVEAR, O. et al. EcoSensor: monitoring environmental pollution using mobile sensors. In: IEEE 17TH INTERNATIONAL SYMPOSIUM ON A WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS (WOWMOM), 2016., 2016, Coimbra, Portugal. **Anais... IEEE**, 2016. n.x, p.1–6.

AMAH, T. E. et al. Towards next-generation routing protocols for pocket switched networks. **Journal of Network and Computer Applications**, [S.l.], p.51–88, 2016.

AMANI, P.; KIHIL, M.; ROBERTSSON, A. NARX-based multi-step ahead response time prediction for database servers. **International Conference on Intelligent Systems Design and Applications, ISDA**, [S.l.], p.813–818, 2011.

AMMARI, H. M. (Ed.). **The Art of Wireless Sensor Networks**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. (Signals and Communication Technology).

ANAGNOSTOPOULOS, C. B.; NTARLADIMAS, Y.; HADJIEFTHYMIADES, S. Situational computing: an innovative architecture with imprecise reasoning. **Journal of Systems and Software**, [S.l.], v.80, n.12, p.1993–2014, 2007.

ANAGNOSTOPOULOS, T.; ANAGNOSTOPOULOS, C.; HADJIEFTHYMIADES, S. An adaptive location prediction model based on fuzzy control. **Computer Communications**, [S.l.], v.34, n.7, p.816–834, May 2011.

AQUINO, R. R. B. de et al. Investigating the use of Echo State Networks for prediction of wind power generation. In: IEEE SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE FOR ENGINEERING SOLUTIONS (CIES), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.148–154.

ARNABOLDI, V. et al. Sensor Mobile Enablement (SME): a light-weight standard for opportunistic sensing services. In: IEEE INTERNATIONAL CONFERENCE ON PERSVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS (PERCOM WORKSHOPS), 2013., 2013. **Anais... IEEE**, 2013. n.March, p.236–241.

ASADI, A.; MANCUSO, V. A Survey on Opportunistic Scheduling in Wireless Communications. **IEEE Communications Surveys & Tutorials**, [S.l.], v.15, n.4, p.1671–1688, 2013.

BALASUBRAMANIAN, A.; LEVINE, B. N.; VENKATARAMANI, A. Replication routing in DTNs: a resource allocation approach. **IEEE/ACM Transactions on Networking**, [S.l.], v.18, n.2, p.596–609, 2010.

BALDAUF, M.; DUSTDAR, S.; ROSENBERG, F. A survey on context-aware systems. **Int. J. Ad Hoc and Ubiquitous Computing**, [S.l.], v.2, n.4, p.263–277, 2007.

BELLAVISTA, P. Pervasive computing at scale: challenges and research directions. In: IEEE SENSORS PROCEEDINGS, 2011., 2011. **Anais...** IEEE, 2011. p.639–642.

BENMILOUD, T. Multioutput adaptive neuro-fuzzy inference system. In: WSEAS INTERNATIONAL CONFERENCE ON NURAL NETWORKS AND 11TH WSEAS INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTING AND 11TH WSEAS INTERNATIONAL CONFERENCE ON FUZZY SYSTEMS (NN'10/EC'10/FS'10), 11., 2010, Stevens Point, Wisconsin, USA. **Anais...** World Scientific and Engineering Academy and Society (WSEAS), 2010. p.94–98.

BENOUARET, K.; VALLIYUR-RAMALINGAM, R.; CHAROY, F. CrowdSC: building smart cities with large-scale citizen participation. **IEEE Internet Computing**, [S.l.], v.17, n.6, p.57–63, Nov. 2013.

BETTINI, C. et al. A survey of context modelling and reasoning techniques. **Pervasive and Mobile Computing**, [S.l.], v.6, n.2, p.161–180, Apr. 2010.

BEZDEK, J. C. What is computational intelligence? In: ZURADA, J. M.; Marks II, R. J.; ROBINSON, C. J. (Ed.). **Computational Intelligence, Imitating Life**. Piscataway: IEEE Computer Society Press, 1994. p.1–12.

BJUREFORS, F. **Measurements in Opportunistic Networks**. 2012. 140p. PhD Thesis — Uppsala University, Sweden.

BLACKSTOCK, M. A. **A Common Model for Ubiquitous Computing**. 2008. 184p. Doctoral — The University of British Columbia.

BOLDRINI, C.; CONTI, M.; PASSARELLA, A. ContentPlace: social-aware data dissemination in opportunistic networks. In: MODELING, ANALYSIS AND SIMULATION OF WIRELESS AND MOBILE SYSTEMS - MSWIM '08, 11., 2008, New York, New York, USA. **Proceedings...** ACM Press, 2008. p.203.

BOLDRINI, C.; CONTI, M.; PASSARELLA, A. Design and performance evaluation of ContentPlace, a social-aware data dissemination system for opportunistic networks. **Computer Networks**, [S.l.], v.54, n.4, p.589–604, Mar. 2010.

BOLDRINI, C. et al. HiBOP: a history based routing protocol for opportunistic networks. In: IEEE INTERNATIONAL SYMPOSIUM ON A WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS, WOWMOM, 2007., 2007. **Anais...** [S.l.: s.n.], 2007.

- BOLDRINI, C. et al. **Deliverable 2.2**: final specification of forwarding paradigms in haggla. [S.l.: s.n.], 2008.
- BOLDRINI, C. et al. Context- and social-aware middleware for opportunistic networks. **Journal of Network and Computer Applications**, [S.l.], v.33, n.5, p.525–541, Sept. 2010.
- BOYLE, D. E.; YATES, D. C.; YEATMAN, E. M. Urban Sensor Data Streams: london 2013. **IEEE Internet Computing**, [S.l.], v.17, n.6, p.12–20, Nov. 2013.
- BOZSIK, J.; ILONCZAI, Z. Echo state network-based credit rating system. In: LINDI 2012 - 4TH IEEE INTERNATIONAL SYMPOSIUM ON LOGISTICS AND INDUSTRIAL INFORMATICS, PROCEEDINGS, 2012. **Anais...** [S.l.: s.n.], 2012. p.185–190.
- BREFELD, U. **Kernels for Automata**. Darmstadt: Technische Universität Darmstadt, 2015. 24p.
- BROUWERS, N.; LANGENDOEN, K. Pogo, a Middleware for Mobile Phone Sensing. In: ACM/IFIP/USENIX 13TH INTERNATIONAL MIDDLEWARE CONFERENCE, MONTREAL, QC, CANADA, DECEMBER 3-7, 2012. PROCEEDINGS, 2012. **Anais...** Springer Berlin Heidelberg, 2012. p.21–40.
- BUJARI, A. A Survey of Opportunistic Data Gathering and Dissemination Techniques. **2012 21st International Conference on Computer Communications and Networks (ICCCN)**, [S.l.], p.1–6, July 2012.
- BUJARI, A. et al. Delay Tolerant Networking over the Metropolitan Public Transportation. **Mobile Information Systems**, [S.l.], p.1–20, 2016.
- BURGESS, J. et al. MaxProp: routing for vehicle-based disruption-tolerant networks. In: IEEE INFOCOM, 2006. **Proceedings...** [S.l.: s.n.], 2006.
- BURLEIGH, S. et al. Delay-tolerant networking: an approach to interplanetary internet. **IEEE Communications Magazine**, [S.l.], v.41, n.6, p.128–136, June 2003.
- BURNS, B. et al. MV Routing and Capacity Building in Disruption Tolerant Networks MV Routing and Capacity Building in Disruption Tolerant Networks. In: IEEE INFOCOM, 2005. **Proceedings...** [S.l.: s.n.], 2005. p.398–408.
- CAMPBELL, A. et al. The Rise of People-Centric Sensing. **IEEE Internet Computing**, [S.l.], v.12, 2008.
- CAMPBELL, A. T. et al. People-centric urban sensing. In: WIRELESS INTERNET - WICON '06, 2., 2006, New York, New York, USA. **Proceedings...** ACM Press, 2006. p.18–es.
- CAN, Z.; DEMIRBAS, M. Smartphone-based data collection from wireless sensor networks in an urban environment. **Journal of Network and Computer Applications**, [S.l.], 2015.
- CAO, L. Support vector machines experts for time series forecasting. **Neurocomputing**, [S.l.], v.51, p.321–339, 2003.

CAO, Y.; SUN, Z. Routing in Delay/Disruption Tolerant Networks: a taxonomy, survey and challenges. **IEEE Communications Surveys & Tutorials**, [S.l.], v.15, n.2, p.654–677, Jan. 2013.

CARDONE, G. et al. Fostering participation in smart cities: a geo-social crowdsensing platform. **IEEE Communications Magazine**, [S.l.], v.51, n.6, p.112–119, June 2013.

CARLSON, D.; SCHRADER, A. Dynamix: an open plug-and-play context framework for android. In: INTERNATIONAL CONFERENCE ON THE INTERNET OF THINGS, IOT 2012, 2012., 2012. **Proceedings...** [S.l.: s.n.], 2012. p.151–158.

CARVALHO, W. V. de. **Mobility and Context Awareness for the Management of Personal Multimedia Documents** : commedia. 2010. 266p. Doutorado — University Joseph Fourier, Grenoble.

CARVALHO, W. V. de et al. CAUS: uma arquitetura sensível ao contexto e orientada a componentes para sistemas de administração de ensino. In: SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO UBIQUA E PERVASIVA 2009, 2009, Bento Gonçalves - RS. **Anais...** [S.l.: s.n.], 2009.

CASS, R.; DEPIETRO, J. Computational intelligence methods for process discovery. **Engineering Applications of Artificial Intelligence**, [S.l.], v.11, n.5, p.675–681, 1998.

CELINO, I.; KOTOULAS, S. Smart Cities [Guest editors' introduction]. **IEEE Internet Computing**, [S.l.], v.17, n.6, p.8–11, Nov. 2013.

CERF, V. et al. **RFC Draft 4878 - Delay-Tolerant Networking Architecture**. [S.l.]: IETF, 2007.

CHEN, F. et al. The network security situation predicting technology based on the small-world echo state network. In: IEEE 4TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND SERVICE SCIENCE, 2013., 2013. **Anais...** IEEE, 2013. p.377–380.

CHEN, G.; KOTZ, D. **A Survey of Context-Aware Mobile Computing Research**. [S.l.]: Dept. of Computer Science, Dartmouth College, 2000.

CHEN, Y. et al. Cruisers: a public automotive sensing platform for smart cities. In: IEEE 36TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS (ICDCS), 2016., 2016. **Anais...** IEEE, 2016. p.767–768.

CHERIAN, J. et al. ParkGauge: gauging the occupancy of parking garages with crowdsensed parking characteristics. **Proceedings - IEEE International Conference on Mobile Data Management**, [S.l.], v.2016-July, p.92–101, 2016.

CIOBANU, R. I.; DOBRE, C.; CRISTEA, V. SPRINT: social prediction-based opportunistic routing. **2013 IEEE 14th International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2013**, [S.l.], 2013.

CONTI, M. et al. From opportunistic networks to opportunistic computing. **IEEE Communications Magazine**, [S.l.], v.48, n.9, p.126–139, Sept. 2010.

- CONTI, M. et al. From MANET to people-centric networking: milestones and open research challenges. **Computer Communications**, [S.l.], v.000, p.1–21, 2015.
- CONTI, M.; GIORDANO, S. Multihop Ad Hoc Networking: the reality. **IEEE Communications Magazine**, [S.l.], v.45, n.4, p.88–95, Apr. 2007.
- CONTI, M.; KUMAR, M. Opportunities in Opportunistic Computing. **Computer**, [S.l.], v.43, n.1, p.42–50, jan 2010.
- COOK, D. J. et al. Ambient Intelligence: technologies, applications, and opportunities. **Electrical Engineering**, [S.l.], p.1–38, 2007.
- CORNELIUS, C. et al. AnonySense : privacy-aware people-centric sensing categories and subject descriptors. In: MOBISYS'08 - PROCEEDING OF THE 6TH INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES, 2008, Colorado, USA. **Anais...** [S.l.: s.n.], 2008.
- COSTA, P. et al. Socially-aware routing for publish-subscribe in delay-tolerant mobile ad hoc networks. **IEEE Journal on Selected Areas in Communications**, [S.l.], v.26, p.748–760, 2008.
- COSTANTINO, G. et al. LoSeRO. In: ANNUAL ACM SYMPOSIUM ON APPLIED COMPUTING - SAC '16, 31., 2016, New York, New York, USA. **Proceedings...** ACM Press, 2016. p.644–650.
- CRAENEN, B.; EIBEN, A. Computational Intelligence. In: **Encyclopedia of Life Support Sciences (EOLSS)**. [S.l.]: EOLSS Publishers Co. Ltd, 2009. p.142.
- CRISTOFARO, E. D.; PIETRO, R. D. Preserving Query Privacy in Urban Sensing Systems. In: DISTRIBUTED COMPUTING AND NETWORKING, 13., 2012. **Proceedings...** [S.l.: s.n.], 2012. n.1, p.218–233.
- CUFF, D.; HANSEN, M.; KANG, J. Urban sensing. **Communications of the ACM**, [S.l.], v.51, n.3, p.24–33, Mar. 2008.
- DALY, E. M.; HAAHR, M. Social network analysis for routing in disconnected delay-tolerant MANETs. In: ACM INTERNATIONAL SYMPOSIUM ON MOBILE AD HOC NETWORKING AND COMPUTING - MOBIHOC '07, 8., 2007. **Proceedings...** [S.l.: s.n.], 2007. p.32–40.
- DAS, T. et al. PRISM: platform for remote sensing using smartphones. In: MOBILE SYSTEMS, APPLICATIONS, AND SERVICES - MOBISYS '10, 8., 2010, New York, New York, USA. **Proceedings...** ACM Press, 2010. p.63–76.
- DATTA, S. K. et al. oneM2M architecture based IoT framework for mobile crowd sensing in smart cities. In: EUROPEAN CONFERENCE ON NETWORKS AND COMMUNICATIONS (EUCNC), 2016., 2016. **Anais...** IEEE, 2016. n.ii, p.168–173.
- DEMMER, M. et al. **Implementing Delay Tolerant Networking**. [S.l.: s.n.], 2004.
- DEY, A. K.; ABOWD, G. D.; SALBER, D. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. **Human-Computer Interaction (HCI) Journal**, [S.l.], v.16, n.2-4, p.97–166, 2001.

DEY, A. K. et al. Towards a Better Understanding of Context and Context-Awareness. In: **HANDHELD AND UBIQUITOUS COMPUTING (HUC '99)**, 1., 1999, London, UK. **Proceedings...** Springer-Verlag, 1999. p.304–307.

DHURANDHER, S. K. et al. EDR: an encounter and distance based routing protocol for opportunistic networks. **2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)**, [S.l.], p.297–302, 2016.

DIACONESCU, E. The Use of NARX Neural Networks to Predict Chaotic Time Series. **WSEAS Trans. Comp. Res.**, [S.l.], v.3, n.3, p.182–191, 2008.

DUCH, W. What is Computational Intelligence and what could it become ? **International Journal of Computational Intelligence and Applications**, [S.l.], 2007.

EAGLE, N.; PENTLAND, A. Reality mining: sensing complex social systems. **Personal and Ubiquitous Computing**, [S.l.], v.10, n.4, p.255–268, may 2006.

EISENMAN, S. B. et al. MetroSense Project : people-centric sensing at scale. In: **WORKSHOP ON WORLD-SENSOR-WEB (WSW 2006)**, 2006, Boulder. **Proceedings...** [S.l.: s.n.], 2006. p.6–11.

ELIASSON, J.; Chen Zhong; DELSING, J. A heterogeneous sensor network architecture for highly mobile users. In: **SIXTH INTERNATIONAL CONFERENCE ON WIRELESS COMMUNICATION AND SENSOR NETWORKS**, 2010., 2010. **Anais...** IEEE, 2010. p.1–6.

ENDSLEY, M. R. Design and evaluation for situation awareness enhancement. In: **HUMAN FACTORS SOCIETY 32ND ANNUAL MEETING**, 1988, Santa Monica, CA, U.S.A. **Proceedings...** [S.l.: s.n.], 1988. p.97–101.

ENDSLEY, M. R. **Toward a Theory of Situation Awareness in Dynamic Systems**. 1995. 32–64p. v.37, n.1.

ENDSLEY, M. R. Theoretical Underpinnings of Situation Awareness: a critical review. In: ENDSLEY, M. R.; GARLAND, D. J. (Ed.). **Situation Awareness Analysis and Measurement**. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 2000.

ENDSLEY, M. R. Situation Awareness. In: SALVENDY, G. (Ed.). **Handbook of Human Factors and Ergonomics**. [S.l.]: John Wiley & Sons Ltd, 2006. p.528–542.

ENDSLEY, M. R.; CONNORS, E. S. Situation Awareness: state of the art. **Energy**, [S.l.], p.13–16, 2008.

ENGEL, P. M. **Redes Multicamadas: o mlp e o algoritmo backpropagation**. Porto Alegre, RS: Lecture Notes, 2011.

ENGELBRECHT, A. P. **Computational intelligence: an introduction**. Second Edi.ed. University of Pretoria, South Africa: John Wiley & Sons Ltd, 2007. 628p.

FERREIRA, A. A. **Um método para design e treinamento de Reservoir Computing aplicado à previsão de séries temporais**. 2011. 126p. Doutorado — Universidade Federal de Pernambuco.

FINK, O.; ZIO, E.; WEIDMANN, U. Anticipating Railway Operation Disruption Events Based on the Analysis of Discrete-Event Diagnostic Data. **Chemical Engineering Transactions**, [S.l.], v.33, p.715–720, 2013.

FORGY, C. L. Rete: a fast algorithm for the many pattern/many object pattern match problem. **Artificial Intelligence**, [S.l.], v.19, n.1, p.17–37, 1982.

FREITAS, E. P. d. et al. Evaluation of coordination strategies for heterogeneous sensor networks aiming at surveillance applications. **Proceedings of IEEE Sensors**, [S.l.], p.591–596, 2009.

GAO, Y. et al. Mosaic: a low-cost mobile sensing system for urban air quality monitoring. **IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications**, [S.l.], p.1–9, 2016.

GELLERSEN, H.; SCHMIDT, A.; BEIGL, M. Multi-Sensor Context-Awareness in Mobile Devices and Smart Artefacts. **Mobile Networks and Applications**, [S.l.], v.7, n.5, p.341–351, 2002.

GIBSON, D. V.; KOZMETSKY, G.; SMILOR, R. W. (Ed.). **Technopolis Phenomenon: smart cities, fast systems, global networks**. [S.l.]: Rowman & Littlefield Publishers, Inc, 1992.

GIRISH, S. Privacy-preserving by a trajectory for participatory sensing in wireless sensor networks. **International Journal of Technical Research and Applications**, [S.l.], v.3, n.3, p.138–141, 2015.

GIRTELSCHMID, S. et al. On the application of Big Data in future large scale intelligent Smart City installations. **International Journal of Pervasive Computing and Communications**, [S.l.], v.10, n.2, p.1–15, 2014.

GIUSEPPE, C. et al. Effective collaborative monitoring in smart cities: converging manet and wsn for fast data collection. In: KALEIDOSCOPE 2011: THE FULLY NETWORKED HUMAN? - INNOVATIONS FOR FUTURE NETWORKS AND SERVICES (K-2011), PROCEEDINGS OF ITU, 2011, Cape Town. **Anais... IEEE**, 2011. p.1 – 8.

GRAY, P.; SALBER, D. Modelling and Using Sensed Context Information in the Design of Interactive Applications. In: IN LNCS 2254: PROCEEDINGS OF 8TH IFIP INTERNATIONAL CONFERENCE ON ENGINEERING FOR HUMAN-COMPUTER INTERACTION (EHCI 2001), 2001, Toronto, Canada. **Anais... Springer**, 2001. p.317–336.

GRAY, P.; SALBER, D. Modelling and using sensed context information in the design of interactive applications. **Computer**, [S.l.], v.2254, n.1, p.317–335, 2010.

HALLAK, R.; Pereira Filho, A. J. Metodologia para análise de desempenho de simulações de sistemas convectivos na região metropolitana de São Paulo com o modelo ARPS: sensibilidade a variações com os esquemas de advecção e assimilação de dados. **Revista Brasileira de Meteorologia**, [S.l.], v.26, n.4, p.591–608, Dec. 2011.

HAYKIN, S. **Redes Neurais - Principios e práticas**. 2.ed. [S.l.: s.n.], 2011.

- HEARST, M. A. et al. Trends and controversies - support vector machines. **IEEE Intelligent Systems**, [S.l.], v.13, n.4, p.18–28, 1998.
- HIGUCHI, T.; YAMAGUCHI, H.; HIGASHINO, T. Context-supported local crowd mapping via collaborative sensing with mobile phones. **Pervasive and Mobile Computing**, [S.l.], v.13, p.26–51, 2014.
- HIGUCHI, T.; YAMAGUCHI, H.; HIGASHINO, T. Mobile Devices as an Infrastructure: a survey of opportunistic sensing technology. **Journal of Information Processing**, [S.l.], v.23, n.2, p.94–104, 2015.
- HUI, P. **People are the network**: experimental design and evaluation of social-based forwarding algorithms. Cambridge: University of Cambridge, Computer Laboratory, 2008.
- HUI, P.; CROWCROFT, J. How small labels create big improvements. In: FIFTH ANNUAL IEEE INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING AND COMMUNICATIONS WORKSHOPS, PERCOM WORKSHOPS 2007, 2007. **Proceedings...** [S.l.: s.n.], 2007. p.65–70.
- HUI, P.; CROWCROFT, J.; YONEKI, E. BUBBLE Rap: social-based forwarding in delay-tolerant networks. **IEEE Transactions on Mobile Computing**, [S.l.], v.10, n.11, p.1576–1589, Nov. 2011.
- HUI, P. et al. Pocket Switched Networking: challenges, feasibility and implementation issues. **Autonomic Communication - Lecture Notes in Computer Science**, Fraunhofer Institut FOKUS, v.3854, p.1–12, 2006.
- IPIÑA, D. L. d. et al. Internet of Things, Linked Data, and Citizen Participation as Enablers of Smarter Cities. **International Journal of Distributed Sensor Networks**, [S.l.], v.2016, p.1–2, 2016.
- IVANCIC, W. et al. **Store, Carry and Forward Problem Statement**. [S.l.]: Internet Engineering Task Force, 2014. Internet-Draft, Work in Progress. (draft-ivancic-scf-problem-statement-01).
- JABEUR, N.; ZEADALLY, S.; SAYED, B. Mobile social networking applications. **Communications of the ACM**, [S.l.], v.56, n.3, p.71, Mar. 2013.
- JAEGER, H. **The echo state approach to analysing and training recurrent neural networks. Technical report GMD report 148**. [S.l.]: German National Research Center for Information Technology, 2001.
- JAEGER, H. **A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. GMD Report 159**. [S.l.]: German National Research Center for Information Technology, 2002.
- JANG, J. S. R. ANFIS: adaptive-network-based fuzzy inference system. **Ieee Transactions On Systems Man And Cybernetics**, [S.l.], v.23, n.3, p.665–685, 1993.
- JASSBI, J. et al. Transformation of a mamdani FIS to first order sugeno FIS. **IEEE International Conference on Fuzzy Systems**, [S.l.], v.516, 2007.

- JAWHAR, I.; MOHAMED, N.; AL-JAROODI, J. Data communication in linear wireless sensor networks using Unmanned Aerial Vehicles. In: INTERNATIONAL CONFERENCE ON UNMANNED AIRCRAFT SYSTEMS (ICUAS), 2014., 2014. **Anais...** IEEE, 2014. p.43–51.
- JOHNS, A. Applications of fuzzy logic in power systems. Part 2: comparison and integration with expert systems, neural networks and genetic algorithms. **Power Engineering Journal**, [S.l.], v.12, n.4, p.185–190, Aug. 1998.
- KANG, J.; CUFF, D. **Pervasive Computing** : embedding the public sphere. Los Angeles, CA: UCLA School of Law, 2005.
- KARIMI, H. A.; LIU, X. A predictive location model for location-based services. In: ACM INTERNATIONAL SYMPOSIUM ON ADVANCES IN GEOGRAPHIC INFORMATION SYSTEMS - GIS 2003, 2003, New York, New York, USA. **Proceedings...** ACM Press, 2003. p.126–133.
- KAUR, A.; KAUR, A. Comparison of Mamdani-Type and Sugeno-Type Fuzzy Inference Systems for Air Conditioning System. **International Journal of Soft Computing and Engineering**, [S.l.], v.2, n.2, p.323–325, 2012.
- KERÄNEN, A.; KÄRKKÄINEN, T.; OTT, J. Simulating Mobility and DTNs with the ONE (Invited Paper). **Journal of Communications**, [S.l.], v.5, n.2, Feb. 2010.
- KERÄNEN, A.; OTT, J.; KÄRKKÄINEN, T. The ONE simulator for DTN protocol evaluation. In: SECOND INTERNATIONAL ICST CONFERENCE ON SIMULATION TOOLS AND TECHNIQUES, 2009. **Proceedings...** ICST, 2009.
- KHAN, A.; IMON, S. K. A.; DAS, S. K. An Energy Efficient Framework for Localization and Coverage in Participatory Urban Sensing. In: ANNUAL IEEE CONFERENCE ON LOCAL COMPUTER NETWORKS, 39., 2014. **Anais...** [S.l.: s.n.], 2014. p.193–201.
- KHAN, J. A.; GHAMRI-DOUDANE, Y. Car rank: an information-centric identification of important smart vehicles for urban sensing. **Proceedings - 2015 IEEE 14th International Symposium on Network Computing and Applications, NCA 2015**, [S.l.], p.184–191, 2016.
- KHAN, W. Z. et al. Mobile Phone Sensing Systems: a survey. **IEEE Communications Surveys & Tutorials**, [S.l.], v.15, n.1, p.402–427, 2013.
- KIM, K.-j. Financial time series forecasting using support vector machines. **Neurocomputing**, [S.l.], v.55, n.1-2, p.307–319, 2003.
- KOCH, F. et al. A Platform for Citizen Sensing in Sentient Cities. **Lecture Notes in Computer Science**, [S.l.], v.7685, p.57–66, 2013.
- KUMAR, S. et al. Urban Street Lighting Infrastructure Monitoring Using a Mobile Sensor Platform. **IEEE Sensors Journal**, [S.l.], v.16, n.12, p.4981–4994, jun 2016.
- LANE, N. D. et al. Urban sensing systems: opportunistic or participatory? In: MOBILE COMPUTING SYSTEMS AND APPLICATIONS - HOTMOBILE '08, 9., 2008, New York, New York, USA. **Proceedings...** ACM Press, 2008. p.11.

LAVRIC, A.; POPA, V.; SFICHI, S. Street lighting control system based on large-scale WSN: a step towards a smart city. In: INTERNATIONAL CONFERENCE AND EXPOSITION ON ELECTRICAL AND POWER ENGINEERING (EPE), 2014., 2014. **Anais...** IEEE, 2014. n.Epe, p.673–676.

LEBRUN, J. et al. Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks. **2005 IEEE 61st Vehicular Technology Conference**, [S.l.], v.4, 2005.

LEE, J.; LEE, H. Developing and validating a citizen-centric typology for smart city services. **Government Information Quarterly**, [S.l.], May 2014.

LI, F.; WU, J. LocalCom: a community-based epidemic forwarding scheme in disruption-tolerant networks. In: ANNUAL IEEE COMMUNICATIONS SOCIETY CONFERENCE ON SENSOR, MESH AND AD HOC COMMUNICATIONS AND NETWORKS, SECON 2009, 2009., 2009. **Anais...** [S.l.: s.n.], 2009.

LI, R.-a.; SHA, X.; LIN, K. Smart greenhouse: a real-time mobile intelligent monitoring system based on wsn. **2014 International Wireless Communications and Mobile Computing Conference (IWCMC)**, [S.l.], p.1152–1156, 2014.

LIEBERMAN, H.; SELKER, T. Out of context: computer systems that adapt to, and learn from, context. **IBM Systems Journal**, [S.l.], v.39, n.3-4, p.617–632, 2000.

LIN, T. et al. Learning long-term dependencies in NARX recurrent neural networks. **IEEE Transactions on Neural Networks**, [S.l.], v.7, n.6, p.1329–1338, 1996.

LINDGREN, A.; DORIA, A.; SCHELÉN, O. Probabilistic routing in intermittently connected networks. **ACM SIGMOBILE Mobile Computing and Communications Review**, [S.l.], v.7, n.3, p.19, July 2003.

LINDGREN, A.; HUI, P. The quest for a killer app for opportunistic and delay tolerant networks. In: ACM WORKSHOP ON CHALLENGED NETWORKS - CHANTS '09, 4., 2009, New York, New York, USA. **Proceedings...** ACM Press, 2009. p.59.

LINO, J. A.; SALEM, B.; RAUTERBERG, M. Responsive environments: user experiences for ambient intelligence. **Journal of Ambient**, [S.l.], v.2, n.4, p.347–367, 2010.

LIU, J. et al. Software-defined internet of things for smart urban sensing. **IEEE Communications Magazine**, [S.l.], v.53, n.9, p.55–63, 2015.

LIU, J.-H. et al. Developed urban air quality monitoring system based on wireless sensor networks. In: FIFTH INTERNATIONAL CONFERENCE ON SENSING TECHNOLOGY, 2011., 2011. **Anais...** IEEE, 2011. p.549–554.

LIU, L. Friend-Based Prediction Routing Protocol in Socially-Aware Opportunistic Networks. **Applied Mechanics and Materials**, [S.l.], v.519-520, p.241–242, Feb. 2014.

LUKOSEVICIUS, M.; JAEGER, H. Reservoir Computing Approaches to Recurrent Neural Network Training. **Preprint submitted to Computer Science Review**, [S.l.], 2010.

MAASS, W.; NATSCHLÄGER, T.; MARKRAM, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. **Neural computation**, [S.l.], v.14, n.11, p.2531–60, Nov. 2002.

MACIAS, E.; SUAREZ, A.; LLORET, J. Mobile sensing systems. **Sensors (Basel, Switzerland)**, [S.l.], v.13, n.12, p.17292–321, Jan. 2013.

MACIEL, L. et al. Exchange rate forecasting using echo state networks for trading strategies. In: IEEE CONFERENCE ON COMPUTATIONAL INTELLIGENCE FOR FINANCIAL ENGINEERING & ECONOMICS (CIFER), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.40–47.

MAKHLOUTA, J. et al. Adaptive Fuzzy Spray and Wait: efficient routing for opportunistic networks. In: INTERNATIONAL CONFERENCE ON SELECTED TOPICS IN MOBILE AND WIRELESS NETWORKING (ICOST), 2011., 2011. **Anais...** IEEE, 2011. p.64–69.

MAQBOOL, B. B.; PEER, M. Classification of Current Routing Protocols for Ad Hoc Networks - A Review. **International Journal of Computer Applications**, [S.l.], v.7, n.8, p.26–32, Oct. 2010.

MATHURAPOJ, A.; PORNAVALAI, C.; CHAKRABORTY, G. Fuzzy-Spray: efficient routing in delay tolerant ad-hoc network based on fuzzy decision mechanism. **2009 IEEE International Conference on Fuzzy Systems**, [S.l.], p.104–109, Aug. 2009.

MENEZES, J. M. P.; BARRETO, G. a. Long-term time series prediction with the NARX network: an empirical evaluation. **Neurocomputing**, [S.l.], v.71, n.16-18, p.3335–3343, 2008.

MILUZZO, E. et al. Sensing Meets Mobile Social Networks : the design , implementation and evaluation of the cenceme application. **SenSys '08**, [S.l.], p.337–350, 2008.

MIN, J.; LEE, Y. Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters. **Expert Systems with Applications**, [S.l.], v.28, n.4, p.603–614, 2005.

MORANDO, S. et al. Fuel Cells Remaining Useful Lifetime Forecasting Using Echo State Network. In: IEEE VEHICLE POWER AND PROPULSION CONFERENCE (VPPC), 2014., 2014. **Anais...** [S.l.: s.n.], 2014. p.1–6.

MORDACCHINI, M. et al. Design and evaluation of a cognitive approach for disseminating semantic knowledge and content in opportunistic networks. **Computer Communications**, [S.l.], v.81, p.12–30, may 2016.

MOREIRA, W.; MENDES, P. **Survey on Opportunistic Routing for Delay / Disruption Tolerant Networks - SITI Technical Report SITI-TR-11-02**. Lisboa, Portugal: Lusófona University, 2010.

MOREIRA, W.; MENDES, P. Impact of human behavior on social opportunistic forwarding. **Ad Hoc Networks**, [S.l.], v.25, p.293–302, Feb. 2015.

- MOREIRA, W.; MENDES, P.; SARGENTO, S. Opportunistic routing based on daily routines. In: IEEE INTERNATIONAL SYMPOSIUM ON A WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS, WOWMOM 2012 - DIGITAL PROCEEDINGS, 2012., 2012. **Anais...** [S.l.: s.n.], 2012.
- MOREIRA, W.; MENDES, P.; SARGENTO, S. Social-Aware Opportunistic Routing Protocol Based on User's Interactions and Interests. **Ad Hoc Networks**, [S.l.], v.129, p.29–31, 2014.
- MORELLI, A. et al. Mobility Pattern Prediction to Support Opportunistic Networking in Smart Cities. **2013 International Conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications**, [S.l.], p.166–175, Nov. 2013.
- MOTA, V. F. et al. Protocols, mobility models and tools in opportunistic networks: a survey. **Computer Communications**, [S.l.], v.48, p.5–19, July 2014.
- MTIBAA, A. et al. PeopleRank: social opportunistic forwarding. In: IEEE INFOCOM, 2010. **Proceedings...** [S.l.: s.n.], 2010.
- MUNOZ, J. H. et al. Smart cities at the forefront of the future internet. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, [S.l.], v.6656, p.447–462, 2011.
- MUSOLESI, M. **Context-aware Adaptive Routing for Delay Tolerant Networking**. 2007. 150p. PhD Thesis — University of London. (May).
- MUSOLESI, M. et al. Writing on the clean slate: implementing a socially-aware protocol in haggie. In: IEEE INTERNATIONAL SYMPOSIUM ON A WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS, WOWMOM2008, 2008., 2008. **Anais...** [S.l.: s.n.], 2008.
- MYLOPOULOS, J. Conceptual modelling and Telos. **Conceptual Modelling, Databases, and CASE: an Integrated View of Information System Development**, John Wiley & Sons, New York, New York, [S.l.], 1992.
- NABHANI, P.; Masoud Bidgoli, A. Adaptive Fuzzy Routing in Opportunistic Network (AFRON). **International Journal of Computer Applications**, [S.l.], v.52, n.18, p.7–11, Aug. 2012.
- NELSON, S. C.; BAKHT, M.; KRAVETS, R. Encounter-based routing in DTNs. In: IEEE INFOCOM, 2009. **Proceedings...** [S.l.: s.n.], 2009. p.846–854.
- NG, C. et al. An RFID-Enabled Wireless Sensor Network (WSN) Monitoring System for Biological and Pharmaceutical Products. **2014 International Symposium on Computer, Consumer and Control**, [S.l.], p.757–760, 2014.
- NGUYEN, H. A.; GIORDANO, S. Routing in Opportunistic Networks. **International Journal of Ambient Computing and Intelligence**, [S.l.], v.1, n.3, p.19–38, Jan. 2009.
- NGUYEN, H. A.; GIORDANO, S. Routing in Opportunistic Networks. In: CURRAN, K. (Ed.). **Ubiquitous Developments in Ambient Computing and Intelligence**. [S.l.]: IGI Global, 2011. p.179–193.

NGUYEN, H. A.; GIORDANO, S. Context information prediction for social-based routing in opportunistic networks. **Ad Hoc Networks**, [S.l.], v.10, n.8, p.1557–1569, Nov. 2012.

NGUYEN, H. A.; GIORDANO, S.; PUIATTI, A. Probabilistic Routing Protocol for Intermittently Connected Mobile Ad hoc Network (PROPICMAN). **2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks**, [S.l.], 2007.

NUGROHO, H. A.; JOELIANTO, E.; WIDIYANTORO, S. Time series estimation of earthquake occurrences in Bali and its surroundings using NARX network model. In: INTERNATIONAL CONFERENCE ON INSTRUMENTATION CONTROL AND AUTOMATION (ICA), 2013., 2013. **Anais...** [S.l.: s.n.], 2013. p.251–256.

OLIVEIRA, C.; GONCALVES, G. Environmental Monitoring Services and wireless sensor networks applied on Urban spaces. In: IEEE INTERNATIONAL CONFERENCE ON AUTOMATION, QUALITY AND TESTING, ROBOTICS (AQTR), 2010., 2010. **Anais...** IEEE, 2010. p.1–6.

OLIVEIRA, C. T. de. **Uma Proposta de Roteamento Probabilístico para Redes Tolerantes a Atrasos e Desconexões**. 2008. 100p. Master — UFRJ.

OLIVEIRA, C. T. et al. **Redes Tolerantes a Atrasos e Desconexões**. 2007. 54p.

OLIVEIRA, E. C. R. de; ALBUQUERQUE, C. V. N. de. CARTOON - Context Aware Routing Over Opportunistic Networks. In: XXX BRAZILIAN SYMPOSIUM ON COMPUTER NETWORKS AND DISTRIBUTED SYSTEMS, 2012, Ouro Preto, MG, Brazil. **Proceedings...** [S.l.: s.n.], 2012. p.872–885.

OLIVEIRA, H. E. **HEWERTON ENES DE OLIVEIRA GRUBIROUTER** : um protocolo de roteamento para redes ~ de. 2014. 87pp. Master — Universidade Federal de Lavras.

ORLINSKI, M.; FILER, N. The rise and fall of spatio-temporal clusters in mobile ad hoc networks. **Ad Hoc Networks**, [S.l.], Mar. 2013.

PALAZZI, C. E. et al. An Overview of Opportunistic Ad Hoc Communication in Urban Scenarios. In: THE 13TH IEEE IFIP ANNUAL MEDITERRANEAN AD HOC NETWORKING WORKSHOP (IEEE/IFIP MEDHOCNET 2014), 2014, Piran, Slovenia. **Proceedings...** [S.l.: s.n.], 2014.

PELLICER, S. et al. A Global Perspective of Smart Cities: a survey. **2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing**, [S.l.], p.439–444, July 2013.

PEREIRA, I. S. A.; COSTA, P. D.; ALMEIDA, J. P. A. A rule-based platform for situation management. **2013 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, CogSIMA 2013**, [S.l.], p.83–90, 2013.

PERTTUNEN, M.; RIEKKI, J.; LASSILA, O. Context Representation and Reasoning in Pervasive Computing: a review. **International Journal of Multimedia and Ubiquitous Engineering**, [S.l.], v.4, n.4, p.1–28, 2009.

PILKA, F.; ORAVEC, M. Multi-step ahead prediction using neural networks. **ELMAR, 2011 Proceedings**, [S.l.], p.269–272, 2011.

PIROZMAND, P. et al. Human mobility in opportunistic networks: characteristics, models and prediction methods. **Journal of Network and Computer Applications**, [S.l.], Mar. 2014.

POONGUZHARSELVI, B.; VETRISSELVI, V. Survey on routing algorithms in opportunistic networks. **2013 International Conference on Computer Communication and Informatics**, [S.l.], p.1–5, Jan. 2013.

PRABHA, C.; KUMAR, S.; KHANNA, R. Analysis of Routing and Forwarding Protocols in Opportunistic Networks. **Procedia Computer Science**, [S.l.], v.85, n.Cms, p.891–898, 2016.

PRZYBYLEK, M. R. Computational Intelligence. **Studies in Computational Intelligence**, [S.l.], v.465, p.119–134, 2013.

RA, M.-r. et al. Medusa: a programming framework for crowd-sensing applications. In: **MOBILE SYSTEMS, APPLICATIONS, AND SERVICES - MOBISYS '12**, 10., 2012, New York, New York, USA. **Proceedings...** ACM Press, 2012. n.Section 2, p.337.

RABIN, M. J. A. et al. Electrical load forecasting using echo state network. **2012 15th International Conference on Computer and Information Technology (ICIT)**, [S.l.], p.50–54, Dec. 2012.

RAHIM, M. S. et al. On the performance of delay - Tolerant routing protocols in underwater networks. In: **OCEANS 2011 IEEE - SPAIN, 2011**. **Anais...** [S.l.: s.n.], 2011.

ROCHA, L. S. **Dependabilidade em Sistemas Ubíquos**. Fortaleza, CE: Universidade Federal do Ceará, 2010.

RODRIGUES, J. G. P.; AGUIAR, A.; BARROS, J. SenseMyCity: crowdsourcing an urban sensor. **CoRR**, [S.l.], v.abs/1412.2070, p.10, dec 2014.

RODRIGUES-SILVA, D.; COSTA, A.; MACEDO, J. Energy Impact Analysis on DTN Routing Protocols. In: **EXTREMECOM' 12, 2012**, Zurich, Switzerland. **Anais...** [S.l.: s.n.], 2012.

ROLIM, C. O. et al. Six degrees of separation to improve routing in opportunistic networks. **International Journal of UbiComp**, [S.l.], v.4, n.3, p.11–22, 2013.

ROLIM, C. O. et al. Towards Predictive Routing Agents in Opportunistic Networks. In: **FIFTH INTERNATIONAL WORKSHOP ON COLLABORATIVE AGENTS RESEARCH & DEVELOPMENT, CARE FOR INTELLIGENT MOBILE SERVICES - PROCEEDINGS OF THE 13TH INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS AND MULTIAGENT SYSTEMS (AAMAS)**, 2014, Paris, France. **Anais...** Springer-Verlag Berlin Heidelberg, 2014.

ROSA, J. H. da; BARBOSA, J. L.; RIBEIRO, G. D. ORACON: an adaptive model for context prediction. **Expert Systems with Applications**, [S.l.], v.45, p.56–70, mar 2016.

RUSLAN, F. A. et al. **Flood water level modeling and prediction using NARX neural network**: case study at kelang river. 2014. 204–207p.

SALIM, F.; HAQUE, U. Urban computing in the wild: a survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and internet of things. **International Journal of Human-Computer Studies**, [S.l.], p.1–18, 2015.

SANCHEZ-IBORRA, R.; CANO, M.-D. JOKER: a novel opportunistic routing protocol. **IEEE Journal on Selected Areas in Communications**, [S.l.], v.34, n.6, p.In press, 2016.

SANTOS, F. N. dos et al. CEDO Content-Centric Dissemination Algorithm for Delay-Tolerant Networks. In: **ACM INTERNATIONAL CONFERENCE ON MODELING, ANALYSIS & SIMULATION OF WIRELESS AND MOBILE SYSTEMS - MSWIM '13, 16.**, 2013, New York, New York, USA. **Proceedings...** ACM Press, 2013. p.377–386.

SCHAFFERS, H. et al. Smart Cities and the Future Internet : towards cooperation frameworks for open innovation. **The Future Internet Lecture Notes in Computer Science**, [S.l.], v.6656, p.431–446, 2011.

SCHILIT, B.; ADAMS, N.; WANT, R. Context-aware computing applications. In: **IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS, 1994**, Santa Cruz, California. **Proceedings...** IEEE Computer Society Press, 1994. p.85–90.

SCOTT, K.; BURLEIGH, S. **RFC 5050 - Bundle Protocol Specification**. 2007.

SHARMA, A.; TAKKAR, S. Routing Protocols in Wireless Ad-hoc Networks : a comparative study. **International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)**, [S.l.], v.3, n.5, p.108–111, 2013.

SHIN, M. et al. AnonySense: a system for anonymous opportunistic sensing. **Pervasive and Mobile Computing**, [S.l.], v.7, n.1, p.16–30, Feb. 2011.

SHU, K. et al. Prediction based on support vector machine for travel choice of high-speed railway passenger in China. In: **INTERNATIONAL CONFERENCE ON MANAGEMENT SCIENCE AND ENGINEERING - ANNUAL CONFERENCE PROCEEDINGS, 2011**. **Anais...** [S.l.: s.n.], 2011. p.28–33.

SIEGELMANN, H. T.; HORNE, B. G.; GILES, C. L. Computational capabilities of recurrent NARX neural networks. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, [S.l.], v.27, n.2, p.208–215, 1997.

SILVA, T. et al. Large-scale study of city dynamics and urban social behavior using participatory sensing. **IEEE Wireless Communications**, [S.l.], v.21, n.1, p.42–51, Feb. 2014.

SOCIEVOLE, A.; De Rango, F.; CAPUTO, A. Opportunistic mobile social networks: from mobility and facebook friendships to structural analysis of user social behavior. **Computer Communications**, [S.l.], 2016.

SOTTARA, D.; MELLO, P.; PROCTOR, M. Adding uncertainty to a Rete-OO inference engine. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, [S.l.], v.5321 LNCS, p.104–118, 2008.

SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. Single-copy routing in intermittently connected mobile networks. In: FIRST ANNUAL IEEE COMMUNICATIONS SOCIETY CONFERENCE ON SENSOR AND AD HOC COMMUNICATIONS AND NETWORKS, 2004. IEEE SECON 2004., 2004., 2004. **Anais...** Ieee, 2004. p.235–244.

SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. Spray and wait. In: PROCEEDING OF THE 2005 ACM SIGCOMM WORKSHOP ON DELAY-TOLERANT NETWORKING - WDTN '05, 2005, New York, New York, USA. **Anais...** ACM Press, 2005. p.252–259.

SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. Efficient routing in intermittently connected mobile networks: the multiple-copy case. **IEEE/ACM Transactions on Networking**, [S.l.], v.16, p.77–90, 2008.

STRANG, T.; LINNHOF-POPIEN, C. A Context Modeling Survey. **Graphical Models**, [S.l.], v.Workshop o, p.1–8, 2004.

SUMATHI, S.; PANEERSELVAM, S. **Computational Intelligence Paradigms: theory & applications using matlab**. 1st.ed. Boca Raton, FL, USA: CRC Press, Inc., 2010. 844p.

THEODORIDIS, E. et al. Large-Scale Participatory Sensing Experimentation Using Smartphones within a Smart City. In: INTERNATIONAL CONFERENCE ON MOBILE AND UBIQUITOUS SYSTEMS: COMPUTING, NETWORKING AND SERVICES, 11., 2014. **Proceedings...** [S.l.: s.n.], 2014. p.178–187.

TORRES, S. et al. BeCity: sensing and sensibility on urban cycling for smarter cities. In: INTERNATIONAL CONFERENCE OF THE CHILEAN COMPUTER SCIENCE SOCIETY (SCCC), 2015., 2015. **Anais...** IEEE, 2015. p.1–4.

TREBATICKÝ, P. Prediction of Dynamical Systems by Recurrent Neural Networks. **Information Sciences and Technologies Bulletin of the ACHM Slovakia**, [S.l.], v.1, n.1, p.47–56, 2009.

VAHDAT, A.; BECKER, D. **Epidemic Routing for Partially-Connected Ad Hoc Networks**. [S.l.]: Duke University, 2000.

VAPNIK, V. N. **The nature of statistical learning theory**. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

VENECIAN, L. R. **EXEHDA-SS: um mecanismo para sensibilidade ao contexto com suporte semântico**. 2010. 109p. Mestrado em Ciência da Computação — UCPEL, Pelotas, RS.

VERMA, A. Integrated Routing Protocol for Opportunistic Networks. **(IJACSA) International Journal of Advanced Computer Science and Applications**, [S.l.], v.2, n.3, p.85–92, 2011.

VERSTRAETEN, D. **Reservoir Computing: computation with dynamical systems**. 2009. 190p. Tese (Doutorado em Ciência da Computação) — .

VERSTRAETEN, D. et al. An experimental unification of reservoir computing methods. **Neural networks : the official journal of the International Neural Network Society**, [S.l.], v.20, n.3, p.391–403, Apr. 2007.

VU, L.; DO, Q.; NAHRSTEDT, K. 3R: fine-grained encounter-based routing in delay tolerant networks. In: IEEE INTERNATIONAL SYMPOSIUM ON A WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS, WOWMOM 2011 - DIGITAL PROCEEDINGS, 2011., 2011. **Anais...** [S.l.: s.n.], 2011.

WANG, H.; ZHANG, M.; WANG, J. Design and implementation of an Emergency Search and Rescue System based on mobile robot and WSN. In: INTERNATIONAL ASIA CONFERENCE ON INFORMATICS IN CONTROL, AUTOMATION AND ROBOTICS (CAR 2010), 2010., 2010. **Anais...** IEEE, 2010. p.206–209.

WANG, S.; CHEN, T.; XU, X. Flue gas turbine condition trend prediction based on improved echo state network. In: INTERNATIONAL CONFERENCE ON MEASURING TECHNOLOGY AND MECHATRONICS AUTOMATION, ICMTMA 2011, 3., 2011. **Proceedings...** [S.l.: s.n.], 2011. v.2, p.242–245.

WARTHMAN, F. **Delay-Tolerant Networks (DTNs) - A tutorial**. 2003. 36p. n.March.

WEISER, M. The computer for the 21st century. **Scientific American**, [S.l.], v.265, n.3, p.94–104, 1991.

WU, D. et al. ADDSEN: adaptive data processing and dissemination for drone swarms in urban sensing. **IEEE Transactions on Computers**, [S.l.], v.XX, n.XX, p.1–1, 2016.

WU, P.; ZHU, J.; ZHANG, J. Y. MobiSens: a versatile mobile sensing platform for real-world applications. **Mobile Networks and Applications**, [S.l.], v.18, n.1, p.60–80, Nov. 2012.

WULFF, N.; SOTTARA, D. Fuzzy reasoning with a rete-oo rule engine. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, [S.l.], v.5858 LNCS, p.337–344, 2009.

XIA, F. et al. Socially Aware Networking: a survey. **IEEE Systems Journal**, [S.l.], v.PP, n.99, p.1–18, 2013.

XIANG, Z.; DEYUN, X. FAULT DIAGNOSIS BASED ON THE FUZZY-RECURRENT NEURAL Zhao Xiang and Xiao Deyun. **Asian Journal of Control**, [S.l.], v.3, n.2, p.89–95, 2001.

XIE, H.; TANG, H.; LIAO, Y. H. Time series prediction based on narx neural networks: an advanced approach. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING AND CYBERNETICS, 2009., 2009. **Proceedings...** [S.l.: s.n.], 2009. v.3, p.1275–1279.

XIE, X. et al. Social relationship enhanced predicable routing in opportunistic network. In: INTERNATIONAL CONFERENCE ON MOBILE AD-HOC AND SENSOR NETWORKS, MSN 2011, 2011., 2011. **Proceedings...** [S.l.: s.n.], 2011. p.268–275.

XU, X.-l.; CHEN, T.; WANG, S.-h. Condition prediction of flue gas turbine based on Echo State Network. In: SIXTH INTERNATIONAL CONFERENCE ON NATURAL COMPUTATION, 2010., 2010. **Anais...** IEEE, 2010. p.1089–1092.

XUE, G. et al. Pothole in the Dark: perceiving pothole profiles with participatory urban vehicles. **IEEE Transactions on Mobile Computing**, [S.l.], v.1233, n.c, p.1–1, 2016.

XUE, J. X. J. et al. **Advanced PROPHET Routing in Delay Tolerant Network**. [S.l.]: Ieee, 2009. 411–413p. n.C.

YAMIN, A. C. B. et al. Towards Merging Context-Aware, Mobile and Grid Computing. **International Journal of High Performance Computing Applications**, [S.l.], v.17, n.2, p.191–203, 2003.

YAU, S. et al. Situation-Awareness for Adaptive Coordination in Service-Based Systems. **29th Annual International Computer Software and Applications Conference (COMPSAC'05)**, [S.l.], p.107–112, 2005.

YAU, S. S. et al. Development and Runtime Support for Situation-Aware Application Software in Ubiquitous Computing Environments. **Computer**, [S.l.], 2004.

YAU, S. S.; KARIM, F.; GUPTA, S. K. S. Reconfigurable context-sensitive middleware for pervasive computing. **Ieee Pervasive Computing**, [S.l.], v.1, n.3, p.33–40, 2002.

YE, J.; DOBSON, S.; MCKEEVER, S. A review of situation identification techniques in pervasive computing. **Pervasive and Mobile Computing**, [S.l.], v.In Press., n.0, 2011.

YIN, K.; JIANG, J. Application of fire monitoring and personnel evacuation in subway station based on wireless sensor network. In: INTERNATIONAL CONFERENCE ON AUDIO, LANGUAGE AND IMAGE PROCESSING, 2014., 2014. **Anais...** IEEE, 2014. p.110–114.

YOGI, M. kumar; CHINTHALA, V. A Study of Opportunistic Networks for Efficient Ubiquitous Computing. **International Journal of Advanced Research in Computer and Communication Engineering**, [S.l.], v.3, n.1, p.5187–5191, 2014.

YU, J. et al. SenSpeed: sensing driving conditions to estimate vehicle speed in urban environments. **IEEE Transactions on Mobile Computing**, [S.l.], v.15, n.1, p.202–216, jan 2016.

YU, P.; MIAO, L.; JIA, G. Clustered complex echo state networks for traffic forecasting with prior knowledge. In: IEEE INTERNATIONAL INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE, 2011., 2011. **Anais...** IEEE, 2011. p.1–5.

YUAN, Q.; CARDEI, I.; WU, J. An Efficient Prediction-Based Routing in Disruption-Tolerant Networks. **IEEE Transactions on Parallel and Distributed Systems**, [S.l.], v.23, n.1, p.19–31, Jan. 2012.

ZADEH, L. a. Is there a need for fuzzy logic? **Information Sciences**, [S.l.], v.178, n.13, p.2751–2779, July 2008.

- ZADEH, L. Fuzzy sets. **Information and Control**, [S.l.], v.8, n.3, p.338–353, June 1965.
- ZHANG, Z. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. **IEEE Communications Surveys & Tutorials**, [S.l.], v.8, n.1, p.24–37, 2006.
- ZHAO, D. et al. On Opportunistic Coverage for Urban Sensing. **2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems**, [S.l.], p.231–239, Oct. 2013.
- ZHAO, D. et al. COUPON: a cooperative framework for building sensing maps in mobile opportunistic networks. **IEEE Transactions on Parallel and Distributed Systems**, [S.l.], v.9219, n.c, p.1–1, 2014.
- ZHAO, D.; MA, H.; TANG, S. COUPON: cooperatively building sensing maps in mobile opportunistic networks. In: IEEE 10TH INTERNATIONAL CONFERENCE ON MOBILE AD-HOC AND SENSOR SYSTEMS, 2013., 2013. **Anais...** IEEE, 2013. p.295–303.
- ZHAO, J. et al. Hybrid neural prediction and optimized adjustment for coke oven gas system in steel industry. **IEEE Transactions on Neural Networks and Learning Systems**, [S.l.], v.23, n.3, p.439–450, 2012.
- ZUBIZARRETA, I.; SERAVALLI, A.; ARRIZABALAGA, S. Smart City Concept: what it is and what it should be. **Journal of Urban Planning and Development**, [S.l.], p.04015005, 2015.
- ZULKEFLEE, S. A.; SATA, S. A.; AZIZ, N. Nonlinear Autoregressive with Exogenous Inputs Based Model Predictive Control for Batch Citronellyl Laurate Esterification Reactor. **Engineering**, [S.l.], n.1992, 2007.

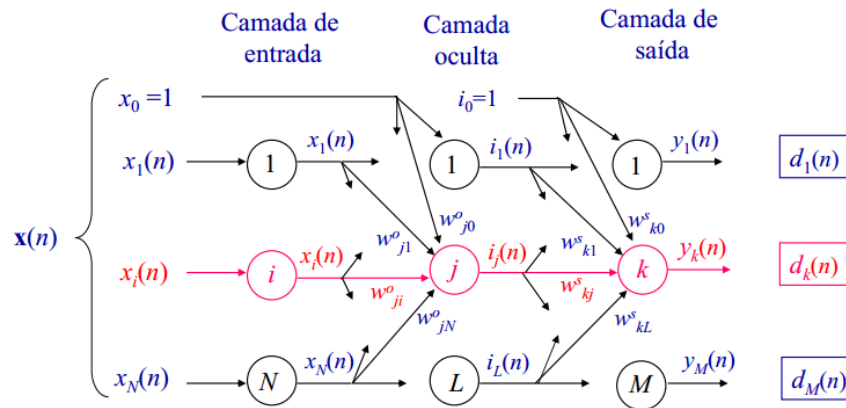
APÊNDICE A — REDES NEURAIS

Esse anexo irá apresentar informações sobre os diferentes tipos de Redes Neurais experimentadas durante o desenvolvimento da presente tese.

Perceptron Multicamadas - MLP

Dentre os vários tipos de Redes Neurais, o MLP (Multilayer Perceptron), ou perceptron multicamadas, é uma das mais utilizada em diferentes domínios e áreas de aplicações. A sua topologia é esquematizada na figura A.1.

Figura A.1: Topologia de um MLP



$i_j(n)$: valor de saída do neurônio genérico (j) da camada oculta gerado por $\mathbf{x}(n)$.

$y_k(n)$: valor de saída do neurônio genérico (k) da camada de saída gerado por $\mathbf{x}(n)$.

$d_k(n)$: valor de saída desejado do neurônio k correspondente a $\mathbf{x}(n)$.

w_{ji}^o e w_{kj}^s : pesos genéricos da camada oculta e de saída, respectivamente.

(??)

Um MLP é composta por no mínimo três camadas, a de entrada, a(s) oculta(s) e a camada de saída:

- **Camada de entrada:** composta por neurônios que distribuem o vetor de entrada para todos os neurônios da camada oculta.
- **Camada oculta:** é responsável por fazer o mapeamento intermediário do problema, dividindo o problema em vetores linearmente separáveis que serão utilizados pela camada de saída. Essa camada pode ser replicada mais de uma vez, sendo que são comuns redes com duas ou três camadas ocultas.
- **Camada de saída:** utiliza neurônios que são responsáveis por rotular as classes ou fazerem o mapeamento desejado.

Em um MLP as informações são processadas em duas fases: na fase de propagação, os sinais de entrada do vetor $x(n)$, são propagados por todas as camadas da rede, gerando os valores intermediários $i(n)$ e as saídas da rede $y(n)$. Na fase de adaptação, são efetuados os ajustes dos pesos $w(n)$ da rede. Para isso é utilizado o erro gerado a partir da diferença entre os valores de saída da rede e os valores desejados. O fluxo das informações acontecem da camada de saída em direção à camada de entrada e os pesos de cada neurônio vão sendo ajustados segundo o algoritmo backpropagation.

Pode ser uma rede bastante utilizada em diferentes aplicações, a MLP foi a primeira rede a ser testada para predição de dados de contexto. Os resultados obtidos demonstraram uma baixa acurácia e alto custo computacional para treinar a rede, o que inviabilizou o seu uso. Isso motivou a busca por outras redes para tratar da predição dos dados.

Máquinas de Vetor de Suporte - SVM

As Máquinas de Vetor de Suporte (Support Vector Machines - SVMs) utilizam modelos lineares para separar classes não lineares através do mapeamento de um conjunto de treinamento de seu espaço original para um novo espaço de maior dimensão, chamado de espaço de características (??). Isso é possível, pois a medida que se aumenta o espaço da dimensão do problema, aumenta também a probabilidade desse problema se tornar linearmente separável em relação a um espaço de baixa dimensão. As SVMs são baseadas na Teoria de Aprendizado Estatístico (??), a qual visa encontrar condições matemáticas apropriadas encontrar um tipo especial de modelo linear chamado de hiperplano de margem máxima (*maximum margin hyperplane*). O hiperplano de margem máxima define a maior distância entre instâncias de classes diferentes. As instâncias de treinamento mais próximas desse valor são chamadas de vetores de suporte (*support vectors*), sendo que existe um vetor de suporte para cada classe.

Para casos linearmente separáveis, o hiperplano que separa as classes de acordo com três atributos pode ser descrito por

$$y = w_0 + w_1x_1 + w_2x_2 + w_3x_3$$

onde x_i são os atributos e w_i são os pesos que devem ser aprendidos pelo algoritmo de aprendizagem e determinam o hiperplano. O hiperplano de margem máxima pode ser representado em termos de vetores de suporte como

$$y = b + \sum \alpha_i y_i x(i) \cdot x,$$

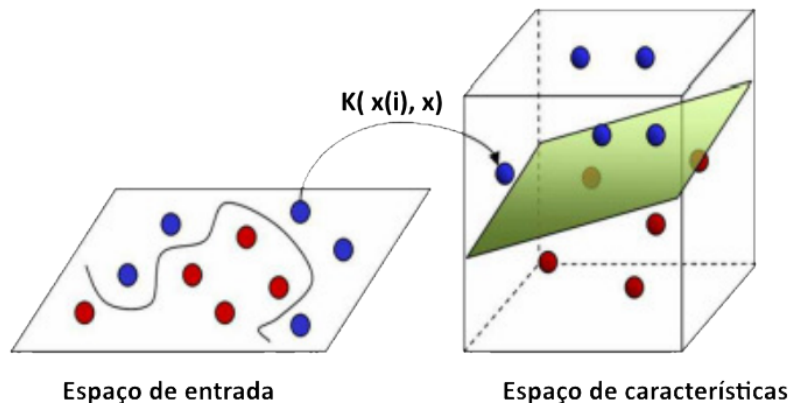
onde x corresponde a uma instância de teste, $x(i)$ corresponde aos vetores de suporte, b e α_i são os parâmetros que determinam o hiperplano.

Para casos não linearmente separáveis uma versão de alta dimensão da equação anterior pode ser representada por

$$y = b + \sum \alpha_i y_i K(x(i), x),$$

onde $K(x(i), x)$ é definido como uma *função de kernel*. As funções de kernel têm por finalidade projetar vetores de características de entrada em um espaço de características de alta dimensão. Esse processo está representado na figura A.2. Existem diferentes tipos de funções de kernel para construir máquinas de vetor de suporte com diferentes tipos de superfícies de decisão no espaço de entrada. Os exemplos mais comuns são o kernel polinomial $K(x, y) = (xy + 1)^d$ e a função Gaussiana $K(x, y) = \exp(-1/\delta^2(x - y)^2)$ onde d é o grau do polinômio e δ^2 é a largura da Gaussiana.

Figura A.2: Aplicação de uma função de kernel



Fonte: (??)

A atratividade de uso das SVMs é atribuída às seguintes características:

- Alta capacidade de generalização, evitando o sobre-treinamento (*overfitting*)
- Robustez para categorização de dados com dimensões altas, como por exemplo imagens. Em outros classificadores, a tendência é que exista sobre-treinamento pois muitas microcaracterísticas são pouco discriminantes.
- Convexidade da função objetivo pois esta é uma função quadrática com apenas um ótimo global.
- Teoria bem definida nas áreas de matemática e estatística.

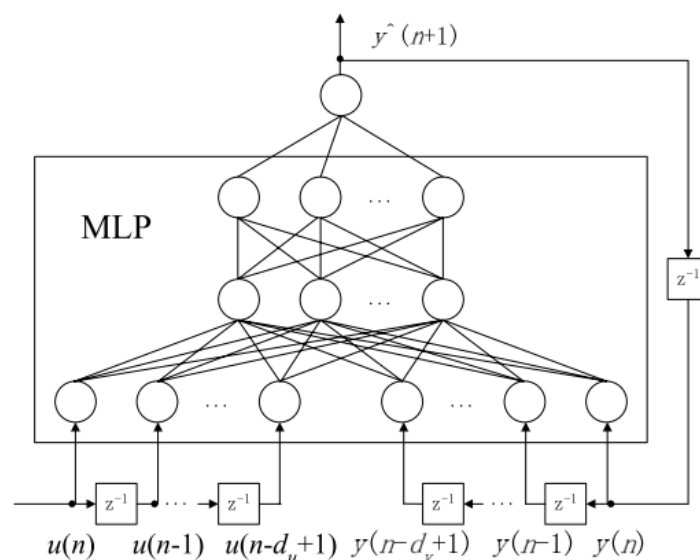
Inicialmente, as SVMs foram propostas para serem aplicadas em tarefas de classificação. Com o tempo demonstraram ser bons regressores, sendo consideradas aproximadores universais. Com relação a tarefa de predição as SVMs podem ser encontradas iniciativas que as aplicaram para a predição de séries temporais (??), predição do valor de ações (??), predizer crises financeiras (??), predição da escolha de passageiros em viagens de trens (??) e conforme apontado por ??) pode ser empregada na área de computação ubíqua como uma técnica base para a detecção de situações.

Modelo Autoregressivo não-linear com entrada exogena - NARX

O Nonlinear Autoregressive model process with eXogenous input (NARX) é um Rede Neural semelhante a uma MLP, porém a entrada da rede consiste na própria saída realimentada com atrasos de tempo e uma entrada exógena com atrasos.

Segundo ??) esse tipo de Rede Neural consiste principalmente de três camadas, entrada, oculta e de saída. A camada de entrada consiste da entrada corrente, anterior e saídas. Elas são ligadas a uma camada oculta que consiste de um ou mais neurônios resultando em um mapeamento não linear da combinação dos pesos da camada de entrada. A camada de saída é formada pela combinação dos valores da camada oculta.

Figura A.3: Arquitetura NARX



Fonte: (??)

O modelo é caracterizado por relações não lineares entre entradas passadas, saídas passadas e saídas preditas e pode ser representada por

$$y(t) = fy(t - 1), \dots y(t - n_y), u(t - 1), \dots u(t - n_u) + e(t)$$

onde $u(t)$ e $y(t)$ representam a entrada e saída do modelo no tempo t no qual a saída corrente $y(t) \in \mathfrak{R}$. As variáveis n_u e n_y representam a ordem de entrada e saída do modelo dinâmico onde $n_u \geq 0, n_y \geq 1$. A função f é uma função não-linear que pode ser determinada por uma MLP (??).

A arquitetura de uma NARX com d_u atrasos de entrada e d_y atrasos de saída é representada na figura A.3 (z^{-1} é a unidade de atraso).

Conforme apontado por ??????) um dos diferenciais da NARX é a sua capacidade atuar na predição de dados que possuem dependências temporais, como por exemplo séries temporais.

Como iniciativas que aplicam NARX nas predições, pode-se apontar o seu uso para predição de séries temporais univariadas(??) e séries temporais caóticas (??), predição de tráfego de rede em aplicações multimídias (??), para prever o comportamento de estruturas expostas a ondas marítimas, predição de enchentes (??), e também estimativa de terremotos (??).

Essas constatações do poder preditivo da NARX que motivaram o desenvolvimento de alguns testes para verificar seu uso na predição de dados de contexto no escopo da presente tese.

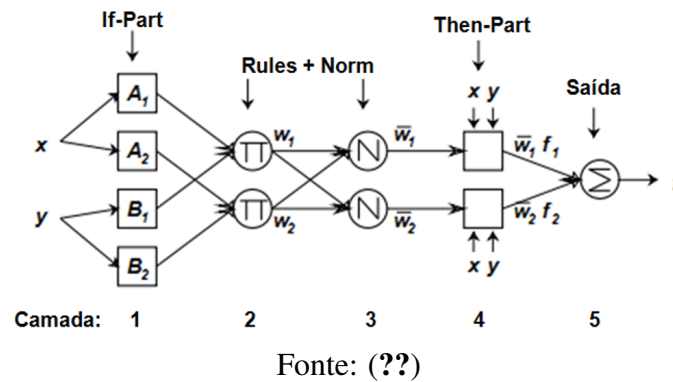
Sistemas de Inferência Neuro-Fuzzy Adaptativos - ANFIS e MANFIS

Um Sistemas de Inferência Neuro-Fuzzy Adaptativos, do inglês Adaptive Neuro-Fuzzy Inference Systems (ANFIS) (??), de acordo com ??), pode ser categorizado como uma abordagem de Inteligencia Computacional híbrida. Um ANFIS é um misto de um sistema fuzzy que é treinado por um algoritmo provido de uma rede Neural. Com isso pretende-se explorar a capacidade de predição e aprendizado de Redes Neurais com a representação e o gerenciamento do conhecimento da Lógica Fuzzy. Como resultados são obtidos sistemas com uma boa capacidade de inferência, mais transparentes no seu funcionamento e capazes de aprender.

A arquitetura de um ANFIS fazendo o uso do modelo Sugeno é apresentada na figura A.4.

Em um ANFIS os neurônios possuem uma estrutura diferente de uma NN normal, sendo que neurônios da mesma camada possuem funções similares. No exemplo da figura, na camada 1 a saída é composta pelo o grau de pertinência do padrão de entrada (x,y). Nesta camada calcula-se o grau de pertinência com que as entradas precisas satisfazem

Figura A.4: Arquitetura de um ANFIS implementando o modelo Sugeno



os termos linguísticos associados a estes nós. Cada nó dessa camada possui uma saída definida pela equação (A.1)

$$O_{1,i} = \mu_{A_i}(x), \quad \text{para } i = 1, 2 \quad \text{ou} \quad O_{1,i} = \mu_{B_{i-2}}(x), \quad \text{para } i = 3, 4 \quad (\text{A.1})$$

Sendo que μ_A e μ_B são definidas por uma função de pertinência fuzzy, como por exemplo, a função sino. Nesse caso os parâmetros da função de pertinência dessa camada (a, b e c no caso da função sino) são chamados de parâmetros de premissa. Tais parâmetros caracterizam como uma premissa é mapeada pela função de pertinência para um conjunto fuzzy.

Na camada 2 cada nó corresponde a uma regra, em tal camada calcula-se o potencial de ativação de cada regra, ou seja, com que grau o consequente da regra está sendo atendido. Os neurônios desta camada executam a operação de norma-t (geralmente produto) conforme a equação (A.2) fornecendo uma saída W_i . Essa saída representa o potencial de ativação da regra.

$$O_{2,i} = W_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \quad \text{para } i = 1, 2 \quad (\text{A.2})$$

Na camada 3 é realizada a normalização do potencial de ativação das regras. Cada nó calcula a razão entre o nível de disparo da regra W_i pela soma dos níveis de disparo de todas as regras. Esta saída será o potencial de ativação normalizado e é dado pela equação (A.3).

$$O_{3,i} = \bar{W}_i = \frac{W_i}{W_1 + W_2}, \quad \text{para } i = 1, 2 \quad (\text{A.3})$$

Já na camada 4 as saídas do neurônios são calculadas pelo produto entre os ní-

veis de disparo normalizados e o valor da regras do consequente da regra em si. Isso é representado pela equação (A.4).

$$O_{4,i} = \overline{W}_i f_i = W_i(p_i x + q_i y + r_i) \quad (\text{A.4})$$

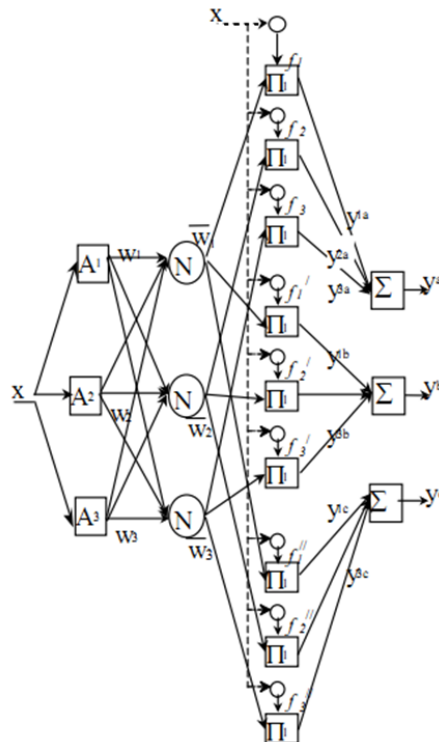
Sendo \overline{W}_i saída da camada 3 e $\{p_i, q_i \text{ e } r_i\}$ o conjunto de parâmetros chamados de parâmetros de consequente.

Por fim a camada 5 efetua o somatório de todos os sinais fornecendo a saída precisa. Sua equação é dada pela equação (A.5)

$$O_{5,i} = \sum \overline{W}_i f_i = \frac{\sum_i W_i f_i}{\sum_i W_i} \quad (\text{A.5})$$

Para o ajuste dos parâmetros de premissa e de consequente que melhor se adaptam aos dados que estão sendo manipulados é utilizado um algoritmo de aprendizado. Os dois algoritmos mais utilizados são o backpropagation ou então uma forma híbrida, a qual combina o backpropagation com o mínimos quadrados, para estimar os valores dos parâmetros de premissa e de consequente. O uso da técnica híbrida proporciona a convergência da rede de forma mais rápida uma vez que as dimensões do espaço de pesquisa é reduzido.

Figura A.5: Arquitetura de um MANFIS implementando o modelo Sugeno de primeira ordem com uma entrada, três regras e três saídas



Um importante detalhe é que um ANFIS fornece somente uma saída. Para fornecer múltiplas saídas, ??) propôs uma extensão do ANFIS chamado de Multioutput Adaptive Neural Fuzzy Inference System (MANFIS). Como o próprio nome diz, MANFIS funciona como vários ANFIS interconectados. Assim como no caso do ANFIS, o potencial do MANFIS está na sua habilidade de construir o mapeamento de entrada-saída usando conhecimento humano (na forma de regras fuzzy if-then) e no aprendizado dos dados fornecidos. Isso é possível, pois cada neurônio do MANFIS implementa um conjunto fuzzy.

A figura A.5 apresenta a arquitetura de um MANFIS.

O funcionamento do MANFIS é semelhante ao ANFIS, porém o seu treinamento precisa ser adaptado para lidar com múltiplas saídas. Mais especificamente, a alteração acontece no passo adiante do backpropagation que ao invés de usar somente um sinal de erro para correção dos pesos passou a usar a soma dos gradientes de erro de todas as saídas da camada anterior. O restante é idêntico.

Experimentos executados

Ao longo do desenvolvimento da tese essas Redes foram experimentadas com vistas a testar a sua capacidade preditiva para serem empregadas na projeção de situações. A seguir são apresentados os artigos publicados (eles também estão listados na seção 7.2) :

- ROLIM, C. O. ; Rossetto, Anubis G.M. ; Leithardt, Valderi R.Q. ; GEYER, C. F. R. Analysis of a hybrid neural network as underlying mechanism for a situation prediction engine. Journal of Applied Computing Research, v. 2, p. 22-31, 2012.
- ROLIM, C. O. ; SCHUBERT, F. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GEYER, C. F. R. ; WESTPHALL, C. B. . Comparison of a Multi output Adaptive Neuro-Fuzzy Inference System (MANFIS) and Multi Layer Perceptron (MLP) in Cloud Computing Provisioning. In: XXX Simposio Brasileiro de Redes de Computadores - X Workshop em Clouds, Grids e Aplicações WCGA, 2012, Ouro Preto - MG. Anais do X Workshop de Computação em Grade e Aplicações - WCGA, 2012.
- ROLIM, C. O. ; SCHUBERT, F. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GEYER, C. F. R. ; WESTPHALL, C. B. . Comparison of a Multi output Adaptive

- ROLIM, C. O. ; ROSSETO, A. G ; LEITHARDT, V. R. Q. ; GEYER, C. F. R. . Análise de uma Rede Neural Híbrida como base para um Mecanismo de Predição de Situação. In: XXXII Congresso da Sociedade Brasileira de Computação - CSBC, 2012, Curitiba. Anais do IV Simpósio Brasileiro de Computação Ubíqua e Pervasiva - SBCUP 2012, 2012.