

UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FACULDADE DE ENGENHARIA

CAMPUS DE ILHA SOLTEIRA

JULIANO MENDES RIBEIRO

**APLICAÇÃO DE REGRAS DE ASSOCIAÇÃO PARA MINERAÇÃO DE DADOS
EM UMA REDE *MESH* DE TRANSDUTORES INTELIGENTES BASEADA NOS
PADRÕES ZIGBEE E IEEE 1451.5**

Ilha Solteira

2020

JULIANO MENDES RIBEIRO

**APLICAÇÃO DE REGRAS DE ASSOCIAÇÃO PARA MINERAÇÃO DE DADOS
EM UMA REDE *MESH* DE TRANSDUTORES INTELIGENTES BASEADA NOS
PADRÕES ZIGBEE E IEEE 1451.5**

Dissertação apresentada a Faculdade de Engenharia – UNESP – Campus de Ilha Solteira-SP como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Elétrica.

Área de Conhecimento: Automação.

Orientador: Prof. Dr. Alexandre César Rodrigues da Silva

Coorientador: Prof. Dr. Tércio Alberto dos Santos Filho

Ilha Solteira

2020

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

R484a Ribeiro, Juliano Mendes.
Aplicação de regras associação para mineração de dados em uma rede mesh de transdutores inteligentes baseada nos padrões ZigBee e IEEE1451.5 / Juliano Mendes Ribeiro. -- Ilha Solteira: [s.n.], 2020
98 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2020

Orientador: Alexandre César Rodrigues da Silva

Coorientador: Tércio Alberto dos Santos Filho

Inclui bibliografia

1. Redes mesh. 2. Mineração de dados. 3. Regras de associação. 4. ZigBee. 5. IEEE1451.5.


Raiane da Silva Santos

Supervisora Técnica de Seção
Seção Técnica de Referência, Atendimento ao usuário e Documentação
Diretoria Técnica de Biblioteca e Documentação
CRB/8 - 9999

CERTIFICADO DE APROVAÇÃO

TÍTULO DA DISSERTAÇÃO: Aplicação de Regras de Associação para Mineração de Dados em uma Rede MESH de Transdutores Inteligentes Baseada nos Padrões ZIGBEE e IEEE 1451.5

AUTOR: JULIANO MENDES RIBEIRO

ORIENTADOR: ALEXANDRE CESAR RODRIGUES DA SILVA

Aprovado como parte das exigências para obtenção do Título de Mestre em ENGENHARIA ELÉTRICA, área: Automação pela Comissão Examinadora:

Prof. Dr. ALEXANDRE CESAR RODRIGUES DA SILVA

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira - UNESP

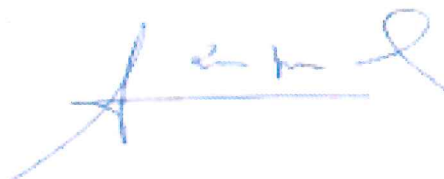
Suely Cunha Amaro Mantovani
Profª. Dra. SUELY CUNHA AMARO MANTOVANI

Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira - UNESP

Profª. Drª. NÚBIA ROSA DA SILVA

Núbia Rosa da Silva
Departamento de Computação - Instituto de Biotecnologia / Universidade Federal de Goiás - UFG

Ilha Solteira, 08 de setembro de 2020



AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, pela minha vida, e por me permitir ultrapassar todos os obstáculos encontrados ao longo da realização deste trabalho.

Obrigado Universidade Estadual Paulista “Júlio de Mesquita Filho” pela oportunidade de fazer o curso de pós-graduação em Engenharia Elétrica. Agradeço por me oferecer professores incríveis, um ambiente de estudo saudável e muitos estímulos para participar de atividades acadêmicas. Sou grato não só aos professores, mas também à direção, ao pessoal do administrativo, da limpeza e demais colaboradores da instituição.

Agradeço a minha mãe Nair Aparecida Mendes Ribeiro, heroína que me deu apoio e incentivo nas horas difíceis de desânimo e cansaço. Agradeço pelo sorriso diário sem mágoas nem rancores. Agradeço pelos meus dias de mau humor e desentendimento, em que me acalmou com suas orações em seu colo.

Ao meu pai Jacy Antônio Ribeiro, pessoa ímpar que diariamente me inspira a ser um homem melhor e mais generoso. Através dos teus exemplos tenho entendido que vale a pena ser uma pessoa honesta e que a dignidade é um dom, uma dádiva concedida àqueles que a buscam.

Ao professor Dr. Alexandre César Rodrigues da Silva, por ter sido meu orientador e ter desempenhado tal função com dedicação e amizade.

Ao professor Dr. Tércio Alberto dos Santos Filho pela oportunidade, paciência, dedicação e credibilidade depositada em mim ao longo desse processo de mestrado.

Ao professor Dr. Sérgio Francisco da Silva pelas correções e ensinamentos que me permitiram apresentar um melhor desempenho no meu processo de formação profissional ao longo do curso.

Agradeço a minha tia, mãe e amiga, Nilva Maria Mendes Correia, que quantas vezes me consolou e ajudou nos percalços da caminhada da vida.

Agradeço ao meu primo Rafael Mendes Correia por ter ajudado minha mãe no momento mais difícil de nossas vidas. Sempre nos encorajou, aconselhando e apoiando em todas as horas, com companheirismo e palavras de incentivo e esclarecimento.

Agradeço a minha namorada, amiga e companheira, Luana Costa de Oliveira, que tem me dado suporte, ajudando muito a encarar as dificuldades que a vida nos proporciona. Muito obrigado pela paciência e companheirismo. Te quero sempre ao meu lado.

Agradeço ao meu amigo Victor Portilho Pereira pela eterna amizade e companheirismo, pela colaboração, apoio e dedicação. Só Deus sabe o que enfrentamos nessa jornada de estudos.

Enfim, grande abraço a todos que de uma forma ou de outra contribuíram para o desenvolvimento desse estudo, que nos momentos de minha ausência dedicados ao estudo de pós-graduação, sempre fizeram entender que o futuro é feito a partir da constante dedicação no presente.

RESUMO

Neste trabalho desenvolveu-se uma rede *mesh* de transdutores inteligentes baseada nos padrões ZigBee e IEEE 1451.5. Após a implantação da rede, realizou-se um estudo sobre a aplicação da mineração de regras de associação nos dados da rede, para descobrir padrões entre as variáveis climáticas em um ambiente urbano. A mineração de regras de associação é um aprendizado de máquina baseado em regras, para descobrir relações entre as variáveis em grandes bases de dados. Na rede desenvolvida existem dois tipos de nós, de acordo com as diretrizes do padrão IEEE 1451.5, denominados de: *Network Capable Application Processor* (NCAP) e *Wireless Transducer Interface Module* (WTIM). Neste trabalho, o NCAP é executado de forma autônoma, realizando a leitura dos sensores conectados aos WTIMs de acordo com períodos de leituras pré-determinados. Após a aquisição dos dados dos sensores realiza-se o processamento e armazenamento no NCAP em memória flash local. Após um período de monitoramento, os dados da rede desenvolvida foram submetidos à mineração de regras de associação por meio do algoritmo Apriori. Constatou-se que esta técnica é capaz de descobrir padrões em dados climáticos com alto grau de complexidade. Logo, confirma-se a viabilidade da aplicação desta técnica em dados adquiridos em uma rede *mesh* de transdutores inteligentes.

Palavras-chave: Redes mesh. Mineração de dados. Regras de associação. ZigBee. IEEE 1451.5.

ABSTRACT

In this work, a mesh network of intelligent transducers based on the ZigBee and IEEE 1451.5 standards was developed. After the implantation of the network, a study was carried out on the application of mining association rules in the network data, to discover patterns between climatic variables in an urban environment. Association rules mining is a rule-based machine learning to discover relationships between variables in large databases. In the developed network there are two types of nodes, according to the guidelines of the IEEE 1451.5 standard, called: Network Capable Application Processor (NCAP) and Wireless Transducer Interface Module (WTIM). In this work, the NCAP is executed autonomously, reading the sensors connected to the WTIMs according to predetermined reading periods. After the acquisition of sensor data, NCAP is processed and stored in local flash memory. After a period of monitoring, the data from the developed network were subjected to mining association rules using the Apriori algorithm. It was found that this technique is able to discover patterns in climatic data with a high degree of complexity. Therefore, the feasibility of applying this technique to data acquired in a mesh network of intelligent transducers is confirmed.

Keywords: Mesh networks. Data mining. Association rules. ZigBee. IEEE 1451.5.

LISTA DE FIGURAS

Figura 1 – Etapas do processo KDD em banco de dados (KDD).	6
Figura 2 – Família de padrões IEEE 802.	16
Figura 3 – Arquitetura IEEE 802.14.5 RM.	17
Figura 4 – Topologias de rede IEEE 802.15.4.	19
Figura 5 – Relacionamento entre as funções dos dispositivos IEEE 802.15.4 e ZigBee.	21
Figura 6 – Topologia estrela para a rede ZigBee.	21
Figura 7 – Topologia <i>mesh</i> para a rede ZigBee.	22
Figura 8 – Topologia <i>Tree</i> para rede ZigBee.	23
Figura 9 – Arquitetura do interfaceamento dos nós NCAP e WTIM.	24
Figura 10 – Mensagem de comando de requisição de leitura de um sensor.	28
Figura 11 – Mensagem de comando de resposta da leitura de um sensor.	28
Figura 12 – Estrutura genérica do <i>frame</i> API “Comando AT”.	30
Figura 13 – <i>Frame</i> API Comando AT com a função <i>Node Discovery</i> (ND).	30
Figura 14 – Estrutura do <i>frame</i> de requisição.	31
Figura 15 – Estrutura do <i>frame</i> de resposta.	32
Figura 16 – Exemplo de aplicação do <i>frame</i> de pedido de transmissão.	32
Figura 17 – Exemplo de aplicação do <i>frame</i> de resposta.	33
Figura 18 – Se o <i>varset</i> {c, d, e} for frequente, então seus subconjuntos são considerados frequentes.	38
Figura 19 – Se o <i>varset</i> {a, b} “não” for frequente, então seus superconjuntos não são frequentes.	39
Figura 20 – Gerando um <i>varset</i> candidato.	40
Figura 21 – Geração dos <i>varsets</i> candidatos C'_2 .	46
Figura 22 – Representação grafica do <i>boxplot</i> , com escala de dados horizontal.	54
Figura 23 – Representação grafica do <i>boxplot</i> com <i>outliers</i> .	55
Figura 24 – <i>Boxplot</i> pré-formado a partir dos quartis calculados.	57
Figura 25 – <i>Boxplot</i> dos exemplos apresentados.	58
Figura 26 – ASUS Tinker Board.	61
Figura 27 – Transceptor XBee Pro S2C utilizado neste projeto.	62
Figura 28 – Prototipagem e esquema elétrico do NCAP.	63

Figura 29 – Fluxograma da aplicação NCAP <i>Request</i> .	64
Figura 30 – Exemplo: Informações dos WTIMs armazenadas no NCAP.	64
Figura 31 – Fluxograma da aplicação NCAP <i>Monitor</i> .	65
Figura 32 – Divisor de tensão.	66
Figura 33 – Regulador de tensão CI AMS1117 H433KE (5V de saída).	67
Figura 34 – WTIM desenvolvido em laboratório	67
Figura 35 – Sensor BME280: temperatura, umidade e pressão barométrica.	68
Figura 36 – Fluxograma da execução do <i>software</i> dos módulos WTIMs.	69
Figura 37 – Topologia atual da rede desenvolvida.	70
Figura 38 – Fases de instalação do NCAP.	71
Figura 39 – Relevô entre o NCAP e WTIM-1.	72
Figura 40 – Instalação do WTIM-1.	72
Figura 41 – Roteador ZigBee instalado no Campus 2 da UFCat.	73
Figura 42 – Relevô entre o NCAP e Roteador ZigBee.	73
Figura 43 – Relevô entre o Roteador ZigBee e WTIM-1.	74
Figura 44 – Relevô entre o Roteador ZigBee e WTIM-2.	74
Figura 45 – Características de instalação do WTIM-2.	75
Figura 46 – Relevô entre o NCAP e WTIM-2.	76
Figura 47 – Estrutura de arquivos do NCAP para o armazenamento dos dados.	77
Figura 48 – <i>Boxplot</i> da base de dados discretizada.	82
Figura 49 – Regras com grau de suporte mais elevado.	83
Figura 50 – Regras geradas com β de 1% e α entre 90% e 100%.	86
Figura 51 – Número de regras R geradas em função do grau mínimo de suporte β .	86

LISTA DE TABELAS

Tabela 1 – Ordem de transmissão dos dados.	25
Tabela 2 – Ordem de transmissão dos bits de um <i>byte</i> .	25
Tabela 3 – Estrutura de uma mensagem de comando.	26
Tabela 4 – Classes de comandos.	26
Tabela 5 – Mensagem de comando de resposta.	27
Tabela 6 – Funções de operação.	27
Tabela 7 – Exemplos de tipos de <i>frames</i> API.	29
Tabela 8 – Base de dados com a média das leituras realizadas.	35
Tabela 9 – Suportes de alguns <i>varsets</i> .	35
Tabela 10 – Gerando as regras de um <i>varset</i> frequente.	43
Tabela 11 – Calculando o grau de confiança de cada regra.	43
Tabela 12 – Banco de dados de observações.	44
Tabela 13 – O conjunto de <i>varsets</i> iniciais contém as variações das variáveis meteorológicas.	45
Tabela 14 – Suporte dos <i>varsets</i> de tamanho 1.	45
Tabela 15 – Poda dos <i>varsets</i> candidatos C'_2 .	46
Tabela 16 – Suporte dos <i>varsets</i> candidatos C_2 .	47
Tabela 17 – Geração dos <i>varsets</i> candidatos C'_3 .	47
Tabela 18 – Poda dos <i>varsets</i> candidatos C'_3 .	48
Tabela 19 – Suporte dos <i>varsets</i> candidatos C_3 .	48
Tabela 20 – Conjuntos de <i>varsets</i> frequentes obtidos nas iterações do algoritmo Apriori.	49
Tabela 21 – Gerando as possíveis regras de associação dos <i>varsets</i> frequentes de L_2 .	49
Tabela 22 – Calculando o grau de confiança de cada regra de L_2 .	50
Tabela 23 – Gerando as possíveis regras de associação dos <i>varsets</i> frequentes de L_3 .	50
Tabela 24 – Calculando o grau de confiança de cada regra de L_3 .	50
Tabela 25 – Regras de associação interessantes obtidas.	51
Tabela 26 – Exemplo de um conjunto de dados ordenado, para a aplicação do <i>boxplot</i> .	55

Tabela 27 – Divisão do conjunto de dados em partes iguais.	55
Tabela 28 – Determinando a mediada Q_2 em um conjunto de dados com um número ímpar de observações.	56
Tabela 29 – Metade superior e inferior do conjunto apresentado na Tabela 27.	56
Tabela 30 – Metade superior e inferior do conjunto apresentado na Tabela 28.	57
Tabela 31 – Arquitetura baseada em camadas.	61
Tabela 32 – Amostra da base de dados obtida pelos WTIMs da rede.	79
Tabela 33 – Discretização dos elementos meteorológicos	80
Tabela 34 – Base de dados discretizada	81
Tabela 35 – Resultado da aplicação do <i>boxplot</i> no banco de dados dos WTIMs	82
Tabela 36 – Número de regras R geradas em função do grau mínimo de suporte β e confiança α através da base de dados do WTIM-1.	85
Tabela 37 – Número de regras R geradas em função do grau mínimo de suporte β e confiança α através da base de dados do WTIM-2.	85

LISTA DE ABREVIATURAS E SIGLAS

ADC	Analog-to-Digital Converter
API	Application Programming Interface
BRB	Belief Rule Base
CI	Circito Integrado
CMU	Carnegie Mellon University
DAC	Digital-to-Analog Converter
DARPA	Defense Advanced Research Projects Agency
Dot5AR	An IEEE 1451.5 Approved Radio
DSN	Distributed Network Sensor
ED	Energy Detection
FFD	Full Function Device
FPGA	Field Programmable Gate Arrays
GPIO	General Purpose Input/Output
GPU	Graphics Processing Unit
GTSs	Guaranteed Time Slots
I ² C	Inter-Integrated Circuit
IBM	International Business Machines Corporation
IDO	Identificador da Observação
IEEE	Institute of Electrical and Electronics Engineers
INMET	Instituto Nacioal de Meteorologia
KDD	Knowledge Discovery in Databases

LGA	Land Grid Array
LOMA	Local Outlier Mining Algorithm
LQI	Link quality indication
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Medium Access Control
MIPS	Million Instructions Per Second
MIT	Massachusetts Institute of Technology
NCAP	Network Capable Application Processor
ND	Node Discovery
NIST	National Institute of Standards and Technology
OSI	Open System Interconnection
PAN ID	PAN Identifier
PCM/I2S	Pulse-code modulation/Inter-IC Sound
PHP	PHP: Hypertext Preprocessor
PLC	Programmable Logic Controllers
PSO	Particle Swarm Optimization
PWM	Pulse Width Modulation
RAM	Random Access Memory
RDF	Resource Description Framework
RF	Radiofrequência
RFD	Reduced Function Device
FID	Radio-Frequency Identification
RISC	Reduced Instruction Set Computer
SOSUS	Sound Surveillance System

SPARQL	Protocol and RDF Query Language
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SSH	Secure Shell
SSW	Semantic Sensor Web
SVM	Support Vector Machine
TEDSs	Transducer Electronic Data Sheets
UART	Universal Asynchrounous Receiver/Transmitter
UFCat	Universidade Federal de Catalão
UTC	Coordinated Universal Time
WEB	World Wide Web
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
WTIM	Wireless Transducer Interface Module

SUMÁRIO

1	INTRODUÇÃO	5
1.1	JUSTIFICATIVA	7
1.2	OBJETIVO DO TRABALHO	8
1.3	TRABALHOS RELACIONADOS	8
1.4	ORGANIZAÇÃO DO TEXTO	12
2	REDE MESH DE TRANSDUTORES INTELIGENTES	14
2.1	DEFINIÇÃO	14
2.2	REDES DE SENSORES SEM FIO	15
2.3	O PADRÃO IEEE 802.15.4	16
2.3.1	Topologias de rede IEEE 802.15.4	18
2.4	O PADRÃO ZIGBEE	20
2.5	O PADRÃO IEEE 1451.5	23
2.5.1	Estrutura das mensagens	25
2.5.2	Mensagem de comando de requisição e resposta	27
2.5.3	Unidade de transferência de dados ZigBee/IEEE 1451.5	28
2.6	CONSIDERAÇÕES SOBRE O CAPÍTULO	33
3	MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO	34
3.1	DEFINIÇÃO DE REGRAS DE ASSOCIAÇÃO	34
3.1.1	Formalização do problema	36
3.1.2	O algoritmo Apriori	37
3.1.2.1	<i>O princípio do algoritmo Apriori</i>	38
3.1.2.2	<i>A fase da geração de candidatos</i>	39
3.1.2.3	<i>A fase da poda</i>	40
3.1.2.4	<i>A fase de cálculo do suporte</i>	41
3.1.2.5	<i>O algoritmo Apriori: A união das três fases</i>	41

3.2	GERANDO AS REGRAS DE ASSOCIAÇÃO	42
3.3	EXEMPLO PRÁTICO DE CRIAÇÃO DE REGRAS DE ASSOCIAÇÃO	44
3.4	CONSIDERAÇÕES SOBRE O CAPÍTULO	51
4	PRÉ-PROCESSAMENTO DOS DADOS	52
4.1	PROCESSO DE DISCRETIZAÇÃO	52
4.2	IDENTIFICANDO <i>OUTLIERS</i> EM UMA BASE DE DADOS	53
4.2.1	O cálculo dos quartis	54
4.2.2	Determinando os <i>outliers</i> da base de dados	57
4.3	CONSIDERAÇÕES SOBRE O CAPÍTULO	59
5	REDE DE TRANSDUTORES DESENVOLVIDA	60
5.1	O DISPOSITIVO NCAP	60
5.1.1	O <i>hardware</i> e <i>interfaces</i>	61
5.1.2	O <i>software</i>	63
5.2	OS DISPOSITIVOS WTIMS	65
5.2.1	Hardware e sensores acoplados	66
5.2.2	Software	68
5.3	APLICAÇÃO DA REDE DE SENSORES EM LOCAL URBANO	69
5.4	COLETA DOS DADOS A SEREM ANALISADOS	76
5.5	CONSIDERAÇÕES SOBRE O CAPÍTULO	77
6	APLICAÇÃO DE MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO PARA A BASE DA REDE <i>MESH</i> DE TRANSDUTORES INTELIGENTES DESENVOLVIDA	78
6.1	A BASE DE DADOS OBTIDA PELOS WTIMS	78
6.2	REALIZANDO O PRÉ-PROCESSAMENTO DA BASE DE DADOS	79
6.2.1	<i>Boxplot</i> : Identificando <i>outliers</i> da base de dados discretizada	81
6.3	EXECUÇÃO DO ALGORITMO APRIORI SOBRE A BASE DE DADOS DISCRETIZADA	83
6.4	CONSIDERAÇÕES SOBRE O CAPÍTULO	87
7	CONCLUSÕES	88

REFERÊNCIAS	90
APÊNDICE A - SUPORTES DE UMA OBSERVAÇÃO	95
APÊNDICE B - ESQUEMÁTICO DO CIRCUITO ELÉTRICO DOS MÓDULOS WTIMS	96
APÊNDICE C - ALGORITMO DE DISCRETIZAÇÃO DA BASE DE DADOS DA REDE <i>MESH</i>.	97
APÊNDICE D - ARTIGOS PULICADOS	98

1 INTRODUÇÃO

As redes de transdutores inteligentes, são constituídas de nós interligados entre si através de uma interface de comunicação. Estas redes são essencialmente sistemas distribuídos, que têm como objetivo supervisionar e monitorar um determinado processo. Tais redes são desenvolvidas, utilizando diversos tipos de tecnologia e padrões de comunicação, sendo que, independente da tecnologia ou do padrão de comunicação, as redes de transdutores devem ser confiáveis, robustas e rápidas.

Quando os transdutores (sensores e atuadores) são aplicados em conjunto com um dispositivo embarcado e um transmissor sem fio, são denominados de nós de rede, que tem como objetivo realizar o controle ou aquisição dos dados no ambiente de aplicação.

Os nós intermediários são denominados de roteadores, no qual o objetivo é receber os dados do coordenador/nó adjacente e reencaminhar para outros nós da rede. Desta forma é possível desenvolver uma rede com vários nós interligados formando uma “malha”, denominando-a de rede *mesh*.

Para criar uma rede *mesh* faz-se necessário seguir algumas diretrizes e ferramentas que são providas por algum padrão de rede. Sendo assim, os padrões de rede *mesh* mais difundidos e utilizados, são: IEEE 802.15.5, ZigBee, Thread, 6LoWPAN, WirelessHART e ISA100.11a. Esses padrões de rede são baseados no padrão IEEE 802.15.4 (*WPAN Low Rate*) que trabalha com baixa taxa de dados, baixo consumo de energia, complexidade baixa e opera na camada física e enlace do modelo *Open System Interconnection* (OSI)¹.

A detecção dos valores dos elementos climáticos de um local por meio de uma rede de sensores sem fio, pode fornecer informações valiosas para o monitoramento da qualidade do ar, e por consequência, da existência de substâncias e gases que causam alterações adversas no meio ambiente, ou seja, poluição atmosférica. A poluição atmosférica afeta a saúde humana, dos animais e plantas, mudando todo o ambiente, causando mudanças

¹Framework concebido pela *International Organization for Standardization* (ISO) para a definição de padrões, visando a interconexão de sistemas heterogêneos (independência de fabricante, sistema operacional e plataforma de hardware) (TANENBAUM, 2011).

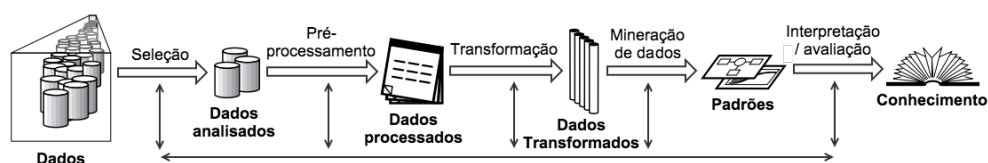
climáticas e o aquecimento global.

Dessa forma, realizar o monitoramento do microclima de uma determinada região utilizando as redes de sensores sem fio, auxilia no estudo das mudanças climáticas, verificando as variações climáticas causadas pela poluição do homem.

Para analisar uma grande quantidade de dados climáticos de longos períodos de monitoramento, pode-se utilizar a mineração de dados. A mineração de dados é uma etapa essencial na área da descoberta de conhecimento em banco de dados de larga escala, também denominada de *Knowledge Discovery in Databases* (KDD). A KDD é o processo que encontra correlações ou padrões de acordo com diferentes perspectivas, em uma grande quantidade de dados. A KDD permite a análise de uma ampla quantidade de dados, categorizando-os e resumindo as relações identificadas. As informações extraídas desse processo podem ser utilizadas para entender algum fenômeno natural, no estudo do diagnóstico de doenças, aumentar a receita e reduzir custos em uma determinada empresa, etc (MITRA; PAL; MITRA, 2002; FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

De acordo com Fayyad, Piatetsky-Shapiro e Smyth (1996), que estão entre os pioneiros na área da KDD, a mineração de dados é uma etapa particular no processo KDD, envolvendo a aplicação de algoritmos específicos para extração de padrões (modelos) a partir dos dados. As etapas do processo KDD, ilustradas na Figura 1, consistem em uma sequência iterativa de seleção, pré-processamento, transformação, mineração de dados e interpretação.

Figura 1 – Etapas do processo KDD em banco de dados (KDD).



Fonte: Fayyad, Piatetsky-Shapiro e Smyth (1996).

Na etapa de *seleção* ocorre a separação dos dados relevantes para a aplicação, usando uma ampla base de dados. Após essa etapa os dados passam pela etapa de *pré-processamento*, na qual ocorre a remoção de ruídos, que são informações incompletas no conjunto de dados. Em seguida ocorre o processo de *transformação*, em que se realiza a redução de dados e a projeção. Nela advém a escolha de recursos úteis para representar os dados, dependendo do objetivo da tarefa, reduz-se o número de variáveis sob considerações ou encontra-se representações invariáveis para os dados.

Após a etapa de transformação, é executada a fase considerada mais importante do

processo KDD, a *mineração de dados*. Nessa fase, ocorre a busca de padrões de interesse em uma forma de representação específica ou em um conjunto de representações, incluindo: regras de associação, classificação, previsão, regressão, clusterização, etc.

Por fim, a etapa de *interpretação* é executada, analisando os padrões descobertos, selecionando as informações relevantes em termos compreensíveis pelos usuários (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Neste trabalho apresenta-se uma rede de sensores do tipo *mesh* desenvolvida em ambiente urbano, capaz de realizar a leitura dos dados climáticos de forma periódica. A rede desenvolvida é baseada no padrão de rede ZigBee e nos padrões de interfaceamento IEEE 1451.1 (NCAP) e IEEE 1451.5 (WTIM).

O NCAP, nó que coordena os nós adjacentes (WTIM's) e os dados que são propagados na rede, é responsável por coletar os dados da rede externa, processá-los e encaminha-los para uma rede interna.

O WTIM é um módulo que possui uma unidade de processamento, transdutores, uma unidade de memória não-volátil (para o armazenamento das TEDS) e um transmissor sem fio para realizar a comunicação com o nó de rede NCAP ou com um roteador adjacente.

Com uma grande quantidade de dados coletados durante um longo período de tempo e com o auxílio da mineração de dados, é possível obter conhecimento das mudanças climáticas da região e também gerar previsões futuras.

1.1 JUSTIFICATIVA

Os avanços dos dispositivos microeletrônicos e da tecnologia de transmissão de dados sem fio, levaram ao desenvolvimento de transdutores de baixa potência e a implantação de redes de transdutores inteligentes em grande escala. Com a capacidade de monitoramento difundida, as redes de transdutores inteligentes têm atraído atenção significativa em muitos domínios de aplicação como, por exemplo, na saúde, na análise climática, nas casas inteligentes, na agricultura de precisão, etc. Essas aplicações geram um enorme volume de dados, geograficamente distribuídos e heterogêneos.

Os dados brutos, ao serem analisados e transformados em informações através da mineração de dados, facilitam a tomada de decisão e as estratégias induzidas pelo homem. Portanto, é essencial desenvolver técnicas para extrair padrões nos dados de uma rede de transdutores inteligentes, a fim de tomar decisões inteligentes.

Aplicar as técnicas de mineração de dados em uma rede de transdutores inteligentes utilizada para a coleta de dados meteorológicos facilita o entendimento das alterações do microclima da região em que a rede está sendo aplicada e desta forma, prevê possíveis mudanças climáticas na região.

O desenvolvimento da rede de sensores sem fio em uma determinada área, facilita a aquisição dos dados e a análise por especialistas, como por exemplo, no trabalho de SANTOS (2020), no qual foram utilizados os dados das estações meteorológicas desenvolvidos neste trabalho para análise do campo higrométrico, os quais referem-se aos corpos sensíveis à variação do grau de humidade ambiente.

1.2 OBJETIVO DO TRABALHO

O objetivo deste trabalho é apresentar uma rede do tipo *mesh* baseada no padrão de rede ZigBee e no padrão de interfaceamento IEEE 1451.5. A rede foi aplicada no ambiente urbano para realizar a coleta de dados ambientais durante longos períodos de tempo. A rede baseada no padrão ZigBee facilita a inserção de novos nós para a coleta de dados tornando-a expansível e de fácil manutenção.

Ao coletar dados durante um longo período de tempo foram utilizadas técnicas de mineração de dados para realizar as análises, com o objetivo de estudar as mudanças climáticas ocorridas durante este período.

1.3 TRABALHOS RELACIONADOS

Na área de mineração de dados aplicada às *Wireless Sensor Networks* (WSNs), ressalta-se o trabalho de Sunitha e Chandrika (2016). Trata-se de um estudo de diferentes métodos de mineração de dados e *soft computing*² para serem implantadas em WSNs, apresentando soluções para vários problemas da área. As WSNs cobrem uma grande área de comunicação, por isso, manter a otimização do processamento e armazenamento de dados se torna uma tarefa bastante desafiadora. Além desse obstáculo, os nós de rede enfrentam o problema do consumo de energia, que é limitada, na detecção de eventos anormais e até mesmo na possível inserção de nós intrusos na rede.

Em Kumar e Chaurasiya (2019) foi apresentada uma abordagem de mineração de

²*Soft computing* lida com modelos aproximados e fornece soluções para problemas complexos da vida real. Diferentemente da computação pesada, a *soft computing* é tolerante à imprecisão, incerteza e aproximações (IBRAHIM, 2016).

dados chamada de *Novel Data Mining* (NDM) para diminuir a redundância nos dados transmitidos em uma WSN. Geralmente há uma grande quantidade de dados detectados pelos nós de uma rede de sensores e alguns são redundantes. Essa redundância prejudica o desempenho da rede, criando alguns deméritos em termos de sobrecarga de computação e transmissão excessiva. O desempenho de um WSN usando a estratégia NDM foi avaliado através de várias métricas de desempenho como, por exemplo, atraso da comunicação ponta-a-ponta, vida útil da rede, eficiência energética, taxa de transferência e redundância de pacotes. Os resultados da simulação mostraram que o esquema proposto superou outras abordagens propostas.

Os dados de um sensor em um nó de uma WSN podem conter valores discrepantes que tendem a ocorrer por acaso, por falha de medição ou por ataques. Essa ocorrência de ruído nos dados, ou *outliers*, pode afetar a confiabilidade do processo em que o nó está encarregado a monitorar. No trabalho de Shenoy e Ameer (2019) são descritas várias técnicas de detecção de *outliers* usando algoritmos de mineração de dados. Das três metodologias adotadas, o *Support Vector Machine* (SVM) tende a fornecer as maiores taxas de detecção (BRAUN; WEIDNER; HINZ, 2011). O método de agrupamento *k-means* fornece boas taxas de detecção e taxas de alarmes falsos quando o número de *clusters* aumenta e a taxa de compactação diminui (ESTEVES; HACKER; RONG, 2013). O *Local Outlier Mining Algorithm* (LOMA) é usado para conjuntos de dados de alta dimensão, o que ajuda a diminuir a complexidade computacional com melhor precisão de detecção (ZHAO; ZHANG; QIN, 2017). Cabe salientar que o LOMA oferece o melhor desempenho quando o limiar esparsa é baixo.

Em Raut e Khandait (2015) apresentaram algumas técnicas de mineração de dados adotadas para as WSNs, examinando-as com diversas classificações e abordagens de avaliação. Apresentam diferentes técnicas usadas na mineração de dados e algoritmos desenvolvidos, o que pode ajudar a construir aplicativos inteligentes para as WSNs. No entanto, observa-se que as técnicas de mineração de dados usadas para abordagens centralizadas e descentralizadas, concentram-se principalmente na extração e análise de informações a partir dos dados das WSNs. Além disso, existem alguns problemas, considerando as limitações das WSNs, como energia limitada, armazenamento, capacidade da CPU, largura de banda de comunicação, etc. Portanto, os autores concluíram que é fundamental atualizar o desempenho de futuras WSNs usando técnicas adequadas de mineração de dados em ambiente centralizado e distribuído.

Alfian et al. (2017) propuseram um sistema eletrônico de rastreabilidade de alimentos, utilizando a tecnologia *Radio-Frequency Identification* (RFID) para rastrear e localizar

produtos, e uma rede de sensores sem fio para coletar temperatura e umidade durante o armazenamento e o transporte. Neste sistema, os dados dos sensores podem ser perdidos ou corrompidos por vários motivos. O sistema proposto utiliza técnicas de mineração de dados para prever informações ausentes do sensor de algum nó da rede. O sistema foi testado em uma indústria de alimentos da Coreia, e mostrou benefícios significativos para gerentes e clientes, fornecendo localização em tempo real e histórico completo de temperatura e umidade. Os autores utilizaram uma técnica de mineração de dados baseada em Redes Neurais Multicamada, que forneceu a melhor precisão de previsão para a falta de dados do sensor em comparação com outros modelos.

Em Wei (2018) apresentaram um método de previsão para tendências oceanográficas. Um esquema de monitoramento on-line foi preparado para coletar dados hidrológicos oceanográficos precisos com base em uma WSN. Em seguida, os dados coletados pela WSN foram processados por um algoritmo de mineração de dados por regressão. Para apoiar o algoritmo de regressão, utilizou-se o algoritmo *Particle Swarm Optimization* (PSO) para procurar a solução ótima global (ANANTATHANAVIT; MUNLIN, 2013). Depois disso, foi criado um sistema de observação e coleta de dados oceanográficos com base na situação hidrológica do porto de Nova York. Em seguida, a regressão tradicional e o método proposto foram aplicados para prever as tendências oceanográficas com base na temperatura da água, salinidade e outros índices. Os resultados mostraram que o algoritmo proposto aprimorou a taxa de utilização dos dados da WSN e obteve boa precisão de previsão. A pesquisa fornece informações importantes sobre a aplicação de tecnologia avançada nas previsões oceanográficas.

Em Harb, Makhoul e Jaoude (2018) foi proposta uma abordagem de redução de dados em dois níveis para redes de sensores. O primeiro nível é operado por nós que contêm sensores e que consiste em compactar os dados coletados ao usar o coeficiente de *Pearson*. O segundo nível é executado em nós intermediários (por exemplo, agregadores, *cluster-head* e assim por diante). O objetivo do segundo nível é eliminar dados redundantes gerados pelos nós vizinhos usando dois métodos de *cluster*, são eles: *EKmeans* e *TopK* (SURESH; ARULANANDAM, 2017; DAS, 2009). Através de simulações e experimentos reais em sensores, evidenciou-se a relevância da abordagem dos autores em termos de minimizar os dados coletados nas WSNs e melhorar a vida útil da rede, em comparação com outras técnicas existentes.

He et al. (2018) propôs um método para corrigir e detectar falhas nos sensores dos nós de uma WSN com o objetivo de melhorar a precisão dos dados. No trabalho, as características dos dados da WSN são analisadas e um método é proposto para o

diagnóstico de falhas baseado no modelo *Belief Rule Base* (BRB) (ALAM et al., 2018). Inicia-se o processo pela descrição do diagnóstico de falhas do sensor com base nas características de um sensor da WSN. Em seguida, as características dos sensores são analisadas a partir dos aspectos de tempo, espaço e atributos. Finalmente, um modelo de diagnóstico de falhas é proposto com base no modelo BRB. Um estudo de caso usando o conjunto de sensores de laboratório da Intel foi desenvolvido para verificar a eficácia do modelo e foi verificado pelos resultados que o método é eficaz no diagnóstico de falhas nas WSNs.

Bhavsar e Arolkar (2014) apresentaram uma técnica de mineração de dados utilizando Regras de Associação Multidimensional para um sistema de monitoramento de saúde do gado, baseado em WSNs. Os autores também expuseram uma visão geral do conceito de mineração de dados, apresentando alguns métodos de mineração embasados em dados de WSNs. Foram discutidos diferentes tipos de doenças e seus sintomas encontrados no gado. Finalmente, os autores propuseram uma técnica de mineração baseada em regras de associação para identificar a doença com base no sintoma.

Em Boukerche e Samarah (2007), os autores apresentaram uma formulação de regras de associação, sendo capaz de gerar informações relacionadas a correlação dos nós de rede de uma WSN, com o objetivo de maximizar o tempo de vida da rede.

Sabit, Al-Anbuky e Gholam-Hosseini (2009) também desenvolveram um trabalho com o objetivo de maximizar o tempo de vida de uma WSN. Eles propuseram um algoritmo clusterização baseado em métodos de mineração de dados e lógica *fuzzy*.

As características mais atraentes das WSNs são: a coleta sistemática de dados e a transmissão das informações coletadas para uma estação base distante. O parâmetro de segurança pode ser interessante se os dados forem transferidos em áreas sensíveis como Defesa, Saúde, Bancos, etc. No entanto, a sensibilidade das WSNs torna-as propensas a ataques que levam à extração, ou até dano nos dados que flui entre nós distintos.

Sendo assim, Singh et al. (2012) construíram um sistema de segurança que é adaptável às mudanças que levam à extração ou dano dos dados que flui em uma WSN. Para detectar nós maliciosos na rede, com base em seus comportamentos, os autores utilizaram algoritmos de mineração de dados, com base em métodos de classificação e clusterização.

Com intuito de melhorar a gestão dos aspectos da agricultura de precisão, o trabalho de Tripathy et al. (2011) apresenta métodos de mineração de dados para transformar os dados de uma WSN em informações (conhecimento, relações e tendências) úteis, com intuito de descobrir a correlação entre o clima e pragas existentes na cultura de

amendoim. A WSN desenvolvida pelos autores realiza leituras de diferentes parâmetros meteorológicos, ao nível micro-climático, em tempo real, como: temperatura e umidade relativa do ar. Dessa forma, a dinâmica das pragas e suas relações ocultas com o clima, foram obtidas e quantificadas utilizando métodos de mineração de dados.

De acordo com o avanço da ciência médica, foi constatado que Taiwan entrou no envelhecimento da sociedade. Devido a isso, há uma demanda maior e gradativa por cadeira de rodas. Portanto, Wu et al. (2011) desenvolveram um sistema para aumentar a segurança da cadeira de rodas. Eles aplicaram sensores em uma cadeira de rodas para medir a posição e a operação dos usuários. Se a postura sentada do usuário for ruim ou se o usuário tiver algum perigo, o sistema mostrará uma mensagem de aviso para informar ao cuidador. De acordo com a posição mostrada, o cuidador poderá ajudar o usuário da cadeira de rodas diretamente. Finalmente, um sistema de mineração de dados foi projetado e implementado para analisar as operações dos usuários de cadeira de rodas. Pode-se descobrir a postura sentada incorreta ou inadequada do usuário da cadeira de rodas por longos períodos.

1.4 ORGANIZAÇÃO DO TEXTO

Incluída a introdução geral no Capítulo 1 que contém a contextualização do trabalho, um breve histórico das redes de sensores sem fio, um resumo sobre a mineração de dados, e os trabalhos mais relevantes na área, o texto foi organizado em 7 capítulos.

No Capítulo 2 é apresentado as redes *mesh* de transdutores inteligentes, enfocando nos padrões de rede utilizados neste trabalho, traçando de forma sucinta suas características e relacionamentos.

No Capítulo 3 são introduzidos os principais conceitos de mineração de regras de associação, apresentando a formalização do problema e os algoritmos que o compõe. É apresentado também um estudo prático e sucinto do emprego da mineração de regras de associação no âmbito meteorológico.

No Capítulo 4 é apresentada a fase de pré-processamento dos dados meteorológicos utilizados neste trabalho.

No Capítulo 5 são apresentadas as características de implementação dos nós WTIMs e do nó de rede NCAP, bem como a descrição do *hardware* e *software* desenvolvidos neste trabalho. A aplicação da rede também é abordada, ilustrando a instalação dos nós no ambiente de aplicação, bem como a topologia física da rede e suas características.

No Capítulo 6 é apresentada a aplicação da mineração de regras de associação nos dados da rede *mesh* de transdutores inteligentes desenvolvida. Os dados da rede foram submetidos a algumas situações de teste. Entretanto, os dados de cada dispositivo da rede foram analisados separadamente pelo algoritmo de mineração.

No Capítulo 7 serão apresentadas as conclusões gerais e os trabalhos futuros a serem desenvolvidos.

2 REDE MESH DE TRANSDUTORES INTELIGENTES

Neste capítulo é apresentado de forma sucinta o histórico das redes de sensores sem fio, com foco no padrão IEEE 80.15.4 que é a base teórica e estrutural do padrão ZigBee. As principais características do padrão ZigBee são destacadas e os tipos de redes por ele formadas. O desenvolvimento de uma rede requer compatibilidade a nível de *software* e de *hardware*, então, neste capítulo é apresentado o padrão IEEE 1451.5. O objetivo desta norma é definir um padrão para acesso aos transdutores através das interfaces sem fio ZigBee, estabelecendo as estruturas das mensagens e comandos utilizados para a troca de dados entre os dispositivos da rede.

2.1 DEFINIÇÃO

Os transdutores (sensores/atuadores), quando aplicados em conjunto com sistemas embarcados, são denominados como transdutores inteligentes, que realizam o controle ou a aquisição dos dados no ambiente de aplicação. As redes de transdutores são essencialmente sistemas distribuídos com o objetivo de monitorar e controlar um determinado processo. Tais redes são desenvolvidas utilizando-se diversos tipos de tecnologia, como *Field Programmable Gate Arrays* (FPGAs), microcontroladores, *Programmable Logic Controllers* (PLCs), microcomputadores, entre outros dispositivos, sendo que, independente de tecnologia, as redes de transdutores devem ser confiáveis, robustas e rápidas (SONG et al., 2019).

Uma rede *mesh* de transdutores inteligentes pode ser definida como uma rede de dispositivos capaz de mensurar aspectos do ambiente e difundir as informações recolhidas por meio da rede. Os dados são transmitidos por intermédio de nós de rede, eventualmente através de múltiplos saltos, para um nó principal, mediante a uma interface de comunicação sem fio.

2.2 REDES DE SENSORES SEM FIO

As tecnologias provenientes das redes *mesh* de transdutores inteligentes, são precedentes de grandes esforços realizados no desenvolvimento das redes de sensores sem fio, denominadas de WSNs. Semelhante a outros sistemas de comunicação, o desenvolvimento das WSNs tem uma ampla variedade de raízes. Como muitas tecnologias, a origem da WSN pode ser vista em avançadas aplicações industriais e militares.

A primeira rede de sensores sem fio que tinha qualquer semelhança com uma WSN moderna foi a *Sound Surveillance System* (SOSUS). O SOSUS é um sistema de vigilância sonora desenvolvido pelos militares dos Estados Unidos da América em meados da década de 1950, com o objetivo de detectar submarinos soviéticos. A rede dispunha de sensores acústicos (hidrofonos), que eram submersos e distribuídos nos oceanos Atlântico e Pacífico. A tecnologia de sensoriamento ainda está sendo utilizada, porém, para outros fins, no monitoramento da fauna marinha e de atividades vulcânicas (SILICON LABS, 2013).

Por intermédio de investimentos realizados em meados das décadas de 1960 e 1970, para desenvolver a Internet que utilizamos hoje em dia, a *Defense Advanced Research Projects Agency* (DARPA) dos EUA iniciou em 1980 o programa *Distributed Network Sensor* (DSN), com a finalidade de explorar formalmente os desafios da implementação das redes de sensores sem fio. Com a ascendência do programa DSN, e sua progressão com ajuda de algumas universidades como a *Carnegie Mellon University* (CMU) e o *Massachusetts Institute of Technology Lincoln Laboratory* (MIT Lincoln Labs), a tecnologia WSN logo encontrou seu espaço no meio acadêmico e no âmbito científico civil.

Os governos e as universidades, eventualmente, começaram a usar as WSNs em aplicações, como, monitoramento da qualidade do ar, detecção de incêndios florestais, prevenção de desastres naturais, estações meteorológicas e monitoramento estrutural (SILICON LABS, 2013; YINBIAO et al., 2014).

Em meados da década de 1980, as empresas *International Business Machines Corporation* (IBM) e *Bell Laboratories* (Bell Labs) começaram a promover o uso das WSNs em aplicações industriais, tais como, distribuição de energia, tratamento de águas residuais e de automação de fábricas. Nessa época, existia uma grande demanda no mercado de WSNs, porém, pelos padrões da época, relacionados à padronização da tecnologia e pelas tecnologias de hardware existentes, as aplicações desta tecnologia eram limitadas. As redes WSNs existentes eram bastantes funcionais e tinham um bom desempenho,

enquanto outros fatores, tais como hardware, custo, padrões de rede, o consumo de energia e escalabilidade não eram tão evoluídos.

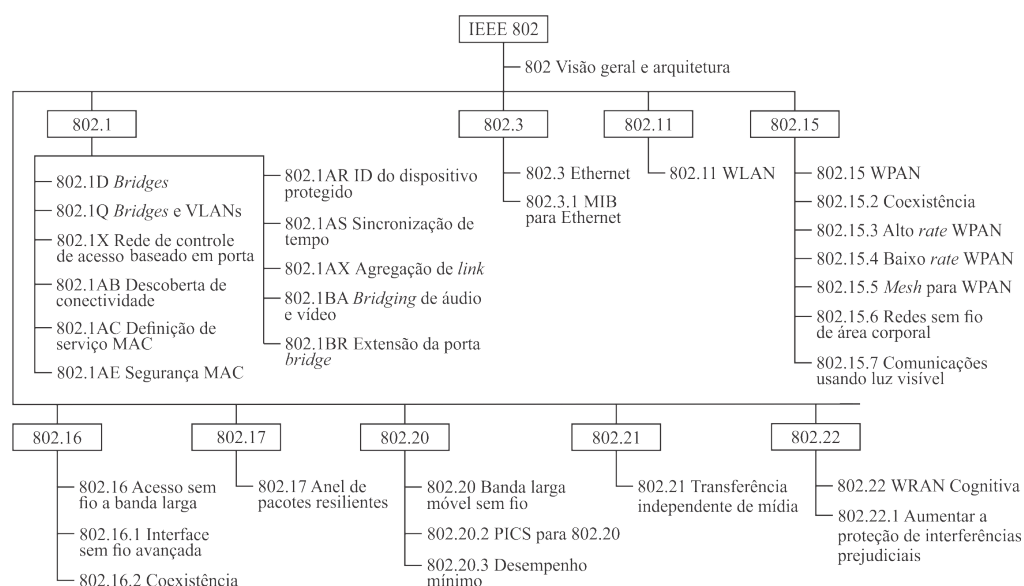
No entanto, no final da década de 1980, os resultados do programa DSN começaram a despertar interesse aos militares dos Estados Unidos. As organizações militares iniciaram programas, afim de adotar a tecnologia de rede de sensores, para fins de guerra. Com as organizações financiando o desenvolvimento da tecnologia WSN, ela começou a fazer rápidos progressos no início da década de 1990 (SILICON LABS, 2013; BINH; DEY, 2018).

Os avanços da computação e da tecnologia de comunicação que foram realizados no final da década de 1990 e no início da década de 2000, levaram à uma nova etapa na evolução da tecnologia WSN. Mais empresas descobriram o enorme potencial das WSNs para aplicações comerciais. Imediatamente a padronização da tecnologia WSN tornou-se importante. Sendo assim, alguns consórcios de empresas foram criados para elaborar padrões referente à tecnologia. Alguns destes padrões são: o ZigBee e o WirelessHART, os quais são baseados na norma IEEE 802.15.4 (BINH; DEY, 2018).

2.3 O PADRÃO IEEE 802.15.4

O padrão IEEE 802.15.4 é parte da família de padrões IEEE 802, apresentada na Figura 2, que é destinada a redes sem fio *Wireless Personal Area Network* (WPAN).

Figura 2 – Família de padrões IEEE 802.

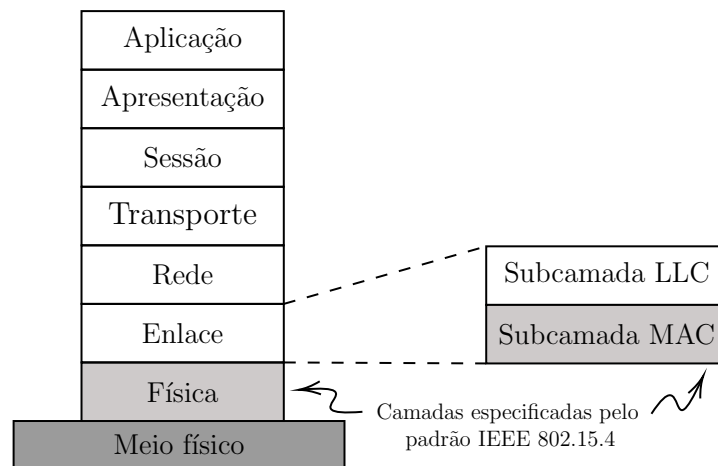


Fonte: ISO/IEC/IEEE (2015).

O objetivo do padrão IEEE 802.15.4 é definir as especificações da subcamada MAC e da camada física de sua pilha de protocolos com o intuito de estabelecer o protocolo e a interconexão compatível para dispositivos de comunicação que utilizam transmissões por radiofrequência (RF) de curto alcance, baixa potência, baixa complexidade e baixa taxa de dados, visando manter rigorosos requisitos de consumo de energia.

Sendo assim, cada dispositivo IEEE 802.15.4 é classificado de acordo com sua funcionalidade, visando reduzir a sua complexidade (IEEE, 2016). Sua pilha de protocolos, ilustrada na Figura 3, é baseada na arquitetura IEEE 802 RM que é uma derivação do modelo de referência OSI/RM.

Figura 3 – Arquitetura IEEE 802.14.5 RM.



Fonte: Versão elaborada pelo autor.

A rede baseada no padrão IEEE 802.15.4 WPAN, ou simplesmente *Low-Rate Wireless Personal Area Network* (LR-WPAN), é uma rede de comunicação simples e de baixo custo, que permite a conectividade de aplicações com potência limitada e requisitos flexíveis. O principal objetivos de uma LR-WPAN é a facilidade de instalação, transferência de dados confiável, baixo custo e consumo reduzido de energia comparado com as redes *Wireless Local Area Network* (WLAN), mantendo um protocolo simples e flexível (IEEE, 2016; GARCIA et al., 2018). Os recursos fornecidos pelo padrão são:

- Operação em topologia estrela ou *peer-to-peer* (ponto a ponto);
- Endereço estendido exclusivo de 64 bits ou endereço curto de 16 bits;
- Alocação opcional de *Guaranteed Time Slots* (GTSs);
- Protocolo de prevenção de colisão CSMA-CA;
- Protocolo com reconhecimento de dados, para confiabilidade na transferência;
- Baixo consumo de energia;
- *Energy detection* (ED);

- *Link quality indication* (LQI);
- 16 canais na banda *Industrial, Scientific and Medical* (ISM¹) de 2.4 GHz, 10 canais na banda ISM de 915 MHz e 1 canal na banda europeia de 868 MHz;
- Taxas de dados de 250 Kb/s (2,4 GHz) e 20/40 Kb/s (868/915 MHz).

Em uma rede LR-WPAN são identificados dois tipos de dispositivos que comunicam entre si para formar diferentes topologias de rede, são eles: dispositivos *Full Function Device* (FFD) e dispositivos *Reduced Function Device* (RFD). Um RFD só pode se comunicar com um FFD, enquanto um FFD pode se comunicar com FFDs e RFDs.

Os dispositivos FFD implementam toda a pilha de protocolos do padrão IEEE 802.15.4, consequentemente eles necessitam de um *hardware* complexo, consumindo assim mais energia. Os dispositivos RFD são mais simples, pois implementam apenas uma parte da pilha de protocolos utilizando menos recursos de *hardware* (GARCIA et al., 2018).

Um dispositivo IEEE 802.15.4 pode assumir três funções diferentes: coordenador, coordenador PAN, e dispositivo. Um coordenador é um dispositivo FFD que é capaz de retransmitir mensagens. Se o coordenador também é o principal controlador de uma LR-WPAN, será denominado de coordenador PAN. Se um dispositivo não está agindo como um coordenador, será simplesmente denominado de “dispositivo”. Um RFD não pode assumir as funções de coordenador, sendo assim, será apenas um dispositivo (GARCIA et al., 2018).

2.3.1 Topologias de rede IEEE 802.15.4

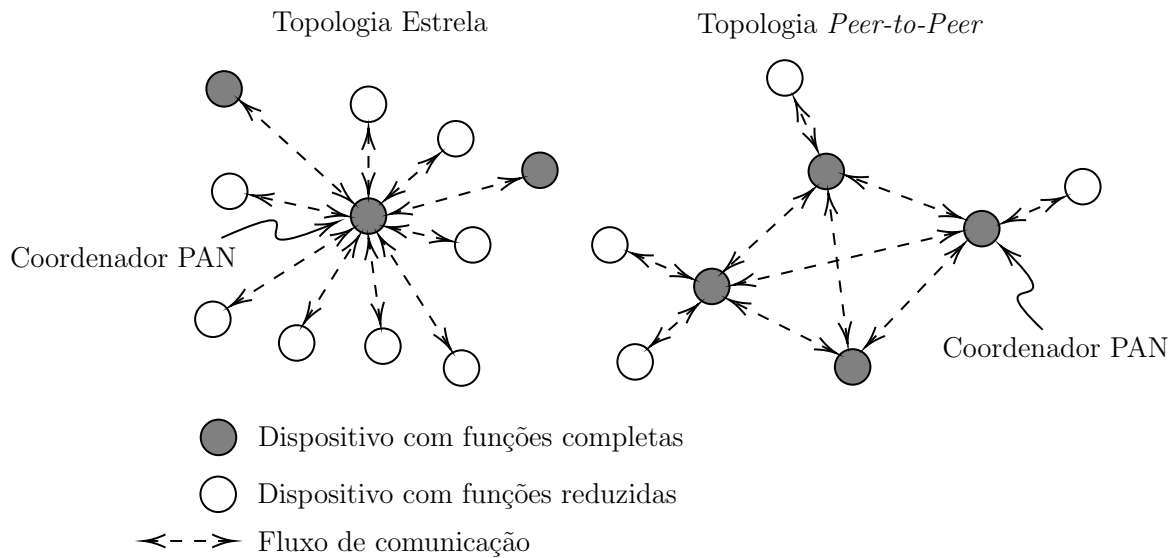
Uma topologia de rede é basicamente o mapa de uma rede. Essencialmente, a estrutura topológica de uma rede pode ser representada fisicamente ou logicamente. A topologia física descreve a disposição, interconexão e localização física de todos os componentes de uma rede, enquanto a topologia lógica ilustra a forma do fluxo de dados de uma rede, independentemente da sua estrutura física. As distâncias entre os dispositivos, as interconexões físicas, as taxas de transmissão, podem ser diferentes entre duas redes, mas suas topologias lógicas podem ser idênticas (GARCIA et al., 2018).

Para superar as limitações na transmissão de dados, como foi descrito na Seção 2.3, topologias de rede com auto-organização e com múltiplos saltos são obrigatórias. Duas topologias básicas são permitidas, mas não são completamente descritas pelo padrão, pois suas funcionalidades são implementadas em camadas mais altas, não fazendo parte do

¹Faixas de rádio, ou seja, porções do espectro de radiofrequências, reservados internacionalmente para fins industriais, científicos e médicos que não sejam no âmbito de telecomunicações.

âmbito da norma, são elas: a topologia estrela e *peer-to-peer* apresentadas na Figura 4, respectivamente (IEEE, 2016).

Figura 4 – Topologias de rede IEEE 802.15.4.



Fonte: Versão elaborada pelo autor.

Cada rede LR-WPAN, independentemente de sua topologia, contém um identificador exclusivo. Este identificador, denominado de *PAN Identifier* (PAN ID), permite a comunicação entre os dispositivos que são configurados com o mesmo identificador.

Uma vez que o identificador PAN é escolhido, o coordenador PAN permite que outros dispositivos, potencialmente ambos os FFDs e RFDs, se juntem a sua rede. Consequentemente, redes que são implementadas em um mesmo local podem operar de forma independente uma da outra. O coordenador PAN é um dispositivo, formado em torno de um FFD, que atua como o nó principal da rede. Trata-se do único dispositivo capaz de se comunicar com mais de um dispositivo, tendo como funcionalidade: a inicialização da rede, distribuição de endereços, manutenção da rede, reconhecimento de todos os dispositivos, entre outras funções (ISO/IEC/IEEE, 2015; GARCIA et al., 2018).

A topologia estrela é usada em aplicações em que a área de abrangência da rede é pequena. Os dispositivos podem se comunicar apenas com o coordenador PAN. Um dispositivo que deseja participar da rede “ouve” uma mensagem de *beacon*² enviada pelo coordenador PAN e, após recebê-la, o dispositivo envia um pedido de associação de volta para o coordenador, o que permite a associação ou não de acordo com uma política pré-definida.

²Pacote de sinalização, que é transmitido periodicamente pelo coordenador da rede, para anunciar a presença de uma LR-WPAN IEEE 802.15.4.

Diferentemente da topologia estrela, a topologia *peer-to-peer* é empregada quando há necessidade de uma grande área de abrangência da rede. Nesta topologia todos os dispositivos podem se comunicar entre si. As mensagens são encaminhadas a partir de qualquer dispositivo a qualquer outro dispositivo na rede através de múltiplos saltos. Os múltiplos saltos, é uma procura proativa a outros dispositivos na rede. No entanto, a implementação de múltiplos saltos requer um dispositivo que contém um *hardware* com memória adicional, neste caso um dispositivo FFD, para a criação de tabelas de roteamento.

Outras estruturas de rede são construídas a partir da topologia *peer-to-peer*, no qual é possível impor restrições sobre a formação topológica da rede. No entanto, tal mecanismo está fora do âmbito desta norma (RAJA; GAVEL; RAGHUVANSHI, 2019; ISO/IEC/IEEE, 2015; LEE et al., 2015).

2.4 O PADRÃO ZIGBEE

Muitas aplicações de monitoramento e controle sem fio são utilizadas em ambientes industriais e domésticos, que exigem maior tempo de operação e menor complexidade em relação a outras tecnologias de rede, como por exemplo: Wi-Fi. O padrão ZigBee foi criado para atender estas necessidades, sendo uma norma desenvolvida por um consórcio crescente de empresas, com mais de 300 membros, que compõem o grupo ZigBee Alliance. O objetivo do ZigBee Alliance é especificar as camadas de rede e aplicação da pilha de protocolos do padrão IEEE 802.15.4, atendendo os requisitos de confiabilidade, segurança, baixo consumo de energia e baixo custo.

Como apresentado na Seção 2.3, existem dois tipos de dispositivos em uma LR-WPAN, os dispositivos FFDs e os RFDs. Sendo assim, o padrão ZigBee especifica os dispositivos seguindo estas diretrizes. Porém, ele utiliza uma terminologia diferente para definir as funções lógicas dos dispositivos, sendo o coordenador, roteador, e dispositivo final. Um coordenador ZigBee é um coordenador PAN IEEE 802.15.4 e roteador ZigBee é um dispositivo que pode agir como um coordenador IEEE 802.15.4. Um dispositivo final ZigBee, é um dispositivo que não é coordenador e nem um roteador, contendo o tamanho mínimo de memória, menor capacidade de processamento e menor número de recursos (FARAHANI, 2008). O relacionamento entre as funções de ambos os padrões é ilustrada na Figura 5.

Figura 5 – Relacionamento entre as funções dos dispositivos IEEE 802.15.4 e ZigBee.

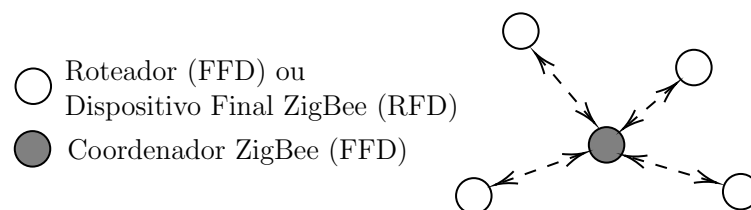


Fonte: Farahani (2008).

O coordenador ZigBee é responsável pela inicialização e manutenção da rede, distribuição de endereços de 16 e 64 bits, reconhecimento de todos os nós permitindo que outros dispositivos entrem ou saiam da rede, encaminhamento de mensagens, entre outras funções. O roteador ZigBee é responsável por configurar e manter *links* de roteamento realizando o encaminhamento de mensagens, proporcionando a expansão da rede. O dispositivo final ZigBee contém apenas a funcionalidade para se comunicar com seu nó pai (o coordenador ou um roteador ZigBee) (PATIL; DATE; KUSHARE, 2014).

A formação da rede é gerenciada pela camada de rede ZigBee. A rede deve ser estabelecida de acordo com uma das duas topologias de rede especificadas no IEEE 802.15.4, a topologia estrela e *peer-to-peer*. Na topologia em estrela, mostrada na Figura 6, cada dispositivo na rede pode se comunicar apenas com o coordenador ZigBee.

Figura 6 – Topologia estrela para a rede ZigBee.



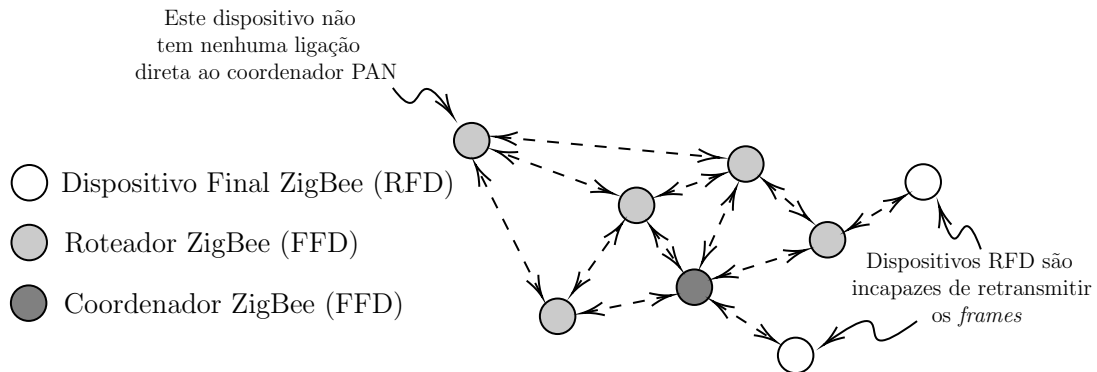
Fonte: Farahani (2008).

Em uma topologia de *peer-to-peer*, cada dispositivo pode se comunicar diretamente com qualquer outro dispositivo, se os dispositivos são colocados perto o suficiente para estabelecer uma ligação de comunicação bem sucedida. Os dispositivos que retransmitem as mensagens são FFDs, ou seja, o coordenador e os roteadores ZigBee, porque RFDs (Dispositivos Finais ZigBee) não são capazes de retransmitir as mensagens. No entanto, um dispositivo final pode fazer parte da rede se comunicando com o coordenador ou algum

roteador.

Uma rede *peer-to-peer* pode assumir diferentes formas usando as restrições sobre os dispositivos que podem se comunicar uns com os outros. Se não há nenhuma restrição, a rede *peer-to-peer* é conhecida como uma topologia *mesh*, ou topologia em malha, ilustrada na Figura 7 (FARAHANI, 2008).

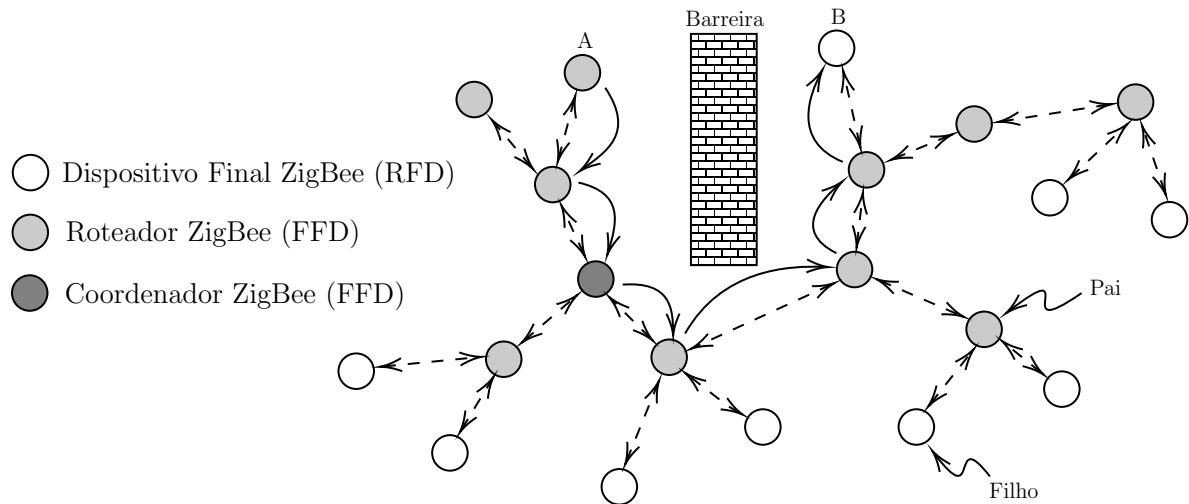
Figura 7 – Topologia *mesh* para a rede ZigBee.



Fonte: Farahani (2008).

Outra forma de rede *peer-to-peer* é a topologia *tree*, ou topologia em árvore, ilustrada na Figura 8. Nesta topologia, um coordenador ZigBee estabelece a rede inicial, os roteadores ZigBee formam os “galhos” e retransmitem as mensagens, e os Dispositivos Finais ZigBee funcionam como “folhas” da árvore e não participam do roteamento de mensagens. Roteadores ZigBee podem fazer com que a rede cresça para além da rede inicial estabelecida pelo coordenador ZigBee. A comunicação é baseada na associação hierárquica entre os dispositivos da rede. Cada nó pai (coordenador ou roteador ZigBee) possui o endereço dos seus respectivos filhos (roteadores ou dispositivos finais ZigBee), e os nós filhos possui o endereço de seu respectivo nó pai. Sendo assim, cada transmissão é feita de forma recursiva na rede, através da análise hierárquica dos nós.

Na Figura 8 é esboçado um exemplo de comunicação entre dois dispositivos em uma rede *tree*. O dispositivo A necessita enviar uma mensagem para o dispositivo B, porém, existe uma barreira física entre os nós; desta maneira, a mensagem é propagada recursivamente na rede, sendo retransmitida em torno da barreira até chegar ao dispositivo B (FARAHANI, 2008; AMEEN; NOURILDEAN, 2013).

Figura 8 – Topologia *Tree* para rede ZigBee.

Fonte: Farahani (2008).

2.5 O PADRÃO IEEE 1451.5

O padrão IEEE 1451, também chamado de “família de padrões IEEE 1451”, tem o objetivo de definir o interfaceamento entre os transdutores inteligentes, ou seja, um protocolo de comunicação padronizado de forma independente da rede de comunicação. A rede *mesh* desenvolvida agrega dois tipos de dispositivos, seguindo as diretrizes do padrão IEEE 1451, sendo: o nó NCAP e os nós WTIMs.

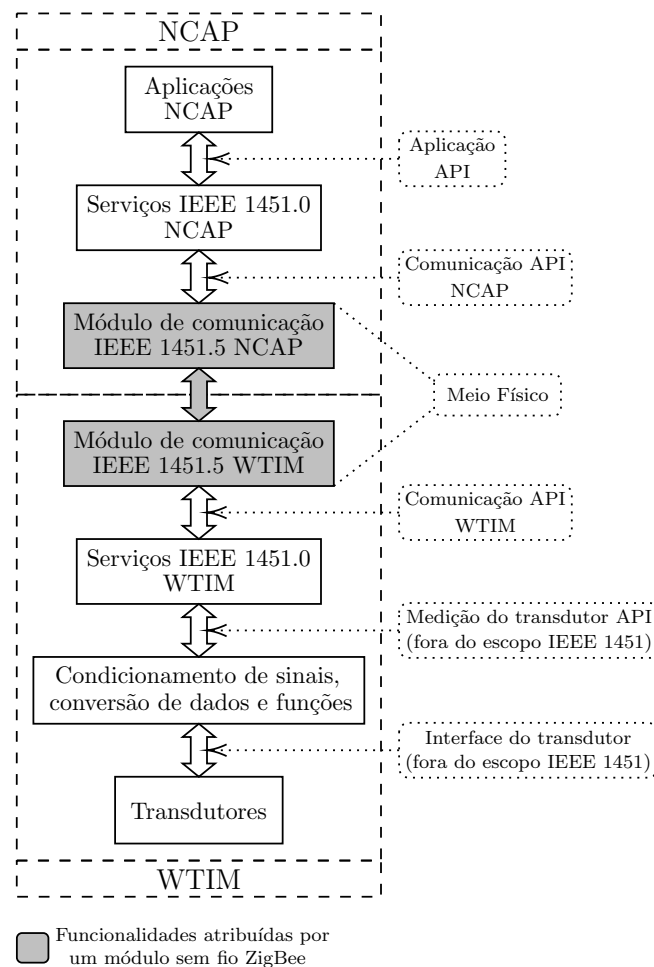
O nó de rede NCAP é especificado pela norma IEEE 1451.1. Tendo em vista que neste trabalho utiliza-se uma rede *mesh* de transdutores inteligentes baseada no padrão de rede ZigBee, a norma IEEE 1451.5 da família de padrões IEEE 1451 foi empregada para definir o interfaceamento entre os nós. O objetivo da norma IEEE 1451.5 é definir um padrão para o acesso aos transdutores através de interfaces de rede sem fio. Sendo assim, é através desta norma que são definidos os nós de rede WTIMs. Ao descrever os WTIMs a norma IEEE 1451.5 baseia-se na norma IEEE 1451.0, que define as TEDS de forma genérica para todas as normas da família de padrões IEEE 1451. Este trabalho não aborda todas as características do padrão IEEE 1451.5. O foco principal foi na comunicação dos transdutores inteligentes referente à troca de mensagens baseadas na norma IEEE 1451.0, não realizando a implementação das TEDS.

A norma IEEE 1451.5 introduz o conceito dos WTIMs conectados a um NCAP, através de uma interface sem fio. As interfaces sem fio aprovadas pelo padrão IEEE 1451.5, chamadas de *An IEEE 1451.5 Approved Radio* (Dot5AR), são: IEEE 802.11,

IEEE 802.15.4, Bluetooth e ZigBee. Um WTIM é um módulo que é composto por um Dot5AR, circuito de condicionamento de sinal, conversor ADC ou/e *Digital-to-Analog Converter* (DAC), e os transdutores. O WTIM pode variar sua complexidade contendo apenas um único transdutor ou vários de acordo com a aplicação (IEEE, 2007b; SONG; LEE; PROCTOR, 2014). O WTIM contém uma interface Dot5AR para a comunicação sem fio, e o NCAP por sua vez contém a mesma interface para completar o *link* de comunicação com o WTIM; esta interface é ilustrada na Figura 9.

Como neste trabalho não há a implementação das TEDS, foram utilizados os serviços de mensagens e comandos da norma IEEE 1451.0. Sendo assim, a arquitetura utilizada neste trabalho é ilustrada na Figura 9. A camada de serviços IEEE 1451.0 provê as funções que definem as mensagens e comandos utilizados na troca de dados com o nó de rede NCAP.

Figura 9 – Arquitetura do interfaceamento dos nós NCAP e WTIM.



Fonte: Versão elaborada pelo autor.

2.5.1 Estrutura das mensagens

Em IEEE (2007a) especifica a norma IEEE 1451.0 que define a estrutura das mensagens (*frames*) enviadas através do módulo de comunicação. Os dados utilizados na transmissão entre o NCAP e o WTIM são descritos em nível de octeto, ou *byte*. Os *bytes* são transmitidos sequencialmente. Na Tabela 1 é ilustrado a ordem de transmissão.

Tabela 1 – Ordem de transmissão dos dados.

1 - <i>byte</i>	1 - <i>byte</i>	1 - <i>byte</i>	1 - <i>byte</i>
8 ^o <i>bit</i> ... 1 ^o <i>bit</i>	8 ^o <i>bit</i> ... 1 ^o <i>bit</i>	8 ^o <i>bit</i> ... 1 ^o <i>bit</i>	8 ^o <i>bit</i> ... 1 ^o <i>bit</i>
1 ^o <i>byte</i> transmitido	2 ^o	3 ^o	4 ^o
5 ^o	6 ^o	7 ^o	8 ^o ...

Fonte: IEEE (2007a).

Para representar uma quantidade numérica, o *bit* mais à esquerda é representado como sendo o mais significativo e, o mais à direita, como sendo o menos significativo. Na Tabela 2 é ilustrado um valor numérico e a ordem de transmissão dos *bits* que correspondem este valor.

Tabela 2 – Ordem de transmissão dos bits de um *byte*.

Ordem	7	6	5	4	3	2	1	0
Valor = 170	1	0	1	0	1	0	1	0

Fonte: IEEE (2007a).

A estrutura de uma mensagem de comando de requisição definida de acordo com a Tabela 3, é formada pelos campos:

- **Número do canal do transdutor:** campo representado por 16 *bits* que define o canal do transdutor de destino;
- **Classe de comando:** campo representado por 8 *bits* que define a classe do comando de acordo com a Tabela 4;
- **Função do comando:** campo representado por 8 *bits* que estabelece a função do comando de acordo com o que foi atribuído na classe de comando;
- **Tamanho:** campo representado por 16 *bits* que determina o tamanho do campo de dados;
- **Campo de dados:** contém as informações a serem transmitidas.

Tabela 3 – Estrutura de uma mensagem de comando.

Mensagem - 10 <i>bytes</i>									
9	8	7	6	5	4	3	2	1	0
<i>byte</i>	Significado								
9	Número do canal do transdutor (<i>byte</i> mais significativo)								
8	Número do canal do transdutor (<i>byte</i> menos significativo)								
7	Classe do comando								
6	Função do comando								
5	Tamanho (<i>byte</i> mais significativo)								
4	Tamanho (<i>byte</i> menos significativo)								
3	Campo de dados (este campo pode expandir de acordo com a necessidade do projeto, aumentando o número de <i>bytes</i> da mensagem)								
2									
1									
0									

Fonte: Versão elaborada pelo autor.

Tabela 4 – Classes de comandos.

Identificação	Atributo	Categoria
0	Reserved	Reservado
1	CommonCmd	Comando comum para o WTIM e o canal do transdutor
2	XdcrIdle	Estado ocioso do transdutor
3	XdcrOperate	Estado de operação do transdutor
4	XdcrEither	Transdutor em estado ocioso ou operando
5	WTIMsleep	Estado de dormência
6	WTIMActive	Comando de estado ativo do WTIM
7	AnyState	Qualquer estado
8 - 127	ReservedClass	Reservada
127 - 255	ClassN	Aberto para os fabricantes

Fonte: IEEE (2007a).

A estrutura de uma mensagem de comando de resposta é definida de acordo com a Tabela 5, sendo formada pelos campos:

- **Sucesso/Falha:** campo representado por 8 *bits* que define se os dados foram enviados com sucesso ou não. Caso este campo for igual a zero, a mensagem foi enviada com sucesso; caso contrário, é considerada como uma mensagem de falha;
- **Tamanho:** campo representado por 16 *bits* que determina o tamanho do campo de dados;
- **Campo de dados:** contém as informações a serem transmitidas.

Na Seção 2.5.2 apresentam-se exemplos de mensagens de requisição e resposta, referente à leitura de um sensor.

Tabela 5 – Mensagem de comando de resposta.

Mensagem - 9 <i>bytes</i>								
8	7	6	5	4	3	2	1	0
<i>byte</i>	Significado							
8	Sucesso/Falha							
7	Tamanho (<i>byte</i> mais significativo)							
6	Tamanho (<i>byte</i> menos significativo)							
5	Campo de dados (este campo pode expandir de acordo com a necessidade do projeto, aumentando o número de <i>bytes</i> da mensagem).							
4								
3								
2								
1								
0								

Fonte: Versão elaborada pelo autor.

2.5.2 Mensagem de comando de requisição e resposta

As funções de operação do transdutor utilizadas neste trabalho são baseadas na classe de comando representado pelo atributo: *XdcrOperate*, empregado para realizar as requisições de leitura dos transdutores. As funções de operação, do atributo *XdcrOperate*, são ilustradas na Tabela 6. Sendo assim, para realizar a requisição de leitura de um determinado sensor, como por exemplo, o sensor localizado no canal 1, a mensagem será representada como ilustrado na Figura 10.

Tabela 6 – Funções de operação.

Identificação	Função	Canal de transdutor	Global/Grupo
0	Reservado	X	-
1	Ler um canal de transdutor	X	-
2	Escrever um canal de transdutor	X	-
3	Comando <i>Trigger</i>	X	X
4	Abortar <i>Trigger</i>	X	X
5 - 127	Reservado	-	-
127 - 255	Aberto para fabricantes	-	-

Fonte: IEEE (2007a).

A classe do comando (Figura 10) igual a 0x03 indica que o atributo é *XdcrOperate*, com a função 0x01 sinalizando a requisição de leitura do canal 1 (0x00 0x01). O canal é representado pelos 2 *bytes* iniciais. O campo tamanho, indica a quantidade de *bytes* reservados para o campo de dados, representado por 2 *bytes*. O campos de dados é um campo de complemento que contém 4 *bytes* reservados, porém, para fazer a requisição de leitura dos sensores eles não são utilizados (0x00 0x00 0x00 0x00).

Figura 10 – Mensagem de comando de requisição de leitura de um sensor.

Mensagem									
00	01	03	01	00	04	00	00	00	00
Identificador do canal		Classe do comando	Função do comando	Tamanho		Campo de dados			

Fonte: Versão elaborada pelo autor.

A mensagem de comando de resposta é utilizada para retornar o valor referente às leituras dos sensores. O comando de resposta da Figura 11 indica que a leitura do sensor ocorreu com sucesso (0x01), caso o valor fosse diferente de 0x01 seria uma mensagem que sinalisaria falha na leitura. O valor obtido da leitura do sensor é localizado no campo de dados, sendo: 0x18, 0x14. O campo de dados contém 6 *bytes* reservados; o valor referente à leitura do sensor sempre é armazenado nos *bytes* menos significativos deste campo.

Figura 11 – Mensagem de comando de resposta da leitura de um sensor.

Mensagem									
01	00	06	00	00	00	00	18	14	
Sucesso	Tamanho		Campo de dados				Leitura do sensor		

Fonte: Versão elaborada pelo autor.

2.5.3 Unidade de transferência de dados ZigBee/IEEE 1451.5

Os padrões ZigBee e IEEE 1451.5 estabelecem regras na comunicação entre os dispositivos vigentes em suas normas. No âmbito da camada de aplicação ZigBee, mais especificamente no *framework* de aplicação, é implementada a unidade de transferência de dados denominada de *frame* API. O *frame* API é a base da comunicação entre os dispositivos da rede *mesh* ZigBee, sendo que, todos os dispositivos se comunicam através de *frames* API. No padrão IEEE 1451.5, a comunicação entre os dispositivos é feita

através de *frames* IEEE 1451.5. Neste trabalho foi implementado os *frames* IEEE 1451.5 de requisição (*request*) e resposta (*reply*), referente às leituras dos sensores.

Para o desenvolvimento da rede *mesh* ZigBee baseando-se no padrão IEEE 1451.5, realizou-se a junção dos *frames* de ambos os padrões. Sendo assim, *frames* IEEE 1451 são encapsulados em *frames* ZigBee e transmitidos através da rede. O *frame* API contém diversas estruturas que são relacionadas ao seu tipo. Cada tipo de *frame* API contém um identificador exclusivo, chamado de Identificador API (ID API), e é através deste identificador que as camadas de rede e aplicação reconhecem as características de um *frame* enviado ou recebido. Alguns tipos de *frames* API, com seus respectivos IDs, são apresentados na Tabela 7.

Tabela 7 – Exemplos de tipos de *frames* API.

Nome do <i>frame</i> API	ID
Comando AT	0x08
Comando AT - Valor do Parâmetro da Fila	0x09
Pedido de Transmissão ZigBee	0x10
Comando AT Remoto	0x17
Comando AT de Resposta	0x88
Transmissão de <i>Status</i> ZigBee	0x8B

Fonte: Versão elaborada pelo autor.

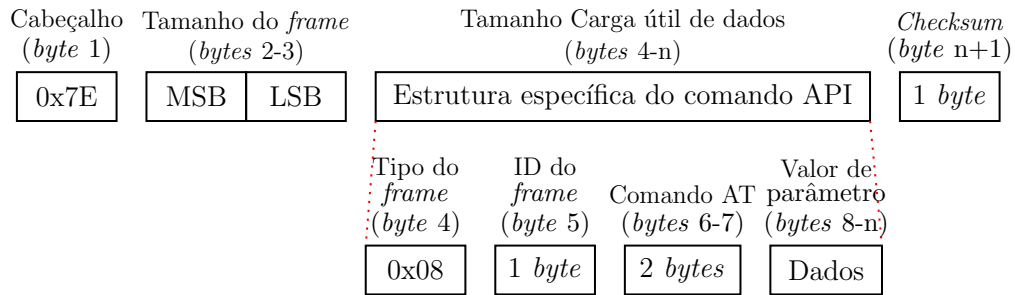
Neste trabalho foram utilizados dois tipos de *frames* API para realizar o gerenciamento e a comunicação entre os dispositivos da rede *mesh* desenvolvida, sendo: Comando AT (0x08) e Pedido de Transmissão ZigBee (0x10). O *frame* “Comando AT” foi utilizado para o reconhecimento e armazenamento dos dados, referentes aos dispositivos da rede *mesh*, sendo que este comando é implementado no nó principal da rede, ou seja, no NCAP. O *frame* “Pedido de Transmissão ZigBee”, foi implementado em todos os dispositivos da rede, sendo a base da comunicação entre os dispositivos, referente à leitura dos sensores.

O *frame* “Comando AT” é utilizado para configurar ou “ler” parâmetros, referentes a um dispositivo local ou remoto na rede *mesh* ZigBee. A estrutura genérica de um *frame* “Comando AT” é ilustrada na Figura 12.

A carga útil de dados do *frame* API contém uma estrutura específica, de acordo com a função do *frame*. Sendo assim, o *byte* 4, localizado após do *byte* LSB (*byte* 3), representará o tipo do *frame*, em qualquer *frame* API. O *byte* 5 armazenará um valor que corresponde o ID do *frame*, sendo utilizado para criar sequências estruturadas de *frames*, como por

exemplo: em uma transmissão de vários *frames* em ordem, contendo dados que devem ser processados de acordo com seu identificador, evitando assim, a entrega de dados de forma errônea. Os *bytes* 6-n serão estruturados de acordo com o tipo do *frame* API (*byte* 4).

Figura 12 – Estrutura genérica do *frame* API “Comando AT”.



Fonte: Versão elaborada pelo autor.

O *byte* que indica o tipo do *frame* é igual a (0x08), sinalizando que este é um *frame* “Comando AT”. Sendo assim, de acordo com o tipo 0x08, os *bytes* 6 e 7 são destinados ao armazenamento do comando AT indicando a função do comando. Neste trabalho, o *frame* “Comando AT” foi utilizado para o reconhecimento dos dispositivos da rede, dessa forma, sua função será: *Node Discovery* (ND: 0x4E, 0x44). O *frame* “Comando AT” com a função ND, tem a estrutura ilustrada na Figura 13.

Figura 13 – *Frame* API Comando AT com a função *Node Discovery* (ND).

0x7E	0x00	0x04	0x08	0x01	0x4E	0x44	0x64
Cabeçalho	Tamanho do <i>frame</i>		Tipo do <i>frame</i> : Comando AT	ID do <i>frame</i>	Comando AT: ND		Checksum

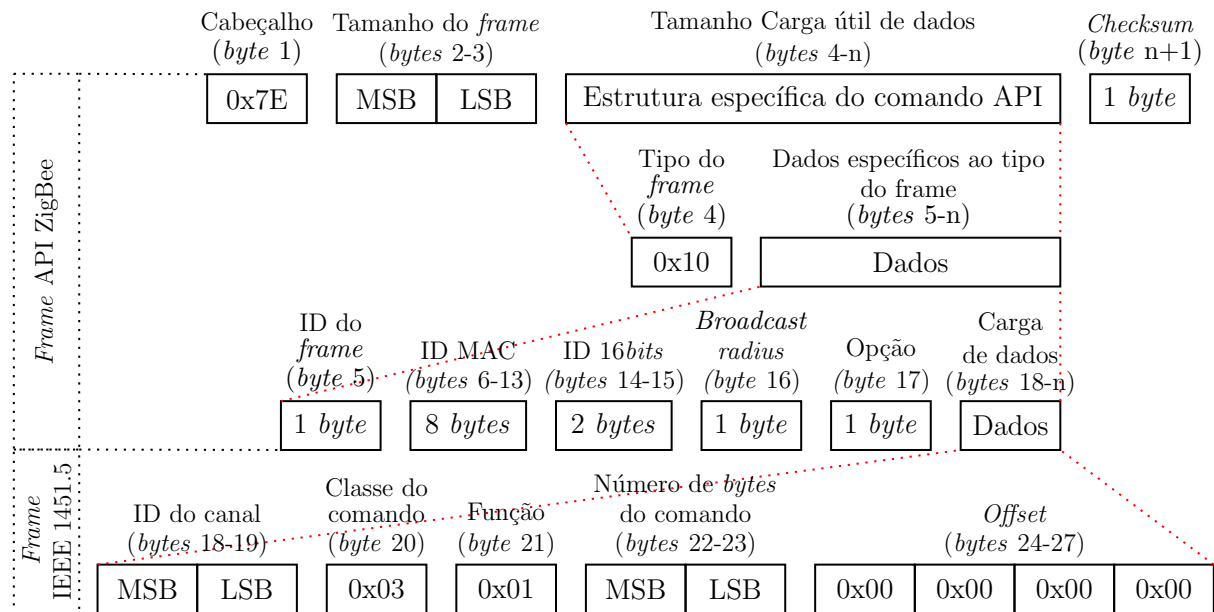
Fonte: Versão elaborada pelo autor.

O *byte checksum* armazena a soma de verificação dos dados contidos no *frame*, com o objetivo de detectar erros em sua estrutura. No âmbito da criação de qualquer *frame* API, é realizado a soma de todos os *bytes* da carga útil de dados (*bytes* 4-n). O resultado da soma é então subtraído pelo *byte* 0xFF, e a diferença é o valor correspondente ao *checksum* do *frame*. Como exemplo, no *frame* ilustrado na Figura 13 o cálculo do *checksum* é realizado da seguinte forma: $[0xFF - (0x08 + 0x01 + 0x4E + 0x44)] = 0x64$.

Utiliza-se o *frame* “Pedido de Transmissão ZigBee” para realizar a troca de mensagens

entre os dispositivos da rede *mesh* ZigBee, com o objetivo de encaminhar os *frames* IEEE 1451.5, realizando o encapsulamento dos dados referente à requisição ou resposta em sua carga útil de dados. A estrutura genérica do *frame* que realiza a requisição de leitura de um sensor é ilustrada na Figura 14.

Figura 14 – Estrutura do *frame* de requisição.

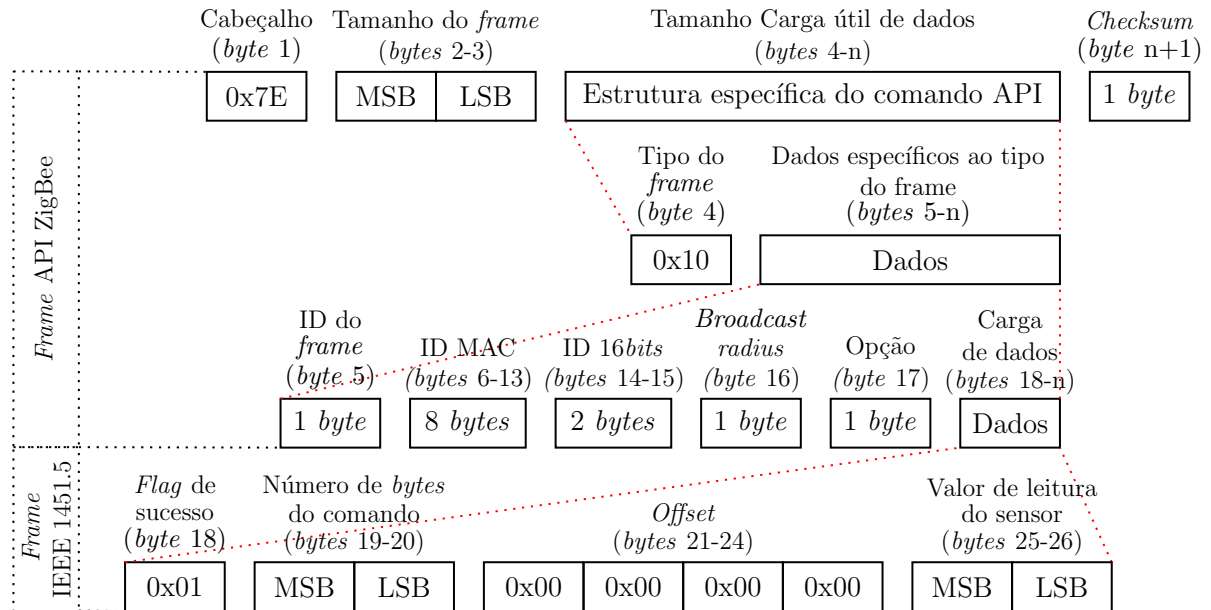


Fonte: Versão elaborada pelo autor.

No *frame*, ilustrado na Figura 14, o *byte* que indica o tipo do *frame* é igual a 0x10. Sendo assim, de acordo com o tipo 0x10, os *bytes* 6-13 são destinados ao MAC ID do dispositivo de destino. Os *bytes* 14-15 representam o endereço curto de 16 *bits* do dispositivo de destino. O *byte* 16 corresponde ao *broadcast radius* da mensagem. O *byte* 17 é o campo de opções de transmissão e os *bytes* 18-n correspondem à carga de dados do *frame*. A carga útil de dados, é o campo onde é armazenado o *frame* de requisição IEEE 1451.5.

A Figura 15 ilustra o encapsulamento de um *frame* de resposta. A diferença na estrutura deste *frame*, em relação ao de requisição, é no campo de carga útil de dados (18-n), onde é armazenado o *frame* de resposta IEEE 1451.5.

Na Figura 16 ilustra-se um exemplo de aplicação do *frame* de pedido de transmissão. Este *frame* encapsula um comando de requisição de leitura IEEE 1451.5, referente a um sensor localizado no canal 0x00 0x01 do dispositivo com MAC ID igual a: 0x00, 0x13, 0xA2, 0x00, 0x40, 0x70, 0x47, 0xC4.

Figura 15 – Estrutura do *frame* de resposta.

Fonte: Versão elaborada pelo autor.

Figura 16 – Exemplo de aplicação do *frame* de pedido de transmissão.

7E	00	18	10	01	00	13	A2	00	40	70	47	C4	FF
FE	00	00	00	01	03	01	00	04	00	00	00	00	78

- Cabeçalho do *frame* API ZigBee
- Tamanho do *frame*
- Tipo do *frame*
- ID do *frame*
- Endereço MAC do dispositivo de destino
- Endereço padrão de 16 bits
- Broadcast radius
- Opções
- Frame IEEE 1451.5 (comando de requisição de leitura de um sensor)
- Checksum

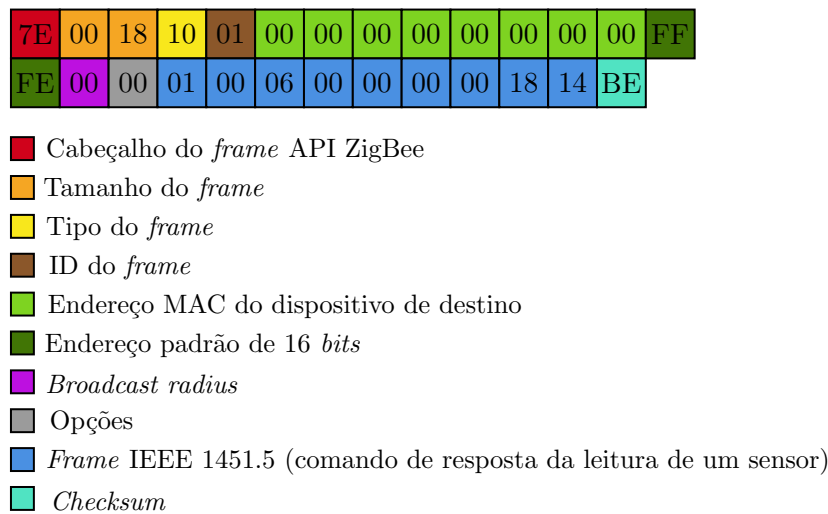
Fonte: Versão elaborada pelo autor.

O endereço de 16 bits é definido a nível de camada de rede; por padrão, seu campo conterá os bytes 0xFF e 0xFE, indicando que este endereço é desconhecido. O campo *broadcast radius* e o campo de opções, são definidos com o byte 0x00, indicando que estes campos não foram utilizados. O *broadcast radius* indica o número máximo de saltos em uma transmissão, ou seja, o número máximo de dispositivos que podem retransmitir as mensagens na rede. Como não foi definido a quantidade máxima de saltos, por padrão, o

número máximo de saltos é indeterminado. No campo de opções é realizada a configuração relacionada aos aspectos de segurança e reconhecimento (*Acknowledgement*) do *frame*. Pelo fato de que neste trabalho não foi empregado nenhuma implementação de controle e segurança no envio dos *frames*, no campo de opções foi utilizado o *byte* 0x00, representando que este campo não é utilizado.

O *frame* de resposta é ilustrado na Figura 17, em sua carga de dados é encapsulado o comando de resposta IEEE 1451.5. Neste trabalho este *frame* foi implementado nos WTIMs da rede, ele é utilizado para enviar ao NCAP informações referentes ao sensor de uma determinada requisição. No campo de endereço MAC do dispositivo de destino é definido os *bytes*: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00. Este endereço indica que o frame é destinado ao coordenador da rede. Por padrão, de acordo com o protocolo ZigBee, se o endereço MAC de destino conter todos os *bytes* igual a 0x00 a mensagem é destinada ao coordenador da rede *mesh*. Os dados do sensor contidos no *frame*, são os *bytes*: 0x18 e 0x14. O NCAP irá manipular os dados do sensor e após a manipulação serão armazenados em uma estrutura de arquivos.

Figura 17 – Exemplo de aplicação do *frame* de resposta.



Fonte: Versão elaborada pelo autor.

2.6 CONSIDERAÇÕES SOBRE O CAPÍTULO

Neste capítulo foram abordados os padrões ZigBee e IEEE 1451.5 que compõe a rede *mesh* desenvolvida nesta pesquisa. É importante destacar que a rede *mesh* apresentada é formada por nós WTIMs e NCAP, e foi exposto a relação entre nós no âmbito da rede ZigBee, de acordo com o interfaceamento definido pela norma IEEE 1451.5.

3 MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO

A mineração de regras de associação trata-se de um relevante ramo da mineração de dados, sendo um dos métodos de mineração de dados mais importantes e amplamente utilizados. O conceito de regras de associação foi proposto pela primeira vez por Agrawal, Imieliński e Swami (1993), com o objetivo de descobrir a relação de uma coleção de dados, sobre as vendas de produtos em um supermercado. Neste capítulo apresenta-se a mineração de regras de associação no âmbito meteorológico, ilustrando a descrição formal do problema, definindo precisamente o que é uma regra de associação e seus conceitos. Também é exposto um exemplo prático do emprego da mineração de regras de associação em uma base de dados.

3.1 DEFINIÇÃO DE REGRAS DE ASSOCIAÇÃO

Nesta seção é realizada a formulação do problema de mineração de regras de associação. Assim, é necessário definir o que são regras de associação e seus conceitos envolvidos. Neste trabalho, a melhor forma de apresentar esta técnica é obtendo conhecimento prático através de dados meteorológicos, como: temperatura, umidade, precipitação, pressão barométrica, etc.

Como exemplo, tem-se: um meteorologista deseja realizar um estudo acerca do relacionamento entre algumas variáveis meteorológicas de um determinado local. Através de uma base de dados que contém observações de um determinado período, o objetivo é extrair alguma correlação entre os valores das variáveis meteorológicas. Uma observação é um conjunto de valores de cada variável meteorológica obtido em um único dia. Cada observação é armazenada no banco de dados, como ilustrado na Tabela 8.

Cada observação diária é denominada de *varset*. Um *varset* com k elementos é denominado de k -*varset*. Suponha que o meteorologista, define que: se um *varset* corresponder pelo menos 50% de todas as observações ele será considerado *frequente*. Por exemplo, considerando o banco de dados representado pela Tabela 8, então o *varset*

{Direção do Vento: 205°, Precipitação: 14mm} é considerado frequente, pois aparece em mais de 60% das observações.

Tabela 8 – Base de dados com a média das leituras realizadas.

IDO	Data	<i>d</i>	<i>v</i>	<i>p</i>	<i>u</i>	<i>t</i>	<i>pb</i>
01	04/10	205	15	14	60	25	9899
02	05/10	205	12	14	45	25	9975
04	06/10	205	12	50	70	22	9987
05	07/10	205	12	14	65	25	9987
06	08/10	205	30	14	70	25	9876
07	09/10	140	12	50	80	23	9987

IDO Identificador da observação

d Direção do vento (°)

v Velocidade do vento (km/h)

p Precipitação (mm)

u Umidade (%)

t Temperatura (°C)

pb Pressão atmosférica (hPa)

Fonte: Versão elaborada pelo autor.

Porém, se o meteorologista mudar sua exigência, passando a considerar que um conjunto frequente esteja pelo menos em 70% das observações, então o *varset* {Direção do Vento: 205°, Precipitação: 14mm} não será considerado frequente. Sendo assim, é definido *suporte* de um *varset* a porcentagem de vezes em que ele aparece no banco de dados de observações. Na Tabela 9 é ilustrado um exemplo que contabiliza os suportes de alguns *varsets* apresentados na Tabela 8.

Tabela 9 – Suportes de alguns *varsets*.

<i>Varset</i>	Suporte
{Direção do vento: 205°, Precipitação: 14mm}	0,6666
{Velocidade do vento: 12km/h, Precipitação: 14mm}	0,3333
{Precipitação: 14mm, Umidade: 60%, Temperatura: 25°C}	0,1666
{Direção do vento: 205°, Velocidade do vento: 12km/h}	0,5

Fonte: Versão elaborada pelo autor.

Caso o meteorologista estabelecer a exigência para que um *varset* seja considerado frequente, correspondendo pelo menos 50% das observações, serão considerados frequentes os *varsets* {Velocidade do vento: 12km/h, Precipitação: 14mm} e {Direção do

vento: 205°, Velocidade do vento: 12km/h}. Na Seção 3.1.1 é apresentado a formulação desse problema.

3.1.1 Formalização do problema

Seja $V = \{v_1, v_2, \dots, v_n\}$ um conjunto de n variáveis meteorológicas diferentes, representando todos os valores das variáveis meteorológicas analisadas por um determinado meteorologista. Dada uma base de dados com observações diárias D , equivalente à apresentada na Tabela 8, em que cada observação diária O é um conjunto de valores das variáveis em V . Isto é, $O \subseteq \text{valores de } V$, e cada O tem um identificador único, correspondente ao identificador IDO (Identificador da Observação) representado na primeira coluna da Tabela 8. Sendo assim, um *varset* de O é subconjunto não vazio dos valores das variáveis de V . Dizemos que O suporta um *varset* o , se $o \subseteq O$. Por exemplo, a primeira observação da Tabela 8 suporta 62 *varsets*, apresentados no Apêndice A. Dessa forma, uma observação suporta $2^k - 2$ *varsets*.

A Regra de Associação é uma implicação $X \Rightarrow Y$, X e Y são *varsets*. Por exemplo, $\{P: 14\text{mm}, U: 60\%\} \Rightarrow \{T: 25^\circ\text{C}\}$ é uma regra de associação. Vamos considerar que esta regra é verdadeira, então a idéia por trás dessa regra, é que: quando precipitar 14mm de chuva e a umidade estiver a 60%, tem-se grande probabilidade da temperatura atingir 25°C. Porém, isso não significa que o inverso dessa regra também é verdadeiro: $\{T: 25^\circ\text{C}\} \Rightarrow \{P: 14\text{mm}, U: 60\%\}$. Quando a temperatura estiver a 25°C não podemos afirmar que irá precipitar 14mm de chuva e que a umidade estará a 60%. Por esse fato, toda regra de associação é associada a um *grau de confiança*. O grau de confiança é simplesmente a porcentagem de observações que suportam X dentre todas as observações que suportam Y , isto é:

$$\text{conf}(X \Rightarrow Y) = \frac{\text{número de observações que suportam } (X \cup Y)}{\text{número de observações que suportam } X} \quad (1)$$

De acordo com a Equação 1, o grau de confiança da regra $\{U: 80\%\} \Rightarrow \{T: 23^\circ\text{C}\}$, com relação ao banco de dados da Tabela 8 é 1 (100%). As variáveis umidade e temperatura assumiram estes valores em apenas uma observação. Desta forma, apenas com o grau de confiança não é possível dizer que a regra anterior é considerada “boa”, já que essa regra representa apenas uma única observação do tipo $\{U: 80\%, T: 23^\circ\text{C}\}$. Com o objetivo de garantir que a regra $X \Rightarrow Y$ seja interessante, é necessário exigir que seu *suporte* seja relativamente alto, além de seu grau de confiança.

Para cada regra de associação $X \Rightarrow Y$ é associado um *suporte*, representado por $sup(X \Rightarrow Y)$. Sendo assim, uma regra de associação r é dita como interessante se $conf(r) \geq \alpha$ e $sup(r) \geq \beta$, onde α é o grau mínimo de confiança e β é o grau mínimo de suporte, ambos definidos pelo usuário. Por exemplo, o suporte da regra $\{VV: 30km/h\} \Rightarrow \{PB: 9876hPa\}$, de acordo com o banco de dados da Tabela 8 é de 0.1666%. Caso, $\alpha = 0.8$ e $\beta = 0.1$ então essa regra é interessante. Porém, caso assumir $\beta = 0.5$ essa regra deixa de ser interessante, independentemente do grau mínimo de confiança α estabelecido.

A criação de regras de associação, a partir de uma banco de dados com inúmeras observações, é um processo muito desafiador e computacionalmente caro. Pois, existem muitas regras que podem ser extraídas a partir de um determinado conjunto de dados (TAN; STEINBACH; KUMAR, 2005). Sendo assim, com o objetivo de diminuir a complexidade da geração de regras de associação, na Seção 3.1.2 é apresentado o algoritmo Apriori, ele foi criado para resolver o problema da mineração de *varsets* frequentes.

3.1.2 O algoritmo Apriori

Através da Seção 3.1.1 foi evidenciado que a mineração de regras de associação é utilizada para encontrar regras confiáveis e representativas entre os *varsets* em um grande banco de dados meteorológico. Desta forma, para calcular o suporte de todos os *varsets* de um banco de dados é necessário de um grande poder computacional. Para contornar este problema, em meados de 1994 foi proposto o algoritmo Apriori. O algoritmo Apriori foi desenvolvido pela equipe de pesquisa do Projeto QUEST da IBM, que originou o *software Intelligent Miner*. Através dele é possível extrair os *varsets* considerados frequentes, facilitando o processo de geração de regras de associação (AMO, 2006; RENLI; YUEMING; LIMING, 2016).

O algoritmo Apriori realiza uma busca por *varsets* frequentes de acordo com o grau mínimo de suporte estabelecido pelo usuário. Desta forma, é necessário fornecer ao algoritmo um banco de dados de observações O e um nível mínimo de suporte β . O retorno do algoritmo é apresentar todos os *varsets* frequentes com relação a O e β . Assim, o algoritmo Apriori é executado através de três fases principais:

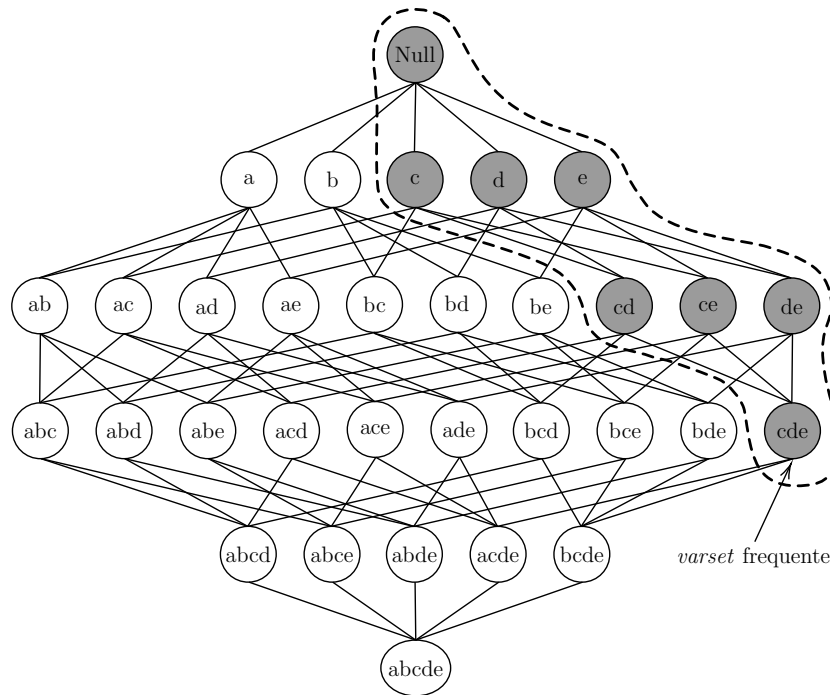
1. fase de geração dos candidatos;
2. fase de poda;
3. fase do cálculo do suporte.

3.1.2.1 O princípio do algoritmo Apriori

O princípio do algoritmo Apriori, utilizado nas fases de geração de candidatos (fase 1) e de poda (fase 2). Esse princípio aborda a propriedade de antimonotonia da relação de inclusão entre os *varsets*.

Uma estrutura de grafo pode ser utilizada para enumerar a lista de todos os *varsets* possíveis. A Figura 18 mostra um grafo de conjuntos de *varsets* para o banco de dados $O = \{a, b, c, d, e\}$. Em geral, um conjunto que tem k observações pode gerar potencialmente $2^k - 1$ conjuntos de *varsets*, excluindo o conjunto vazio (*null*).

Figura 18 – Se o *varset* $\{c, d, e\}$ for frequente, então seus subconjuntos são considerados frequentes.



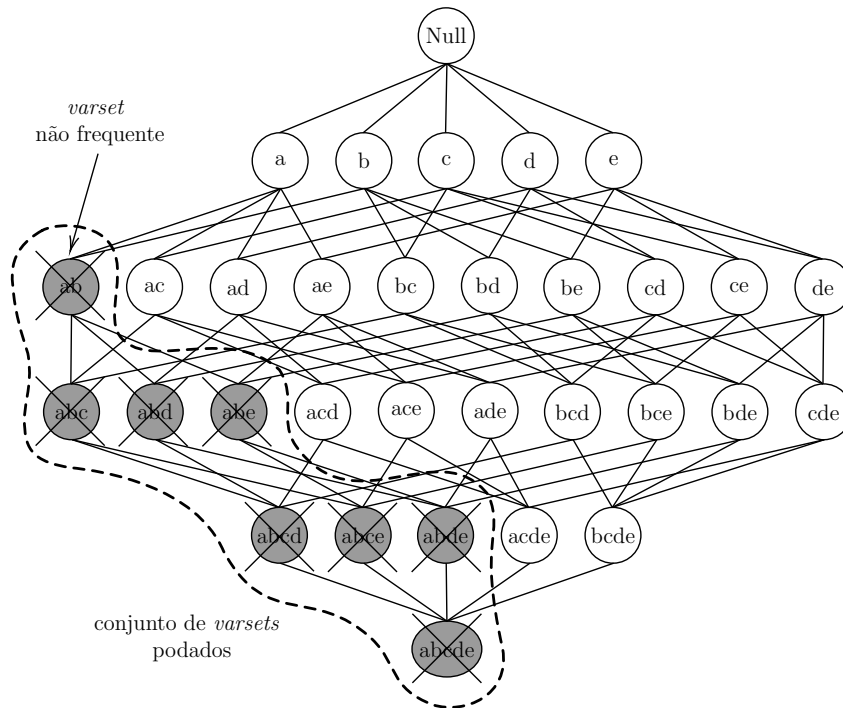
Fonte: Versão elaborada pelo autor.

Segundo Tan, Steinbach e Kumar (2005), o princípio do algoritmo Apriori diz que se um *varset* é frequente, então todos os seus subconjuntos também devem ser frequentes. Para ilustrar, considere a estrutura de *varsets* apresentada na Figura 18. Supondo que o conjunto $\{c, d, e\}$ é um *varset* frequente, então, qualquer observação que contenha $\{c, d, e\}$ também deve conter seus subconjuntos, $\{c, d\}$, $\{c, e\}$, $\{d, e\}$, $\{c\}$, $\{d\}$ e $\{e\}$. Como resultado, se $\{c, d, e\}$ é frequente, então todos os subconjuntos de $\{c, d, e\}$ (ou seja, os conjuntos de *varsets* sombreados na Figura 18) também devem ser frequentes.

Por outro lado, se um *varset* como $\{a, b\}$ “não” for frequente, todos os seus

superconjuntos também não devem ser frequentes. Conforme ilustrado na Figura 19, todo o subgrafo contendo os superconjuntos de $\{a, b\}$ pode ser podado imediatamente, uma vez que $\{a, b\}$ não é considerado frequente. Essa estratégia de “aparar” o espaço de pesquisa exponencial, baseado na medida de suporte, é conhecida como poda baseada em suporte. Essa estratégia é plausível por uma propriedade chave da medida de suporte, ou seja, que o suporte para um *varset* nunca excede o suporte de seus subconjuntos. Essa propriedade também é conhecida como a propriedade de antimonotonia da relação de inclusão entre os *varsets* (TAN; STEINBACH; KUMAR, 2005).

Figura 19 – Se o *varset* $\{a, b\}$ “não” for frequente, então seus superconjuntos não são frequentes.



Fonte: Versão elaborada pelo autor.

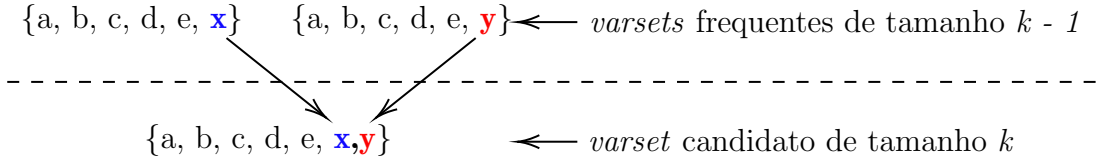
3.1.2.2 A fase da geração de candidatos

Na fase de geração de candidatos é gerado os *varsets* candidatos de tamanho k a partir de conjuntos de tamanho $k - 1$, originados de iterações anteriores. Como o objetivo é gerar somente *varsets* que tenham chance de serem frequentes, devido ao princípio do algoritmo Apriori, todos os *varsets* de tamanho $k - 1$ são frequentes. Assim, um *varset* candidato de tamanho k é construído juntando-se pares de *varsets* de tamanho $k - 1$ e que tem $k - 2$ elementos em comum. Dessa forma temos certeza que um *varset* é gerado por dois subconjuntos frequentes, de tamanho $k - 1$. Mas isso não garante que o *varset*

candidato gerado seja frequente, sendo analisado na segunda fase do algoritmo, descrita na Seção 3.1.2.3 (AMO, 2006).

A Figura 20 ilustra a construção de um *varset* na fase de geração de candidatos. O Algoritmo 1 apresenta o funcionamento da fase de geração de candidatos representado por uma operação de seleção (*select*) utilizada em banco de dados relacionais, gerando o conjunto de *varsets* candidatos C'_k .

Figura 20 – Gerando um *varset* candidato.



Fonte: Versão elaborada pelo autor.

Algoritmo 1 Algoritmo de geração de candidatos

```

1: função geraCandidatos( $L_{k-1}$ )
2:
3: insert into  $C'_k$ 
4: select  $p.var_1, p.var_2, \dots, p.var_{k-2}, p.var_{k-1}, q.var_{k-1}$ 
5: from  $L_{k-1}p, L_{k-1}q$ 
6: where  $p.var_1 = q.var_1, p.var_2 = q.var_2, \dots, p.var_{k-2} = q.var_{k-2}, p.var_{k-1} < q.var_{k-1}$ 
7:
8: retorne  $C'_k$ 
9:
10: fim função

```

Na Seção 3.1.2.3 será abordada a fase de poda do algoritmo Apriori, que realiza a avaliação se um determinado *varset* é frequente ou não.

3.1.2.3 A fase da poda

Na fase de poda são analisados os *varsets* candidatos gerados na fase anterior (geração de candidatos) C'_k . Os *varsets* candidatos que tiverem um subconjunto *varset*, ou um *subvarset*, de tamanho $k - 1$ que não está presente no conjunto de *varsets* frequentes anterior L_{k-1} (de tamanho $k - 1$), é podado do conjunto de *varsets* candidatos C'_k . Esse processo é expressado pela seguinte equação:

$$C_k = C'_k - \{I \mid \text{existe } J \subseteq I \text{ tal que } |J| = k - 1 \text{ e } J \notin L_{k-1}\} \quad (2)$$

Por exemplo, na segunda iteração do algoritmo é gerado um conjunto de *varsets* frequentes de tamanho 2, $L_2 = \{\{a, c\}, \{a, e\}, \{a, d\}, \{b, c\}, \{c, d\}, \{b, d\}\}$. Na

próxima iteração, na fase de geração de *varsets* candidatos, é gerado o conjunto de *varsets* candidatos $C'_3 = \{\{a, c, e\}, \{a, c, d\}, \{a, d, e\}, \{b, c, s\}\}$. Ao analisar o conjunto C'_3 na fase de poda, tem-se os seguintes *varsets* frequentes: $\{\{a, c, d\}, \{b, c, d\}\}$. Neste modo, de acordo com a Equação 2, temos: $C_3 = C'_3 - \{\{a, d, e\}, \{a, c, e\}\} = \{\{a, c, d\}, \{b, c, d\}\}$. Os *varsets* $\{\{a, d, e\}$ e $\{a, c, e\}\}$ foram podados porque eles contém, respectivamente, os *subvarsets* $\{d, e\}$ e $\{c, e\}$ que não estão presentes em L_2 (AMO, 2006).

O Algoritmo 2, descreve todo o processo de poda exposto na Equação 2.

Algoritmo 2 O algoritmo de poda

```

1: função realizaPoda( $C'_k, L_{k-1}$ )
2:
3: para todo  $c \in C'_k$  faça
4:   para todo subconjuntos de  $k - 1$  itens de  $s \in c$  faça
5:     se  $s \notin L_{k-1}$  então
6:       remova  $s$  de  $C'_k$ 
7:     fim se
8:   fim para
9: fim para
10:
11:  $C_k = C'_k$ 
12: retorne  $C_k$ 
13:
14: fim função

```

3.1.2.4 A fase de cálculo do suporte

Na fase de cálculo do suporte é apurado o suporte para cada um dos *varsets* do conjunto C_k originado no processo de poda. O banco de dados de observações O é percorrido, analisando quais são os *varsets* do conjunto C_k que suportam uma observação $o \in O$. Assim, é incrementado o contador de suporte para estes *varsets* (AMO, 2006).

Na Seção 3.1.2.5 será exposto a estrutura completa do algoritmo Apriori, englobando as três fases descritas nas Seções 3.1.2.2, 3.1.2.3 e 3.1.2.4.

3.1.2.5 O algoritmo Apriori: A união das três fases

O pseudocódigo do algoritmo Apriori, que realiza a geração de *varsets* frequentes através das três fases descritas nas Seções 3.1.2.2, 3.1.2.3 e 3.1.2.4, é apresentado no Algoritmo 3.

Nas três fases do algoritmo Apriori são gerados todos os *varsets* frequentes possíveis de um determinado problema. Agora, é necessário obter as regras de associação através

destes *varsets* frequentes. Sendo assim, na Seção 3.2 é apresentado essa etapa, que é muito importante na mineração de regras de associação.

Algoritmo 3 O algoritmo Apriori

```

1: função Apriori( $V, O, \beta$ )
2:
3:  $k = 1$  ▷ Primeira iteração.
4:
5: para todo varset  $v \in V$  faça ▷ Selecciona cada varset.
6:   para toda observação  $o \in O$  tal que  $v \subseteq o$  faça ▷ Selecciona os varsets contidos na
observação  $o$ .
7:      $v.cont = v.cont + 1$  ▷ Incrementa o contador de suporte.
8:   fim para
9: fim para
10:
11:  $L_k = \{v \mid v \in V \text{ tal que } sup(v) \geq \beta; |v| = 1\}$  ▷ Filtra os varsets frequentes, tendo suporte
maior ou igual ao mínimo definido  $\beta$ .
12:
13: faça
14:
15:    $k = k + 1$  ▷ Próxima iteração.
16:    $C'_k = \text{geraCandidatos}(L_{k-1})$  ▷ Gerando os varsets candidatos (fase 1).
17:    $C_k = \text{realizaPoda}(C'_k, L_{k-1})$  ▷ Processo de poda dos varsets candidatos (fase 2).
18:
19:   para toda observação  $o \in O$  faça } ▷ Realiza o cálculo do suporte para os varsets
20:      $C_o \leftarrow \text{subConjuntos}(C'_k, o);$  candidatos do conjunto  $C_k$ , de acordo com
21:     para todo varset  $c \in C_o$  faça cada observação (fase 3);
22:        $c.cont = c.cont + 1$ 
23:     fim para
24:   fim para } ▷ E extrai os varsets frequentes, tendo suporte
25:    $L_k = \{c \mid c \in C_k \text{ tal que } sup(c) \geq \beta\}$  maior ou igual ao mínimo definido  $\beta$  (fase 3).
26:
27:
28: enquanto  $L_k \neq \emptyset$  ▷ Enquanto gerar varsets frequentes em  $L_k$  este laço é executado.
29:
30: retorne  $L_k$  ▷ Retorna todos os varsets frequentes.
31:
32: fim função

```

3.2 GERANDO AS REGRAS DE ASSOCIAÇÃO

Ao finalizar a geração dos *varsets* frequentes pelo algoritmo Apriori, é necessário definir as regras de associação do tipo $X \Rightarrow Y$. Uma vez que X e Y são *varsets*. Sendo assim, para cada *varset* frequente $l = \{l_1, l_2, \dots, l_k\}$ de L com $k \geq 2$, pode-se gerar no máximo $2^k - 2$ regras diferentes através dos itens do conjunto l . O conjunto X da regra $X \Rightarrow Y$ será um subconjunto de l , uma vez que $X \subset l$. Agora, o conjunto Y também será subconjunto de l , porém os itens contidos neste conjunto serão os elementos $l - X$. Por exemplo, considerando que o *varset* $\{DV: 205^\circ, P: 14\text{mm}, 25^\circ\text{C}\}$ seja classificado como frequente pelo algoritmo Apriori. Então, pode-se gerar 6 ($2^3 - 2$) regras diferentes, obtendo as regras de associação demonstradas na Tabela 10.

Tabela 10 – Gerando as regras de um *varset* frequente.

$l = \{\text{DV: } 205^\circ, \text{P: } 14\text{mm}, \text{T: } 25^\circ\text{C}\}$		$k = 3$
X	$Y = l - X$	Regra
$\{\text{DV: } 205^\circ\}$	$\{\text{P: } 14\text{mm}, \text{T: } 25^\circ\text{C}\}$	$\{\text{DV: } 205^\circ \Rightarrow \text{P: } 14\text{mm}, \text{T: } 25^\circ\text{C}\}$
$\{\text{P: } 14\text{mm}\}$	$\{\text{DV: } 205^\circ, \text{T: } 25^\circ\text{C}\}$	$\{\text{P: } 14\text{mm} \Rightarrow \text{DV: } 205^\circ, \text{T: } 25^\circ\text{C}\}$
$\{\text{T: } 25^\circ\text{C}\}$	$\{\text{DV: } 205^\circ, \text{P: } 14\text{mm}\}$	$\{\text{T: } 25^\circ\text{C} \Rightarrow \text{DV: } 205^\circ, \text{P: } 14\text{mm}\}$
$\{\text{DV: } 205^\circ, \text{P: } 14\text{mm}\}$	$\{\text{T: } 25^\circ\text{C}\}$	$\{\text{DV: } 205^\circ, \text{P: } 14\text{mm} \Rightarrow \text{T: } 25^\circ\text{C}\}$
$\{\text{DV: } 205^\circ, \text{T: } 25^\circ\text{C}\}$	$\{\text{P: } 14\text{mm}\}$	$\{\text{DV: } 205^\circ, \text{T: } 25^\circ\text{C} \Rightarrow \text{P: } 14\text{mm}\}$
$\{\text{P: } 14\text{mm}, \text{T: } 25^\circ\text{C}\}$	$\{\text{DV: } 205^\circ\}$	$\{\text{P: } 14\text{mm}, \text{T: } 25^\circ\text{C} \Rightarrow \text{DV: } 205^\circ\}$

T: Temperatura, P: Precipitação, U: Umidade, DV: Direção do vento

Fonte: Versão elaborada pelo autor.

Após obter todas as regras de associação, de acordo com o *varset* frequente $\{\text{DV: } 205^\circ, \text{P: } 14\text{mm}, 25^\circ\text{C}\}$, é necessário definir o grau de confiança de cada regra. O grau de confiança é calculado de acordo com a Equação 1 apresentada na Seção 3.1.1. O cálculo do grau de confiança de cada regra da Tabela 10 é apresentado na Tabela 11. Os dados referentes as transações foram obtidos através da Tabela 8 da Seção 3.1.

Tabela 11 – Calculando o grau de confiança de cada regra.

Regra	Transações ($X \cup Y$)	Transações (X)	$conf(X \Rightarrow Y)$
$\{\text{DV: } 205^\circ \Rightarrow \text{P: } 14\text{mm}, \text{T: } 25^\circ\text{C}\}$	4	5	0.8
$\{\text{P: } 14\text{mm} \Rightarrow \text{DV: } 205^\circ, \text{T: } 25^\circ\text{C}\}$	4	4	1
$\{\text{T: } 25^\circ\text{C} \Rightarrow \text{DV: } 205^\circ, \text{P: } 14\text{mm}\}$	4	4	1
$\{\text{DV: } 205^\circ, \text{P: } 14\text{mm} \Rightarrow \text{T: } 25^\circ\text{C}\}$	4	4	1
$\{\text{DV: } 205^\circ, \text{T: } 25^\circ\text{C} \Rightarrow \text{P: } 14\text{mm}\}$	4	4	1
$\{\text{P: } 14\text{mm}, \text{T: } 25^\circ\text{C} \Rightarrow \text{DV: } 205^\circ\}$	4	4	1

Fonte: Versão elaborada pelo autor.

A próxima etapa é definir qual regra de associação será considerada “interessante”. De acordo com a Seção 3.1.1, uma regra é interessante se $conf(r) \geq \alpha$ e $sup(r) \geq \beta$, onde α é o grau mínimo de confiança e β é o grau mínimo de suporte, ambos definidos pelo usuário. O *varset* $\{\text{DV: } 205^\circ, \text{P: } 14\text{mm}, 25^\circ\text{C}\}$ é frequente de acordo com o algoritmo Apriori. Então, a condição $sup(r) \geq \beta$ é verdadeira. Sendo que, r é a regra imposta e β o grau mínimo de suporte definido pelo usuário; o cálculo do suporte no algoritmo Apriori é exibido na linha 26 do Algoritmo 3. Assim, o grau de confiança da primeira regra foi 0.8 (80%) e das demais regras foi 1 (100%).

Um meteorologista, pode definir: para uma regra ser interessante ela precisa ter no mínimo 70% de confiança. Sendo assim, todas as regras geradas serão consideradas interessantes. Porém, se o meteorologista mudar sua exigência, passando a considerar uma regra interessante com pelo menos 90% de confiança. Então, a primeira regra não será considerada interessante, pois, contém o grau de confiança de 80%. As demais regras sempre serão interessantes, independentemente do grau mínimo de confiança imposto pelo meteorologista, pois tem o valor máximo de confiança (100%).

3.3 EXEMPLO PRÁTICO DE CRIAÇÃO DE REGRAS DE ASSOCIAÇÃO

Nesta seção é exibido um exemplo prático e sucinto do emprego da mineração de regras de associação no âmbito meteorológico. Cada observação diária é ilustrada na Tabela 12.

Tabela 12 – Banco de dados de observações.

IDO	Data	p	u	t
01	04/10	14	60	25
02	05/10	14	45	25
04	06/10	50	70	22
05	07/10	14	65	25
06	08/10	14	70	25
07	09/10	50	80	23
08	10/10	14	70	25

IDO Identificador da observação

p Precipitação (mm)

u Umidade (%)

t Temperatura ($^{\circ}\text{C}$)

Fonte: Versão elaborada pelo autor.

Foram extraídos do conjunto de observações as variações de cada variável meteorológica e essa variação é armazenada em uma estrutura de dados para executar a primeira iteração do algoritmo Apriori. Essa estrutura de dados é ilustrada pela Tabela 13. Nesse exemplo, o suporte mínimo β é definido como 0.2, ou seja, de 20%.

Primeiramente, executa-se a primeira iteração do algoritmo Apriori ($k = 1$), que é descobrir quais são os *varsets* frequentes de tamanho 1. De acordo com a Tabela 13, existem apenas 10 variações no conjunto de dados, então é necessário identificar o suporte de cada variação. Nesta iteração, cada variação desta lista é chamada de *varset*, e os suportes referentes são apresentados pela Tabela 14.

Tabela 13 – O conjunto de *varsets* iniciais contém as variações das variáveis meteorológicas.

Variáveis Meteorológicas	<i>varsets</i> $k = 1$
Precipitação	{P: 14mm}
	{P: 50mm}
Umidade	{U: 45%}
	{U: 60%}
	{U: 65%}
	{U: 70%}
	{U: 80%}
Temperatura	{T: 22°C}
	{T: 23°C}
	{T: 25°C}

Fonte: Versão elaborada pelo autor.

Tabela 14 – Suporte dos *varsets* de tamanho 1.

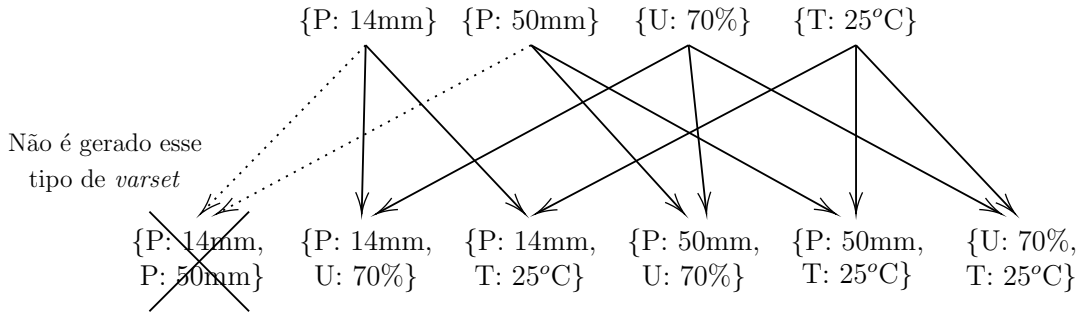
<i>varsets</i> $k = 1$	Número de ocorrências	Suporte	Frequente
{P: 14mm}	5	0.71	✓
{P: 50mm}	2	0.28	✓
{U: 45%}	1	0.14	-
{U: 60%}	1	0.14	-
{U: 65%}	1	0.14	-
{U: 70%}	3	0.42	✓
{U: 80%}	1	0.14	-
{T: 22°C}	1	0.14	-
{T: 23°C}	1	0.14	-
{T: 25°C}	5	0.71	✓

Fonte: Versão elaborada pelo autor.

Após calcular o suporte é necessário definir quais *varsets* são frequentes. Um *varset* é considerado frequente quando seu suporte é maior ou igual ao mínimo definido β . Sendo assim, apenas 4 *varsets* de tamanho 1 são considerados frequentes, de acordo com a quarta coluna da Tabela 14. Logo, o conjunto de *varsets* frequentes L_k , denominado de L_1 , é: $\{\{P: 14mm\}, \{P: 50mm\}, \{U: 70\% \}, \{T: 25^\circ C\}\}$.

No início da segunda iteração ($k = 2$) é efetuada a geração dos *varsets* candidatos, a partir do conjunto de *varsets* frequentes identificado na iteração anterior (L_{k-1}), denominado de L_1 . Sendo assim, na Figura 21 é apresentada a geração dos *varsets* candidatos. A fase de geração dos *varsets* candidatos é apresentada na Seção 3.1.2.2.

Figura 21 – Geração dos *varsets* candidatos C'_2 .



Fonte: Versão elaborada pelo autor.

Em uma mesma observação é armazenado apenas um dado de cada variável meteorológica. Desta forma, ao gerar um *varset* candidato, não é permitido a associação de uma mesma variável meteorológica, como por exemplo: $\{P: 14mm, P: 50mm\}$.

Logo, o conjunto C'_k de *varsets* candidatos gerados na iteração $k = 2$, denominado de C'_2 , é: $\{\{P: 14mm, U: 70\%\}, \{P: 14mm, T: 25^\circ C\}, \{P: 50mm, U: 70\%\}, \{P: 50mm, T: 25^\circ C\}, \{U: 70\%, T: 25^\circ C\}\}$. Em seguida, é executado o processo de poda no conjunto de *varsets* candidatos C'_2 . Este processo é apresentado na Seção 3.1.2.3. Caso haja um *varset* candidato contendo algum subconjunto, de tamanho $k - 1$, que não está presente no conjunto de *varsets* frequentes anterior L_1 , haverá a poda deste *varset* do conjunto de *varsets* candidatos C'_2 . Deste modo, o subconjunto de cada *varset* candidato é apresentado na Tabela 15.

Tabela 15 – Poda dos *varsets* candidatos C'_2 .

<i>Varset</i> candidato C'_2	Subconjunto ($k - 1$)	Poda
$\{P: 14mm, U: 70\%\}$	$\{\{P: 14mm\}, \{U: 70\%\}\}$	-
$\{P: 14mm, T: 25^\circ C\}$	$\{\{P: 14mm\}, \{T: 25^\circ C\}\}$	-
$\{P: 50mm, U: 70\%\}$	$\{\{P: 50mm\}, \{U: 70\%\}\}$	-
$\{P: 50mm, T: 25^\circ C\}$	$\{\{P: 50mm\}, \{T: 25^\circ C\}\}$	-
$\{U: 70\%, T: 25^\circ C\}$	$\{\{U: 70\%\}, \{T: 25^\circ C\}\}$	-

Fonte: Versão elaborada pelo autor.

Como todos os subconjuntos obtidos, apresentados na Tabela 15, estão presentes no

conjunto de *varsets* frequentes L_1 , não há nenhuma poda nos *varsets* candidatos de C'_2 . Logo, o conjunto de *varsets* candidatos após o processo de poda $C_2 = C'_2$. Posteriormente é calculado o suporte dos *varsets* candidatos de C_2 , para identificar os *varsets* frequentes. Sendo assim, na Tabela 16 é exibido o suporte de cada *varset* candidato.

Tabela 16 – Suporte dos *varsets* candidatos C_2 .

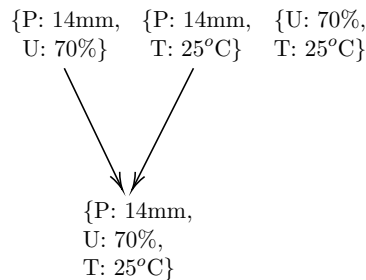
<i>Varset</i> candidato C_2	Número de ocorrências	Suporte	Frequente
{P: 14mm, U: 70%}	2	0.28	✓
{P: 14mm, T: 25°C}	5	0.71	✓
{P: 50mm, U: 70%}	1	0.14	-
{P: 50mm, T: 25°C}	0	0	-
{U: 70%, T: 25°C}	2	0.28	✓

Fonte: Versão elaborada pelo autor.

Os *varsets* candidatos com suporte maior ou igual ao mínimo β , são: {P: 14mm, U: 70%}, {P: 14mm, T: 25°C} e {U: 70%, T: 25°C}, logo estes *varsets* são considerados frequentes. Então, o conjunto de *varsets* frequentes L_k obtido na iteração $k = 2$, denominado de L_2 , é: {{P: 14mm, U: 70%}, {P: 14mm, T: 25°C}, {U: 70%, T: 25°C}}.

Na segunda iteração ($k = 2$) houve a geração de *varsets* frequentes, então todos os passos da segunda iteração são novamente executados na próxima iteração ($k = 3$). Pois, de acordo com a linha 28 do Algoritmo 3 apresentado na Seção 3.1.2.5, a condição de execução da estrutura de repetição é $L_2 \neq \emptyset$. Portanto, na terceira iteração ($k = 3$) é inicialmente realizada a geração dos *varsets* candidatos C'_k , denominados de C'_3 , através do conjunto de *varsets* frequentes obtidos na iteração anterior L_{k-1} , denominado de L_2 , ilustrados na Tabela 16. Sendo assim, na Figura 17 é apresentada a criação dos *varsets* candidatos desta iteração.

Tabela 17 – Geração dos *varsets* candidatos C'_3 .



Fonte: Versão elaborada pelo autor.

Logo, o conjunto C'_3 é formado por apenas um elemento, sendo: $\{\{P: 14\text{mm}, U: 70\%, T: 25^\circ\text{C}\}\}$. Como foi realizado na iteração anterior ($k = 2$), é executado o processo de poda no conjunto C'_3 . Sendo assim, o subconjunto do *varset* candidato, com elementos de tamanho $k - 1$, é apresentado na Tabela 18.

Tabela 18 – Poda dos *varsets* candidatos C'_3 .

Varset candidato C'_3	Subconjunto ($k - 1$)	Poda
$\{P: 14\text{mm}, U: 70\%, T: 25^\circ\text{C}\}$	$\{\{P: 14\text{mm}, U: 70\% \},$ $\{P: 14\text{mm}, T: 25^\circ\text{C} \},$ $\{U: 70\%, T: 25^\circ\text{C} \} \}$	-

Fonte: Versão elaborada pelo autor.

Através da Tabela 18 compreende-se que o *varset* candidato $\{P: 14\text{mm}, U: 70\%, T: 25^\circ\text{C}\}$ não foi podado. Pois, todos os elementos de seu subconjunto pertencem ao conjunto de *varsets* frequentes obtidos na iteração anterior (L_{k-1}), denominado de L_2 .

Como não houve poda no conjunto C'_3 que contém apenas um *varset* candidato, então o conjunto C_3 , é: $\{\{P: 14\text{mm}, U: 70\%, T: 25^\circ\text{C}\}\}$. Agora, é necessário definir se o *varset* candidato de C_3 é frequente, desta forma, é necessário calcular seu grau de suporte. Na Tabela 19 é apresentado o suporte, e se é considerado frequente.

Tabela 19 – Suporte dos *varsets* candidatos C_3 .

Varset candidato C_3	Número de ocorrências	Suporte	Frequente
$\{P: 14\text{mm}, U: 70\%, T: 25^\circ\text{C}\}$	2	0.28	✓

Fonte: Versão elaborada pelo autor.

Logo, o *varset* candidato de C_3 é considerado frequente, pois, contém suporte maior ou igual ao mínimo definido β . Assim, o conjunto de *varsets* frequentes L_k na iteração $k = 3$, denominado de L_3 , é: $\{\{P: 14\text{mm}, U: 70\%, T: 25^\circ\text{C}\}\}$. Como o conjunto de *varsets* frequentes $L_3 \neq \emptyset$, então haverá outra iteração na estrutura de repetição. Porém, na quarta iteração não será gerado nenhum *varset* candidato e consequentemente nenhum *varset* frequente, pois na iteração $k = 3$ foi gerado apenas um *varset* frequente. Logo, o algoritmo Apriori será finalizado, pois $L_4 = \emptyset$.

Ao finalizar a execução do algoritmo Apriori, os *varsets* frequentes obtidos serão analisados com a finalidade de gerar regras de associação para o problema proposto. O

processo de geração de regras de associação é descrito na Seção 3.2. A Tabela 20 apresenta todos os *varsets* frequentes gerados através do algoritmo Apriori. Os *varsets* da primeira iteração, que possuem tamanho 1, não serão utilizados nesta etapa, pois não existe a associatividade de itens em um único elemento do conjunto. Desta forma, as regras são obtidas a partir dos *varsets* frequentes relacionados a segunda e terceira iteração.

Tabela 20 – Conjuntos de *varsets* frequentes obtidos nas iterações do algoritmo Apriori.

Iteração k	Conjunto de <i>varsets</i> frequentes L_k
1	$\{\{P: 14\text{mm}\}, \{P: 50\text{mm}\}, \{U: 70\%\}, \{T: 25^\circ\text{C}\}\}$
2	$\{\{P: 14\text{mm}, U: 70\%\}, \{P: 14\text{mm}, T: 25^\circ\text{C}\}, \{U: 70\%, T: 25^\circ\text{C}\}\}$
3	$\{P: 14\text{mm}, U: 70\%, T: 25^\circ\text{C}\}$

Fonte: Versão elaborada pelo autor.

Inicialmente, serão analisados os *varsets* frequentes obtidos na segunda iteração ($k = 2$). Sendo assim, as regras obtidas destes *varsets* são ilustradas na Tabela 21. Como todas as possíveis regras foram criadas, então é necessário identificar o grau de confiança de cada regra. Neste exemplo, a confiança mínima α é definida como 0.6, ou seja, de 60%.

Tabela 21 – Gerando as possíveis regras de associação dos *varsets* frequentes de L_2 .

<i>Varset</i> frequente (l)	X	$Y = l - X$	Regra
$\{P: 14\text{mm}, U: 70\%\}$	$\{P: 14\text{mm}\}$	$\{U: 70\%\}$	$\{P: 14\text{mm}\} \Rightarrow \{U: 70\%\}$
	$\{U: 70\%\}$	$\{P: 14\text{mm}\}$	$\{U: 70\%\} \Rightarrow \{P: 14\text{mm}\}$
$\{P: 14\text{mm}, T: 25^\circ\text{C}\}$	$\{P: 14\text{mm}\}$	$\{T: 25^\circ\text{C}\}$	$\{P: 14\text{mm}\} \Rightarrow \{T: 25^\circ\text{C}\}$
	$\{T: 25^\circ\text{C}\}$	$\{P: 14\text{mm}\}$	$\{T: 25^\circ\text{C}\} \Rightarrow \{P: 14\text{mm}\}$
$\{U: 70\%, T: 25^\circ\text{C}\}$	$\{U: 70\%\}$	$\{T: 25^\circ\text{C}\}$	$\{U: 70\%\} \Rightarrow \{T: 25^\circ\text{C}\}$
	$\{T: 25^\circ\text{C}\}$	$\{U: 70\%\}$	$\{T: 25^\circ\text{C}\} \Rightarrow \{U: 70\%\}$

Fonte: Versão elaborada pelo autor.

A Tabela 22 apresenta o cálculo de confiança de cada possível regra de associação, com base nas observações armazenadas pelo meteorologista e apresentadas na Tabela 12. As regras consideradas interessantes, que contém grau de confiança maior ou igual ao mínimo α , de acordo com a Tabela 22, são: $\{U: 70\%\} \Rightarrow \{P: 14\text{mm}\}$, $\{P: 14\text{mm}\} \Rightarrow \{T: 25^\circ\text{C}\}$, $\{T: 25^\circ\text{C}\} \Rightarrow \{P: 14\text{mm}\}$ e $\{U: 70\%\} \Rightarrow \{T: 25^\circ\text{C}\}$.

Agora, é necessário analisar os *varset* frequente obtido na terceira iteração ($k = 3$). Dessa forma, as regras obtidas do *varset* são ilustradas na Tabela 23; o cálculo de cada regra é feito com base nas obserações da Tabela 12.

Tabela 22 – Calculando o grau de confiança de cada regra de L_2 .

Regra	$(X \cup Y)$	X	$conf(X \Rightarrow Y)$	Interessante
$\{P: 14mm\} \Rightarrow \{U: 70\%\}$	2	5	0.4	-
$\{U: 70\%\} \Rightarrow \{P: 14mm\}$	2	3	0.66	✓
$\{P: 14mm\} \Rightarrow \{T: 25^\circ C\}$	5	5	1	✓
$\{T: 25^\circ C\} \Rightarrow \{P: 14mm\}$	5	5	1	✓
$\{U: 70\%\} \Rightarrow \{T: 25^\circ C\}$	2	3	0.66	✓
$\{T: 25^\circ C\} \Rightarrow \{U: 70\%\}$	2	5	0.4	-

Fonte: Versão elaborada pelo autor.

Tabela 23 – Gerando as possíveis regras de associação dos *varsets* frequentes de L_3 .

<i>Varset</i> frequente (l)	X	$Y = l - X$	Regra
{P: 14mm, U: 70%, T: 25°C}	{P: 14mm}	{U: 70%, T: 25°C}	$\{P: 14mm\} \Rightarrow \{U: 70\%, T: 25^\circ C\}$
	{U: 70%}	{P: 14mm, T: 25°C}	$\{U: 70\%\} \Rightarrow \{P: 14mm, T: 25^\circ C\}$
	{T: 25°C}	{P: 14mm, U: 70%}	$\{T: 25^\circ C\} \Rightarrow \{P: 14mm, U: 70\%\}$

Fonte: Versão elaborada pelo autor.

Como todas as possíveis regras foram criadas, então é necessário identificar o grau de confiança de cada regra. De acordo com a Tabela 24, apenas uma regra foi considerada interessante, sendo: $\{U: 70\%\} \Rightarrow \{P: 14mm, T: 25^\circ C\}$, possuindo grau de confiança maior ou igual ao mínimo α .

Tabela 24 – Calculando o grau de confiança de cada regra de L_3 .

Regra	$(X \cup Y)$	X	$conf(X \rightarrow Y)$	Interessante
$\{P: 14mm\} \Rightarrow \{U: 70\%, T: 25^\circ C\}$	2	5	0.4	-
$\{U: 70\%\} \Rightarrow \{P: 14mm, T: 25^\circ C\}$	2	3	0.66	✓
$\{T: 25^\circ C\} \Rightarrow \{P: 14mm, U: 70\%\}$	2	5	0.4	-

Fonte: Versão elaborada pelo autor.

Finalmente, todas as regras de associação interessantes do problema apresentado foram reveladas. Todas as regras obtidas são apresentadas na Tabela 25.

Tabela 25 – Regras de associação interessantes obtidas.

Regras de associação obtidas
$\{U: 70\%\} \Rightarrow \{P: 14\text{mm}\}$
$\{P: 14\text{mm}\} \Rightarrow \{T: 25^\circ\text{C}\}$
$\{T: 25^\circ\text{C}\} \Rightarrow \{P: 14\text{mm}\}$
$\{U: 70\%\} \Rightarrow \{T: 25^\circ\text{C}\}$
$\{U: 70\%\} \Rightarrow \{P: 14\text{mm}, T: 25^\circ\text{C}\}$

Fonte: Versão elaborada pelo autor.

3.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo foi apresentado detalhadamente o processo de mineração de regras de associação e o algoritmo Apriori, utilizado para auxiliar na exploração por regras de associação confiáveis e representativas. O algoritmo Apriori proporciona baixa complexidade temporal na pesquisa por *varsets* frequentes, por meio do princípio Apriori aplicado na fase de geração de *varsets* candidatos e na fase de poda, diminuindo o espaço de busca.

4 PRÉ-PROCESSAMENTO DOS DADOS

Neste capítulo apresenta-se o processo de pré-processamento, que realiza uma análise inicial nos dados com o objetivo de organizá-los para o processo de mineração de dados. A técnica de discretização utilizada para organizar os dados é então destacada, bem como a técnica de análise descritiva *boxplot*, que expõe as características do conjunto de dados com o objetivo de apresentar suas variações e possíveis anomalias.

4.1 PROCESSO DE DISCRETIZAÇÃO

Discretização é uma etapa do processo de pré-processamento dos dados, que é executada antes da aplicação da técnica de mineração de regras de associação. Através da discretização pode-se representar os valores de uma variável, através de um conjunto de intervalos discretos. Geralmente, é mais fácil entender dados contínuos quando são divididos e armazenados em categorias ou em grupos que transmitam o mesmo significado. Por exemplo, podemos dividir a variável contínua “direção do vento” e armazená-la nos seguintes grupos: entre 0° e 45° (nordeste), entre 46° e 90° (leste), entre 91° e 135° (sudeste), entre 136° e 180° (sul), entre 181° e 225° (sudoeste), entre 226° e 270° (oeste), entre 271° e 315° (noroeste) e entre 316° e 360° (norte).

Consideramos que a discretização é útil se não for evidenciado nenhuma diferença objetiva entre as variáveis que estão na mesma faixa de valores. De acordo com o exemplo anterior as direções 98° e 115° transmitem as mesmas informações, ou seja, que a direção está à sudeste. Sendo assim, a discretização trata-se de uma ferramenta que auxilia no melhor entendimento dos dados e neste trabalho, para gerar as regras de associação de dados meteorológicos, utiliza-se a discretização no processo de pré-processamento dos dados com objetivo de gerar regras mais consistentes e de melhor entendimento.

Quando discretiza-se um conjunto de dados eles são ajustados em faixas de valores, acarretando na redução do impacto de pequenas flutuações nos dados. Mesmo com a discretização é possível evidenciar valores atípicos contidos na base de dados discretizada,

também chamados de ruídos. Esses ruídos são denominados de *outliers*. Um *outlier* é uma observação atípica dentre o conjunto de dados de cada variável do problema, que contém um valor muito afastado da maioria dos dados. (SUN et al., 2017).

De acordo com Ping-guang (2010), a descoberta de valores atípicos evidencia um comportamento inesperado no conjunto de dados, devido à variabilidade na medição ou por indicar erro experimental, podendo ser excluídos do conjunto de dados. Segundo Kohail e El-Halees (2011), *outliers* em dados meteorológicos podem ocorrer devido a problemas na aquisição dos dados e por instrumentos de coleta de dados defeituosos. Podendo também representar mudanças anormais nas condições atmosféricas, através de eventos anormais repentinos, como: furacões e incêndios florestais.

A detecção de valores atípicos pode ser considerada como uma etapa do pré-processamento. Por esta razão, remover os *outliers* da base de dados é necessário para evitar impactos sobre o método de mineração. Portanto, encontrar um comportamento anormal na base de dados, através da pesquisa de *outliers*, tem um significado muito importante ao aplicar a mineração de regras de associação.

4.2 IDENTIFICANDO *OUTLIERS* EM UMA BASE DE DADOS

Na estatística descritiva, o *boxplot* é uma maneira conveniente de representar graficamente grupos de dados numéricos através de seus “quartis”. Os quartis de um conjunto de dados, são os três pontos que dividem o conjunto de dados em quatro grupos iguais, cada grupo compreendendo um quarto dos dados (HUBERT; VANDERVIEREN, 2008). Além dos quartis, é traçado no *boxplot* os valores máximos e mínimos. Desse modo, o *boxplot* é um gráfico construído com base nas seguintes variantes:

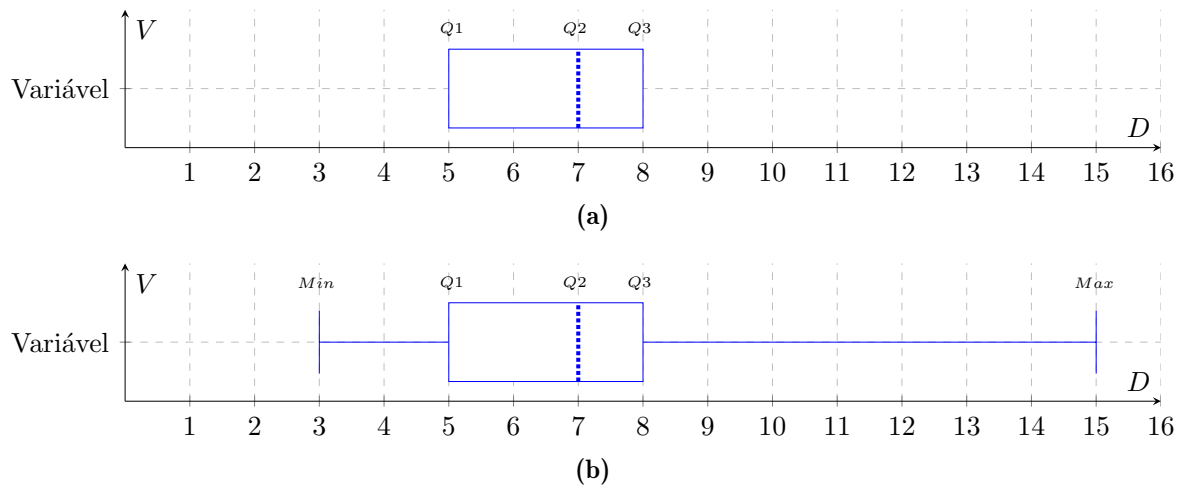
- Valor mínimo;
- Primeiro quartil ($Q1$);
- Segundo quartil ($Q2$), ou mediana;
- Terceiro quartil ($Q3$);
- Valor máximo.

O primeiro quartil ($Q1$), é o número que deixa 25% das observações abaixo e 75% acima, enquanto que o terceiro quartil ($Q3$), deixa 75% das observações abaixo e 25% acima. Já o segundo quartil ($Q2$) é a mediana, deixa 50% das observações abaixo e 50% das observações acima. Os valores máximos e mínimos correspondem aos valores extremos dos dados.

Como apresentado na Figura 22(a), o gráfico é estruturado por uma caixa, construída

paralelamente ao eixo da escada dos dados, podendo ser horizontal ou vertical. Essa caixa estende-se desde o quartil $Q1$ até o quartil $Q3$, e nela traça-se o quartil $Q2$ que é uma linha na posição da mediada dos dados. Essa caixa, que descreve os 50% centrais da distribuição, é comum a todas as variantes do *boxplot* (FARIAS, 2020). Na Figura 22(a) é apresentado um gráfico com escala de dados horizontal, com os seguintes quartis: $Q1 = 5$, $Q2 = 7$ e $Q3 = 8$. Na Figura 22(b) os valores extremos dos dados são traçados, interligando-os através de uma linha paralela à escala dos dados. Nela é apresentado um gráfico contendo as cinco variantes, o valor mínimo (Min) é 3 e o valor máximo (Max) é 15.

Figura 22 – Representação grafica do *boxplot*, com escala de dados horizontal.



Fonte: Versão elaborada pelo autor.

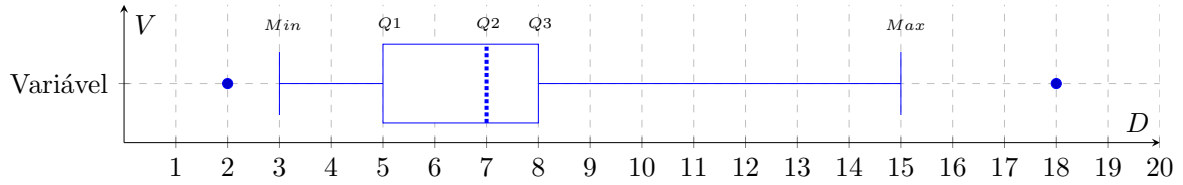
Outra variante muito comum, trata de forma especial os valores atípicos ou ruídos, que são os *outliers* da base de dados. No exemplo exibido na Figura 22(b), todas as observações são maiores ou iguais a 3 e menores ou iguais a 15. Suponha-se que, existe uma observação igual a 2 e outra igual a 18. Então, nesse contexto, 2 e 18 são *outliers*. Para retratar esses valores atípicos, na Figura 23 são ilustradas marcações em suas posições no gráfico.

4.2.1 O cálculo dos quartis

Os quartis são valores que dividem um conjunto de dados em quatro partes, como apresentado na Seção 4.2, cada parte contém o mesmo número de observações. Sendo assim, 25% das observações são menores que o primeiro quartil, 50% são menores que o segundo quartil e 75% são menores que o terceiro quartil. No *boxplot* o conjunto de dados

são dispostos em uma escala, dessa forma, para realizar o cálculo dos quartis os dados terão que ser ordenados.

Figura 23 – Representação grafica do *boxplot* com *outliers*.



Fonte: Versão elaborada pelo autor.

O cálculo se inicia com a mediana, ou segundo quartil (Q_2). A mediana é o valor que divide o conjunto de dados em partes iguais. Considerando o conjunto ordenado A , apresentado na Tabela 26. A mediana divide o conjunto em duas partes iguais, cada parte com 9 observações. Essa divisão é ilustrada na Tabela 27.

Tabela 26 – Exemplo de um conjunto de dados ordenado, para a aplicação do *boxplot*.

Ordem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	3.0	3.5	4.5	5.0	5.0	5.5	6.5	6.5	6.5	7.5	7.6	7.9	8.0	8.0	9.0	9.5	10.0	15.0

Fonte: Versão elaborada pelo autor.

Tabela 27 – Divisão do conjunto de dados em partes iguais.

Ordem	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
A	3.0	3.5	4.5	5.0	5.0	5.5	6.5	6.5	6.5	7.5	7.6	7.9	8.0	8.0	9.0	9.5	10.0	15.0

Fonte: Versão elaborada pelo autor.

A mediana (Q_2) é a média dos dois valores centrais, de acordo com a Equação 3.

$$Q_2 = \frac{6.5 + 7.5}{2} = 7.0 \quad (3)$$

Caso haja um número ímpar de observações, a mediana será o valor central. Considerando o conjunto B apresentado na Tabela 28, a mediana $Q_2 = 25.0$. A mediana divide o conjunto em duas partes iguais, cada parte com 6.

O primeiro e terceiro quartis (Q_1 e Q_3), são calculados através das medianas das duas partes obtidas no cálculo do quartil Q_2 . O quartil Q_1 é a mediana da metade inferior e o quartil Q_3 é a mediana da metade superior. No cálculo dos quartis Q_1 e Q_3

desconsidera-se a mediana $Q2$. Sendo assim, considerando o conjunto A ; cada metade do conjunto tem um número ímpar de observações, totalizando 9 observações. A metade inferior e a superior são ilustradas e divididas separadamente nas Tabelas 29(a) e 29(b), respectivamente. Nesse contexto, a mediana deixará 4 observações abaixo e 4 observações acima. Logo, a mediana de cada uma das partes será a quinta observação. São elas: $Q1 = 5.0$ e $Q3 = 8.0$.

Tabela 28 – Determinando a mediana $Q2$ em um conjunto de dados com um número ímpar de observações.

Ordem	1	2	3	4	5	6	7	8	9	10	11	12	13
B	15.0	17.0	18.0	19.0	19.0	20.0	25.0	26.0	26.0	28.0	30.0	32.0	42.0

Fonte: Versão elaborada pelo autor.

Tabela 29 – Metade superior e inferior do conjunto apresentado na Tabela 27.

(a)										
Ordem	1	2	3	4	5	6	7	8	9	
A	3.0	3.5	4.5	5.0	5.0	5.5	6.5	6.5	6.5	

(b)										
Ordem	10	11	12	13	14	15	16	17	18	
A	7.5	7.6	7.9	8.0	8.0	9.0	9.5	10.0	15.0	

Fonte: Versão elaborada pelo autor.

Considerando o conjunto B , cada metade do conjunto tem 6 observações. A metade inferior e a superior são ilustradas e divididas separadamente nas Tabelas 30(a) e 30(b), respectivamente.. A mediana de cada uma das partes, será a média das terceira e quarta observações. Assim, $Q1$ é a mediana da metade inferior, sendo apresentada pela Equação 4 e $Q3$ é a mediana da metade superior, sendo apresentada pela Equação 5.

$$Q1 = \frac{18.0 + 19.0}{2} = 18.5 \quad (4)$$

$$Q3 = \frac{28.0 + 30.0}{2} = 29.0 \quad (5)$$

Após o cálculo dos quartis o *boxplot* poderá ser pré-formado. Sendo assim, o gráfico obtido pelos dados apresentados, respectivamente, pelos conjuntos A e B , é exposto na Figura 24. Note-se que os valores mínimo e máximo não estão ilustrados no gráfico, sendo definidos após o cálculo dos limites inferior e superior de ambos os conjuntos de dados. Então, na Seção 4.2.2 é apresentado a metodologia para definir esses valores, utilizando

os dados obtidos no cálculo dos quartis. Através dos limites, pode-se definir também os *outliers* dos conjuntos.

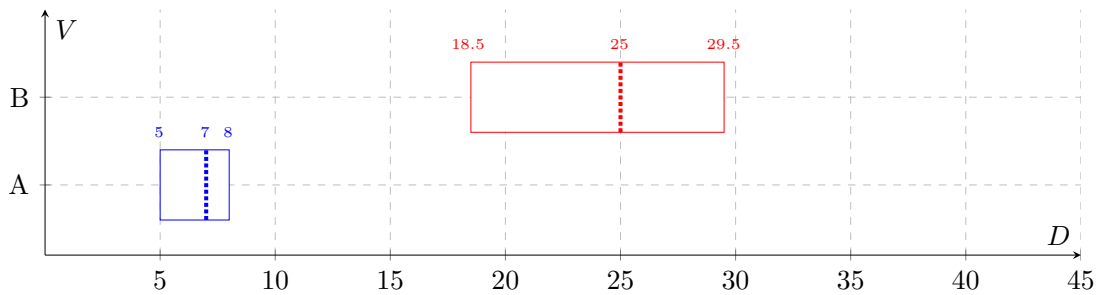
Tabela 30 – Metade superior e inferior do conjunto apresentado na Tabela 28.

(a)						
Ordem	1	2	3	4	5	6
<i>B</i>	15.0	17.0	18.0	19.0	19.0	20.0

(b)						
Ordem	8	9	10	11	12	13
<i>B</i>	26.0	26.0	28.0	30.0	32.0	42.0

Fonte: Versão elaborada pelo autor.

Figura 24 – *Boxplot* pré-formatado a partir dos quartis calculados.



Fonte: Versão elaborada pelo autor.

4.2.2 Determinando os *outliers* da base de dados

Para determinar os *outliers* da base de dados, utiliza-se a amplitude interquartil (*AIQ*), que é a distância entre o primeiro e terceiro quartis. A Equação 6 exibe o cálculo de *AIQ*.

$$AIQ = Q3 - Q1 \quad (6)$$

Note-se que *AIQ* é o comprimento da caixa. Os valores abaixo de $Q1$ ou acima de $Q3$ por mais de $1.5 \times AIQ$, serão considerados valores atípicos. Esses valores, como foram apresentados na Seção 4.2, tem uma representação especial no *boxplot*. Dessa forma, os valores que serão *outliers* deverão estar fora dos limites superior e inferior, definidos pela Equação 7, sendo x um valor atípico.

$$\begin{cases} x < Q1 - 1.5 \times AIQ & \triangleright \text{Limite inferior.} \\ x > Q3 + 1.5 \times AIQ & \triangleright \text{Limite superior.} \end{cases} \quad (7)$$

Os valores que se encontram entre os limites inferior e superior são chamados de valores adjacentes. A representação gráfica se completa traçando uma linha de $Q1$ até o menor valor adjacente, e outra de $Q3$ até o maior valor adjacente. Esse processo exclui-se os *outliers*, respeitando os limites encontrados. Como exemplo, a partir do conjunto A , apresentado na Tabela 26, na Seção 4.2.1, obteve-se os limites inferior e superior ilustrados pela Equação 8.

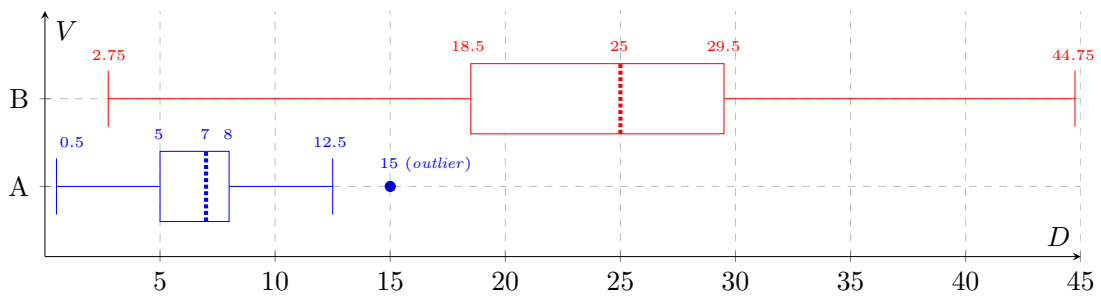
$$\begin{cases} Q1 - 1.5 \times AIQ = 5 - 1.5 \times (8 - 5) = 0.5 & \triangleright \text{Limite inferior.} \\ Q3 + 1.5 \times AIQ = 8 - 1.5 \times (8 - 5) = 12.5 & \triangleright \text{Limite superior.} \end{cases} \quad (8)$$

Em relação ao conjunto B apresentado na Tabela 28, na Seção 4.2.1, obteve-se os limites inferior e superior ilustrados pela Equação 9.

$$\begin{cases} Q1 - 1.5 \times AIQ = 18.5 - 1.5 \times (29 - 18.5) = 2.75 & \triangleright \text{Limite inferior.} \\ Q3 + 1.5 \times AIQ = 29 - 1.5 \times (29 - 18.5) = 44.75 & \triangleright \text{Limite superior.} \end{cases} \quad (9)$$

Finalmente, os limites inferior e superior podem ser representados no gráfico, sendo os valores máximo e mínimo de cada conjunto de dados. Na Figura 25 é apresentado o *boxplot* dos exemplos retratados através dos conjuntos A e B . Foi encontrado apenas um *outlier* no conjunto A , que é o valor 15. Esse valor está acima do limite superior, ou seja, o valor máximo do conjunto no *boxplot*. Sendo assim, é considerado um valor atípico no conjunto de dados. Por outro lado, não foi encontrado nenhum *outlier* no conjunto B . Sendo assim, esse conjunto não contém nenhum valor atípico.

Figura 25 – *Boxplot* dos exemplos apresentados.



Fonte: Versão elaborada pelo autor.

4.3 CONSIDERAÇÕES SOBRE O CAPÍTULO

Neste capítulo foi apresentado a etapa de pré-processamento dos dados por meio da discretização e da técnica de análise descritiva *boxplot*. Com a discretização dos dados, as regras geradas na etapa de mineração de regras de associação são mais consistentes e de melhor entendimento. E com a utilização do *boxplot* podemos analisar a variação dos dados e possíveis ruídos no conjunto de dados, evitando impactos sobre o método de mineração de regras de associação.

5 REDE DE TRANSDUTORES DESENVOLVIDA

A rede *mesh* desenvolvida tem como finalidade detectar parâmetros ambientais, como, velocidade e direção do vento, temperatura, umidade, pressão barométrica e precipitação. A rede possui dois tipos de dispositivos, o NCAP e o WTIM, de acordo com o padrão IEEE 1451.5. Os WTIMs foram desenvolvidos utilizando o módulo de rádio XBee PRO S2C, sensores e um microcontrolador.

Para realizar a leitura dos dados dos sensores (temperatura, umidade, pressão barométrica e precipitação) pelo microcontrolador, utilizou-se a interface de comunicação *Inter-Integrated Circuit* (I²C), a qual transmite os dados de forma serial para o dispositivo de aquisição. Após finalizar a aquisição e processamento, os dados obtidos pelos sensores são encapsulados em *frames* ZigBee, para serem enviados ao nó de rede NCAP, de acordo com as diretrizes de transmissão e recepção de dados dos padrões ZigBee e IEEE 1451.5.

5.1 O DISPOSITIVO NCAP

O NCAP foi desenvolvido utilizando um módulo de rádio XBee PRO S2C e uma placa de desenvolvimento ASUS Tinker Board. Ao receber os dados, o NCAP realiza o armazenamento das informações em um cartão de memória. O NCAP desenvolvido tem o objetivo de criar uma rede autônoma e expansível, que possa realizar a coleta de dados por um longo período de tempo. A funcionalidade do NCAP inclui a descoberta automática dos WTIMs, o gerenciamento de leituras dos sensores por meio da comunicação autônoma entre os WTIMs em intervalos de tempo pré-definidos, o armazenamento dos dados e também a transmissão dos dados para uma rede externa.

Na Tabela 31 apresenta-se a arquitetura em camadas do NCAP, ilustrando as interfaces de comunicação e serviços usados neste projeto. O NCAP também pode ser conectado a *World Wide Web* (WEB) através de uma interface externa Ethernet. Portanto, o desenvolvimento do NCAP foi dividido em dois níveis: *hardware/interfaces* e *software*.

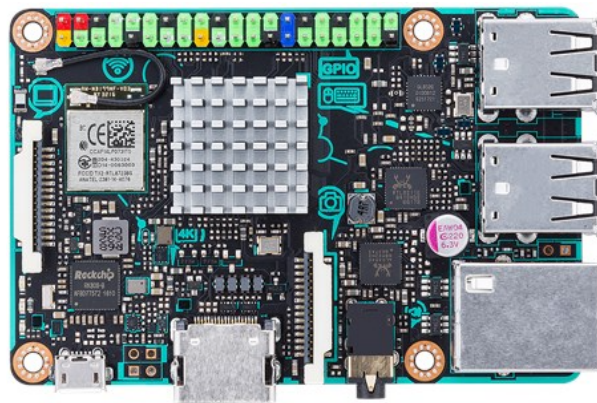
Tabela 31 – Arquitetura baseada em camadas.

Protocolo de gerenciamento do NCAP	Editor de Texto	JSON	
Servidor Apache	MySQL	Aplicações	PHP
Sistema Operacional			
Ethernet IEEE 802.3	Hardware	Interface UART	ZigBee IEEE 802.15.4

Fonte: Versão elaborada pelo autor.

5.1.1 O hardware e interfaces

O NCAP foi desenvolvido usando a plataforma *Single Board Computer* da ASUS, Tinker Board, lançada no início de 2017. Na Figura 26 é apresentado o Tinker Board.

Figura 26 – ASUS Tinker Board.

Fonte: ASUS (2020).

As dimensões e a *General Purpose Input/Output* (GPIO) foram projetadas para serem compatíveis com os modelos de segunda geração e da plataforma Raspberry Pi. O Tinker board utilizado possui um processador quad-core baseado em ARM, o Rockchip RK3288, operando a 1,8 GHz com 2 GB de memória *Random Access Memory* (RAM) DDR3. Também está equipado com uma interface SD 3.0 que oferece velocidades de leitura e gravação significativamente mais rápidas para cartões microSD expansíveis usados para o sistema operacional, aplicativos e armazenamento de arquivos. Outras interfaces inclusas no Tinker Board, são:

1. Processador Gráfico Integrado ARM® Mali™-T764 GPU;
2. Gigabit Ethernet;
3. Rede sem fio 802.11 b/g/n;
4. Bluetooth V4.0 + EDR;

5. Áudio RTL ALC4040 CODEC;
6. 4 portas USB 2.0;
7. 28 portas GPIO;
8. 2 portas *Serial Peripheral Interface* (SPI);
9. 2 portas *Inter-Integrated Circuit* (I²C);
10. 4 portas *Universal Asynchrounous Receiver/Transmitter* (UART);
11. 2 portas *Pulse Width Modulation* (PWM);
12. 1 porta *Pulse-code modulation/Inter-IC Sound* (PCM/I2S).

O módulo transceptor utilizado foi o XBee Pro S2C, ilustrado na Figura 27, é fabricado pela Digi International. O módulo contém 2,4 GHz de banda de frequência e taxa de transmissão de 250 Kb/s, podendo atingir um alcance de até 3200 m de distância de comunicação de acordo com a especificação do fabricante.

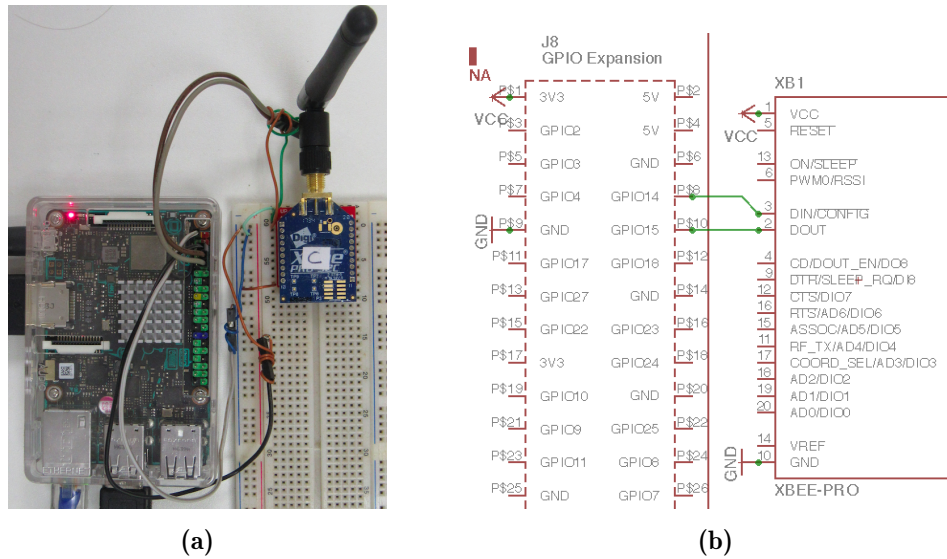
Figura 27 – Transceptor XBee Pro S2C utilizado neste projeto.



Fonte: XBEE STORE (2020).

Os módulos XBee Pro S2C são compatíveis com outros dispositivos fabricados pela Digi International que utilizam a tecnologia XBee ZigBee, incluindo gateways, adaptadores, extensores de rede, sensores, entre outros. Então, os módulos XBee e XBee-PRO ZigBee, da série S2C, são compatíveis pino-a-pino com os módulos da série S2. Nesse trabalho, os nós NCAP e WTIMs utilizam módulos XBee Pro S2C, pois esta versão incorpora diversas melhorias em relação às versões anteriores, como a série S2 e S2B. Um comparativo entre as versões dos módulos Xbee pode ser obtido em Digi (2020).

Na Figura 28(a) apresenta-se o nó NCAP desenvolvido em fase de prototipagem e a Figura 28(b) o circuito.

Figura 28 – Prototipagem e esquema elétrico do NCAP.

Fonte: Versão elaborada pelo autor.

De acordo com a Figura 28(b), o Tinker Board e o módulo transceptor se comunicam diretamente através da interface UART, devido ao Tinker Board possuir saída UART com o mesmo nível de tensão do módulo XBee Pro S2C. Neste caso não houve a necessidade de desenvolvimento de um divisor de tensão, pois, ambos trabalham em 3,3 V.

5.1.2 O software

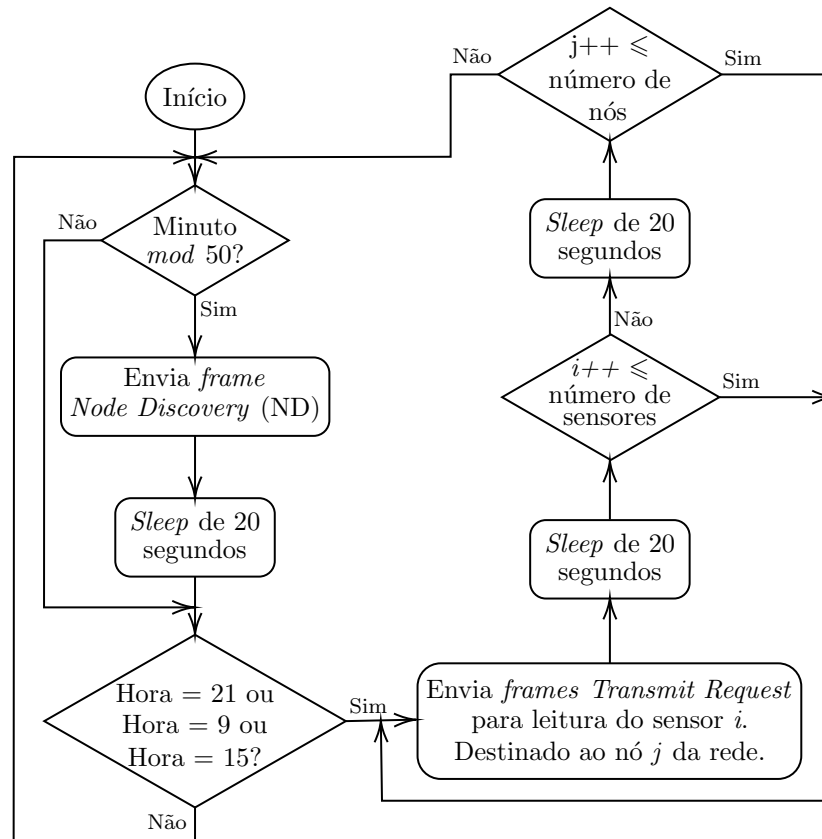
Para o desenvolvimento do NCAP foi instalado o sistema operacional TinkerOS-Debian, que é uma versão específica do Debian neste dispositivo, disponibilizada no próprio site do fabricante. Em seguida, foi instalado um conjunto de ferramentas para o ambiente de desenvolvimento, contendo: apache server, PHP 7 e o servidor de banco de dados MySQL.

O programa do NCAP foi dividido em duas aplicações desenvolvidas com o uso da Linguagem C. A primeira aplicação, denominada “NCAP *Monitor*”, monitora a entrada de dados da interface UART a procura de *frames* ZigBee, enviados pelos WTIMs. A segunda aplicação, denominada “NCAP *Request*”, envia *frames* de requisição para os WTIMs que fazem parte da rede, em intervalos de tempos pré-definidos. As aplicações NCAP *Request* e NCAP *Monitor* são apresentadas nos fluxogramas, Figuras 29 e 31, respectivamente.

Na concepção da rede, o NCAP não têm conhecimento de quais WTIMs estão ativos. O reconhecimento dos dispositivos é realizado fazendo o envio, em *broadcast*, de um *frame*

ZigBee com a função *Node Discovery*, através da aplicação *NCAP Request*. Desta forma, todos os nós ativos irão enviar um *frame* de resposta, indicando ao coordenador que encontra-se ativo.

Figura 29 – Fluxograma da aplicação *NCAP Request*.



Fonte: Versão elaborada pelo autor.

As respostas de cada WTIM, referente ao *Node Discovery* são obtidas através da aplicação *NCAP Monitor*, armazenando em um arquivo de texto o nome do dispositivo transceptor e seu respectivo MAC ID. Na Figura 30 apresenta-se como os dados dos dispositivos são armazenados.

Figura 30 – Exemplo: Informações dos WTIMs armazenadas no NCAP.

```

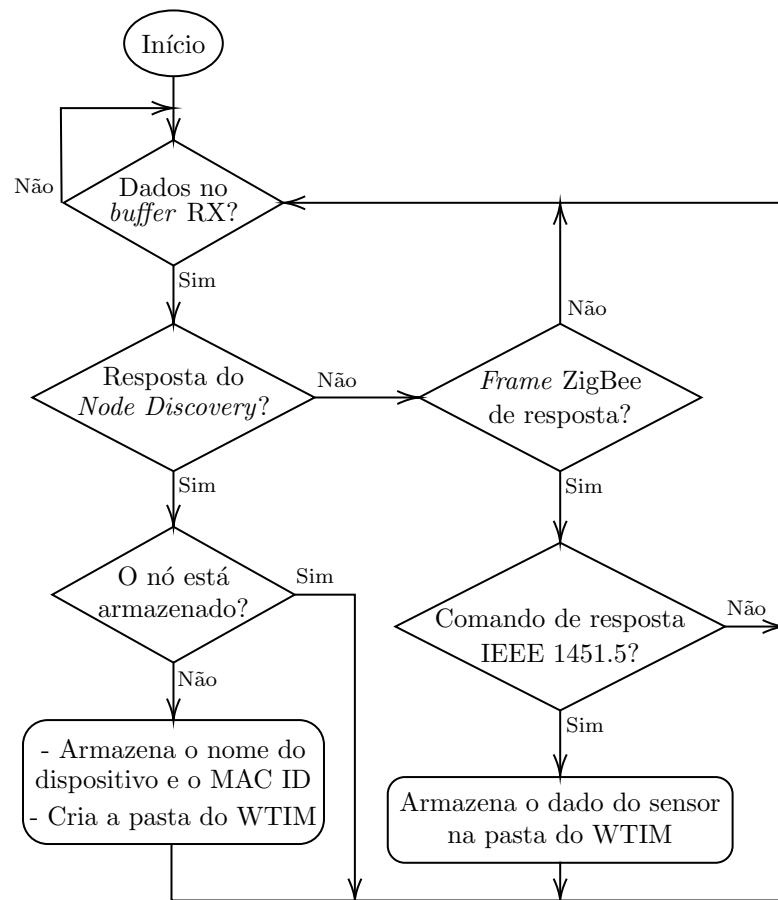
Roteador-1 0 13 A2 0 41 76 61 66
Roteador-2 0 13 A2 0 41 86 31 64
  
```

Fonte: Versão elaborada pelo autor.

Ao obter o MAC ID de cada dispositivo ativo, a aplicação *NCAP Request* solicita a leitura dos sensores instalados em cada WTIM. Essa solicitação está condicionada a períodos de tempo pré-definidos na aplicação. Nesse trabalho, os períodos de leitura, são: 9h, 15h e 21h.

Na estrutura de arquivos do NCAP, cada dispositivo WTIM contém um diretório, que é criado automaticamente no momento do reconhecimento de cada nó. O diretório armazena planilhas no formato *comma-separated values* (.csv), com os dados dos sensores. O nome da planilha contém o endereço MAC ID do WTIM e o canal do sensor, por exemplo: 013A2041766166-canal-7.csv. Cada sensor contém sua respectiva planilha. O processo de criação dos diretórios e arquivos é gerenciado pela aplicação *NCAP Monitor*, ilustrado no fluxograma da Figura 31.

Figura 31 – Fluxograma da aplicação *NCAP Monitor*.



Fonte: Versão elaborada pelo autor.

5.2 OS DISPOSITIVOS WTIMS

Os dispositivos WTIM, também denominados de módulos de aquisição de dados, são constituídos por vários componentes, como, transdutores, unidade de processamento, transceptor sem fio e uma fonte de alimentação.

5.2.1 Hardware e sensores acoplados

A arquitetura dos WTIMs (Apêndice B) foi desenvolvida com o auxílio da ferramenta *EAGLE PCB*. O *hardware* é constituído por um microcontrolador Atmega8 de 8 *bits*, fabricado pela Atmel, que é baseado na arquitetura *Reduced Instruction Set Computer* (RISC), com desempenho de 16 *Million Instructions Per Second* (MIPS), possuindo 8KB de memória *flash* e 1KB de memória *Static Random-Access Memory* (SRAM).

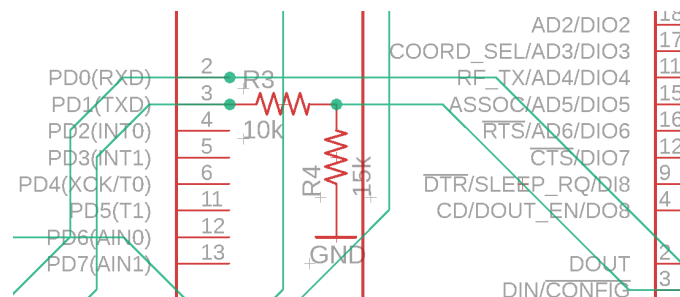
Em termos de periféricos, o microcontrolador possui 6 canais *Analog-to-Digital Converter* (ADC). Porém, neste projeto foram utilizados 4 canais ADC, 2 canais digitais e a interface I²C.

O circuito de condicionamento de sinal foi projetado fora do âmbito do módulo de sensor sem fio, sendo que cada transdutor pode possuir uma característica específica, da mesma forma o seu condicionamento de sinal. Assim, os módulos WTIMs foram elaborados de forma genérica ao transdutor.

O módulo transceptor utilizado nos WTIMs foi o XBee Pro S2C, fabricado pela *Digi International*, que opera na faixa de frequência de 2.4 GHz, com uma taxa de transmissão de 250 Kbps, com um alcance de até 3200 m, dependendo do ambiente aplicado.

O microcontrolador e o módulo transceptor se comunicam através de suas interfaces UART, e ambos trabalham com tensão diferente para a comunicação UART. O microcontrolador trabalha com 5 V e o módulo transceptor com 3.3 V. Sendo assim, os sinais enviados através do microcontrolador foram condicionados utilizando um divisor de tensão, ou divisor resistivo, ilustrado na Figura 32. No pino Tx do microcontrolador foi utilizado um resistor de 10 K Ω e outro de 15 K Ω . Sendo assim, todos os sinais recebidos no módulo XBee Pro S2C serão de 3.3V.

Figura 32 – Divisor de tensão.

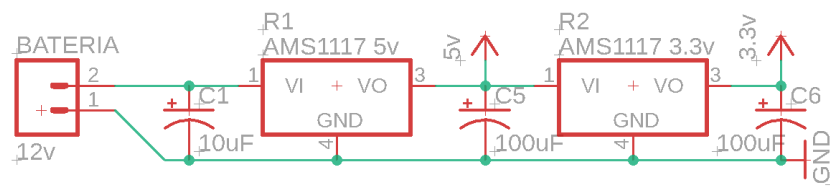


Fonte: Versão elaborada pelo autor.

Para a alimentação dos dispositivos foram utilizados 2 Circuitos Integrados (CIs)

reguladores de tensão, que tem por finalidade manter a tensão do circuito contínua, mesmo havendo variações na tensão da fonte de alimentação (bateria), ou na corrente inerente ao circuito, Figura 33. Foi utilizado capacitores que eliminam ruídos de RF e dão maior estabilidade na tensão. O microcontrolador dispôs do CI (Circuito Integrado) AMS1117 H433KE (5 V de saída) e o módulo transceptor do CI AMS1117 H433PE (3.3 V de saída).

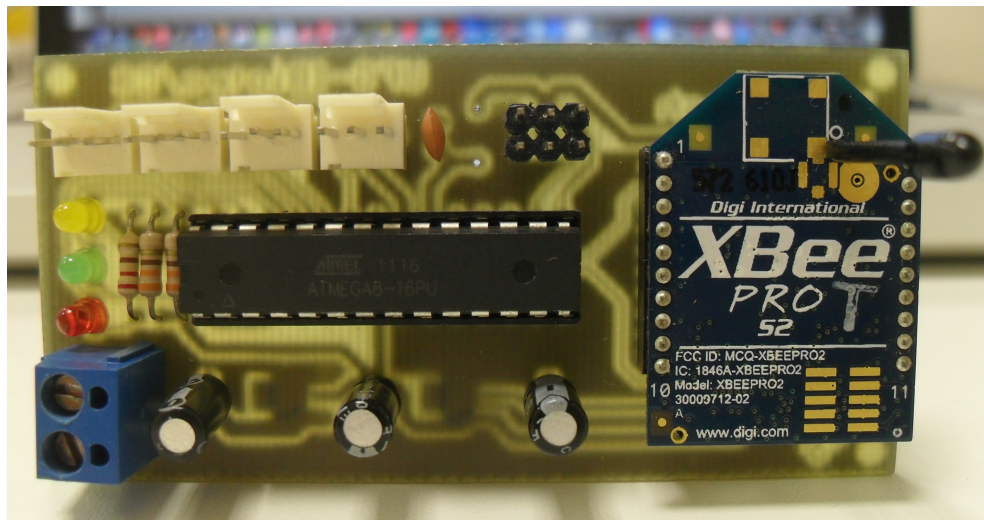
Figura 33 – Regulador de tensão CI AMS1117 H433KE (5V de saída).



Fonte: Versão elaborada pelo autor.

A plataforma de *hardware* foi confeccionada através do processo fotográfico. O resultado obtido após o processo de montagem do circuito é apresentado na Figura 34.

Figura 34 – WTIM desenvolvido em laboratório



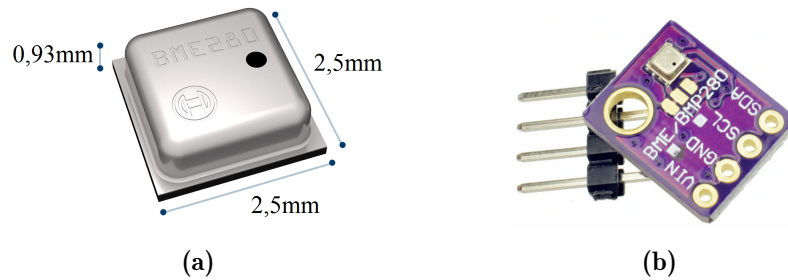
Fonte: Versão elaborada pelo autor.

Para a concepção da rede, o sensor utilizado nos módulos WTIMs é o BME280 produzido pela Bosh, apresentado na Figura 35(a). O BME280 é um sensor ambiental desenvolvido especificamente para aplicações móveis em que tamanho e baixo consumo de energia são as principais restrições. O sensor combina características individuais de alta linearidade e alta precisão para pressão, umidade e temperatura em um único pacote *Land Grid Array* (LGA¹).

¹Um pacote LGA é uma “embalagem” que pode integrar sistemas inteiros - incluindo componentes

O sensor BME280 provê as interfaces digitais SPI e I²C. Neste trabalho, utiliza-se a interface I²C para realizar a conexão entre o sensor e o WTIM. Sendo assim, o sensor BME280 é atribuído a uma placa PCB com uma interface I²C, como ilustrado na Figura 35(b).

Figura 35 – Sensor BME280: temperatura, umidade e pressão barométrica.



Fonte: Mouser (2020), Pomykal (2020).

O sensor de umidade possui um rápido tempo de resposta, que suporta requisitos de desempenho para aplicações emergentes, como reconhecimento de contexto e alta precisão em uma ampla faixa de temperatura. O sensor de pressão barométrica absoluta contém recursos de precisão e alta resolução com ruído muito baixo. O sensor de temperatura integrado é otimizado para obter ruído muito baixo e alta resolução. É usado principalmente para a compensação da temperatura dos sensores de pressão e umidade, e também pode ser usado para estimar a temperatura ambiente.

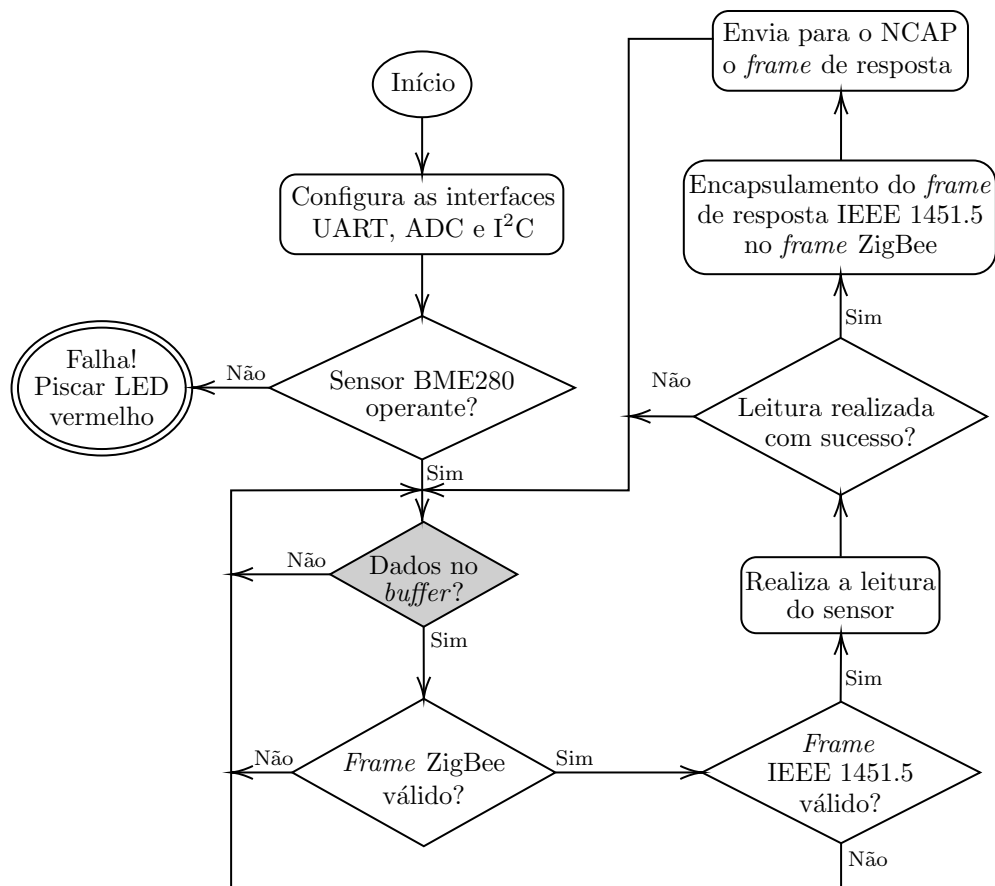
5.2.2 Software

O *software* dos módulos WTIMs foi desenvolvido com o uso da linguagem C embarcada, fornecendo suporte a 7 sensores. Sendo que, 4 sensores são analógicos conectados na porta ADC do microcontrolador, 2 sensores digitais e 1 sensor digital conectado especificamente na interface I²C. Todas as funções, de aquisição de dados e comunicação, é controlada pelo NCAP. A execução do *software* é ilustrada através do fluxograma da Figura 36.

Neste fluxograma, ao ativar o WTIM é efetuada a inicialização das interfaces UART, ADC e I²C. Logo após, é verificado se o sensor BME280 está respondendo normalmente, caso contrário as funções do WTIM ficarão inativas. Após o sucesso da verificação do sensor as funções de comunicação e aquisição de dados ficarão aguardando alguma passivos, ICs e sensores.

requisição do NCAP. As requisições e respostas são feitas através de *frames* ZigBee. Desta forma, *frames* baseados no padrão IEEE 1451.5 são encapsulados em *frames* ZigBee e transmitidos através da rede. Quando o NCAP envia um *frame* de requisição (*request*) de leitura de um sensor ao WTIM, o *frame* baseado no padrão IEEE 1451 é desencapsulado deste *frame* ZigBee. Após o desencapsulamento, é feita a identificação do comando para realizar a leitura do sensor desejado. O dado do sensor é então inserido em um *frame* de resposta (*reply*) baseado no padrão IEEE 1451, sendo encapsulado em um *frame* ZigBee e transmitido para o NCAP.

Figura 36 – Fluxograma da execução do *software* dos módulos WTIMs.

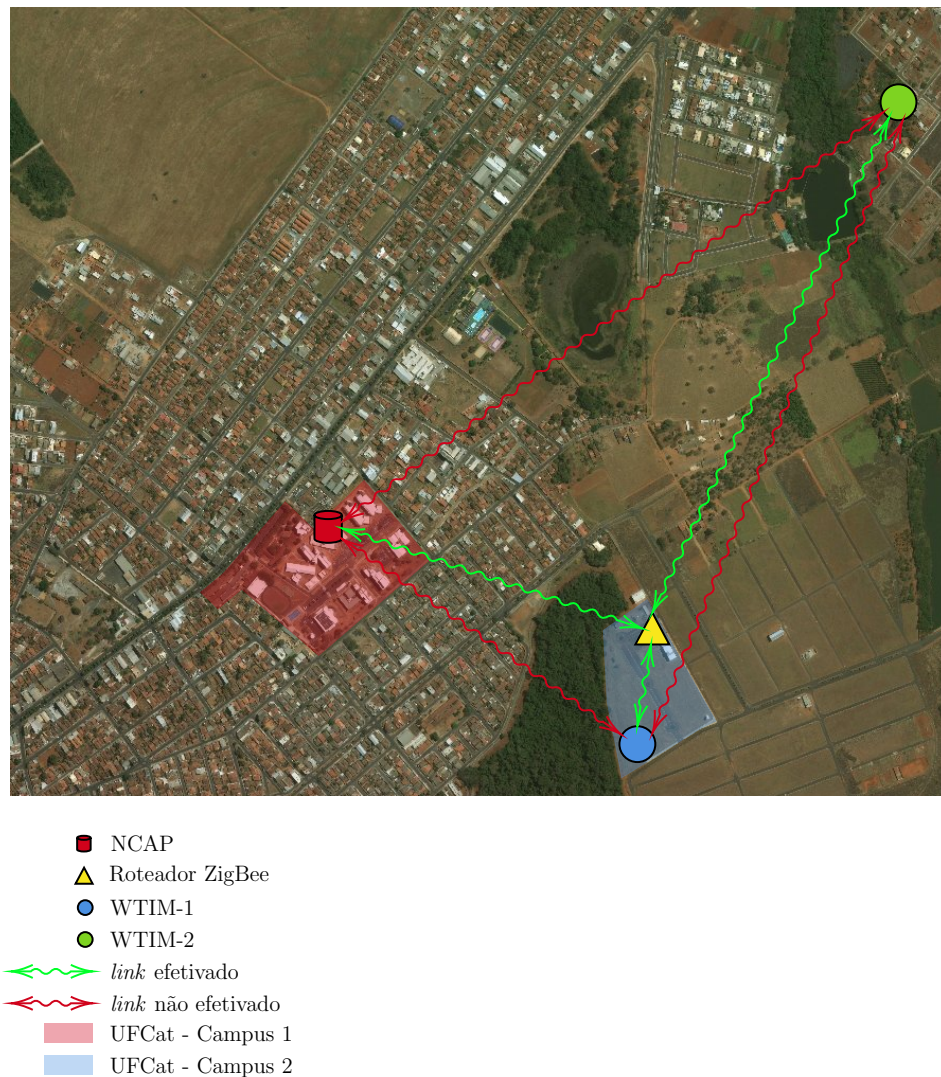


Fonte: Versão elaborada pelo autor.

5.3 APLICAÇÃO DA REDE DE SENSORES EM LOCAL URBANO

O local escolhido para realizar a implantação da rede foi em um ambiente urbano, localizado nas proximidades dos Câmpus 1 e 2 da Universidade Federal de Catalão (UFCat). Esse ambiente e a topologia atual da rede são apresentados na Figura 37, e esta topologia está em fase de expansão. Existem em funcionamento o NCAP, dois WTIMs e um dispositivo chamado Roteador ZigBee.

Figura 37 – Topologia atual da rede desenvolvida.



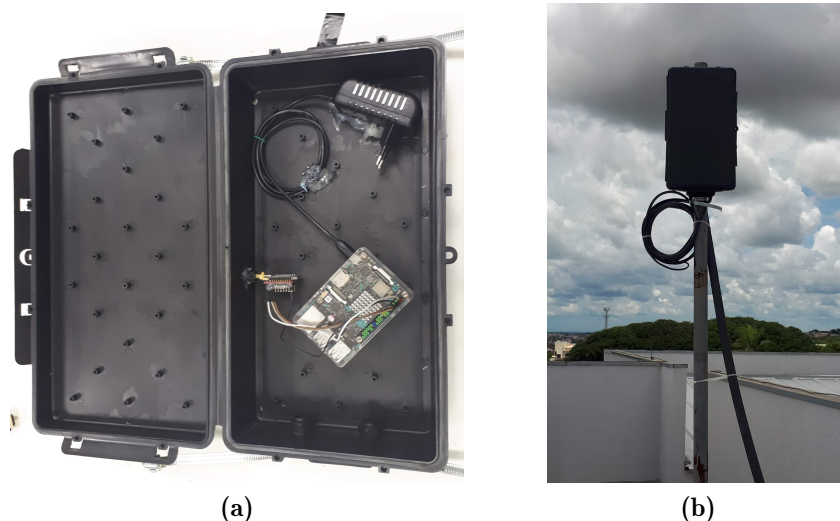
Fonte: Versão elaborada pelo autor.

Por ser o *gateway* da rede, o NCAP foi o primeiro dispositivo a ser instalado, pois a rede ZigBee depende do nó coordenador ativo para inicializar a rede com os outros dispositivos. O NCAP está situado aproximadamente a 16 metros de altura em um prédio administrativo do Campus 1 da UFCat. Seu *hardware* foi alojado em uma caixa hermeticamente fechada à prova de intempéries, como apresentado na Figura 38(a). Na Figura 38(b) apresenta-se o NCAP já instalado no prédio.

Além da instalação do *hardware* do NCAP no local desejado, foi instalado o circuito elétrico para alimentação do sistema e o ponto de acesso para a rede Ethernet da universidade. Para evitar a suspensão do NCAP, por queda no fornecimento de energia, foi instalado um nobreak que é um sistema secundário de energia elétrica.

O acesso ao NCAP deve ser feito através da internet, via conexão *Secure Shell* (SSH), via *web browser* através do IP: 200.137.238.25 ou pelo domínio: meioambiente.catalao.ufg.br. Através da conexão SSH pode-se configurar o Tinker Board do NCAP, executar as aplicações NCAP Monitor e NCAP Request, e coletar os dados obtidos pela rede.

Figura 38 – Fases de instalação do NCAP.



Fonte: Versão elaborada pelo autor.

O NCAP não estabeleceu um link de comunicação direto com o WTIM-1 que está localizado no Campus 2 da UFCat, sendo assim, houve a necessidade da instalação do Roteador ZigBee para realizar a propagação das mensagens na rede. Na Figura 39 mostra-se algumas características do relevo entre os nós, inclusive a principal barreira que impede a comunicação. A Figura 39 foi criada com dados obtidos pelo Google Earth.

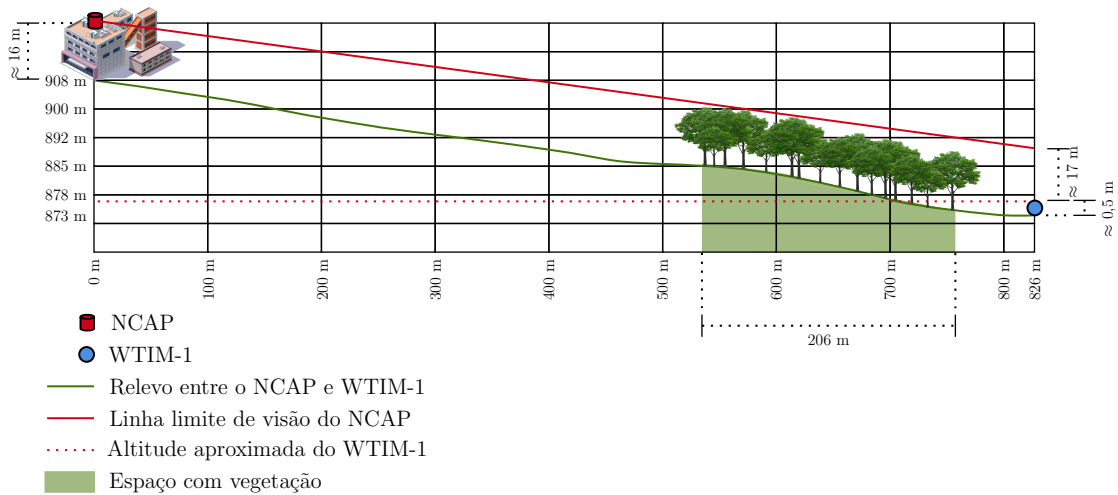
O WTIM-1 está localizado aproximadamente 826 metros de distância do NCAP com 50 centímetros de altura. Entre os dois dispositivos contém uma reserva legal de cerradão², ou catanduva, abrigando árvores com cerca de 15 metros de altura em uma faixa de 148 metros. A vegetação nesse local dificultou a construção de uma torre para a instalação do WTIM-1, ficando viável financeiramente e construtivamente a instalação do Roteador ZigBee para conectar ambos os dispositivos.

O WTIM-1 é ilustrado na Figura 40, o *hardware* foi instalado em uma caixa hermeticamente fechada e o sensor BME280 foi instalado separadamente em um alojamento especial do lado de fora da caixa. O alojamento fornece a proteção necessária

²Cerradão, ou catanduva, é a denominação de uma formação florestal do cerrado, com árvores podendo alcançar até 15 metros de altura.

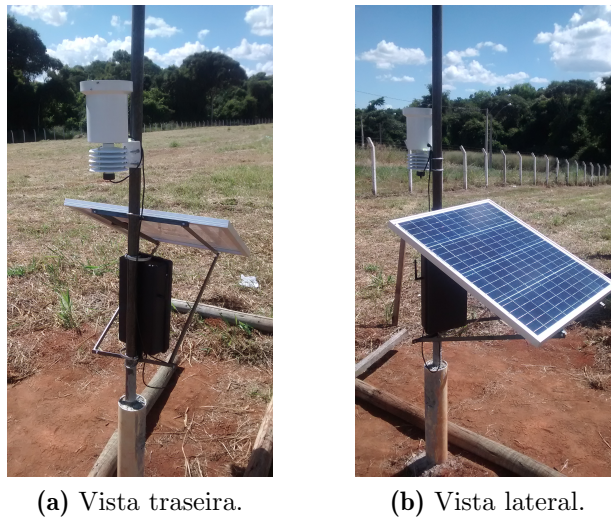
para que o sensor execute sua função de modo que não atrapalhe as medições do ambiente externo.

Figura 39 – Relevo entre o NCAP e WTIM-1.



Fonte: Versão elaborada pelo autor.

Figura 40 – Instalação do WTIM-1.



Fonte: Versão elaborada pelo autor.

Para alimentação elétrica do sistema foi utilizado uma bateria de 12v de 9A, sendo recarregada por meio de painel fotovoltaico de 90W. Um controlador de carga foi utilizado para administrar a corrente elétrica que é adicionada ou extraída da bateria. Isso evita sobrecargas e pode proteger contra sobretensão, o que pode reduzir o desempenho da bateria ou sua expectativa de vida.

O Roteador ZigBee é ilustrado na Figura 41, o dispositivo não é um WTIM, pois

tem apenas um módulo transceptor XBee Pro S2C e uma fonte de alimentação em seu *hardware*. O módulo foi instalado aproximadamente a 16 metros de altura no prédio do Campus 2 da UFCat e sua função é expandir a rede criada, roteando as informações dos nós adjacentes para o destino especificado no *frame* ZigBee.

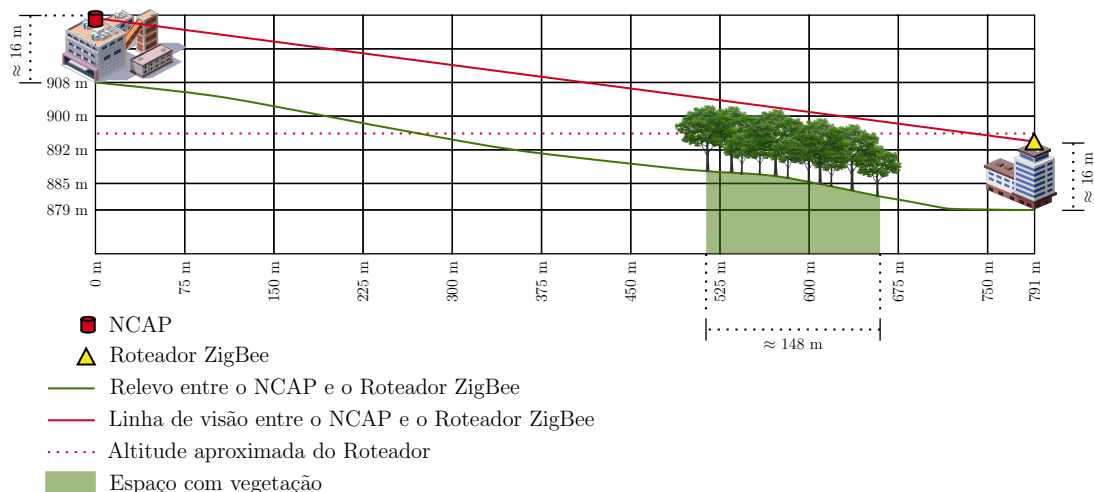
Figura 41 – Roteador ZigBee instalado no Campus 2 da UFCat.



Fonte: Versão elaborada pelo autor.

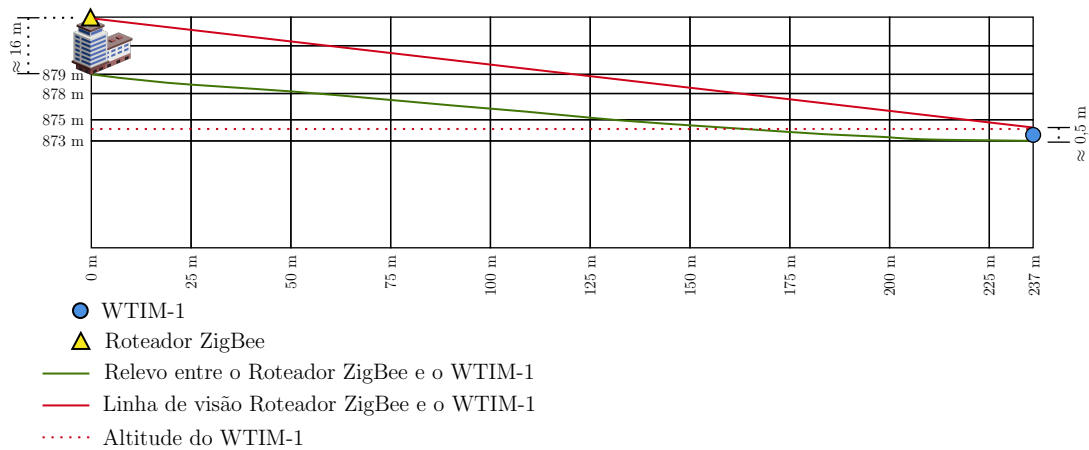
O Roteador ZigBee está localizado aproximadamente 791 metros de distância do NCAP e 237 metros de distância do WTIM-1. Na Figura 42 são mostradas as características do relevo entre o NCAP e o Roteador ZigBee. Devido a elevação deste dispositivo, foi possível transpor a barreira de vegetação, obtendo um link de comunicação satisfatório com o NCAP.

Figura 42 – Relevo entre o NCAP e Roteador ZigBee.



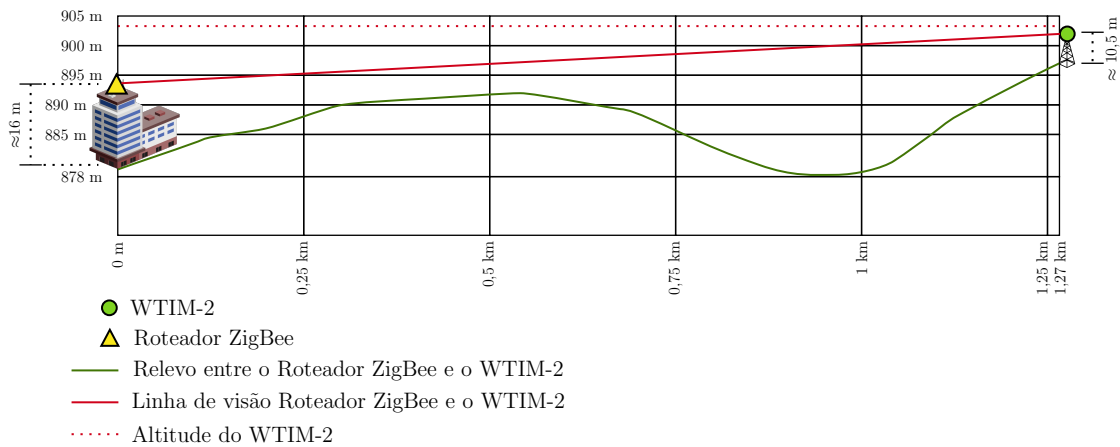
Fonte: Versão elaborada pelo autor.

Sem barreira e com uma distância relativamente pequena, a comunicação direta entre o Roteador ZigBee e o WTIM-1 foi obtida. Na Figura 43 apresenta-se a descrição do relevo e a disposição de ambos os dispositivos.

Figura 43 – Relevo entre o Roteador ZigBee e WTIM-1.

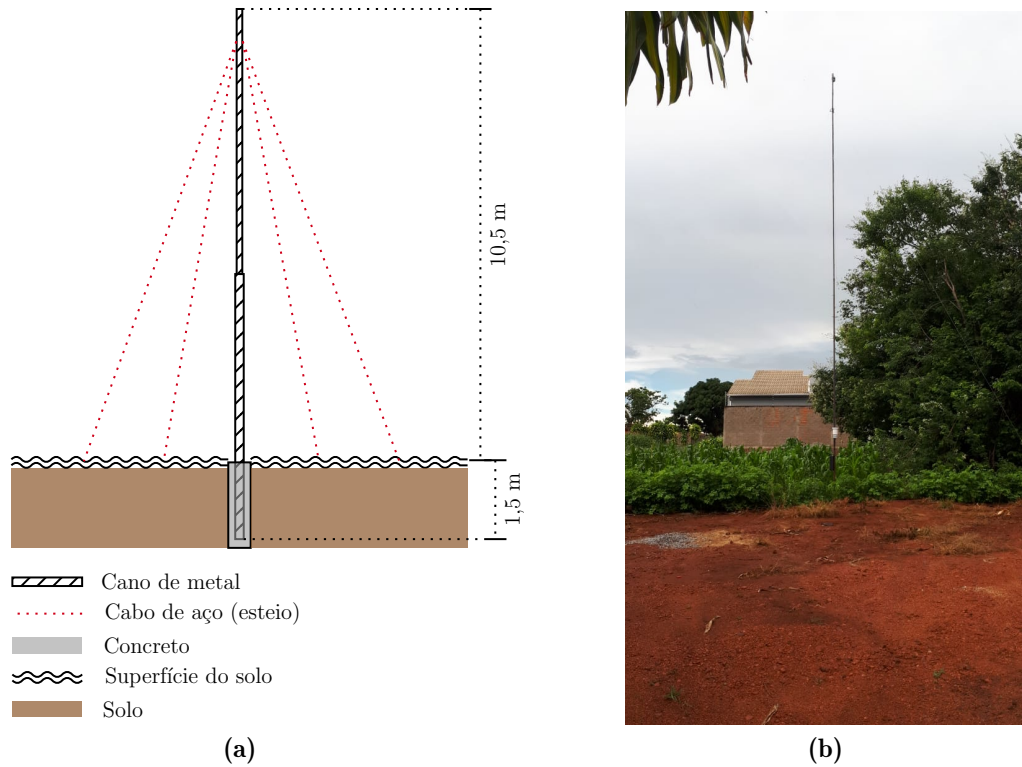
Fonte: Versão elaborada pelo autor.

O fato de localizar-se em altitude elevada, com o Roteador ZigBee foi possível expandir a rede para outras localidades. Sendo assim, foi estabelecido o WTIM-2, o segundo nó WTIM da rede e o quarto nó instalado. Na Figura 44 é ilustrado o relevo entre o Roteador ZigBee e o WTIM-2.

Figura 44 – Relevo entre o Roteador ZigBee e WTIM-2.

Fonte: Versão elaborada pelo autor.

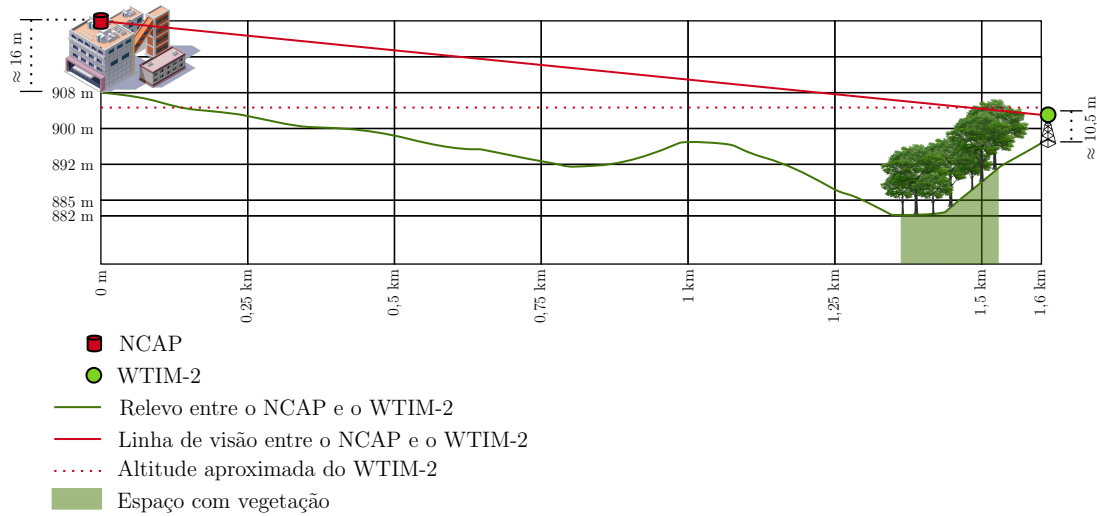
Para transpor algumas barreiras que impediam a comunicação entre o Roteador ZigBee e o WTIM-2, foi necessário a construção de uma torre estaiada para a instalação do WTIM-2. O projeto da torre é ilustrado na Figura 45(a). A torre contém aproximadamente 10,5 metros de altura e está localizada em uma área particular com 1,27 km de distância do Campus 2 da UFCat, onde foi instalado o Roteador ZigBee. A instalação final da torre é apresentada na Figura 45(b).

Figura 45 – Características de instalação do WTIM-2.

Fonte: Versão elaborada pelo autor.

Assim como no WTIM-1, o *hardware* do WTIM-2 foi acomodado em uma caixa hermeticamente fechada e o sensor BME280 foi instalado separadamente em um alojamento especial do lado de fora da caixa. Porém, o módulo transceptor do WTIM-2 foi instalado fora da caixa na parte mais alta da torre, deixando apenas o circuito na parte de baixo da torre. Essa forma de instalação facilita possíveis manutenções no equipamento desenvolvido, como por exemplo, em atualizações de *software* e na adição de novos sensores.

O WTIM-2 não obteve comunicação direta com o NCAP. Na Figura 46 podemos observar que a distância entre os dispositivos é satisfatória, porém, obstáculos como árvores e edificações atrapalham a comunicação.

Figura 46 – Relevo entre o NCAP e WTIM-2.

Fonte: Versão elaborada pelo autor.

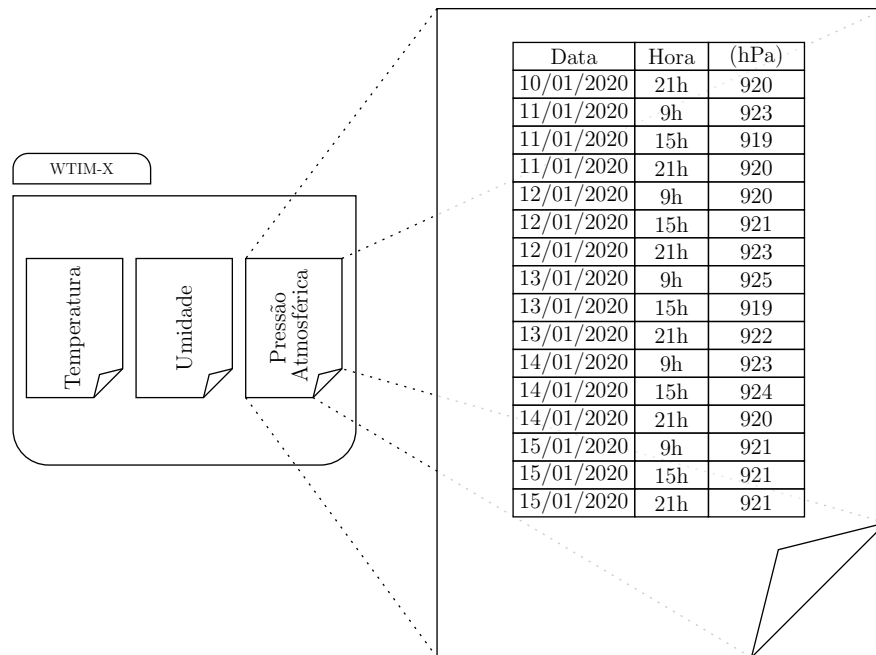
5.4 COLETA DOS DADOS A SEREM ANALISADOS

A coleta dos dados foi realizada por meio de horários pré-definidos no *software* NCAP *Request*. Neste trabalho, o padrão de leitura dos dados dos sensores foi definido de acordo com o Instituto Nacional de Meteorologia (INMET), sendo assim, as requisições foram realizadas às 21h, 9h e 15h.

Cada WTIM possui uma pasta na estrutura de arquivos do NCAP para o armazenamento dos dados, e cada sensor contém um arquivo específico nesta pasta. Na Figura 47 é apresentada a estrutura de arquivos de forma genérica. O arquivo é uma planilha eletrônica do tipo .csv que possui três colunas, sendo que na primeira e na segunda coluna contém, respectivamente, a data e a hora atuais e a terceira coluna está destinada para o armazenamento dos dados do sensor.

Após um período de monitoramento, aplica-se a técnica de mineração de regras de associação estudada, para determinar associações entre as variáveis climáticas. Os dados de todos os WTIMs foram analisados para gerar regras no âmbito geral da rede.

Figura 47 – Estrutura de arquivos do NCAP para o armazenamento dos dados.



Fonte: Versão elaborada pelo autor.

5.5 CONSIDERAÇÕES SOBRE O CAPÍTULO

Neste capítulo foi abordada as características da rede *mesh* de transdutores inteligentes desenvolvida, baseada nos padrões ZigBee e IEEE 1451.5. Assim, é importante destacar que a rede *mesh* desenvolvida é formada por nós WTIMs e NCAP, foi salientado o desenvolvimento do *hardware* destes dispositivos, bem como uma abordagem sucinta da comunicação dos nós no âmbito da rede *mesh* ZigBee de acordo com o interfaceamento definido pela norma IEEE 1451.5. A rede está sendo aplicada em ambiente urbano e está em fase de expansão, sua topologia atual contém 4 nós. Foram abordadas algumas características sobre o local de implantação de cada nó, bem como a distância e o relevo entre os nós.

6 APLICAÇÃO DE MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO PARA A BASE DA REDE *MESH* DE TRANSDUTORES INTELIGENTES DESENVOLVIDA

No presente capítulo apresenta-se a aplicação de mineração de regras de associação nos dados da rede *mesh* de transdutores inteligentes desenvolvida. Os dados da rede foram submetidos a algumas situações de teste. Entretanto, os dados de cada WTIM foram analisados separadamente. É importante salientar que não faz parte do objetivo deste experimento a interpretação das informações resultantes do processo de descoberta de conhecimento.

6.1 A BASE DE DADOS OBTIDA PELOS WTIMS

Uma amostra da base de dados utilizada neste experimento é apresentada na Tabela 32, sendo obtida através do monitoramento realizado pelos WTIMS (WTIM-1 e WTIM-2) da rede de transdutores inteligentes desenvolvida.

As leituras referentes a temperatura instantânea (t), umidade reativa do ar (u) e pressão atmosférica (ps) compõe as variáveis de observação tomadas diariamente. Assim, para cada variável são realizadas 3 leituras diárias solicitadas pelo NCAP. As leituras são realizadas nos seguintes horários, padronizados pelo *Coordinated Universal Time* (UTC) em UTC-3, 21h, 9h e 15h. A base de dados contém informações decorrentes de 91 dias de monitoramento, dessa forma, a base de dados ilustrada pela Tabela 32 contém 273 registros.

Ao minerar as regras de associação mediante a base de dados apresentada na Tabela 32, observa-se que primeiro deve-se passar pelo pré-processamento com o objetivo de discretizar os valores das variáveis de observação, de acordo com as características da base de dados apresentada.

Tabela 32 – Amostra da base de dados obtida pelos WTIMs da rede.

Data	Hora	WTIM-1			WTIM-2		
		t	u	ps	t	u	ps
03/06/19	21	18.22	86.067	919.95	20.80	87.511	922.96
03/06/19	9	27.55	46.059	919.87	28.36	38.851	923.06
03/06/19	15	29.21	39.700	917.34	30.63	35.483	920.33
04/06/19	21	17.28	78.752	921.35	17.90	83.654	924.40
04/06/19	9	17.73	92.377	922.01	19.60	89.211	925.15
04/06/19	15	26.10	48.855	919.52	26.77	45.762	922.51
05/06/19	21	15.57	73.032	920.09	15.45	93.394	923.23
05/06/19	9	16.63	80.784	921.93	20.29	65.645	925.03
05/06/19	15	27.33	32.439	918.97	27.79	31.359	922.02
06/06/19	21	17.98	74.867	919.61	17.87	57.119	922.76
06/06/19	9	20.19	68.915	921.51	21.58	57.119	924.64
06/06/19	15	27.62	39.861	918.34	27.32	42.316	921.43
⋮							
t = temperatura ($^{\circ}C$)							
u = umidade (%)							
ps = pressão atmosférica (hPa)							

Fonte: Versão elaborada pelo autor.

6.2 REALIZANDO O PRÉ-PROCESSAMENTO DA BASE DE DADOS

Neste experimento, a discretização que é uma das funções do pré-processamento, foi realizada através da média diária de cada variável climática. Considerou-se apenas as faixas de valores discretas que compreende as variáveis climáticas estudadas, e um novo banco de dados foi criado para armazenar os dados pré-processados.

O método de discretização é apresentado na Tabela 33, sendo que, cada elemento climático é subdividido e cada subdivisão contém uma identificação única e um significado. Ao realizar a discretização, haverá uma verificação de todos os valores das variáveis climáticas de acordo com a tabela de discretização. Por exemplo, ao discretizar os dados referentes à temperatura haverá uma análise referente a este elemento climático de acordo com cada valor obtido e sua ocorrência na tabela de discretização. Quando houver leituras entre 0 e 10°C, o intervalo será classificado como uma temperatura “muito fria” e sua identificação será “23” em um novo banco de dados criado para armazenar os dados pré-processados.

Tabela 33 – Discretização dos elementos meteorológicos

Elemento meteorológico	Identificação	Intervalo de valores	Significado
Direção do vento ($^{\circ}$)	1	(0, 45]	Nordeste
	2	(45, 90]	Leste
	3	(90, 135]	Sudeste
	4	(135, 180]	Sul
	5	(180, 225]	Sudoeste
	6	(225, 270]	Oeste
	7	(270, 315]	Noroeste
	8	(315, 360]	Norte
Velocidade do vento (km/h)	9	(0, 2]	Sem vento
	10	(2, 12]	Leve
	11	(12, 30]	Moderado
	12	(30, 40]	Fresco
	13	(40, 62]	Forte
Precipitação (mm)	14	(0, 0.5]	Sem chuva
	15	(0.5, 1]	Vestígio
	16	(1, 49]	Leve
	17	(49, 250]	Moderada
	18	(250, 1000]	Pesada
Umidade (%)	19	(0, 50]	Muito seco
	20	(50, 70]	Seco
	21	(70, 90]	Meio úmida
	22	(90, 100]	Úmida
Temperatura ($^{\circ}C$)	23	(0, 10]	Muito frio
	24	(10, 16]	Frio
	25	(16, 22]	Suave
	26	(22, 28]	Caloroso
	27	(28, 50]	Quente
Pressão barométrica (hPa)	28	(0, 9950]	Muito baixa
	29	(9950, 10000]	Baixa
	30	(10000, 10050]	Moderada
	31	(10050, 10100]	Ligeiramente alta
	32	(10100, 10150]	alta
	33	(10150, 20000]	Muito alta

Fonte: Nandagopal, Karthik e Arunachalam (2010).

Após a etapa de discretização, a base de dados compreenderá as características ilustradas na Tabela 34. Por exemplo: no dia 3 de Junho de 2019, a temperatura estava “calorosa” (26), a umidade estava “seca” (20) e a pressão atmosférica estava “muito baixa” (28). Observa-se que neste mesmo dia as características das variáveis climáticas são as mesmas para os dados de ambos os WTIMs. Como o processo de discretização foi realizado no âmbito diário, após este processo, a base de dados conterà 91 registros.

Tabela 34 – Base de dados discretizada

	WTIM-1			WTIM-2		
Data	t	u	ps	t	u	ps
03/06/19	26	20	28	26	20	28
04/06/19	25	21	28	25	21	28
05/06/19	25	20	28	25	20	28
06/06/19	25	20	28	26	20	28
			$\vdots$			

t = temperatura

u = umidade

ps = pressão atmosférica

Fonte: Versão elaborada pelo autor.

O Algoritmo 4 do Apêndice C apresenta o processo que realiza a discretização da base de dados. A discretização é aplicada nas informações de cada WTIM separadamente.

O processo de discretização evidenciou que a pressão atmosférica sempre é “muito baixa” (28) no período de observação. Na quarta e sétima coluna da Tabela 34 nota-se essa característica. Entre os dias 3 de Junho de 2019 e 2 de Setembro de 2019 a pressão atmosférica manteve a mesma relação entre as faixas discretas de valores.

A próxima etapa é encontrar algum comportamento anormal na base de dados, através da pesquisa de *outliers*.

6.2.1 *Boxplot*: Identificando *outliers* da base de dados discretizada

O *boxplot* foi aplicado no conjunto de dados discretizados de cada variável climática. As informações de cada WTIM foram analisadas separadamente e as variantes obtidas neste processo são apresentadas na Tabela 35.

Os dados da Tabela 35 são representados pela associação de cada variável climática com sua variante no gráfico, sendo na Figura 48(a) o gráfico do WTIM-1 e na Figura 48(b), o gráfico do mesmo processo para o WTIM-2.

Observa-se que, nos gráficos das Figuras 48(a) e (b) as variáveis de temperatura (t) e pressão atmosférica (ps) mantiveram a mesma variação nas duas localidades onde os WTIMS estão instalados. A temperatura média no período de observação foi “suave” (25) nos dois conjuntos de dados, mantendo uma variação entre “frio” (24) e “quente” (27).

Porém, a maioria das observações se concentraram entre “suave” (25) e “caloroso” (26). A pressão atmosférica assume o mesmo valor no período de observação. Nota-se nos gráficos esta característica, sendo traçado apenas uma linha no *boxplot* do conjunto *ps*. Sendo assim, suas variantes são constantes, como também é possível observar na Tabela 35.

Tabela 35 – Resultado da aplicação do *boxplot* no banco de dados dos WTIMs

	WTIM-1			WTIM-2		
	<i>t</i>	<i>u</i>	<i>ps</i>	<i>t</i>	<i>u</i>	<i>ps</i>
<i>min</i>	24	19	28	24	19	28
<i>Q1</i>	25	20	28	25	19	28
<i>Q2</i>	25	20	28	25	21	28
<i>Q3</i>	26	21	28	26	21	28
<i>max</i>	27	22	28	27	22	28

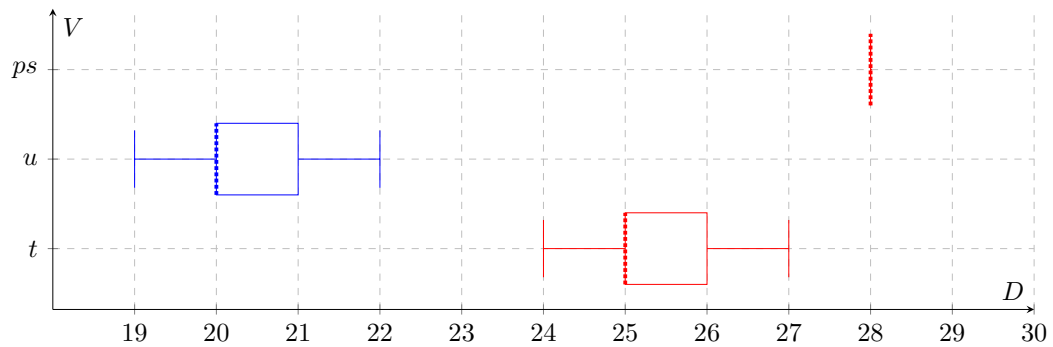
t = temperatura

u = umidade

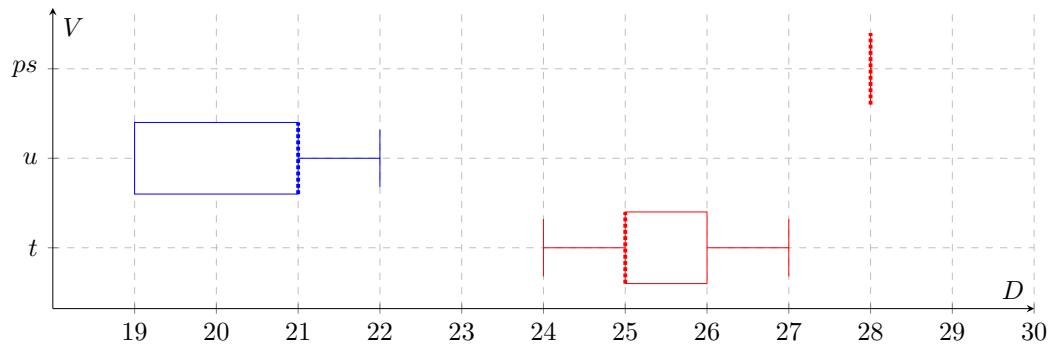
ps = pressão atmosférica

Fonte: Versão elaborada pelo autor.

Figura 48 – *Boxplot* da base de dados discretizada.



(a)



(b)

Fonte: Versão elaborada pelo autor.

A umidade também manteve a mesma variação no período de observação nas duas bases de dados, variando entre “muito seca” (19) e “úmida” (22). Porém, o valor médio da umidade na base de dados do WTIM-1 foi “seca” (20), com a maioria das observações entre “seca” (20) e “meio úmida” (21). Por outro lado, o valor médio da umidade na base de dados do WTIM-2 foi “meio úmida” (21), com a maioria das observações entre “muito seca” (19) e “meio úmida” (21).

O *boxplot* não evidenciou nenhum *outlier* na base de dados estudada. Sendo assim, nenhuma informação é considerada atípica na base de dados. Após esse processo, a mineração de regras de associação pode ser aplicada na base de dados.

6.3 EXECUÇÃO DO ALGORITMO APRIORI SOBRE A BASE DE DADOS DISCRETIZADA

A técnica de mineração de regras de associação foi aplicada na base de dados discretizada de cada dispositivo separadamente. Primeiramente, o algoritmo Apriori foi executado com o objetivo de encontrar regras com o grau de suporte mínimo β mais elevado, definindo o grau de confiança mínima α com um valor muito pequeno. Os parâmetros definidos nos testes foram os seguintes: $\beta = 20\%$ e $\alpha = 1\%$. Sendo assim, ao executar o algoritmo Apriori na base de dados do WTIM-1 gerou-se 24 regras e na base de dados do WTIM-2 gerou-se 16 regras. As regras geradas nestes testes são ilustradas nas Figuras 49(a) e (b).

Figura 49 – Regras com grau de suporte mais elevado.

```

1 Rule: ('28',) ==> ('25', '21'), 0.218
2 Rule: ('28',) ==> ('19',), 0.232
3 Rule: ('28',) ==> ('25', '20'), 0.266
4 Rule: ('28',) ==> ('20',), 0.317
5 Rule: ('25',) ==> ('21',), 0.395
6 Rule: ('25',) ==> ('28', '21'), 0.395
7 Rule: ('25', '28') ==> ('21',), 0.395
8 Rule: ('28',) ==> ('21',), 0.418
9 Rule: ('25',) ==> ('20',), 0.482
10 Rule: ('25',) ==> ('28', '20'), 0.482
11 Rule: ('25', '28') ==> ('20',), 0.482
12 Rule: ('21',) ==> ('25',), 0.522
13 Rule: ('21',) ==> ('25', '28'), 0.522
14 Rule: ('28', '21') ==> ('25',), 0.522
15 Rule: ('28',) ==> ('25',), 0.551
16 Rule: ('20',) ==> ('25',), 0.838
17 Rule: ('20',) ==> ('25', '28'), 0.838
18 Rule: ('28', '20') ==> ('25',), 0.838
19 Rule: ('21',) ==> ('28',), 1.000
20 Rule: ('25',) ==> ('28',), 1.000
21 Rule: ('20',) ==> ('28',), 1.000
22 Rule: ('19',) ==> ('28',), 1.000
23 Rule: ('25', '21') ==> ('28',), 1.000
24 Rule: ('25', '20') ==> ('28',), 1.000

```

(a)

```

1 Rule: ('28',) ==> ('26',), 0.235
2 Rule: ('28',) ==> ('19',), 0.257
3 Rule: ('28',) ==> ('25', '21'), 0.385
4 Rule: ('28',) ==> ('21',), 0.458
5 Rule: ('28',) ==> ('25',), 0.566
6 Rule: ('25',) ==> ('21',), 0.680
7 Rule: ('25',) ==> ('28', '21'), 0.680
8 Rule: ('25', '28') ==> ('21',), 0.680
9 Rule: ('21',) ==> ('25',), 0.841
10 Rule: ('21',) ==> ('25', '28'), 0.841
11 Rule: ('28', '21') ==> ('25',), 0.841
12 Rule: ('26',) ==> ('28',), 1.000
13 Rule: ('19',) ==> ('28',), 1.000
14 Rule: ('25',) ==> ('28',), 1.000
15 Rule: ('21',) ==> ('28',), 1.000
16 Rule: ('25', '21') ==> ('28',), 1.000

```

(b)

Fonte: Versão elaborada pelo autor.

As regras obtidas não tem restrição de tamanho, ou seja, foram geradas regras triviais de 2 itens e regras de até 3 itens, determinada pelo número total de variáveis meteorológicas utilizadas (neste caso 3).

Cada regra gerada contém seu grau de confiança, que é apresentado na frente de cada regra. As regras são armazenadas em ordem crescente com relação ao grau de confiança. A primeira regra contém o menor grau de confiança e a última regra contém o maior grau de confiança.

A primeira regra gerada através da base de dados do WTIM-1 contém 21,8% de confiança, como apresentado na Figura 49(a). Esta regra evidencia, que: Quando a pressão atmosférica está “muito baixa” (28), a temperatura fica “suave” (25) e a umidade fica “meio úmida” (21). A última regra gerada através da base de dados do WTIM-1 contém 100% de confiança. Esta regra evidencia, que: Quando a temperatura está “suave” (25) e a umidade está “seca” (20), a pressão atmosférica fica “muito baixa” (28).

A primeira regra gerada através da base de dados do WTIM-2 é trivial e contém 23,5% de confiança, como ilustrado na Figura 49(b). Esta regra evidencia, que: Quando a pressão atmosférica está “muito baixa” (28), a temperatura fica “calorosa” (26). A última regra gerada através da base de dados do WTIM-2 contém 100% de confiança. Esta regra evidencia, que: Quando a temperatura está “suave” (25) e a umidade está “meio úmida” (21), a pressão atmosférica fica “muito baixa” (28).

Foram realizados mais 110 testes com a base de dados de cada WTIM, com parâmetros de suporte mínimo β e confiança mínima α distintos. Por convenção, β foi definido com valores pequenos, para não diminuir a qualidade das regras geradas.

Na Tabela 36 é apresentado a quantidade de regras geradas nesse experimento através dos dados do WTIM-1 e a Tabela 37 apresenta a quantidade de regras geradas nesse experimento através dos dados do WTIM-2. O suporte mínimo β variou dentro do intervalo de 1% a 10%, e para cada um desses valores foi alterado o grau de confiança mínimo α , dentro do intervalo de 90% a 100%. As regras obtidas, como foi definido no teste anterior, não tem restrição de tamanho.

Tabela 36 – Número de regras R geradas em função do grau mínimo de suporte β e confiança α através da base de dados do WTIM-1.

β (%)	α (%)										
	90	91	92	93	94	95	96	97	98	99	100
1	19	19	19	19	19	19	19	19	19	19	19
2	17	17	17	17	17	17	17	17	17	17	17
3	16	16	16	16	16	16	16	16	16	16	16
4	14	14	14	14	14	14	14	14	14	14	14
5	13	13	13	13	13	13	13	13	13	13	13
6	13	13	13	13	13	13	13	13	13	13	13
7	12	12	12	12	12	12	12	12	12	12	12
8	11	11	11	11	11	11	11	11	11	11	11
9	10	10	10	10	10	10	10	10	10	10	10
10	10	10	10	10	10	10	10	10	10	10	10

Fonte: Versão elaborada pelo autor.

Tabela 37 – Número de regras R geradas em função do grau mínimo de suporte β e confiança α através da base de dados do WTIM-2.

β (%)	α (%)										
	90	91	92	93	94	95	96	97	98	99	100
1	18	18	18	18	18	18	18	18	18	18	18
2	17	17	17	17	17	17	17	17	17	17	17
3	16	16	16	16	16	16	16	16	16	16	16
4	16	16	16	16	16	16	16	16	16	16	16
5	14	14	14	14	14	14	14	14	14	14	14
6	13	13	13	13	13	13	13	13	13	13	13
7	12	12	12	12	12	12	12	12	12	12	12
8	12	12	12	12	12	12	12	12	12	12	12
9	12	12	12	12	12	12	12	12	12	12	12
10	10	10	10	10	10	10	10	10	10	10	10

Fonte: Versão elaborada pelo autor.

Pode-se observar neste teste que para cada grau de suporte, independentemente do grau mínimo de confiança, o número de regras geradas se manteve constante nas duas bases de dados. Isto indica que, nesta faixa de confiança, todas as regras encontradas contém 100% de confiança para cada grau de suporte. Sendo assim, para cada grau mínimo de suporte, utilizando o grau mínimo de confiança entre 90% e 100%, obtêm-se como resultado as mesmas regras. Essa característica é ilustrada na Figura 50.

Nas Figuras 50(a) e (b). contém regras geradas através da base de dados do WTIM-

1 com grau mínimo de suporte de 1% e grau mínimo de confiança de 90% e 91%, respectivamente. Nota-se que foram geradas as mesmas regras.

Figura 50 – Regras geradas com β de 1% e α entre 90% e 100%.

resultS0.01C0.9.data	
1	Rule: ('22',) ==> ('28',) , 1.000
2	Rule: ('24',) ==> ('28',) , 1.000
3	Rule: ('21',) ==> ('28',) , 1.000
4	Rule: ('25',) ==> ('28',) , 1.000
5	Rule: ('20',) ==> ('28',) , 1.000
6	Rule: ('27',) ==> ('28',) , 1.000
7	Rule: ('26',) ==> ('28',) , 1.000
8	Rule: ('19',) ==> ('28',) , 1.000
9	Rule: ('24', '22') ==> ('28',) , 1.000
10	Rule: ('19', '27') ==> ('28',) , 1.000
11	Rule: ('25', '21') ==> ('28',) , 1.000
12	Rule: ('24', '21') ==> ('28',) , 1.000
13	Rule: ('27', '21') ==> ('28',) , 1.000
14	Rule: ('25', '20') ==> ('28',) , 1.000
15	Rule: ('24', '20') ==> ('28',) , 1.000
16	Rule: ('26', '20') ==> ('28',) , 1.000
17	Rule: ('26', '21') ==> ('28',) , 1.000
18	Rule: ('19', '26') ==> ('28',) , 1.000
19	Rule: ('19', '25') ==> ('28',) , 1.000

(a)

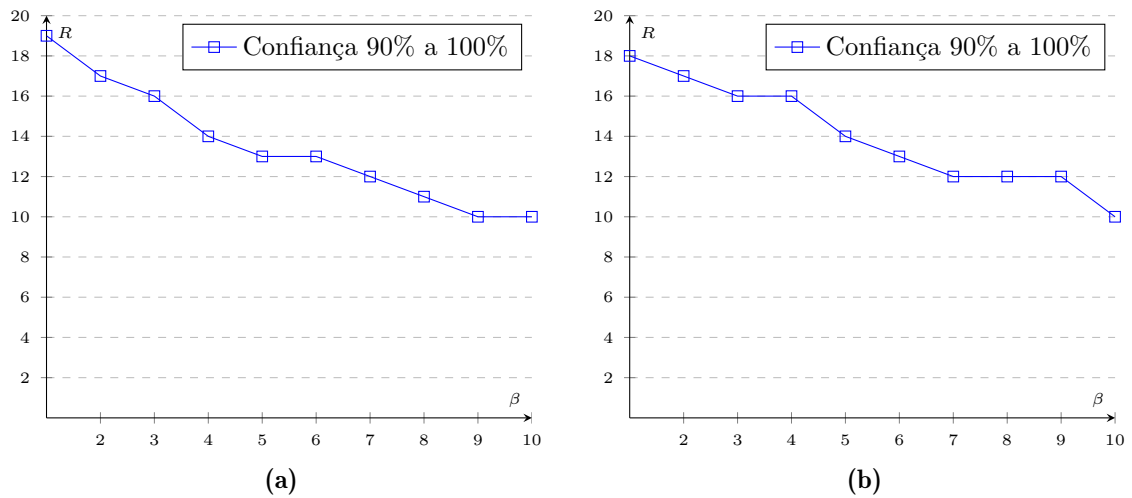
resultS0.01C0.91.data	
1	Rule: ('22',) ==> ('28',) , 1.000
2	Rule: ('24',) ==> ('28',) , 1.000
3	Rule: ('21',) ==> ('28',) , 1.000
4	Rule: ('25',) ==> ('28',) , 1.000
5	Rule: ('20',) ==> ('28',) , 1.000
6	Rule: ('27',) ==> ('28',) , 1.000
7	Rule: ('26',) ==> ('28',) , 1.000
8	Rule: ('19',) ==> ('28',) , 1.000
9	Rule: ('24', '22') ==> ('28',) , 1.000
10	Rule: ('19', '27') ==> ('28',) , 1.000
11	Rule: ('25', '21') ==> ('28',) , 1.000
12	Rule: ('24', '21') ==> ('28',) , 1.000
13	Rule: ('27', '21') ==> ('28',) , 1.000
14	Rule: ('25', '20') ==> ('28',) , 1.000
15	Rule: ('24', '20') ==> ('28',) , 1.000
16	Rule: ('26', '20') ==> ('28',) , 1.000
17	Rule: ('26', '21') ==> ('28',) , 1.000
18	Rule: ('19', '26') ==> ('28',) , 1.000
19	Rule: ('19', '25') ==> ('28',) , 1.000

(b)

Fonte: Versão elaborada pelo autor.

Os gráficos apresentados na Figura 51 ilustram o número de regras obtidas para cada grau de confiança mínima α , alterando-se o grau de suporte mínimo β . Percebe-se que o número de regras encontradas diminuiu na medida em que o β aumentou, como é normal na mineração de regras de associação.

Figura 51 – Número de regras R geradas em função do grau mínimo de suporte β .



Fonte: Versão elaborada pelo autor.

Através da base de dados do WTIM-1 foi constatado a geração de regras iguais com o

suporte mínimo β assumindo valores diferentes. Esse processo é ilustrado na Figura 51(a) ao gerar regras com β igual a 5% e 6% e ao gerar regras com β igual a 9% e 10%. Isso também foi constatado nas regras geradas através da base de dados do WTIM-2. Esse processo é ilustrado na Figura 51(b) ao gerar regras com β igual a 3% e 4% e ao gerar regras com β igual a 7%, 8% e 9%.

6.4 CONSIDERAÇÕES SOBRE O CAPÍTULO

Nesse capítulo foi apresentado a aplicabilidade da mineração de regras de associação na base de dados da rede *mesh* de transdutores inteligentes desenvolvida. Comprova-se a eficiência desta técnica ao gerar regras de associação nos dados da rede. A mineração de regras de associação foi aplicada, separadamente, nos dados de cada WTIM da rede. Os dados foram coletados em um período de 91 dias de monitoramento. As regras foram obtidas de acordo com as características dos dados, relacionando-as com os parâmetros utilizados no algoritmo Apriori.

7 CONCLUSÕES

Para realizar a aquisição dos dados climáticos foi desenvolvida uma rede *mesh* de transdutores inteligentes com base no padrão de rede ZigBee e no padrão de interfaceamento IEEE 1451.5. A rede foi implantada em torno da Universidade Federal de Catalão (UFCat) e atualmente contém 4 nós capazes de obter informações acerca da temperatura, umidade e pressão atmosférica.

A escolha da tecnologia ZigBee para a criação da rede *mesh* justifica-se pelo fato de possibilitar a construção de uma rede de baixo custo, auto-organizável e confiável. Outra característica que deve ser ressaltada é a facilidade de configuração e interfaceamento com a tecnologia embarcada. Utilizando essa tecnologia, por meio do uso de microcontroladores e da linguagem C, as funcionalidades dos dispositivos podem ser facilmente alteradas para atender quaisquer necessidades que possam aparecer. Além de prover a interface com a tecnologia de rádio, o microcontrolador fornece a interface com os sensores.

Outro aspecto importante está no interfaceamento dos nós da rede ZigBee por meio da tecnologia IEEE 1451.5. Que fornece uma interface padronizada entre os nós utilizando-se um meio sem fio de comunicação. Com o emprego desta tecnologia no âmbito da rede ZigBee a comunicação com os transdutores inteligentes se torna mais dinâmica e flexível.

Ao idealizar a implantação da rede no local definido, constatou-se a dificuldade da comunicação entre os dispositivos no ambiente urbano devido a problemas de relevo, edificações, vegetação e interferências eletromagnéticas. Para ter comunicação satisfatória é necessário que os módulos de rádio tenham visada direta, respeitando a distância máxima de comunicação entre os módulos. Sendo assim, é importante obter um local adequado para a instalação de cada nó da rede. Neste trabalho, houve a necessidade da instalação de nós em prédios da UFCAT para garantir linhas de visada em vários locais da cidade, e a construção de uma torre para a instalação de um nó adjacente.

Um aspecto importante a ser ressaltado é que, obter os dados de uma rede de sensores é indispensável para definir características do microclima da área monitorada.

Em microclimas urbanos a emissão de poluentes e edificações dão origem ao aumento de temperatura e na mudança da composição natural do ar, provocando diferenças de umidade, precipitação e outros componentes do clima.

Finalmente, os informações da rede desenvolvida foram submetidas à mineração de regras de associação. A mineração de dados foi executada na base de dados de cada dispositivo da rede após a etapa de pré-processamento. A etapa de pré-processamento não evidenciou nenhuma anomalia, ou ruído nos dados. Isso comprova que não houve erros de leitura ou ineficiência no sensor utilizado. Então, a mineração de regras de associação foi aplicada usando o algoritmo Apriori, por meio de alguns cenários de teste. Pelos resultados obtidos, comprova-se a eficiência desta técnica de mineração de dados ao gerar regras nos dados de uma rede *mesh* de transdutores inteligentes.

A partir da pesquisa realizada, originaram-se diretamente os trabalhos apresentados no Apêndice D que foram publicados em anais de congressos.

Como proposta de trabalho futuros, outros tipos de variáveis climáticas podem ser identificadas com a instalação de mais sensores. A rede também pode ser modificada, podendo conter nós WTIMs com diferentes tecnologias embarcadas, como: Raspberry e dispositivos Espressif. Outra possibilidade é a implementação de um banco de dados relacional no NCAP para o armazenamento das informações dos sensores de cada nó, isso facilitaria a manipulação dos dados e sua apresentação em uma plataforma Web.

REFERÊNCIAS

- AGRAWAL, R.; IMIELIŃSKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. **SIGMOD Rec.**, New York, v. 22, n. 2, p. 207–216, 1993.
- ALAM, M. E.; KAISER, M. S.; HOSSAIN, M. S.; ANDERSSON, K. An iot-belief rule base smart system to assess autism. In: 2018 4TH INTERNATIONAL CONFERENCE ON ELECTRICAL ENGINEERING AND INFORMATION COMMUNICATION TECHNOLOGY (ICEEICT), 4., 2018, Dhaka. **Anais [...]**. 2018. p. 672–676.
- ALFIAN, G.; RHEE, J.; AHN, H.; LEE, J.; FAROOQ, U.; IJAZ, M. F.; SYAEKHONI, M. A. Integration of rfid, wireless sensor networks, and data mining in an e-pedigree food traceability system. **Journal of Food Engineering**, Netherlands, v. 212, n. 2, p. 65–75, 2017.
- AMEEN, S.; NOURILDEAN, S. Coordinator and router investigation in IEEE802.15.14 zigbee wireless sensor network. In: 2013 INTERNATIONAL CONFERENCE ON ELECTRICAL COMMUNICATION, COMPUTER, POWER, AND CONTROL ENGINEERING (ICECCPCE), 1., 2013, Mosul. **Anais [...]**. 2013. p. 130–134.
- AMO, S. A. **Curso Introductório de Mineração de Dados**. 2006. Disponível em: <<http://www.deamo.prof.ufu.br/CursoDM2014.html>>. Acesso em: 26 Set. 2020.
- ANANTATHANAVIT, M.; MUNLIN, M. Radius particle swarm optimization. In: 2013 INTERNATIONAL COMPUTER SCIENCE AND ENGINEERING CONFERENCE (ICSEC), 17., 2013, Nakorn Pathom. **Anais [...]**. 2013. p. 126–130.
- ASUS. **ASUS Tinker Board**. 2020. Disponível em: <<https://www.asus.com/br/Single-Board-Computer/Tinker-Board/gallery/>>. Acesso em: 26 Set. 2020.
- BHAVSAR, A. R.; AROLKAR, H. A. Multidimensional association rule based data mining technique for cattle health monitoring using wireless sensor network. In: 2014 INTERNATIONAL CONFERENCE ON COMPUTING FOR SUSTAINABLE GLOBAL DEVELOPMENT (INDIACOM), 1., 2014, New Delhi. **Anais [...]**. 2014. p. 810–814.
- BINH, H. T. T.; DEY, N. **Soft Computing in Wireless Sensor Networks**. 1. ed. Boca Raton: Chapman and Hall/CRC, 2018.
- BOUKERCHE, A.; SAMARAH, S. An efficient data extraction mechanism for mining association rules from wireless sensor networks. In: 2007 IEEE INTERNATIONAL CONFERENCE ON COMMUNICATIONS, 2007, Glasgow. **Anais [...]**. 2007. p. 3936–3941.
- BRAUN, A. C.; WEIDNER, U.; HINZ, S. Support vector machines, import vector machines and relevance vector machines for hyperspectral classification - a comparison. In: 2011 3RD WORKSHOP ON HYPERSPECTRAL IMAGE AND SIGNAL PROCESSING: EVOLUTION IN REMOTE SENSING (WHISPERS), 3., 2011, Lisbon. **Anais [...]**. 2011. p. 1–4.

DAS, G. Top-k algorithms and applications. In: INTERNATIONAL CONFERENCE ON DATABASE SYSTEMS FOR ADVANCED APPLICATIONS, 2009, Brisbane. **Anais [...]**. 2009. v. 5463, p. 789–792.

DIGI. **XBee Family Features Comparison**. 2020. Disponível em: <https://media.digikey.com/pdf/Other%20Related%20Documents/Digi%20International%20Other%20Doc/Xbee_RF_Features_Chart.pdf>. Acesso em: 26 Set. 2020.

ESTEVEES, R. M.; HACKER, T.; RONG, C. Competitive k-means, a new accurate and distributed k-means algorithm for large datasets. In: 2013 IEEE 5TH INTERNATIONAL CONFERENCE ON CLOUD COMPUTING TECHNOLOGY AND SCIENCE, 5., 2013, Bristol. **Anais [...]**. 2013. p. 17–24.

FARAHANI, S. **ZigBee Wireless Networks and Transceivers**. Burlington: Newnes, 2008.

FARIAS, A. M. L. de. **O BOXPLOT**. 2020. Disponível em: <<http://www.professores.uff.br/anafarias/wp-content/uploads/sites/210/2020/09/estdesc-0-light.pdf>>. Acesso em: 26 Set. 2020.

FAYYAD, U.; PIATETSKY-SHAPIO, G.; SMYTH, P. The kdd process for extracting useful knowledge from volumes of data. **Commun. ACM**, New York, v. 39, n. 11, p. 27–34, 1996.

GARCIA, L.; JIMENEZ, J.; TAHA, M.; LLORET, J. Wireless technologies for iot in smart cities. **Network Protocols and Algorithms**, Las Vegas, v. 10, p. 23–64, 2018.

HARB, H.; MAKHOUL, A.; JAOUDE, C. A. A real-time massive data processing technique for densely distributed sensor networks. **IEEE Access**, Piscataway, v. 6, p. 56551–56561, 2018.

HE, W.; QIAO, P.; ZHOU, Z.; HU, G.; FENG, Z.; WEI, H. A new belief-rule-based method for fault diagnosis of wireless sensor network. **IEEE Access**, Piscataway, v. 6, p. 9404–9419, 2018.

HUBERT, M.; VANDERVIEREN, E. An adjusted boxplot for skewed distributions. **Computational Statistics & Data Analysis**, Netherlands, v. 52, n. 12, p. 5186–5201, 2008.

IBRAHIM, D. An overview of soft computing. **Procedia Computer Science**, Netherlands, v. 102, p. 34–38, 2016.

IEEE. Ieee standard for a smart transducer interface for sensors and actuators - common functions, communication protocols, and transducer electronic data sheet (teds) formats. **IEEE Std 1451.0-2007**, Piscataway, p. 1–335, 2007.

IEEE. Ieee standard for a smart transducer interface for sensors and actuators wireless communication protocols and transducer electronic data sheet (teds) formats. **IEEE Std 1451.5-2007**, Piscataway, p. 1–236, 2007.

IEEE. Ieee standard for low-rate wireless networks. **IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)**, Piscataway, p. 1–709, 2016.

ISO/IEC/IEEE. Iso/iec/ieee international standard - information technology – telecommunications and information exchange between systems – local and metropolitan area networks – overview and architecture. **ISO/IEC/IEEE 8802-A:2015(E)**, Piscataway, p. 1–80, 2015.

KOHAIL, S. N.; EL-HALEES, A. M. Implementation of data mining techniques for meteorological data analysis. **International Journal of Information and Communication Technology Research**, Singapore, v. 1, n. 3, p. 96–100, 2011.

KUMAR, S.; CHAURASIYA, V. K. A strategy for elimination of data redundancy in internet of things (iot) based wireless sensor network (wsn). **IEEE Systems Journal**, Piscataway, v. 13, n. 2, p. 1650–1657, 2019.

LEE, S. R.; BACK, J.; OH, J.; JEONG, M. A mesh topology formation scheme for ieee 802.15.4 based wireless sensor networks. In: 2015 SEVENTH INTERNATIONAL CONFERENCE ON UBIQUITOUS AND FUTURE NETWORKS, 7., 2015, Sapporo. **Anais [...]**. 2015. p. 150–152.

MITRA, S.; PAL, S. K.; MITRA, P. Data mining in soft computing framework: a survey. **IEEE Transactions on Neural Networks**, Piscataway, v. 13, n. 1, p. 3–14, 2002.

MOUSER. **Bosch BME280 Humidity and Pressure Sensor**. 2020. Disponível em: <<https://br.mouser.com/new/bosch/bosch-bme280/>>. Acesso em: 26 Set. 2020.

NANDAGOPAL, S.; KARTHIK, S.; ARUNACHALAM, V. P. Mining of meteorological data using modified apriori algorithm. **European Journal of Scientific Research**, Victoria, Mahé, v. 47, p. 295–308, 2010.

PATIL, R.; DATE, T.; KUSHARE, B. Zigbee based parameters monitoring system for induction motor. In: 2014 IEEE STUDENTS' CONFERENCE ON ELECTRICAL, ELECTRONICS AND COMPUTER SCIENCE, 2014, Bhopal. **Anais [...]**. 2014. p. 1–6.

PING-GUANG, C. Research on outlier data mining algorithms based on subspace. In: 2010 3RD INTERNATIONAL CONFERENCE ON ADVANCED COMPUTER THEORY AND ENGINEERING(ICACTE), 3., 2010, Chengdu. **Anais [...]**. 2010. v. 2, p. V2–355–V2–357.

POMYKAL, L. **BME280 Pressure measurement**. 2020. Disponível em: <<http://www.lpomykal.cz/bme280-pressure-measurement/>>. Acesso em: 26 Set. 2020.

RAJA, N.; GAVEL, S.; RAGHUVANSHI, A. S. Hardware architecture for csma/ca algorithm for ieee 802.15.4 lr-wpans. In: 2019 9TH ANNUAL INFORMATION TECHNOLOGY, ELECTROMECHANICAL ENGINEERING AND MICROELECTRONICS CONFERENCE (IEMECON), 9., 2019, Jaipur. **Anais [...]**. 2019. p. 174–178.

RAUT, A. R.; KHANDAIT, S. P. Review on data mining techniques in wireless sensor networks. In: 2015 2ND INTERNATIONAL CONFERENCE ON ELECTRONICS AND COMMUNICATION SYSTEMS (ICECS), 2., 2015, Coimbatore. **Anais [...]**. 2015. p. 390–394.

RENLI, W.; YUEMING, D.; LIMING, D. The application of apriori-bso algorithms in medical records data mining. In: 2016 IEEE INFORMATION TECHNOLOGY,

NETWORKING, ELECTRONIC AND AUTOMATION CONTROL CONFERENCE, 2016, Chongqing. **Anais [...]**. 2016. p. 827–832.

SABIT, H.; AL-ANBUKY, A.; GHOLAM-HOSSEINI, H. Distributed wsn data stream mining based on fuzzy clustering. In: 2009 SYMPOSIA AND WORKSHOPS ON UBIQUITOUS, AUTONOMIC AND TRUSTED COMPUTING, 2009, Brisbane. **Anais [...]**. 2009. p. 395–400.

SANTOS, M. D. S. **Clima Urbano: Análise do Campo Higrométrico nos Bairros Setor Universitário e Residencial Barka em Catalão (GO)**. 101 f. Dissertação (Mestrado em Geografia) — UFCAT, Universidade Federal de Catalão, Catalão, 2020.

SHENOY, A. P.; AMEER, P. M. Anamoly detection in wireless sensor networks. In: TENCON 2019 - 2019 IEEE REGION 10 CONFERENCE (TENCON), 2019, Kochi. **Anais [...]**. 2019. p. 1504–1508.

SILICON LABS. **The Evolution of Wireless Sensor Networks**. 2013. Disponível em: <<https://www.silabs.com/documents/public/white-papers/evolution-of-wireless-sensor-networks.pdf>>. Acesso em: 26 Set. 2020.

SINGH, M.; MEHTA, G.; VAID, C.; OBEROI, P. Detection of malicious node in wireless sensor network based on data mining. In: 2012 INTERNATIONAL CONFERENCE ON COMPUTING SCIENCES, 2012, Phagwara. **Anais [...]**. 2012. v. 1, p. 291–294.

SONG, E. Y.; BURNS, M.; PANDEY, A.; ROTH, T. Ieee 1451 smart sensor digital twin federation for iot/cps research. In: 2019 IEEE SENSORS APPLICATIONS SYMPOSIUM (SAS), 2019, Sophia Antipolis. **Anais [...]**. 2019. p. 1–6.

SONG, E. Y.; LEE, K. B.; PROCTOR, F. Testing system for ieee 1451.5-802.11 standard-based wireless sensors. In: 2014 IEEE INTERNATIONAL INSTRUMENTATION AND MEASUREMENT TECHNOLOGY CONFERENCE (I2MTC) PROCEEDINGS, 2014, Montevideo. **Anais [...]**. 2014. p. 862–867.

SUN, D.; ZHAO, H.; YUE, H.; ZHAO, M.; CHENG, S.; HAN, W. St td outlier detection. **IET Intelligent Transport Systems**, Charity, v. 11, n. 4, p. 203–211, 2017.

SUNITHA, R.; CHANDRIKA, J. A study on detecting and resolving major issues in wireless sensor network by using data mining and soft computing techniques. In: 2016 INTERNATIONAL CONFERENCE ON EMERGING TRENDS IN ENGINEERING, TECHNOLOGY AND SCIENCE (ICETETS), 2016, Pudukkottai. **Anais [...]**. 2016. p. 1–6.

SURESH, N.; ARULANANDAM, K. An efficient k-means clustering algorithm using euclidean distance techniques. **International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)**, Chennai, v. 6, n. 11, p. 259–266, 2017.

TAN, P.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. 1. ed. London: Pearson, 2005.

TANENBAUM, A. S. **Redes de Computadores**. 5. ed. São Paulo: Pearson, 2011.

TRIPATHY, A. K.; ADINARAYANA, J.; SUDHARSAN, D.; MERCHANT, S. N.; DESAI, U. B.; VIJAYALAKSHMI, K.; REDDY, D. R.; SREENIVAS, G.; NINOMIYA, S.; HIRAFUJI, M.; KIURA, T.; TANAKA, K. Data mining and wireless sensor network for agriculture pest/disease predictions. In: 2011 WORLD CONGRESS ON INFORMATION AND COMMUNICATION TECHNOLOGIES, 2011, Mumbai. **Anais** [...]. 2011. p. 1229–1234.

WEI, Z. Application of data mining technology based on wireless sensor networks in oceanographic forecasting. **International Journal of Online Engineering**, Vienna, v. 14, n. 4, p. 137–148, 2018.

WU, Y. H.; WANG, C. C.; CHEN, T. S.; LI, C. Y. An intelligent system for wheelchair users using data mining and sensor networking technologies. In: 2011 IEEE ASIA-PACIFIC SERVICES COMPUTING CONFERENCE, 2011, Jeju Island. **Anais** [...]. 2011. p. 337–344.

XBEE STORE. **XBee Pro S2C**. 2020. Disponível em: <<https://www.xbeestore.com.br/>>. Acesso em: 26 Set. 2020.

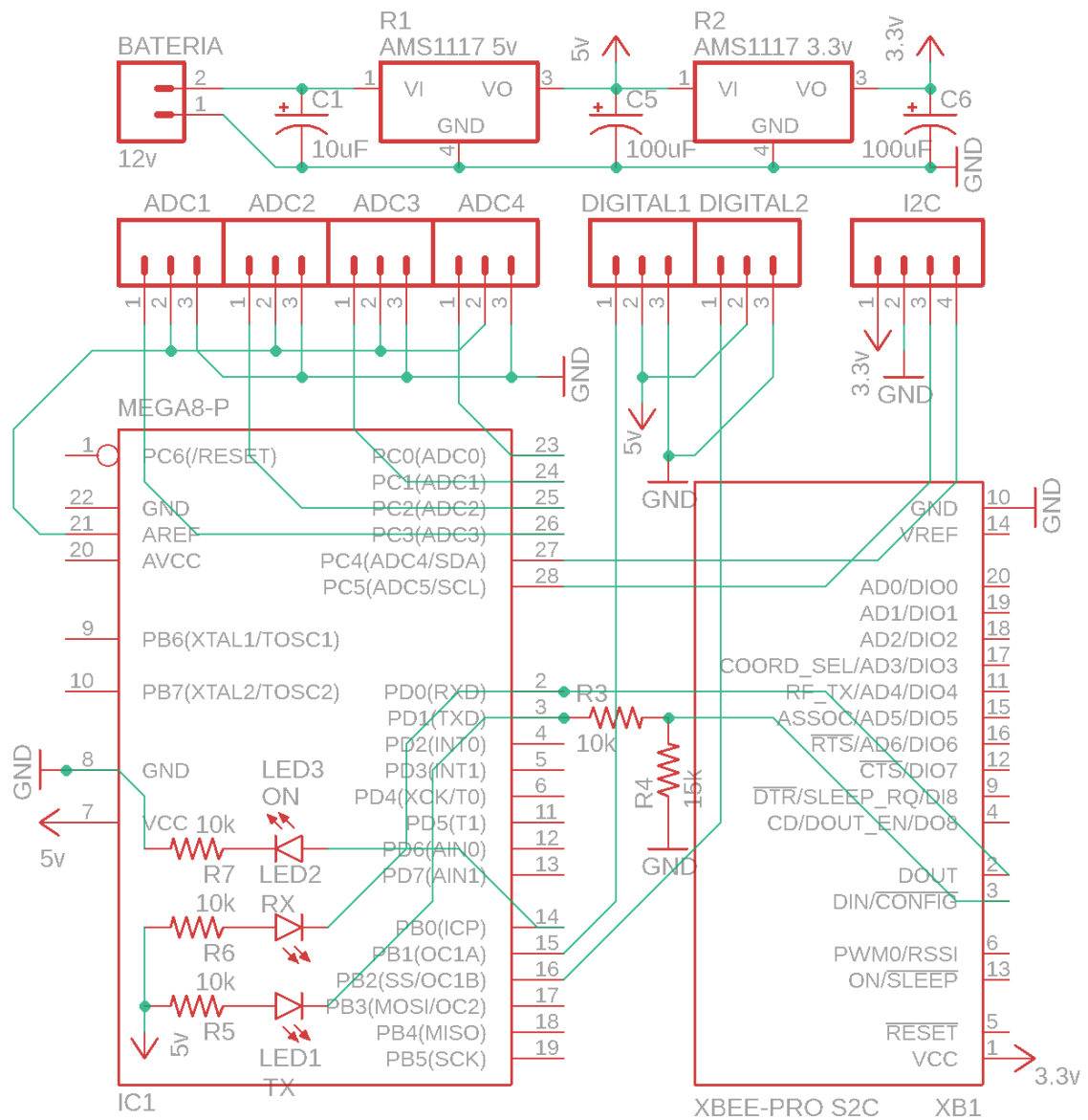
YINBIAO, S.; LEE, K.; LANCTOT, P.; JIANBIN, F.; HAO, H.; CHOW, B.; DESBENOIT, J.; STEPHAN, G.; HUI, L.; GUODONG, X.; CHEN, S.; FAULK, D.; KAISER, T.; SATOH, H.; JINSONG, O.; LINKUN, W.; SHOU, W.; YAN, Z.; JUNPING, S.; HAIBIN, Y.; PENG, Z.; DONG, L.; QIN, W. **Internet of Things: Wireless Sensor Networks**. Geneva: IEC, 2014.

ZHAO, X.; ZHANG, J.; QIN, X. Loma: A local outlier mining algorithm based on attribute relevance analysis. **Expert Systems with Applications**, Netherlands, v. 84, p. 272–280, 2017.

APÊNDICE A - SUPORTES DE UMA OBSERVAÇÃO

Considerando, que: DV = direção do vento, VV = velocidade do vento, P = precipitação, U = umidade, T = temperatura e PB = pressão barométrica. A observação {DV: 205°, VV: 15km/h, P: 14mm, U: 60%, T: 25°C, PB: 9899hPa} suporta os seguintes *varsets*: {DV: 205°}, {VV: 15km/h}, {P: 14mm}, {U: 60%}, {T: 25°C}, {PB: 9899hPa}, {DV: 205°, VV: 15km/h}, {DV: 205°, P: 14mm}, {DV: 205°, U: 60%}, {DV: 205°, T: 25°C}, {DV: 205°, PB: 9899hPa}, {VV: 15km/h, P: 14mm}, {VV: 15km/h, U: 60%}, {VV: 15km/h, T: 25°C}, {VV: 15km/h, PB: 9899hPa}, {P: 14mm, U: 60%}, {P: 14mm, T: 25°C}, {P: 14mm, PB: 9899hPa}, {U: 60%, T: 25°C}, {U: 60%, PB: 9899hPa}, {T: 25°C, PB: 9899hPa}, {DV: 205°, VV: 15km/h, P: 14mm}, {DV: 205°, VV: 15km/h, U: 60%}, {DV: 205°, VV: 15km/h, T: 25°C}, {DV: 205°, VV: 15km/h, PB: 9899hPa}, {DV: 205°, P: 14mm, U: 60%}, {DV: 205°, P: 14mm, T: 25°C}, {DV: 205°, P: 14mm, PB: 9899hPa}, {DV: 205°, U: 60%, T: 25°C}, {DV: 205°, U: 60%, PB: 9899hPa}, {DV: 205°, T: 25°C, PB: 9899hPa}, {VV: 15km/h, P: 14mm, U: 60%}, {VV: 15km/h, P: 14mm, T: 25°C}, {VV: 15km/h, P: 14mm, PB: 9899hPa}, {VV: 15km/h, U: 60%, T: 25°C}, {VV: 15km/h, U: 60%, PB: 9899hPa}, {VV: 15km/h, T: 25°C, PB: 9899hPa}, {P: 14mm, U: 60%, T: 25°C}, {P: 14mm, U: 60%, PB: 9899hPa}, {P: 14mm, T: 25°C, PB: 9899hPa}, {U: 60%, T: 25°C, PB: 9899hPa}, {DV: 205°, VV: 15km/h, P: 14mm, U: 60%}, {DV: 205°, VV: 15km/h, P: 14mm, T: 25°C}, {DV: 205°, VV: 15km/h, P: 14mm, PB: 9899hPa}, {DV: 205°, VV: 15km/h, U: 60%, T: 25°C}, {DV: 205°, VV: 15km/h, U: 60%, PB: 9899hPa}, {DV: 205°, VV: 15km/h, T: 25°C, PB: 9899hPa}, {DV: 205°, P: 14mm, U: 60%, T: 25°C}, {DV: 205°, P: 14mm, U: 60%, PB: 9899hPa}, {DV: 205°, P: 14mm, T: 25°C, PB: 9899hPa}, {DV: 205°, U: 60%, T: 25°C, PB: 9899hPa}, {VV: 15km/h, P: 14mm, U: 60%, T: 25°C}, {VV: 15km/h, P: 14mm, U: 60%, PB: 9899hPa}, {VV: 15km/h, P: 14mm, T: 25°C, PB: 9899hPa}, {VV: 15km/h, U: 60%, T: 25°C, PB: 9899hPa}, {P: 14mm, U: 60%, T: 25°C, PB: 9899hPa}, {DV: 205°, VV: 15km/h, P: 14mm, U: 60%, T: 25°C}, {DV: 205°, VV: 15km/h, P: 14mm, U: 60%, PB: 9899hPa}, {DV: 205°, VV: 15km/h, P: 14mm, T: 25°C, PB: 9899hPa}, {DV: 205°, VV: 15km/h, U: 60%, T: 25°C, PB: 9899hPa}, {DV: 205°, P: 14mm, U: 60%, T: 25°C, PB: 9899hPa}, {VV: 15km/h, P: 14mm, U: 60%, T: 25°C, PB: 9899hPa}.

APÊNDICE B - ESQUEMÁTICO DO CIRCUITO ELÉTRICO DOS MÓDULOS WTIMS



APÊNDICE C - ALGORITMO DE DISCRETIZAÇÃO DA BASE DE DADOS DA REDE *MESH*.

Algoritmo 4 Discretização das variáveis climáticas da base de dados da rede *mesh*.

```

1: arquivo = abrir('wtim.data', 'r')
2: linha = arquivo.lerLinha()
3:  $k = 0$ 
4:
5: enquanto linha faça
6:   para  $i = 1$  até 3 faça
7:      $t = t + \text{filtraTemp}(linha)$ 
8:      $u = u + \text{filtraUmid}(linha)$ 
9:      $ps = ps + \text{filtraPres}(linha)$ 
10:
11:   linha = arquivo.lerProxLinha()
12: fim para
13:
14:    $tD_k = \text{discretizaT}(t/3)$ 
15:    $uD_k = \text{discretizaU}(u/3)$ 
16:    $psD_k = \text{discretizaPs}(ps/3)$ 
17:
18:   linha = arquivo.lerProxLinha()
19:    $k = k + 1$ 
20: fim enquanto
21: novoArquivo = abrir('wtim.csv', 'w')
22:
23: para  $i = 0$  até  $k - 1$  faça
24:   novoArquivo.esqueva( $tD[i] + ',' + uD[i] + ',' + psD[i] + ',' + '\n'$ )
25: fim para

```

APÊNDICE D - ARTIGOS PULICADOS

RIBEIRO, J. M.; SILVA, A. C. R.; SANTOS FILHO, T. A.; SILVA, S. F. Aplicação de Regras de Associação em Mineração de Dados para Análise Climática na Região de Catalão - GO - Brasil. In: SEMINÁRIO EM TECNOLOGIA DA INFORMAÇÃO INTELIGENTE, 2018, São Paulo. **Anais** [...]. São Paulo: UNINOVE, 2018. p. 1-478.

SANTOS FILHO, T. A.; RIBEIRO, J. M.; SILVA, A. C. R.; SILVA, S. F.; OLIVEIRA JUNIOR, A. C. Analysis of a ZigBee Wireless Sensor Network Based on the IEEE 1451 Standard. In: XII CONGRESSO BRASILEIRO DE AGROINFORMÁTICA, 2019, Indaiatuba. **Anais** [...]. Indaiatuba: Embrapa, 2019. p. 321-330.