UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM

INSTITUTO DE COMPUTAÇÃO - ICOMP

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

MINERAÇÃO DESCENTRALIZADA DE CORRELAÕES ENTRE DISPOSITIVOS IOT EM AMBIENTES INTELIGENTES

JULHO, 2018

MANAUS - AM

UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM

INSTITUTO DE COMPUTAÇÃO - ICOMP

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

MINERAÇÃO DESCENTRALIZADA DE CORRELAÕES ENTRE DISPOSITIVOS IOT EM AMBIENTES INTELIGENTES

MÁRCIO ANDRÉ DA COSTA ALENCAR

Proposta de dissertação apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas, submetida a exame de qualificação.

ORIENTADOR: PROF. DR. RAIMUNDO DA SILVA BARRETO

JULHO, 2018

MANAUS - AM

Resumo

Identificar os padrões de interação dos usuários em relação aos dispositivos IoT é uma característica esperada dos ambientes inteligentes. Destacar esses padrões, e usa-los como conhecimento para tomada de decisões, pode proporcionar facilidade, conforto, praticidade e autonomia na execução de atividades rotineiras. Embora seja comum em ambientes inteligentes centralizados, a extração informações em ambiente descentralizado e embarcado é um desafio computacional relevante, principalmente quando se considera as restrições de armazenamento e processamento dos dispositivos IoT. Para tanto, esta dissertação descreve um método para minerar correlações implícitas entre os padrões de mudanças de estados dos dispositivos por meio de análises associativas embarcadas. Fundamentado sobre as métricas de suporte (*support*), confiança (*confidence*) e sustentação (*lift*), o método identifica as correlações mais relevantes entre as ações de dois dispositivos e sugere ao usuário a integração entre os mesmos por meio de gatilhos. Os resultados parciais mostram que, em média, as regras mais relevantes para ambas arquiteturas coincidiram em serem as mesmas em 84,25% das ocorrências, Além disto o método proposto identificou correlações relevantes que não foram identificadas pela arquitetura centralizada. Como contribuição, este método salienta que analisar os padrões de mudanças de estado dos dispositivos é uma abordagem eficiente para proporcionar ambiente altamente integrado e inteligente.

Palavras-chaves: Sistemas distribuídos, Regras de associação, Internet das Coisas, Mineração embarcada, Integração inteligente de dispositivos.

Abstract

Identifying patterns and user behavior are an expected feature of IoT devices. Highlighting these patterns and using them as knowledge for decision-making can provide easiness, comfort, practicality and autonomy to execute daily activities. Although it is common in centralized intelligent environments, perform a decentralized and embedded knowledge extractions is still a relevant computational challenge considering the storage and processing constraints of IoT devices. To accomplish that, this dissertation describes a method to mine implied correlations among the patterns of state changes of the devices through embedded associative analyzes. Based on the support, confidence and lift metrics, the method identifies the most relevant correlations between the actions of two devices and suggest the integration between them to users by means of triggers. Partial results show that, on average, the most relevant rules for both architectures coincided with being the same in 84.25% of the occurrences. In addition, the proposed method identified relevant correlations that were not identified by the centralized architecture. As a contribution, this method emphasizes that analyzing the patterns of device state changes is an efficient approach to providing highly integrated and intelligent environment.

***Keywords**: Distributed systems, Association rules, Internet of Things, Embedded data mining, Intelligent integration.*

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

API	APPLICATION PROGRAMMING INTERFACE
AVG	AVARAGE
CR	THE CURVE REGRESSION
FARM	FRESHNESS ASSOCIATION RULE MINING
FHUT	FREQUENT HEAD UNION TAIL
FMUT	MAXIMAL FREQUENT PATTERN
FP-GROWTH	FREQUENT PARTTERN GROWTH
GQM	GOAL, QUESTION AND METRICS
HAC	HIERARCHICAL AGGLOMERATIVE CLUSTERING
HTTP	HYPERTEXT TRANSFER PROTOCOL
ID3	ITERATIVE DICHOTOMISER 3
IOT	INTERNET OF THINGS
JSON	JAVASCRIPT OBJECT NOTATION
JSON-LD	JSON FOR LINKING DATA
KNN	K-NEAREST NEIGHBORS
LFP	LOCAL FREQUENT PARTTERNS
LSHAP	LOOP SCHEDULING WITH HETEROGENEOUS ASSIGNMENT WITH PROBABILITY
MLR	MULTIPLE LINEAR REGRESSION
OACCR	OBTAINING ACCURATE AND COMPREEHENSIBLE CLASSIFICATION RULES
PCA	PRINCIPALS COMPONENTS ANALYSIS
PEP	PARENT EQUIVALENCE PRUNING
PICO	POPULATION, INTERVENTION, COMPARISON, OUTCOMES
RFID	RADIO-FREQUENCY IDENTIFICATION
SLR	SIMPLE LINEAR REGRESSION
SVM	SUPORT VECTOR MACHINE
TITARL	TEMPORAL INTERVAL TREE ASSOCIATION RULE LEARNING
URL	UNIFORM RESOURCE LOCATOR
WOT	WEB OF THINGS
WS	WEB SOCKET
WT	WEB THING
WTC	WEIGHTED TRANSITIVE CLUSTERING
WTE	WEB THING ESTENDIDA
WTM	WEB THING MODEL
WTS	WEB THING SEMANTICA

Capítulo 1

Introdução

Embora não haja uma definição universal, o termo Internet das Coisas (em inglês *Internet of Things* - IoT) geralmente se refere ao cenário em que coisas (objetos, dispositivos, construções, seres, etc.) são capazes de se conectar a uma rede de comunicação e realizar o sensoriamento, processamento, geração, consumo e troca de dados sem nenhuma intervenção humana. Suas aplicações atendem a diversas áreas, como por exemplo, casas inteligentes, agricultura, transportes, segurança, educação, cuidados com a saúde, indústrias e outros, com intuito de melhorar a qualidade de vida das pessoas (EVANS, 2011; ROSE et al., 2015; AL-FUQAHA et al., 2015; MAKOSHENKO e ENKOVICH, 2017; PULIAFITO e ANASTASI, 2017; CAI et al., 2017;).

Segundo a *International Data Corporation,* empresa especializada em consultoria empresarial em tecnologia da informação, em 2017 a indústria investiu US$ 800 bilhões em pesquisas sobre IoT e estima-se que até 2021 irá investir US$1,4 trilhões. Com tais investimentos, projeta-se que até 2020 haja mais de 50 bilhões de dispositivos conectados à internet (EVANS, 2011; ROSE et al., 2015; AL-FUQAHA et al., 2015; CAI et al., 2017).

O contínuo desenvolvimento de tecnologias, componentes, sistemas e infraestrutura ampliam a capacidade de conectividade, armazenamento e processamento em dispositivos inteligentes. Tais capacidades, aliadas às técnicas de mineração de dados e aprendizagem de máquinas, proporcionam mais robustez, autonomia a esses dispositivos. A inserção no mercado de produtos (televisão, micro-ondas, lâmpadas, geladeiras, etc.) com capacidades de conectividade e processamento de dados são suporte a este cenário.

As atividades diárias de uma pessoa, e a forma como ela interage com as coisas, geram um volume de dados nos quais estão inseridas informações relevantes quanto aos seus padrões de uso e preferências. A identificação e extração destas informações implícitas vai ao encontro do potencial esperado para a Internet das Coisas, possibilitando integração entre dispositivos, automatização de atividades e predição de ações ressaltando assim a importância dos estudos nesta área.

1.1. Contexto

A capacidade de prover inteligência e autonomia na IoT recai principalmente sobre a necessidade de identificar atividades, padrões e correlações implícitas nos registros de atividades de cada dispositivo que a compõe. A arquitetura sobre a qual é implementado o ambiente inteligente implica diretamente na técnica de aprendizagem de máquina e mineração de dados utilizados para a extração de conhecimento.

Em uma arquitetura centralizada, as informações coletadas são armazenadas unicamente no dispositivo central. Embora tal arquitetura simplifique o processo de análise de dados, a centralização das informações exige que este dispositivo tenha recursos suficientes para armazenar e processar todos os dados do ambiente, além disso, a relação de dependência para com este nó central apresenta-se como um ponto de vulnerabilidade, o qual inviabilizaria o funcionamento adequado do ambiente inteligente em caso de falha.

A arquitetura descentralizada não demanda a presença de um elemento único de armazenamento e processamento de dados. Problemas de dependência e exigência de muitos recursos seriam inexistentes, em contrapartida, cada dispositivo é responsável oferecer ao usuário todos os mecanismos necessários para configuração, conectividade, controle, armazenamento e processamento de dados. Neste cenário, a falha de um dos dispositivos não afetaria diretamente os demais, destacando-se como um ambiente mais favorável para implementação do ambiente inteligente, além de suportar melhor o acréscimo de novos dispositivos sendo, portanto, uma solução escalável.

O estado da arte em algoritmos de aprendizagem de máquinas (por exemplo, Redes Neurais Profundas) tem o custo de armazenamento e processamento elevados, apresentando-se como desafio computacional significativo em todo o espectro de dispositivos de computação, desde clientes com poucos recursos até servidores em nuvem (CHEN et al., 2015; MOONS e VERHELST, 2017). Uma estratégia para contornar tais adversidades é a utilização de técnicas de mineração de dados, mas especificamente das regras de associação, que buscam identificar padrões de uso similares em um conjunto dados de modo que satisfaça critérios de confiabilidade mínimos (CHEN et al., 2015; TAN et al., 2006). Embora sejam aplicadas frequentemente em bases de dados volumosas, a simplicidade de seus algoritmos possibilita que suas implementações sejam suportadas em sistemas com poucos recursos, tanto de processamento quanto de memória.

Visando explorar as vantagens de uma arquitetura distribuída, além da simplicidade e praticidade dos algoritmos de regras de associação, os estudos conduzidos nesta dissertação buscam desenvolver um mecanismo descentralizado para identificar correlações implícitas entre dispositivos de um ambiente inteligente, por meio de regras de associação, e oferecer ao usuário sugestões de integração entre os mesmos.

1.2. Definição do Problema

Considerando a busca de reduzir os custos de implementação e, até mesmo, evitar o desperdício de recursos (por exemplo, não faz sentido ter um Raspberry Pi para cada lâmpada de uma casa), é essencial que o ambiente inteligente utilize diversos dispositivos de baixo custo. Com os custos mais baixos, é mais fácil integrar muito mais dispositivos IoT. A relação entre custo e quantidade de recursos é diretamente proporcional e, dessa forma, os

dispositivos, como consequência, terão baixa capacidade de armazenamento e de processamento.

Tendo como premissa o cenário exposto, realizar a mineração de dados embarcada em um dispositivo IoT apresenta desafios relevantes tais como:

a) Armazenamento de dados;

b) Conectividade;

c) Comunicação entre dispositivos (interoperabilidade);

d) Processamento de dados para extração de correlações; e

e) Interface de gerenciamento para o usuário.

Cada desafio citado implica em uma série de problemas adjacentes tais como: tipo de estrutura de dados, tecnologia e protocolos usados, sintaxe de comunicação, identificação de outros dispositivos na rede, registros de mudança de estado dos dispositivos, tratamento de requisições, confiabilidade da regra e alta sensibilidade a novos registros.

1.3. Motivação

O interesse da indústria no desenvolvimento de Internet das Coisas impulsiona a fabricação de produtos e tecnologias voltadas para a melhoria da qualidade de vida das pessoas. Através destas tecnologias é possível levar segurança, praticidade e comodidade nas atividades do dia a dia dos usuários. Destaca-se ainda que, para grupos de pessoas que necessitem de cuidados especiais, como idosos e pessoas com restrições física, a Internet das Coisas torna-se mais relevante dando suporte à autonomia e independência dessas pessoas.

Outro fator motivador é a exploração da fronteira do conhecimento em relação a Internet das Coisas. Embora haja muitos estudos voltados para o avanço de ambiente inteligentes utilizando uma arquitetura centralizada, explorar técnicas de mineração de dados embarcadas em arquiteturas descentralizadas para análise de dados limitados é uma abordagem pouco usual no contexto atual em que as redes neurais profundas exigem um grande volume de dados.

1.4. Objetivos

O objetivo principal desta dissertação é demonstrar a capacidade de integração inteligente entre dispositivos na internet das coisas, por meio da identificação descentralizada de correlações implícitas entre seus padrões de mudanças de estados.

Para que tal objetivo seja alcançado, faz-se necessário o cumprimento dos seguintes objetivos específicos:

I. Desenvolvimento um mecanismo descentralizado para identificação das regras de associação mais relevantes (confiáveis) entre dispositivos IoT presentes em uma rede local;

II. Avaliação experimental do mecanismo proposto, em ambiente controlado, quanto à formação de regras de associação entre dispositivos IoT;

III. Demonstração da similaridade entre as regras de associação geradas em ambiente descentralizado e as geradas em ambiente centralizado; e

IV. Avaliação experimental do mecanismo proposto, em cenário real de uso.

1.5. Organização do Trabalho

No **Capítulo 1** é apresentada introdução deste trabalho através da contextualização, definição do problema, as motivações e os objetivos desta dissertação.

O **Capítulo 2** enfatiza a fundamentação teórica, explorando assuntos acerca de sistemas embarcados, arquiteturas, modelos, técnicas e algoritmos utilizados durante o desenvolvimento desta dissertação, de tal forma que sejam explanados conceitos e definições básicas para compreensão da metodologia apresentada.

Os **Capítulos 3 e 4** descrevem, respectivamente, o protocolo da revisão sistemática e a discussão dos trabalhos correlatos referentes ao objeto de pesquisa nesta dissertação. Ambos capítulos possibilitam uma visão geral das técnicas e algoritmos usado para solucionar problemas similares aos apresentados durante a definição do problema.

O método proposto é detalhado no **Capítulo 5** de tal forma que é possível observar, inicialmente, uma visão geral do mecanismo e, em um segundo momento, as particularidades do funcionamento individual e colaborativo dos dispositivos.

Para a validação do modelo, no **Capítulo 6**, são apresentados o método de avaliação, os parâmetros definidos e os *datasets* usados. Além destes, apresentam os resultados parciais obtidos pela aplicação das técnicas definidas no Capítulo 5.

Durante o **Capítulo 7** são discutidas as considerações finais na qual são apresentadas as limitações do método proposto, os próximos passos, descrevendo os ambientes nos quais serão realizados os experimentos em cenários reais de uso bem como o cronograma para execução de tais atividades

Capítulo 2

Conceitos e Definições

Para melhor compreensão dos elementos que compõem este estudo, faz-se necessário o entendimento prévio de alguns mecanismos, técnicas e tecnologias, como estas se adequam para o desenvolvimento dos experimentos e como corroboram alcançar os objetivos desta pesquisa.

Neste capítulo apresentam-se conceitos sobre Sistemas Embarcados, apontando suas características, particularidades e desafios de planejamento, desenvolvimento e implementação. Posteriormente, apresenta-se uma consequência do contínuo avanço dos Sistemas Embarcados, a Internet das Coisas. São apresentados conceitos, arquiteturas e soluções deste novo paradigma o qual objetiva melhorar a qualidade de vida das pessoas. Também é incorporado a este capítulo conceitos sobre Mineração de dados e Aprendizagem de Máquinas e suas aplicações no contexto de Internet das Coisas e Sistemas Embarcados, focando-se principalmente em algoritmos de Regras de Associação. Finalmente, uma breve descrição sobre análise probabilística de eventos.

2.1. Sistemas Embarcados

Embora haja diversas definições, para Health (2003), sistemas embarcados é um sistema baseado em microprocessador desenvolvido para controlar uma funcionalidade ou um conjunto de funções específicas não programadas pelo usuário final. Consoante a tal definição, Barr e Massa (2009) definem como uma combinação de *hardware* e *software* - e talvez periféricos adicionais - projetados para executar uma função dedicada.

Os sistemas embarcados estão presentes em muitos objetos de nosso dia a dia tais como, micro-ondas, televisão, condicionador de ar, geladeiras entre outros. Por se tratarem de sistemas com funcionalidades específicas é possível encontrar algumas restrições, como as citadas por Berguer (2002):

- São sensíveis aos custos: dimensão, quantidade e qualidade de componentes, conectores e periféricos usados;
- Possuem restrições quanto ao consumo de energia: devem trabalhar confiavelmente por longos períodos com fonte limitada de energia;
- Operam em ambientes com condições extremas: estão em todas as partes, logo, estão sujeitos às variações ambientais (calor, frio, vibrações, humidade, etc.);
- Armazenam todo seu código na ROM: A capacidade de armazenamento não volátil destes sistemas é mais limitada que a de computadores de propósito geral;

- Requerem ferramentas e métodos para serem projetados de forma eficiente: emuladores, simuladores, *debuggers* são ferramentas que auxiliam durante o processo de desenvolvimento e validação do sistema embarcado;

Além das limitações destacadas por Berguer (2002), podemos ressaltar a limitação de processamento. Por se tratar de um sistema que atende uma funcionalidade específica, o uso de micro controladores com alto poder de processamento é financeiramente desfavorável, além de haver desperdício de recursos.

O contínuo desenvolvimento de sistemas embarcados, a facilidade de acesso à componentes e a crescente acessibilidade de serviços de rede impulsionaram o crescimento no número de dispositivos conectados à internet. Este crescimento deu suporte a este novo paradigma conhecido como Internet das Coisas, o qual, abre as portas para inovações gerando novas formas de interação entre as pessoas e os seres humanos e possibilitando a construção de cidades inteligentes, infraestruturas e serviços para melhorar a qualidade de vida e a utilização dos recursos.

Para Buuya e Dasterdi (2016), a concretização da interoperabilidade entre esses diversos sistemas embarcados e dispositivos de fabricantes diferentes é necessário que as empresas entrem em acordo quanto à pilha de protocolos, que envolve uma série de aspectos, tecnologias e padrões. Embora tal processo seja complexo por envolver muitas variáveis, como quantidade de empresas, economia global, interesses empresariais e custos de fabricação, vários consórcios, órgãos e grupos de pesquisas apresentam modelos que possibilitam tal interoperabilidade.

Um destes modelos é a *Web* das Coisas (em inglês *Web of Things* - WoT), apresentado pela W3C (*World Wide Web Consortium*), que busca descrever um modelo e uma API (*Application Programming Interface*) *Web* a serem seguidas por quaisquer fabricantes que desejem desenvolver um produto, dispositivo, serviço ou aplicação para WoT.

2.2. Web das Coisas (WoT)

Distanciando-se das demais propostas que buscam gerar novos protocolos e padrões para interoperabilidade, a WoT faz uso de uma estrutura amplamente difundida, reduzindo sua complexidade e expandindo sua compatibilidade por meio dos padrões *web*. Cada objeto inteligente na WoT é identificado como *Thing* ou *Web Thing* (WT), que é uma representação digital de um objeto físico acessível por meio de uma *API RESTful*, embarcada ou não. *RESTful* é um estilo arquitetônico que possibilita, através de uma representação de uma interface simples e o protocolo HTTP, a interoperabilidade entre sistemas.

2.2.1. Padrões de Integração

Embora a WoT especifique três formas de conectividades entre dispositivos - *Direct Connectivity, Gateway Based Connectivity, Cloud Based Connectivity* – apenas a primeira apresenta a mesma estrutura contemplada nesta proposta.

O padrão *Direct Connectivity* define que cada *Web Thing* possui sua API para a qual os clientes devem enviar suas solicitações. O cliente e a WT podem estar na mesma rede ou em redes diferentes, modificando-se apenas a URL para a qual o cliente deve enviar sua requisição. A Figura 1 apresenta este modelo de conectividade.

Figura 1. Modelo *Direct Connectivity* entre clientes *Web Thing*. [Adaptado de: http://model.webofthings.io/]

Embora sua topologia de rede seja centralizada (*Wi-Fi Router*), os serviços oferecidos por cada *Web Thing* são independentes uns dos outros possibilitando aos clientes acessarem suas funcionalidades diretamente através da rede, local ou não. Outra característica importante é que a interrupção de um dos dispositivos inteligentes não afeta os demais.

O padrão *Gateway Based Connectivity (Figura 2)* geralmente é usado quando o dispositivo (coisa) não dispõe dos recursos necessários para prover uma API embarcada. Dessa forma uma WT intermediária expõe a API atuando como um *proxy* ou *gateway* (dependendo da complexidade do sistema), intermediando a comunicação entre a coisa e outros sistemas.

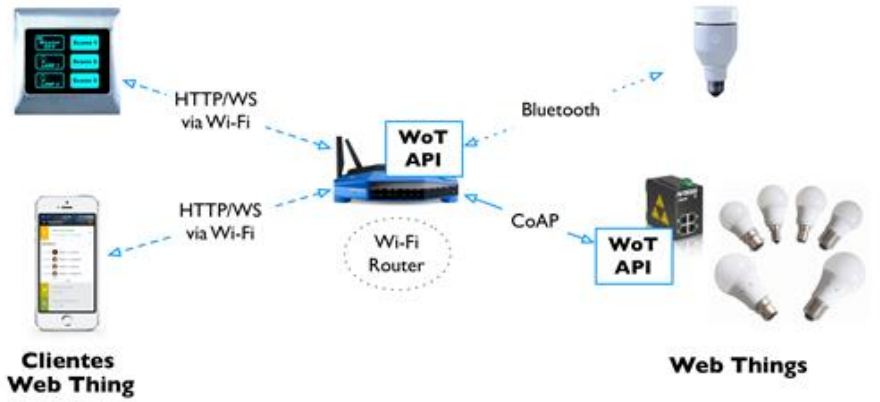

Figura *2.* Modelo *Gateway Based Connectivity* entre dispositivos. [Adaptado de: http://model.webofthings.io/]

O terceiro padrão, *Cloud Based Connectivity* (Figura 3) é similar ao *Gateway Based Connectivity,* porém, neste caso, o *gateway* é um serviço em nuvem e sua comunicação exige conectividade com a internet.

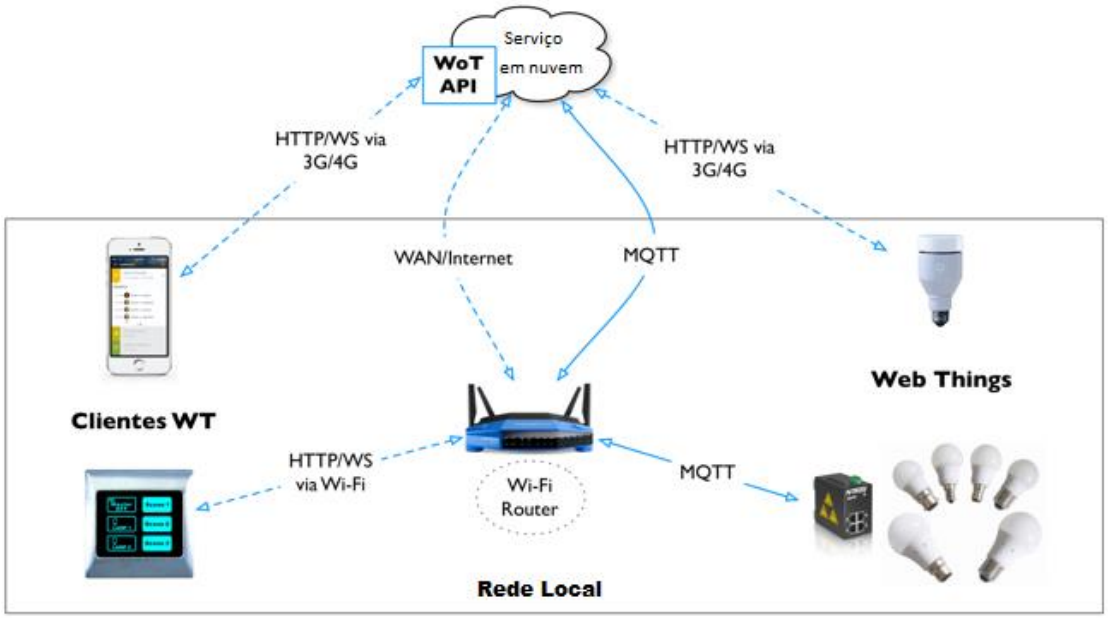

Figura 3. Modelo *Cloud Based Connectivity*. [Adaptado de: http://model.webofthings.io/]

Os modelos de conectividade descritos na *Web* das Coisas possibilitam aos desenvolvedores atender diversos cenários de aplicações as quais são essenciais para implementação de um ambiente inteligente composto de diversos agentes.

2.2.2. Requisitos para WoT

Algumas especificações, convenções de nomes e recomendações para transformar um *web server* em *Web Thing* são apresentadas a seguir:

- Deve suportar ao menos HTTP/1.1 e, quando possível, suportar HTTP/2;

- Deve ter seus recursos acessíveis via URL HTTP exclusiva, sendo capaz de responder uma requisição HTTP GET para seu endereço raiz (IP ou Nome) sobre a porta padrão (80 para HTTP, 443 para HTTPS);
- Deve suportar requisições HTTP do tipo *GET*, *POST*, *PUT* e *DELETE* para operações de leitura, criação, atualização e remoção respectivamente;
- Deve implementar os códigos de status HTTP 200 (*Success*), HTTP 400 (*Bad Request*) e HTTP 500 (*Internal Server Error*);
- Deve suportar JSON;
- Deve suportar *HTTP GET* em sua URL raiz.

2.2.3. Modelo *Web Things* (WTM)

Seguidas as convenções da Seção 2.2.2, é possível ler e trocar dados com qualquer outra entidade na WoT, porém, ainda não é possível a compreensão desses dados. Dessa forma, a *Web Thing Model* (WTM) especifica um modelo, o conteúdo JSON e API REST que uma WT deve implementar.

Seguir as especificações do WTM transforma uma *Web Thing* em *Web Thing Estendida* (WTE) e a implementação de sua semântica a transforma também em uma *Web Thing Semântica* (WTS). A Figura 4 apresenta os escopos evolutivos (níveis do modelo WTM) de um simples Servidor *Web* a uma *Web Thing* Semântica.

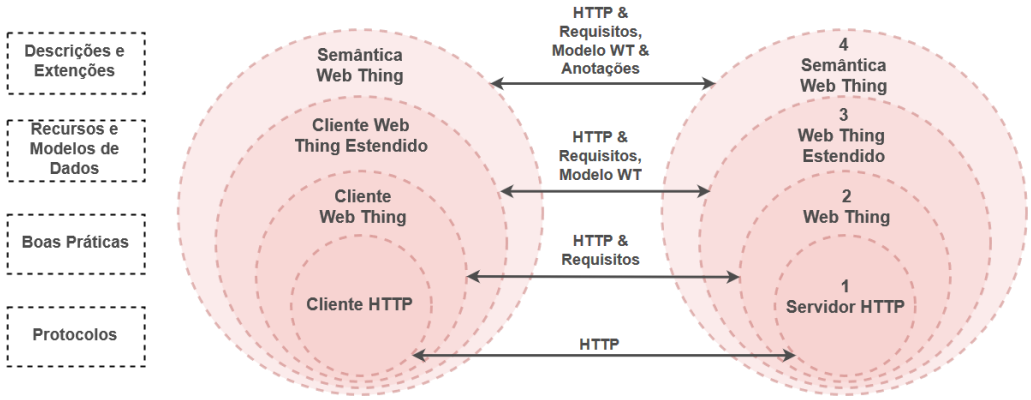

Figura 4. Níveis do Modelo *Web Thing.* [Adaptado de: http://model.webofthings.io/]

Cada WTE deve apresentar uma URL específica para cada um dos seus recursos, conforme apresentado na Tabela 1. É encorajado o uso de uma estrutura lógica de árvores que permitam a navegação pelos recursos mantendo pequeno o volume de dados no pacote JSON. O conteúdo JSON pode conter campos adicionais, dependendo do tipo de recursos que está sendo considerado.

Tabela 1. Recursos exigidos como resposta pelo WTM para um *Web Thing* Estendido.

CAMPO	TIPO	DESCRIÇÃO
Id	*string*	Identificação única da *Web Thing*
createAt	*string*	*Timestamp* de criação
updateAt	*string*	*Timestamp* da última atualização
name	*string*	Nome amigável da *Web Things*
description	*string*	Informações gerais
tags	*[string]*	Lista de *tags* associadas ao dispositivo
customFields	*object*	Objeto JSON com par "chave-valor" para armazenamento de informações personalizadas
links	*object*	Objeto JSON que lista os sub recursos a que o recurso atual se liga

Por meio desses links relativos é possível acessar informações, propriedade, ações e outros recursos, da *Web Thing*. Os links exigidos para uma WTE são apresentados na Tabela 2. Para o suporte para WTS deve ser implementado por meio de JSON-LD reverenciado via HTTP *Link Header*.

Tabela 2. URL REST exigidos para um Web Thing Estendido

URL REST	AÇÃO
{wt}	Retorna um objeto que é sua representação
{wt}/model	Retorna um objeto contendo o modelo da Web Thing
{wt}/properties	Retorna um vetor de propriedade que o recurso inicial possui
{wt}/properties/{id}	Retorna uma lista de valores recentes da propriedade
{wt}/actions	Retorna um vetor de descrições ações que o recurso pode realizar
{wt}/actions/{id}	Retorna um vetor que lista as execuções recentes de uma ação específica
{wt}/actions/{id}/{actionId}	Retorna o status de uma ação ou 404 caso a id da ação não for encontrada.

2.3. Mineração de Dados

Mineração de dados, também conhecida como descoberta de conhecimento a partir de dados, é o processo de automatizado, ou por conveniência, de extração de padrões que representam conhecimento implicitamente armazenados ou capturados em base de dados (HAN et al., 2012).

Esse processo, como apresentado por CHEN et al. (2015), é dividido em várias etapas:

I. Preparação de dados: que consiste em realizar a limpeza de ruídos, seleciona um conjunto de dados dentro de uma base maior ou integrar esses dados com outros dados;

II. Mineração de dados: Neste processo são executados os algoritmos que buscam encontrar e validar padrões de conhecimentos descobertos; e

III. Apresentação de dados: Visualizar os dados e representar o conhecimento extraído para o usuário.

As funcionalidades da mineração de dados incluem classificação, agrupamento, análise associativa, análise de séries temporais e análises de *outliers*. Destas funcionalidades podemos destacar a análise associativa, utilizada neste trabalho.

2.3.1. Análise Associativa

Segundo Chen et al. (2015), a análise associativa é a descoberta de regras de associação que exibem condições de atributo-valor que frequentemente ocorrem juntas em um determinado conjunto de dados. Tais condições pode ser representadas por meio de regras expressas no formado "$A \Rightarrow B[métricas]$" onde A define a premissa da regra (antecedente), B é a conclusão da regra (consequente) e "[métricas]" são os valores que permitem quantificar a inferência $A \Rightarrow B$.

Geralmente as uma regra de associação é considerada interessante se ela satisfaz um limiar mínimo de suporte e confiança e podem ser representadas formalmente por:

$$support\ (A \Rightarrow B) = P(A \cup B)$$

$$confidence\ (A \Rightarrow B) = P(B \mid A)$$

Suporte de uma regra faz referência frequência de um conjunto de dados (A e B) em uma base de dados, enquanto a confiança expressa a probabilidade condicional de B em relação à A, ou seja, quão frequente B é em relação à A. Uma terceira métrica chamada "*lift",* apresentada por Han *et al.* (2012), define o grau de correlação entre o antecedente e o consequente. Tal métrica é calculada por meio do suporte e confiança como segue:

$$lift(A \Rightarrow B) = \frac{confidence(A \Rightarrow B)}{support(B)}$$

Esta métrica sugere que a ocorrência do itemset A é independente da ocorrência do itemset B se $P(A \cup B) = P(A)$ do contrário, A e B são dependentes e correlacionadas.

2.3.2. Regra de Associação Apriori

Proposto por Agrawal e Srikant (1994), o algoritmo de regras de associação *apriori* busca identificar *itemsets* frequentes para regras de associação booleanas. Este algoritmo adota a propriedade *apriori* na qual considera que os *k-itemsets* (*itemset* de tamanho k) são usados para explorar (k+1)-*itemsets.*

Formalmente, esta propriedade é definida pela seguinte observação: Se um *itemset I* não satisfaz um limite mínimo de suporte então *I* não é frequente, ou seja, *P(I) < min_sup*. Se um *itemset A* é adicionado ao *itemset I* então o conjunto resultante *I U A* não pode ocorrer com uma frequência maior que *I*, logo *I U A* também não é um *itemset* frequente, ou seja, *P (I U A) < min_sup*. Esta propriedade pertence à classe de propriedades conhecidas como anti-monotonicidade a qual define que se um conjunto *(I)* não passar em um dado teste então um

superconjunto *(I U A)* também não passará. Em síntese, um *itemset* é frequente “se e somente se” o subconjunto deste *itemset* também for frequente. Dessa forma o algoritmo de Regra de Associação Apriori executa duas operações básicas:

1. ***Join Step***: Para encontrar $L_{k,}$, um conjunto de candidatos de *k-itens* é gerado unindo-se L_{k-1} com si mesmo.
2. ***Prune Step***: Em cada iteração todo os itens do conjunto devem atender ao limiar mínimo das métricas, caso contrário não pertencerá ao superconjunto.

Embora simples, o algoritmo de regra de associação possui um custo computacional elevado, no sentido de que o primeiro passo do algoritmo, *Join Step,* realiza a combinação de todos os itens do conjunto para formação dos superconjuntos em cada iteração. Dessa forma o passo 2, *Prune Step*, ameniza tal complexidade reduzindo o número de candidatos para a próxima iteração. Tal processo é ilustrado na Figura 5.

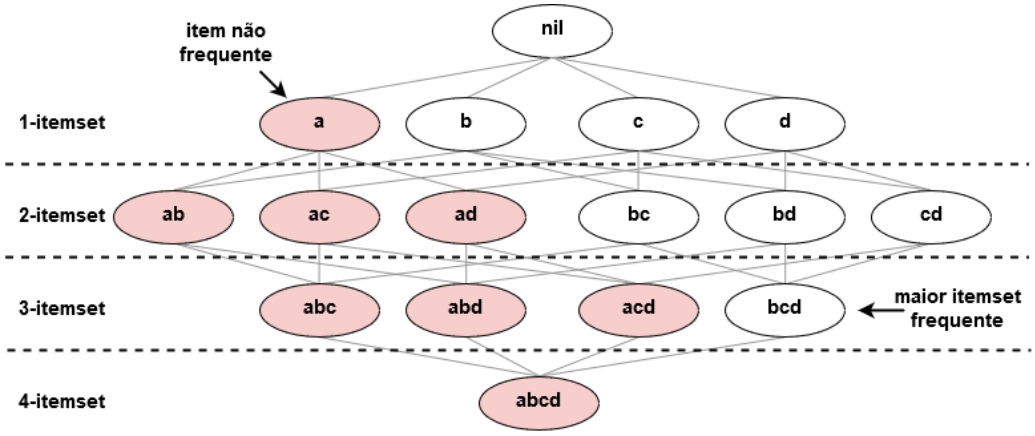

Figura 5. Ilustração a execução do algoritmo de regra de associação.

No exemplo, é possível identificar o processo de *Join Step,* em que cada linha representa uma iteração do algoritmo iniciando de um conjunto vazio até a geração do maior itemset mais frequente (3-itemset). É possível observar também o impacto do processo de *Prune Step* em que todos os conjuntos que possuem o item não frequente (a) são desconsiderados, reduzindo o custo de processar mais da metade de conjuntos possíveis para as próximas iterações. O processo se repete à cada iteração até que o maior itemset frequente (bcd) seja encontrado.

2.4. Análise Probabilística

A teoria matemática da probabilidade nos dá as ferramentas básicas para a construção e análise de modelos matemáticos para fenômenos aleatórios. Ao estudar um fenômeno aleatório, estamos lidando com um experimento cujo resultado não é previsível antecipadamente. Experiências deste tipo que imediatamente vêm à mente são as que surgem nos jogos de azar. De fato, o primeiro desenvolvimento da teoria da probabilidade nos séculos XV e XVI foi motivado por problemas desse tipo (SOON, 2004).

A importância do estudo de eventos aleatórios está nas inferências que são possíveis realizar sobre os eventos que podem ocorrer. Caeiro (2009) define que uma experiência aleatória é um evento cujo resultado é desconhecido (antes de sua realização), apesar de se conhecerem todos os possíveis resultados, como por exemplo:

E_1 :*Lançamento de uma moeda e observação da face voltada para cima*;

O espaço de resultados Ω é discreto se tem um número numerável de elementos, se Ω contém um intervalo de números reais, então o espaço de resultados é contínuo.
Considerando o exemplo $E_{1,}$ temos:

$$E_{1:}: \Omega = \{cara, coroa\}$$

Diz-se que o espaço de resultado Ω é discreto se há um número, finito ou infinito, numerável de elementos. Se Ω contém um intervalo, finito ou infinito, de números reais, então o espaço de resultados é contínuo numerável de elementos. Se Ω contém um intervalo (finito ou infinito) de números reais.

Cada elemento de um espaço de resultados Ω é identificado como ponto amostral ou acontecimento. Em muitos experimentos é necessário identificar qual a probabilidade de ocorrência de um dado acontecimento.

O Regra de Laplace define que, em um experimento que possua N possíveis resultados (|Ω|) mutuamente exclusivos, é possível calcular a probabilidade de um acontecimento A de tal forma que:

$$P(A) = \frac{Na}{N} = \frac{números\ de\ elementos\ em\ \Omega\ iguais\ à\ A}{número\ de\ elementos\ em\ \Omega}$$

A combinação de dois eventos gera também uma combinação de seus espaços de resultados, como exemplo: Temos o lançamento de duas moedas e a observação de suas faces voltadas para cima, seu espaço de resultados é:

$$E_1 \cup E_2 = \Omega = \{(cara, cara), (coroa, coroa), (cara, coroa), (coroa, cara)\}$$

Considerando a combinação de espaços de resultados para o experimento, temos que $\Omega = (E_1 \cup E_2)$, logo, |Ω| = 4. A probabilidade de ambas moedas caírem com a face 'cara' voltada para cima é:

$$P(A) = \frac{Na}{N} = \frac{1}{4} = 0{,}25 = 25\%$$

A probabilidade é de 25% pois há apenas um único acontecimento em Ω igual ao resultado dentre os 4 possíveis. Esta análise possibilita avaliar em um conjunto de espaço amostral a probabilidade de um dado acontecimento ocorrer. Este espaço amostral pode se ampliar ou reduzir dada as características do experimento, o número de repetições e as combinações dos espaços de resultados individuais.

2.5. Resumo

No Capítulo 2 foram descritos todos os conceitos, definições e teorias que dão suporte ao desenvolvimento da solução apresentada nesta dissertação. Foram apresentadas as principais características de sistemas embarcados e suas restrições quanto ao armazenamento, processamento e consumo de energia. Apresentou-se também o modelo Web das Coisas que possibilita a para integração e interoperabilidade entre dispositivos na Internet das Coisas, além destes, explanou-se sobre as técnicas de mineração de dados por meio de regras de associação *apriori* e análise probabilística de eventos.

Capítulo 3

Revisão Sistemática da Literatura

Como base para definição do algoritmo de regra de associação utilizado no método proposto (Capítulo 5), realizou-se uma revisão sistemática na qual o objeto da pesquisa era avaliar estudos que aplicassem algoritmos de regras de associação em sistemas embarcados no contexto da Internet das Coisas. Não se limitando a isto, uma busca por aproximação possibilitou conduzir o leitor a ter uma compreensão sobre como é possível realizar a extração de conhecimento em sistemas embarcados abordando técnicas de mineração em base de dados limitadas.

3.1. Protocolo

O protocolo foi elaborado conforme especificado em: BIOLCHINI et al. (2005), MAFRA E TRAVASSOS (2006), e KITCHENHAM (2004) e tem suas questões de pesquisas esquematizada a partir do paradigma GQM (*goal, question, and metric*) descrito em BASILI et al. (1994):

Tabela 3. Objetivo definido a partir do paradigma *goal, question and metric*.

Analisar	Algoritmos de regras de associação
Com o propósito de	Identificar métodos, mecanismos, técnicas e plataformas.
No que diz respeito a	Aplicações em ambiente embarcado e/ou com base de dados limitadas
Do ponto de vista do	Pesquisador
No contexto	Acadêmico

3.2. Questões de Pesquisa

Baseado nas informações definidas na Tabela 3 foram levantadas 5 questões de pesquisas além da questão principal.

- **Questão principal:** Quais os principais algoritmos de regras de associação usados em ambiente embarcado ou técnicas de extração de conhecimento em base de dados limitadas?
- **Questão 1**: Quais algoritmos possuem experimentos em ambiente embarcado?
- **Questão 2**: Quais algoritmos possuem experimentos em base de dados limitadas?
- **Questão 3**: Sobre quais as plataformas / hardwares foram aplicados estes algoritmos?
- **Questão 4**: Quais os mecanismos / técnicas foram usados para otimização das regras de associação?

- **Questão 5**: Quais algoritmos/mecanismos/técnicas foram usados para correlacionar estados de dispositivos?

3.3. Fontes e *String* de Busca

A biblioteca digital usada para obtenção dos artigos para esta revisão sistemática foi a SCOPUS: <https://www.scopus.com/>.

Os critérios para sua seleção foram:

- Consulta de artigos em biblioteca digitais;
- Disponibilidade de consulta de artigos através da web;
- Presença de mecanismos de busca através de palavras-chaves e que suportem a *string* de busca;
- Ter os estudos disponíveis na língua inglesa;

A *string* de busca foi definida a partir das questões de pesquisa e do padrão PICO (*population, intervention, comparison, outcomes*) (KITCHENHAM e CHARTERS, 2007), conforme a estrutura abaixo:

- **População**: Algoritmos de regras de associação.
- **Intervenção**: Em ambiente embarcado ou base de dados limitadas.
- **Comparação**: Não se aplica.
- **Resultados**: Algoritmos, métodos, mecanismos e técnicas.

A Tabela 4 apresenta os resultados obtidos durante execução processo iterativo para obtenção da *string* de busca ideal

Tabela 4. Número de artigos obtidos durante a definição da *string* final

ID	STRING DE BUSCA	ARTIGOS
1	("association rules" OR "associative analysis" OR "associative rule mining" OR "temporal association rule" OR "temporal relation")	16309
2	("association rules" OR "associative analysis" OR "associative rule mining" OR "temporal association rule" OR "temporal relation") AND ("embedded" OR "constrain* data*" OR "limit* data*" OR "small data*" OR "tiny data*")	261
3	("association rules" OR "associative analysis" OR "associative rule mining" OR "temporal association rule" OR "temporal relation") AND ("embedded" OR "constrain* data*" OR "limit* data*" OR "small data*" OR "tiny data*") AND ("algorithm*" OR "mechanism*" OR "techniq*" OR "method*")	227

A seguir é apresentada a *string* final, no padrão SCOPUS, utilizada no dia 18/06/2018 para obtenção dos artigos avaliados nesta revisão sistemática:

(TITLE-ABS-KEY ("association rules" OR "associative analysis" OR "associative rule mining" OR "temporal association rule" OR "Temporal relation") AND TITLE-ABS-KEY ("embedded" OR "constrain data*" OR "limit* data*" OR "small data*" OR "tiny data*") AND TITLE-ABS-KEY ("algorithm*" OR "mechanism*" OR "techniq*" OR "method*"))*

3.4. Critérios de Inclusão e Exclusão

Para a seleção dos artigos foram adotados os seguintes critérios de inclusão:

- Mineração de dados em ambiente embarcado;
- Extração de conhecimento em bases com poucos dados;
- Regras de associação para correlacionar dispositivos;

Para os critérios de exclusão, foram consideradas as seguintes características:

- Uso de um extenso volume de dados;
- Coletânea de publicações, exceto *surveys*;
- Trabalho não aplicável à nenhum critério de inclusão;
- Publicação não disponível;

É importante destacar que, embora o estudo desta dissertação seja dentro do contexto de *Internet das Coisas*, durante o processo de revisão sistemática, os critérios de inclusão consideram estudos que são de outras áreas. Isso permitiu explorar estudos que possuem características similares às apresentadas na Seção 1.2

3.5. Extração de Informações

Os critérios de inserção e exclusão foram aplicados em dois momentos de filtragem:

I. Avaliação do Título, Resumo e Palavras-chaves;
II. Título, Resumo, Palavras-chaves, Fonte de publicação, Autores, Objetivos, Ano de publicação, Número de citações, Fator de impacto, Algoritmos usados, Técnicas envolvidas, Otimização, Plataforma/Ambiente de execução, Tamanho da base de dados, Comentários, Trabalhos futuros;

3.6. Resultados

Após a definição da *string* de busca, sua aplicação na ferramenta de busca da biblioteca Scopus retornou 227 dos quais 25 eram duplicados. Estudos válidos (202) apresentam sua distribuição ao longo dos anos na Figura 6.

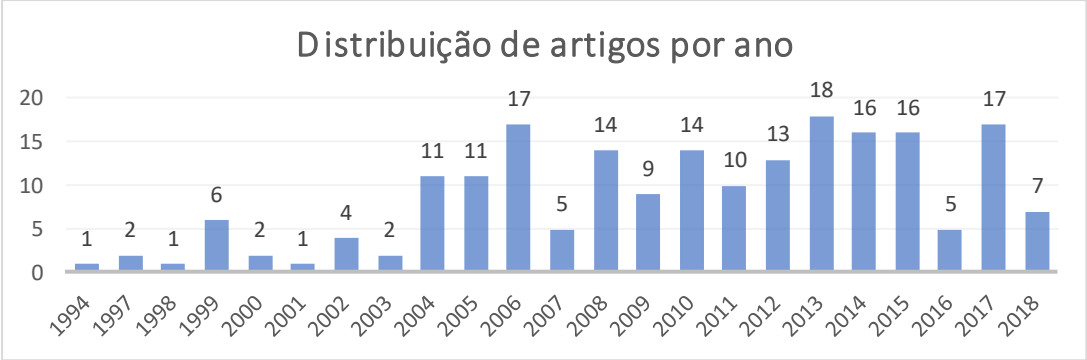

Figura 6. Distribuição de publicações ao longo dos anos.

É possível observar que a partir de 2004 a quantidade de artigos cresceu significativamente e apresenta um pico a cada dois anos aproximadamente.

Durante a aplicação dos critérios de inclusão e exclusão para o primeiro filtro, dos 202 artigos válidos, 114 se enquadraram nos critérios de seleção enquanto 88 foram rejeitados por se enquadrarem em algum critério de exclusão. Durante a análise mais profunda dos estudos no segundo filtro, 15 estudos apresentaram temáticas consoantes às questões de pesquisas da revisão.

A execução da revisão sistemática foi executada com auxílio da ferramenta Start (Zamboni *et al*, 2010). A revisão sistemática completa pode ser encontrada em (ALENCAR, 2018). Neste endereço é possível obter a lista de artigos (*.*bib*) obtida através da string de busca no Scopus, o arquivo de revisão sistemática (*. start), as fichas de extrações e o relatório geral da revisão gerado automaticamente pelo Start.

Como principal contribuição a revisão foi capaz de responder as questões de pesquisas definidas na Seção 3.2:

- **Questão principal: Quais os principais algoritmos de regras de associação usados em ambiente embarcado ou que tratam de bases de dados limitadas?**

  - Apriori, TITArl, JRip, FP-Growth, LFP, FARM, LSHAP

- **Questão 1: Quais algoritmos possuem experimentos em ambiente embarcado?**

  - Apriori, TITArl, JRip, FP-Growth, LFP, FARM, LSHAP

- **Questão 2: Quais algoritmos possuem experimentos em base de dados limitadas?**

  - Apriori, TITArl

- **Questão 3: Sobre quais as plataformas / hardwares foram aplicados estes algoritmos?**

- Cartão RFID, Processadores, Computador de propósito geral

- **Questão 4: Quais os mecanismos/técnicas foram usados para otimização das regras de associação?**

  - Fila Circular, CVI (*Cluster Validity Index*), *Weighted transitive clustering* (WTC), Regressão Linear, *Random Forest*, PCA, *TreeNet*, OACCR (*Obtaining Accurate and Comprehensible Cassification Rules*), GPDCM (G*enetic Programming Data Construction Method*), Effrom's Bootstrap, TinyDB, SPIRIT, WARM, Escalonamento Circular, *Prefetching*, *K-Means Clustering*, Análise de conceito formal

- **Questão 5: Quais algoritmos/mecanismos/técnicas foram usados para correlacionar estados de dispositivos?**

  - Apriori, TITArl

As questões de pesquisa foram respondidas satisfatoriamente, havendo um ponto de convergência entre os estudos analisados. Dos quinze trabalhos identificados, nove utilizaram diretamente o algoritmo de Regra de Associação Apriori, geralmente associado a técnicas probabilísticas e/ou de reamostragem para validarem seus resultados enquanto os demais desenvolveram algoritmos baseado no Apriori, otimizando desempenhos e ou ajustando-o a uma instância do problema de padrões frequentes.

3.7. Resumo

O processo de revisão sistemática, base deste artigo, possibilitou a identificação de técnicas, métodos e mecanismos relevantes para extração de conhecimentos em base de dados limitadas. Não se limitando a isto, foi possível observar a versatilidade dos algoritmos de regras de associação os quais se estendem à solução de problemas em áreas distintas do conhecimento como exatas, humanas, biológicas e outras. Tal levantamento possibilitou a definição do algoritmo ideal para a proposta apresentada nesta dissertação.

Capítulo 4

Trabalhos Correlatos

Os trabalhos correlatos apresentam os resultados obtidos durante a revisão sistemática, descrita no Capítulo 3, que possibilitou a avaliação de estudos cujos autores fizeram uso de algoritmos de regras de associação para extração de correlações em base de dados limitadas (com poucos registros), independentemente de sua área de aplicação. Os critérios levaram em consideração, estudos cuja implementações buscavam identificar correlações de sensores em ambientes inteligentes e/ou embarcados. Tais critérios possibilitaram uma observação ampla e representativa dos mecanismos utilizados para este tipo de problema.

Os estudos identificados foram divididos em duas seções: estudos com regras de associação apriori e estudos com algoritmos baseados em árvore. Em cada seção os estudos foram subdivididos em de acordo com sua otimização e técnicas envolvidas.

Nas considerações finais deste capítulo serão destacados quais métodos, técnicas e algoritmos possuem maior relevância para o objetivo geral desta pesquisa.

4.1. Estudos com Regras Associação *Apriori*

Usado em 9 (nove) dos 15 (quinze) estudos para extração de informações de uma base de dados limitadas da Revisão Sistemática da Literatura, os estudos deste subitem foram agrupados considerando suas contribuições, ou seja, os que apresentaram algum tipo otimização em relação a outras técnicas (comparativos) e os que não apresentaram (não comparativos) para solução dos problemas especificados.

A Tabela 5 apresenta os estudos que aplicaram regras de associação *apriori,* porém, que não realizaram estudos comparativos com outras técnicas.

Tabela 5. Estudos não comparativos que aplicaram regras de associação apriori.

AUTOR	TÉCNICAS ENVOLVIDAS
Pal . *et al*. (2017)	-
Lynden, S. (2017)	-
Karimi-Majd, A.-M. and Mahootchi, M. (2015)	*Cluster Validity Index*
Smith, M.R *et al.* (2009)	*Efron's Bootstrap*
Mori, T. *et al*. (2005)	*K-means Clustering*
McArthur, D. P. *et al*. (2012)	Análise de conceito formal

Dos estudos apresentados na Tabela 5 apenas um faz a aplicação direta do algoritmo enquanto os demais buscaram usar outras técnicas (pré ou pós mineração) para otimizar, validar e/ou explorar melhor os resultados.

Pal *et al*. (2017) buscaram, por meio de geração de invariantes, identificar em um sistema ciber-físico (ambiente controlado por sistemas computacionais) possíveis ataques à uma estação de tratamento d'água. Tal estudo não apresentou otimização ou alguma técnica envolvida uma vez que se limitou à geração de invariantes, por meio do algoritmo de regra de associação *apriori*, correlacionando os estados dos sensores durante um ataque ao sistema. Durante os experimentos foram identificadas 11.500 regras de associação para os 51 sensores, logrando sucesso em responder sua questão de pesquisa ao constatar tais invariantes por meio das regras geradas em um dado espaço de tempo (durante o ataque).

Lynden (2017) apresentou uma análise de URLs semânticas para suportar a vinculação automatizada de dados estruturados na Web. Embora seu estudo tenha usado técnicas de aprendizagem de máquina, as mesmas não foram usadas para otimizar ou complementar os resultados da aplicação das regras de associação *apriori*, pelo contrário, o algoritmo *apriori* foi aplicado para correlacionar o conhecimento obtido com uma base de dados de URLs (DBPedia) possibilitando, na web semântica, identificar páginas correlatas de forma mais eficiente. Os testes foram executados em uma base de dados com 5.000 URLs, uma quantidade limitada quando se comparada ao volume de URLs existentes na internet.

Seguindo a mesma estratégia de Lynden, Karimi-Majd e Mahootchi (2015), Smith *et al.* (2009), Mori *et al.* (2005) e McArthur *et al.* (2012) também apresentaram estudos que, embora não ofereçam otimização, combinam outras técnicas à regra de associação *apriori* para serem bem-sucedidos em suas pesquisas.

Karimi-Majd e Mahootchi (2015) apresentaram uma metodologia (ilustrada na Figura 7) que possibilita correlacionar informações de múltiplas fontes de dados para geração de novos serviços aos consumidores. Para tal, realizou-se a fusão (pré-processamento) de três bases de dados distintas e, posteriormente, a mineração de regras usando o algoritmo *apriori*. Finalmente, as regras extraídas são agrupadas usando a técnica *Cluster Validity Index*

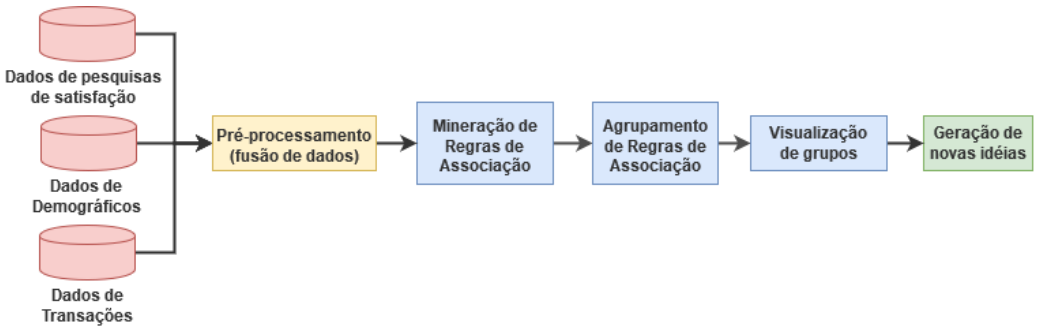

Figura 7. Visão esquemática da metodologia (as cinzas representam os passos principais da metodologia). [Adaptado de Karimi-Majd e Mahootchi (2015)]

Os resultados obtidos por Karimi-Majd e Mahootchi (2015) são grupos de interesses em comum que correlacionam pesquisas de satisfação, dados demográficos e transações (produtos adquiridos pelos clientes). Com base nesses grupos é possível identificar quais os

interesses mais relevantes dos consumidores por região, possibilitando novos serviços com base nos *clusters* de interesse.

Smith *et al*. (2009) apresentam um estudo no qual avaliaram a sensibilidade e confiabilidade das regras de associação geradas a partir de base de dados pequenas (poucos registros). Para validação do modelo as regras foram avaliadas pelo método *Effron's bootstrap.* Durante os experimentos foram correlacionados os dados obtidos pelas mamografias e suas respectivas avaliações laboratoriais (biopsia das massas/nódulos identificados pelas mamografias). A Figura 8 ilustra como é feita a avaliação das regras geradas.

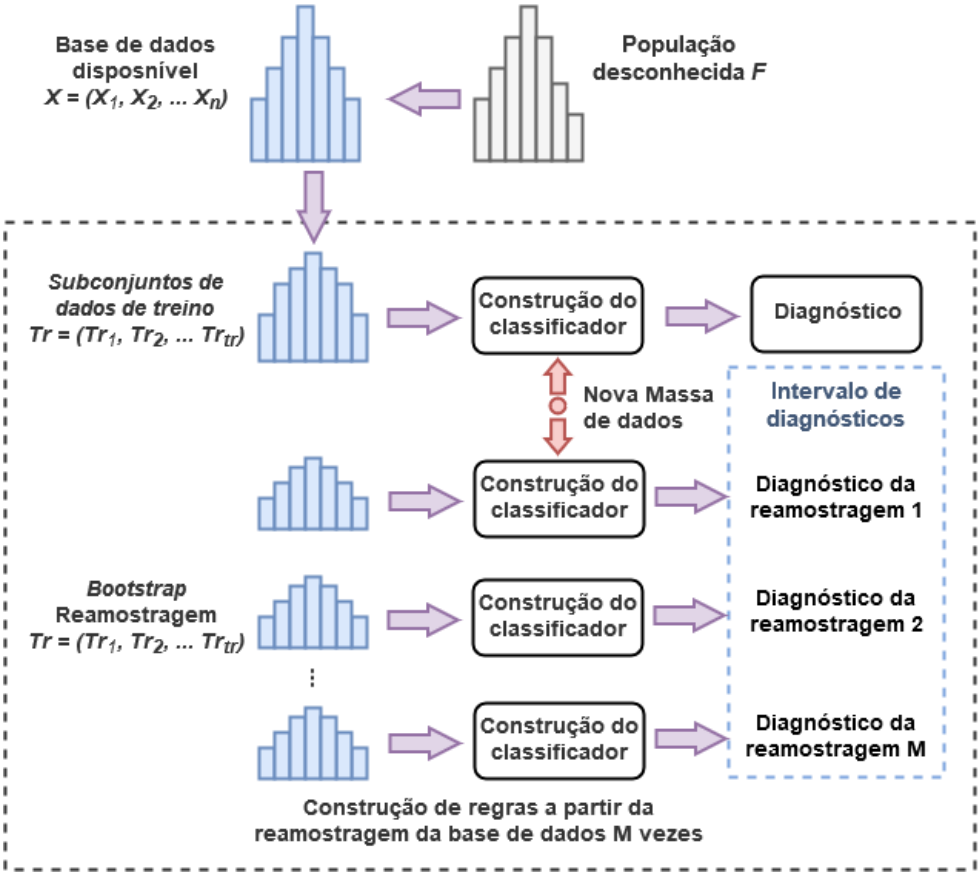

Figura 8. Visão geral do método proposto o qual gera novos modelos de diagnósticos usando novas amostras aleatórias. [Adaptado de Smith et al. (2009)]

Inicialmente um modelo é gerado usando uma base limitada de treino. Posteriormente são gerados novos modelos a partir de novas amostragens da base de treino. A quantidade de diagnósticos gerados pelos modelos obtidos com as amostragens determina o quão confiável o modelo/diagnóstico inicial é, ou seja, quanto menor a quantidade de diagnósticos diferentes, mais confiável.

Mori *et al.* (2005) buscaram, através de regras de associação temporal (regra de associação a qual agrupa transações individuais em uma única transação em um pré-determinado espaço de tempo), identificar correlações entre os estados de um conjunto de sensores para determinar (ou antecipar) uma ação em ambiente residencial. Como

metodologia, os autores definiram que a mudança de estados de uma sequência de sensores em um dado espaço de tempo determina um evento. Este espaço de tempo foi definido usando a técnica de agrupamento *K-Means,* a qual classifica as instâncias em grupos baseados no ponto médio entre as *K* instâncias mais próximas ao centroide, esse processo se repete até que não haja mais alteração de instâncias entre os grupos.

McArthur *et al.* (2012) exploram o uso de regras de associação *apriori* em base de dados pequenas com o intuito identificar correlações entre a taxa de desemprego e fatores socioeconômico em regiões do sudoeste da Noruega. O auxílio da técnica de Análise de Conceito Formal possibilitou o estudo da correlação entre os atributos e as regiões de uma ótica sistemática validando seus estudos. Análise de conceito formal é um método baseado em princípios de derivar uma hierarquia conceitual ou ontologia formal de uma coleção de objetos e suas propriedades. Cada conceito na hierarquia representa o conjunto de objetos que compartilham os mesmos valores para um determinado conjunto de propriedades; e cada subconjunto na hierarquia contém um subconjunto dos objetos nos conceitos acima dele.

Além dos estudos descritos anteriormente, durante a revisão sistemática foi possível identificar outros 3 (três) estudos (Tabela 6) que buscaram otimizar os resultados obtidos pelo algoritmo *apriori* com uso de técnicas adicionais e comparar os resultados com outras técnicas de aprendizagem e mineração.

Tabela 6. Estudos o algoritmo *apriori* e apresentaram alguma otimização.

AUTOR	OTIMIZAÇÃO	TÉCNICAS ENVOLVIDAS
Wang, X. *et al.* (2013)	Redução/Compactação da Base de dados	-
Karthik, K. (2015)	Redução de complexidade	Fila Circular
Paul, R. *et al.* (2012)	Desempenho melhor que KNN, SVN, *NaiveBayes* e *Random Forest*	Dempster-Shafer

Wang *et al.* (2013) realizaram a redução da base de dados removendo registros duplicados, o que impacta na redução do custo de processamento para a busca de *itemset* frequentes. Além disso, outra otimização proposta pelos autores é a compactação dos dados onde, considerando valores binários para representar a ausência (0) e presença (1) de um determinado item em uma transação, a compactação dá-se através da representação por uma *string* contendo uma lista de itens presentes e o seu índice em uma lista de controle (que contém todos os possíveis itens). Exemplificando, suponha-se que a lista de controle seja *X={pão, manteiga, queijo, presunto, café, leite, ovo...}*, e o conjunto *Dj* seja uma transação específica na qual possui *Dj={pão, manteiga, ovo},* logo, seus índices seriam $1_2 1_3 1_7$, de posse de tais índices é possível compactar a informação no seguinte código: “0110001”. Este mecanismo de compactação possibilita a otimização do uso do espaço de armazenamento em sistemas embarcado durante os experimentos.

Embora os estudos apresentados por Karthik (2015) não sejam focados em extração de conhecimentos, os princípios de regras de associação foram aplicados para tornar a

identificação de RFID mais segura. Sua proposta foi embarcar, nos cartões RFID, uma sequência de códigos, ordenados em uma fila circular, que pode ser identificada como um *itemset.* Quando estimulado, o cartão envia um *footprint* de parte desta sequência, ampliando a quantidade de identificações possíveis em um cartão. Além disso é possível remover um dos elementos do *itemset* como mecanismos de renovação da segurança sem disponibilizar o cartão. Tal comportamento só é possível por conta do princípio da anti-monotonicidade (ver em 2.3.2).

Paul *et al.* (2012) buscou combinar as regras de associação *apriori* com a Teoria de Dempster-Shafer para identificar associações probabilísticas entre um conjunto de características clínicas e o diagnóstico de displasia óssea. Teoria de Dempster-Shafer é uma teoria matemática que permite combinar evidências de diferentes fontes para chegar a um grau de confiabilidade (representada por uma função de credibilidade) que leva em conta todas as evidências possíveis. Durante os experimentos, para reduzir a complexidade do problema, o *itemset* foi limitado ao máximo de 10 itens, no entanto não houve uma definição para o suporte mínimo uma vez que toda evidência (regra de associação) é válida para a Teoria de Dempster-Shafer. A metodologia proposta mostrou-se mais eficiente que árvores de decisão (ID3), *Random Forest, Naive Bayes*, SVM e kNN além de ser um pouco superior também aos diagnósticos clínicos.

4.2. Algoritmos Baseados em Árvores

Os algoritmos apresentados nesta seção organizam seus dados em forma de árvore para identificar os *itemsets* mais frequente. Ressalta-se que dos 3 (três) estudos identificados, apenas 1 (um) não apresentou otimização em relação a outra técnica, ou seja, foi estudo não comparativo, embora tenha feito uso de técnica adicional. Os demais estudos propuseram algoritmos e/ou fizeram uso de outras técnicas e apresentaram uma análise comparativa validando a eficiência de suas propostas em relação aos outros estudos.

A Tabela 5 relaciona os estudos, os algoritmos utilizados, suas otimizações e as técnicas envolvidas. Assim como a sessão anterior, os estudos foram classificados inicialmente por seus algoritmos e posteriormente por sua contribuição em relação à otimização apresentada.

Tabela 7. Estudos com algoritmos baseados em árvores e suas contribuições e técnicas envolvidas.

AUTOR	ALGORITMO	OTIMIZAÇÃO	TÉCNICAS ENVOLVIDAS
Shanmuganathan, S et al. (2014)	JRip	-	Regressão Linear
Gonzalez, L.I.L. e Amft, O. (2015)	TITArl	Desempenho melhor que HAC	*Weighted Transitive Clustering*
Ali, S.H. (2012)	FP-Growth	Extração de conhecimento em base de dados pequenas.	*Random Forest*, PCA e *TreeNet*

O estudo conduzido por Shanmuganathan et al. (2014) buscou, identificar os efeitos da variação de temperatura na produção de óleo de palma na Malásia. Através de uma abordagem híbrida, os autores analisaram uma pequena base de dados contendo informações sobre temperatura e o rendimento da produção de óleo de palma. As técnicas usadas foram árvores de decisão (J48), regressão estatística e regras de associação obtendo 78.9% de acurácia na identificação do período mais crítico para plantações (durante a abertura e permanência da flor totalmente aberta).

Gonzalez L.I.L. e Amft, O (2015) buscaram explorar a correlação entre grupos de sensores e atuadores em um edifício para comprovar sua hipótese de que há correlação nas alterações de estados dos sensores de ambiente específico em um dado espaço tempo. O agrupamento de variáveis proposto por Gonzalez L.I.L. e Amft, O (2015), *Weighted Transitive Clustering* (WTC), baseia-se na correlação temporal existente entre a mudança de estados de sensores que monitoram um mesmo ambiente, ou seja, as variáveis agrupadas terão uma forte correlação durante suas mudanças de estado, do contrário, variáveis que pertencem a outros ambientes não serão correlacionadas. Com isso, baseado nas regras de relacionamentos, é possível inferir quais variáveis estão relacionadas ao mesmo espaço e poderão ser agrupadas. Este processo pode ser observado na Figura 9.

Figura 9. Visão esquemática do agrupamento de variáveis. [Adaptado de Gonzalez et al. (2015)]

As regras de relacionamentos são obtidas através do algoritmo *Temporal Interval Tree Association rule learning* (TITArl), proposto por Guillem-Bert and Crowley (2012), o qual busca identificar quantas vezes um evento *A* implica em um evento *B* considerando um limite de tempo máximo *r* para que *B* ocorra após *A.* Sua metodologia buscou comparar o algoritmo TITArl em relação aos métodos *Hierarchical Agglomerative Clustering* (HAC)*, Random Choises Based* e *Mannually Rules Based*, obteve um rendimento superior a todos, de forma que foi possível agrupar 75% dos registros, considerando um intervalo de tempo *r* de 15 segundos. Os demais algoritmos não identificaram os grupos corretamente ou criaram grupos com variáveis pertencentes a outros ambientes, apresentando correlações que não atendiam ao propósito do estudo.

Já Ali (2012) realizou a extração de correlações com uso do algoritmo FP-Growth (*Frequent Patterns Growth*). Este algoritmo faz parte de uma metodologia mais ampla (*Obtaining Accurate and Comprehensible Classification Rules - OACCR*) para extração de correlações em base de dados (pequenas e grandes) com informações médicas. A metodologia híbrida OACCR combina o uso de *Random Forest* para tratar valores ausentes, PCA (*Principle Component Analysis*) para redução de dimensionalidade da base de dados, FP-Growth para extração de regras de associação e, finalizando, com *TreeNet* para classificar as regras de associação geradas. A Figura 10 apresenta o diagrama de blocos da OACCR.

Uma extensão é proposta no modelo para tratamento de base de dados limitadas onde um pré-processamento é permitindo a geração de instâncias virtuais baseadas na população original. Uma vez concluída, os dados (originais e virtuais) são enviados para o processo de extração, citado anteriormente.

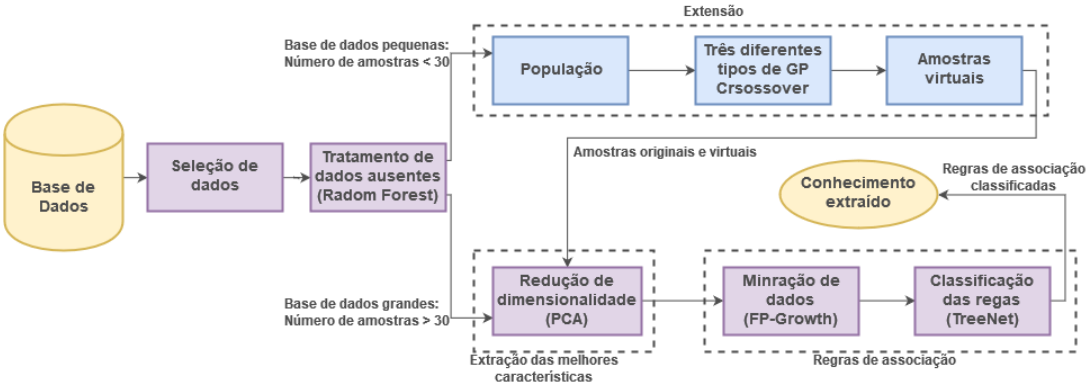

Figura 10. Diagrama de blocos OACCR. [Adaptado de Ali (2012)]

Durante a revisão foram identificados também estudos (apresentados na Tabela 8) que desenvolveram seus próprios algoritmos ou realizaram alguma otimização em algoritmos existentes. Estes, por sua vez, demonstraram a eficácia de suas propostas apresentando estudos comparativos em relação a outras técnicas.

Tabela 8. Estudos comparativos de implementação/otimização de algoritmos.

AUTOR	ALGORITMO	OTIMIZAÇÃO	TÉCNICAS ENVOLVIDAS
Xu, T. et al. (2008)	LFP	Redução de *itemsets* não frequentes	Algoritmos similares o *apriori*
Gruenwald, L. et al (2007)	FARM	Redução de complexidade de tempo e compactação de fluxo de dados	*TinyDB, Spirit*, WARM
Qiu et al. (2006)	LSHAP	Economia de custos (energia) com tratamento probabilístico de leituras em loops	Escalonamento circular e *prefetching*

Xu *et al*. (2008) apresentaram uma estratégia que reduz a quantidade de *itemsets* inválidos, ou seja, não frequentes. O algoritmo *Local Frequent Patterns* (LFP) pode ser aplicado à quaisquer algoritmos baseados no *apriori* e possibilita reduzir significativamente a geração de *itemsets* desnecessários. Embora exija um consumo maior de memória, o LFP possibilita podar um espaço de busca inválido de forma eficiente. Sua estratégia está baseada na seguinte premissa:

Dado um padrão frequente *p* e seu conjunto de candidatos *C*. Supõe-se que um item *a* é o último item do padrão *p*. Para cada item em *b* pertencente à *C*, qualquer padrão *p'*, se *b* concatenado à *a* não é frequente, então *b* concatenado à *p'* concatenado à *p* também é não frequente.

Para validação da hipótese, Xu *et al*. (2008) implementaram dois algoritmos de regras de associação (MAFIA e SPAM), baseado no *apriori*, usando o LFP e outras três propostas semelhantes FHUT, MHUT e PEP. Os resultados obtidos demonstraram que, para bases

pequenas, o MAFIA+LFP obtiveram um desempenho 30% melhor que MAFIA+FHUT e MAFIA+MHUT. Já para base de dados densa, o desempenho foi similar, porém não tão rápido quanto o MAFIA+PEP. Comparando o SPAM+LFP com SPAM, para bases de dados pequenas, o primeiro obteve um desempenho 10 vezes melhor que o segundo enquanto que em base de dados grandes este desempenho reduziu para 30% a 50% melhor. Xu et at, concluem que o LFP é eficiente quando tratando em base de dados pequena, porém seu rendimento cai quando aplicados a bases grandes.

O estudo de Gruenwald *et al*. (2007) buscou apresentar uma solução para estimar valores ausentes, corrompidos ou atrasados de leituras de um ou vários sensores em qualquer turno. O algoritmo *Freshness Association Rule Mining* (FARM) realiza a estimativa de uma leitura de sensor ausente baseada em uma média ponderada da leitura atual dos sensores a ele correlacionados. Cada peso participante na média é derivado diretamente da força da associação correspondente do sensor. Ao atribuir a cada rodada um peso diferente que cresce de acordo com sua ordem, é possível definir um mapeamento reversível entre um histórico inteiro de fluxo de um sensor e o conjunto de números reais. Isso permite que os dados sejam compactos e ainda suficientes para estimar.

Gruenwald *et al.* (2007) compararam o desempenho do FARM com outros três algoritmos, WARM, SPIRIT e *Tiny*DB além de outros quatro métodos estatísticos, *Simple Linear Regression* (SLR), *Multiple Linear Regression* (MLR), *The Curve Regression* (CR) e estimativas por média (AVG). As bases de dados possuíam 15% de dados ausentes. Em relação ao tempo de execução, o FARM é menos que 1 milissegundo mais lento que os demais métodos. A capacidade de estimar valores superou os 80% para FARM e WARM. Quanto a acurácia de classificação, o FARM obteve o melhor desempenho entre todos os métodos.

Qiu *et al.* (2006) propuseram um algoritmo para minimizar o custo total de execução de programas para sistemas embarcados em tempo real. O *Loop Scheduling with Heterogeneous Assignment with Probability* (LSHAP), busca, por meio de análise probabilística, estimar o tempo necessário para execução de uma tarefa. O algoritmo primeiramente correlaciona as entradas aos seus tempos de execução da tabela de histórico, posteriormente, faz-se uso do escalonamento rotativo como forma de melhorar o processo de atribuição e minimizar o custo total do processo. Finalmente, é realizado o *prefetching* para adiantar a preparação dos dados em tempo de execução.

4.3. Resumo

Conforme apresentado nas Seção 4.1 e 4.2, diversos estudos buscam identificar correlações implícitas em bases de dados limitadas. Alguns pesquisadores recorreram a métodos auxiliares, pelos quais buscam reforçar a confiabilidade das regras obtidas pelos algoritmos de análise associativa. Dos 15 estudos apresentados, 9 fizeram o uso das regras

de associação *Apriori*, enquanto os demais estudos desenvolveram seus algoritmos baseados no *Apriori* destacando assim sua relevância para este tipo de análise de dados.

Pal *et al.* (2017) e Lynden (2017), fizeram a aplicação direta deste algoritmo sem uso de técnicas auxiliares, sendo este suficiente para obtenção dos resultados esperados pelos autores.

Os estudos de Karimi-Majad e Mahootchi (2015) Mori *et al.* (2012) usaram métodos de agrupamento em momentos distintos no processo de extração de conhecimento. O primeiro realizou o agrupamento das regras de associação para visualizar grupos de interesses comum, ou seja, procedimento pós-extração, e o segundo realizou o agrupamento de registros individuais em um dado espaço de tempo para gerar transações (identificadas pelo autor como eventos) antes da extração de dados, ou seja, procedimento pré-extração. Nos cenários expostos pelos autores o agrupamento pré ou pós influenciou diretamente na análise dos resultados de forma que foi possível agrupar registros que possuíam correlações (temporais e/ou frequentes).

Nos estudos realizados por Ali (2012), Smith *et al.* (2009) os autores obtiveram modelos (regras de associações) mais confiáveis por meio das técnicas de reamostragens de dados. Embora o primeiro tenha gerado reamostragem de dados reais e o segundo gerou amostras virtuais baseadas no padrão de registros dos dados originais, é possível notar que esse método se mostrou bastante eficiente uma vez que o aumento no volume de dados possibilita identificar padrões e correlações mais precisas, embora exijam mais armazenamento e processamento.

Ali (2012) destaca-se, assim como Xu T. *et al*. (2008), Shammuganathan *et al.* (2014) e Gonzalez e Amft, (2015), por optar em realizar o uso de um algoritmo de regra de associação baseado em árvores (FP-Growth, JRip e TITArl respectivamente), este tipo de algoritmo, embora tenha um custo de processamento mais elevado, apresenta significativa otimização quanto ao espaço usado para armazenamento os padrões frequentes durante o processo de extração das regras. Xu T. et al. (2008) otimizam ainda mais o uso de espaço de armazenamento realizando a podagem das árvores, retirando os *itemsets* menos frequentes.

Qiu *et al*. (2006) e Gruenwald *et al*. (2007) apresentaram métodos alternativos para análise associativa, porém, seus algoritmos são baseados, assim como os demais, no *apriori*. Ambos autores apresentaram suas próprias variações do algoritmo base (*apriori*) e realizaram seus experimentos em base de dados embarcadas, buscando tratar problemas essenciais como baixa capacidade de armazenamento, processamento e memória. O primeiro apresentou um método de compactação e extração de conhecimento da base de dados, enquanto o segundo propôs um mecanismo para otimizar o uso de processadores em sistemas embarcados através de uma associação entre suas entradas e seus históricos de tempo de execução.

Capítulo 5

Método proposto

Este capítulo tem por objetivo discorrer sobre o método proposto destacando as abordagens utilizadas nos componentes para solucionar os problemas apresentados na Seção 1.2. Concomitante a isto, também é apresentado como os mesmos interagem entre si para satisfazem os objetivos definidos na seção 1.4.

Para ampla compreensão da proposta é necessário o conhecimento prévio de seus componentes, sendo assim, este capítulo está dividido em Dispositivo Inteligente (Seção 5.1), Base de Dados (Seção 5.2), Conectividade e Integração (Seção 5.3), Mineração de Correlações (Seção 5.4), Base de Correlações (Seção 5.5) e Visão Geral da Arquitetura (Seção 5.6).

5.1. Dispositivo Inteligente

Um ambiente inteligente geralmente é composto por diversos objetos fortemente integrados à sistemas embarcados que os controlam/monitoram e possibilitam o processamento, recebimento e envio de dados. Esses sistemas têm por objetivo representar o objeto controlado em meio digital de tal forma que seja possível identificar seus atributos, ações, estados e outras características necessárias para sua interação com outros sistemas. O conjunto formado pela integração entre o objeto e o sistema embarcado é identificada nesta proposta como dispositivo inteligente.

Cada dispositivo possui um conjunto de estados que representa sua interação com o ambiente, bem como um conjunto de ações que lhes permite transitar entre os estados. Por exemplo, uma lâmpada possui os estados “ligada” e “desligada” e as ações disponíveis são “ligar” e “desligar”. Vale ressaltar que, para os dispositivos atuadores, estas ações podem ser requisitadas por meio de estímulos físicos (Ex.: interruptor) e ou lógicos (Ex.: requisição ao sistema embarcado).

Sendo $S = \{s_1, s_2, \dots s_i\}$ um conjunto finito de *i* itens que representam todos os possíveis estados que o dispositivo pode assumir e $A = \{a_1, a_2, \dots, a_i\}$ um conjunto de *i* ações disponíveis para transitar entre os estados de *S*. É possível definir formalmente cada dispositivo do ambiente inteligente por meio de uma Máquina de Estado Finito (MEF) conforme o ilustrado na Figura 11.

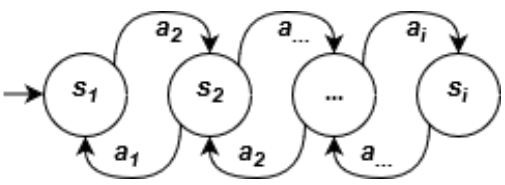

Figura 11. Máquina de Estados Finitos com i estados e i ações.

Alguns dispositivos, mais especificamente os sensores, monitoram o ambiente coletando dados com valores reais. Para estes dispositivos é necessário discretizar os valores para definição dos conjuntos de ações e estados. À exemplo disto, os sensores de temperatura podem gerar infinitos valores reais dependendo do grau de precisão do sensor, neste caso, o processo de discretização vincula os estados “frio”, “moderado” e “quente” às ações de leituras “inferiores à 20ºC”, “entre 20ºC e 28ºC” e “superior à 28ºC” respectivamente. Vale observar que, diferente dos atuadores, os sensores geralmente não permitem o acesso a suas ações por meio de estímulos lógicos, geralmente são estímulos físicos de um sensor que coleta dados de um certo fenômeno (ex.: mudança climática, chuvas, neve) ou um evento (ex.: movimento, som, fumaça).

5.2. Base de Dados

Outra especificação deste método define a estrutura de armazenamento de dados embarcada, uma vez destacadas as características do problema e as restrições de recursos. Diferentemente das abordagens de mineração de dados e aprendizagem de máquinas o modelo proposto ressalta a importância dos registros e análise das mudanças de estados dos dispositivos, ou seja, a quantidade de ocorrências de uma determinada ação que resultaram em uma mudança de estado na MEF.

Assumindo $T = \{t_1, t_2, \dots, t_j\}$ um conjunto finito de intervalo de tempos discretos (*slots*), é possível definir uma base de dados embarcada como uma matriz de contadores $M_{ij} = A \times T$ onde cada elemento c_{nm} é um contador associado à ação $a_n \in A$ no slot $t_m \in T$.

$$M_{ij} = \begin{bmatrix} c_{11} & \cdots & c_{1j} \\ \vdots & \ddots & \vdots \\ c_{i1} & \cdots & c_{ij} \end{bmatrix} \qquad \begin{array}{l} i\text{: número de itens em } A\ (\ |A|\ ); \\ j\text{: número de slots em } T\ (\ |T|\ ); \\ c\text{: contador para ação } a_i \text{ no slot } t_j; \end{array}$$

Esta organização de armazenamento permite a redução da quantidade de dados que precisam ser pré-processados para análise de dados embarcada. Também é possível definir qual ação possui mais probabilidade de ocorrer, identificando o contador com maior valor em um determinado *slot*.

Seja C_j um conjunto que possua todos os contadores da coluna *j* pertencentes à matriz M_{ij}, e uma função $max\_action(C_j, A)$ que retorne a ação correspondente ao contador de maior valor em C_j (ou *nulo* caso haja mais de um contador), então é possível definir um conjunto $P = \{(p_1, p_2, \dots, p_n) \in A \mid 1 \leq n \leq |T| \ \wedge \forall p_x \leftarrow \max\_action(C_j, A)\}$ que define as ações mais prováveis de ocorre em cada slot, sendo este o padrão de mudança de estados do dispositivo.

Vale ressaltar que embora os contadores sejam incrementados à cada mudança de estado, ao final do processo de mineração, os mesmos passam por uma transformação logarítmica onde $\forall c_{nm} \in M_{ij} \mid 1 \leq n \leq |A| \wedge\ 1 \leq m \leq |T| \ \wedge\ c_{nm} \leftarrow log_{|A|}^{(C_{nm}+1)}$. Esta transformação permite os registros de dados antigos não exerçam forte influência na extração do padrão de uso para minerações futuras, além disto, a transformação evita que os

contadores assumam valores elevados que não são capazes de serem representados em alguns sistemas embarcados.

5.3. Conectividade e Integração

Visando a independência e autonomia, cada dispositivo se comporta como uma *Web Thing* (Seção 2.2.2) e satisfaz os requisitos especificados no padrão *Direct Connectivity* (Seção 2.2.1) ou seja, uma vez conectado à rede local, o dispositivo disponibiliza acesso aos seus recursos por meio de uma API REST. Dessa forma é possível interagir com a WT por meio de requisições HTTP para execução de ações, coletar dados, definir configurações e outros recursos conforme a funcionalidade do dispositivo e as especificações do *Web Thing Model* (Seção 2.2.3).

Outra característica essencial para o modelo é a capacidade dos dispositivos ingressarem uma rede *multicast* e tratar pacotes *Multicast Echo Request/Reply.* Essa especificação otimiza o processo de descoberta de dispositivos inteligentes em rede e possibilita agrupar os mesmos de acordo com um ambiente e/ou categoria se for necessário. Por meio do envio de um pacote *Echo Request* ao endereço de *multicast* é possível obter o *IP* de todos os dispositivos que fazem parte do grupo (associado ao endereço de *multicast*).

5.4. Mineração de Regra de Associação

Uma vez satisfeita as especificações citadas de representação, armazenamento e conectividade é possível discorrer sobre o processo de extração de correlações os dispositivos. Que, para tal, usou-se o algoritmo de Regras de Associação Apriori (AGRAWAL e SRIKANT,1994) assumindo como métricas a elevação (*lift*), confiança (*confidence*) e suporte (*support*), conforme apresentados na seção 2.3.2.

Embora o algoritmo de Regras Associação Apriori busque identificar os maiores *itemsets* frequentes em uma base de transações, esta proposta identifica as correlações mais pertinentes entre dois dispositivos as quais satisfazem os limiares mínimos das métricas, ou seja, identificar quais *itemsets* de tamanho 2 possuem o valor de *support*, *lift* e *confidence* mais elevados e satisfazendo o suporte mínimo.

A extração de correlações é baseada no padrão de mudança de estado dos dispositivos, dessa forma é possível correlacioná-los acordo com os padrões que o usuário mais interage com os dispositivos. A Figura 12 ilustra, de forma simplificada, como é realizado o processo de extração de correlação entre dois dispositivos.

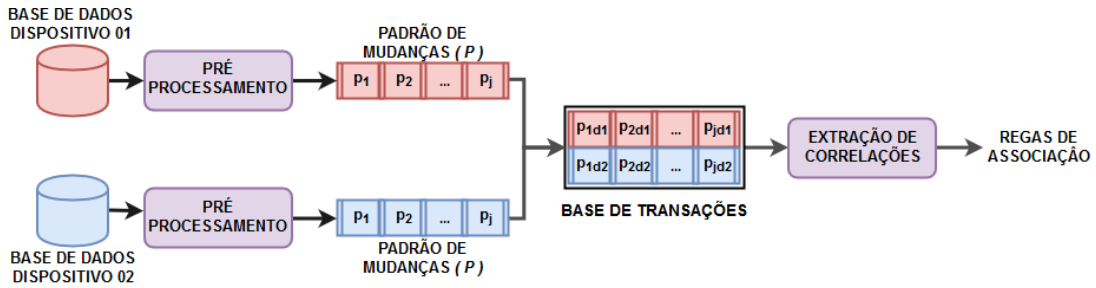

Figura 12. Ilustração do processo de extração de correlações entre dois dispositivos.

Cada dispositivo possui uma base de dados embarcada de onde são extraídos os padrões de mudanças de estados, conforme o especificado na Seção 5.2. Posteriormente é feita a fusão de dados que une padrões de ambos padrões gerando uma base de transações com *itemsets* de tamanho máximo iguais a 2. Nesta etapa, o algoritmo de Regra de Associação Apriori analisa a Base de Transações preservando apenas as regras com métricas mais elevadas que serão posteriormente apresentadas ao usuário para aceitação ou não.

Essas regras podem definir, por exemplo, que a ação *"A"* no Dispositivo 01 infere na ação *"B"* no Dispositivo 02 e a qualidade desta inferência é mensurada de acordo com as métricas *support*, *lift* e *confidence.* Em outras palavras, é possível afirmar que, dado um grau de precisão, geralmente, quando a ação *"A"* é requisitada no Dispositivo 01, a ação *"B"* também é requisitada no Dispositivo 02.

Uma observação importante neste modelo é que o cálculo das métricas leva em consideração o tamanho máximo que a base de transações pode assumir, ou seja, número máximo de slots (|*T*|). Isto permite que as métricas sejam calculadas corretamente mantendo as devidas proporções em relação ao conjunto de dados analisado viabilizado a comparação justa entre as regras de dispositivos diferentes.

5.5. Base de Correlações

Além da Base de Dados os dispositivos dispõem de um a Base de Correlações a qual armazena das regras de associação obtidas durante o processo de mineração.

Cada registro da Base de Correlação deve possuir as seguintes informações:

- ***Antecedent Action***: Ação que define a premissa da regra que vincula uma ação do dispositivo local a ação em outro dispositivo remoto (consequente);
- ***Consequent Action (URL)***: URL que permite executar a ação correspondente à consequência da regra no dispositivo correlacionado;
- ***Lift*****:** Valor da métrica de correlação entre a ação do antecedente e o consequente;
- ***Confidence***: Valor da métrica de confiança da regra; e
- ***Support****:* Valor da métrica de suporte para esta correlação.

Esta base tem por finalidade armazenar as regras mais relevantes para que o usuário possa consultar e ativar as que mais atendem aos seus interesses. Por meio delas é possível

correlacionar os dispositivos através de requisições HTTP para estimular a mudança de estado em um outro dispositivo para que a regra seja satisfeita.

Vale ressaltar que, dada as limitações de armazenamento, apenas uma regra de correlação pode ser ativada para cada ação possível do dispositivo. Uma vez ativas, as demais regras (da mesma ação) são descartadas para reduzir o consumo de espaço de armazenamento.

5.6. Visão Geral do Método Proposto

O método proposto nesta dissertação, identificado como *Decentralized Mining of Paired States Changes* (DMPSC), possibilita a mineração decentralizadas de mudanças de estados pareados, ou seja, é possível identificar mudanças de estados que frequentemente ocorrem juntas.

A integração dos componentes e técnicas apresentadas nas seções anteriores ocorre em um ambiente embarcado, ou seja, cada dispositivo desempenha as mesmas funções independentemente uns dos outros. Estas características possibilita um ambiente inteligente flexível, com alta tolerância a falhas e que se molda ao padrão do usuário.

A Figura 13 sintetiza os componentes apresentados nas seções anteriores e ilustra sua interação entre si e com o ambiente.

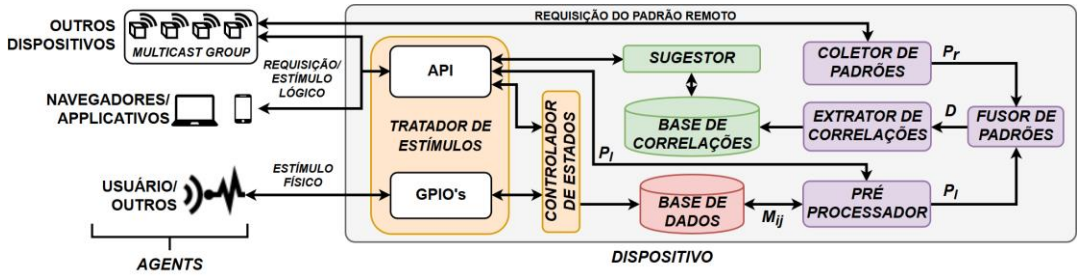

Figura 13.Visão geral do mecanismo proposto.

Sendo cada dispositivo uma *Web Thing* a interação com o mesmo pode ocorrer através de estímulos lógicos (requisições HTTP) ou físicos por meio das GPIO's (General Port Intput/Output) no sistema embarcado. Estes estímulos são analisados pelo Tratador de Estímulos que verifica se a ação requisitada gera uma mudança de estado no dispositivo, neste caso, é enviado um sinal ao componente Controlador de Estados que incrementa o contador na Base de Dados Embarcada após transitar do estado atual para o estado solicitado. Previamente ao incremento, o Controlador de Estados verifica se a ação solicitada é a que possui mais probabilidade de ocorrer entre as demais ações no *slot* atual, em caso afirmativo, o Controlador de Estados envia um sinal de volta ao tratador de Estímulos que consulta a Base de Correlações e então envia uma requisição HTTP para a URL que estimula a ação no consequente da que vincula ambas ações.

As regas da base de correlações são atualizadas em intervalos de tempos fixos, definido pelo usuário. Por meio do envio de um pacote para o endereço de *multicast* é possível obter o

endereço de todos os dispositivos do grupo e, em posse desta lista, inicia-se o processo interativo que consiste de quatro passos:

I. **Pré-Processamento**: O dispositivo gera o padrão de mudança de estados (P_l) a partir da base de dados embarcada (M_{ij});

II. **Coletor de padrões**: Este componente coleta o padrão de mudança de estados do (próximo) dispositivo na lista, ou seja, do grupo de *multicast*. Este padrão é representado por P_r (Padrão remoto);

III. **Fusor de padrões**: Em posse de ambos padrões (P_l e P_r) este componente gera uma base de transações $D = \{\{(p_{l1}, p_{r1}), (p_{l2}, p_{r2}) \dots (p_{lj}, p_{rj})\} \mid p_{ln} \in P_l \wedge p_{rn} \in P_r \wedge 1 \leq n \leq |T|\}$;

IV. **Extrator de Correlações**: Após a geração da Base de Transação é aplicado o Algoritmo de Regra de Associação Apriori em *D.* Os itemsets de tamanho 2 são armazenados caso suas métricas sejam maiores ou iguais às já armazenadas, caso contrário, são descartadas.

Esses quatro passos repetem-se até que todos os dispositivos da lista sejam analisados e restem na Base de Correlações apenas as regras de associação mais relevantes para cada ação do dispositivo. Por fim, o dispositivo irá sugerir ao usuário, por meio de sua interface de gerenciamento, as regras obtidas as quais podem ser ativas ou descartadas pelo usuário.

5.7. Resumo

Neste capítulo foi apresentado o método que visa atender aos objetivos de pesquisas desta dissertação. Por meio de uma arquitetura simplificada, o comportamento individual e colaborativo dos dispositivos possibilitará oferecer ao usuário um conjunto de regras de associação dos dispositivos para proporcionar um ambiente inteligente integrado baseado no padrão de uso do usuário.

Capítulo 6

Experimentos

Este capítulo descreve os experimentos executados para validação do método proposto e está dividido em: Metodologia (Seção 6.1), Descrição dos Dados (Seção 6.2), Preparação dos Dados (Seção 6.3), Execução dos Experimentos (Seção 6.4) e Resultados Parciais (Seção 6.5).

Todos os códigos fontes, scripts e resultados referentes aos experimentos podem ser obtidos em (ALENCAR, 2018).

6.1. Metodologia de Experimentação

A metodologia adotada para avaliar o modelo compara as regras geradas pela simulação do ambiente descentralizado, conforme especificado o Método Proposto (Seção 5), em relação as regras obtidas em cenário centralizado.

As extrações de regras, ou análise associativa, ambiente centralizado foram executadas por meio da biblioteca “arules” (Hahsler *et al*. 2011) no software de análises estatísticas R e são identificadas como R-Rules. A extração em ambiente descentralizado é referenciada como DMPSC sendo esta implementada em Python 3 para a execução de testes.

A execução da análise associativa consiste em executar o DMPSC e o R-Rules em Bases de Transações geradas a partir de diferentes *datasets*. As regras extraídas seriam comparadas para identificar quão similar são os conjuntos de regras produzidos. A geração de regras para o DMPSC é executada por meio da implementação em Python3 seguindo as especificações do modelo.

A avaliação foi definida de acordo com o seguinte critério: “*A regra mais relevante para cada dispositivos em cada experimento deve coincidir em ambas análises (R-Rules e DMPSC)”.* Para isto foram definidas três métricas:

I. **Taxa de Acerto (*hits rate***): Indica o percentual de regras identificadas como as mais relevantes pelo DMPSC em relação às geradas pelo R-Rules.
II. **Taxa de Falsos Negativos (*false negative rate*)**: Percentual de regras identificadas pelo DMPSC que também foram identificadas pelo R-Rules porém, durante a seleção aleatória do DMPSC, a escolha divergiu da primeira ocorrência das regras do R-Rules.
III. **Taxa de Regas incomparáveis (*unmatched rules rate)***: Percentual de regras identificadas pelo DMPSC que não foram identificadas pelo R-Rules.

Para se aproximar o máximo possível de um ambiente real de uso, alguns parâmetros foram pré-definidos para este experimento, sendo eles:

I. Todos os dispositivos criaram uma base de dados exclusiva para cada dia da semana (segunda à domingo) preservando a independência das correlações ao longo de 7 dias;
II. As bases de dados foram segmentadas em 96 slots (24 horas divididos em intervalos de 15 minutos);
III. Os experimentos consideraram extrações em intervalos de 7, 14, 28 dias para cada dia da semana;
IV. Suporte (*support*) mínimo: 1% do padrão de mudança de estado, ou seja, basta a ação ser a mais pertinente em apenas 1 dos 96 slots;
V. Confiança (*confidence*) mínima: 90%
VI. Elevação (*lift*) mínima: 1.1, busca-se apenas correlações positiva entre os dispositivos;

Os experimentos foram executados em uma máquina virtual no Virtual Box com 1 processador e 2GB de RAM usando Debian Linux (v.9.0)

6.2. Descrição dos Dados

Foram utilizados cinco *datasets* públicos disponibilizado pelo projeto WSU CASAS (Crandall *et al.,* 2013) os quais possuem registros de sensores de baterias, sensores magnéticos de portas, interruptores de lâmpadas, sensores de luminosidade, sensores infravermelhos de movimento e sensores de temperatura.

A Tabela 9 apresenta informações gerais sobre algumas características dos *datasets,* tais como*:* ambiente monitorado, quantidade de participantes, número de dispositivos, quantidade de dias monitorados e número de registros.

Tabela 9. Informações gerais sobre os *datasets*

DATASET	**AMBIENTE**	**PARTICIPANTES**	**DISPOSITIVOS**	**DIAS**	**REGISTROS**
hh107	Residencial	2	110	371	3.369.689
hh123	Residencial	1	88	588	2.907.282
hh129	Residencial	1	86	668	12.303.984
shib009	Residencial	n/d	8	847	3.187.940
tokyo	Trabalho	9	67	115	802.534

Os registros são compostos por quatro atributos conforme apresentados na Tabela 10.

Tabela 10. Campos que compõe os registros dos *datasets*

ATRIBUTO	**FORMATO**	**DESCRIÇÃO**
DATA	<ano>**-**<mês>**-**<dia>	dia da criação do registro
HORÁRIO	<hora>**:**<minuto>**:**<segundo>**.**<milésimos>	hora da criação do registro
DISPOSITIVO	*String*	nome ou lugar do dispositivo
VALOR	*string / int / float*	estado (rótulo) ou valor contínuo

6.3. Preparação dos Dados

Por meio de um programa desenvolvido em *Python3*, a base de dados foi pré-processada para que atendesse todas as especificações do modelo. Este pré-processamento consistiu em tratar os dados ausentes, remover registros duplicados, discretizar o intervalo de tempo em *slots* e os valores contínuos (leituras dos sensores) além de segmentar a base de dados em vários arquivos respeitando o dia da semana e os intervalos de verificações (*checkpoints*).

Os registros que não possuíam o atributo valor (dados ausentes) foram descartados. Esta estratégia foi definida por ser um comportamento plausível em um cenário real de uso, por exemplo, quando um equipamento está danificado ou desligado por alguns dias.

A abordagem adotada para a discretização dos valores contínuos consistiu em:

I. Identificar o valor máximo e mínimo lidos pelo dispositivo;
II. Obter a média aproximada de leitura dividindo por dois a soma dos valores máximo e mínimo;
III. Substituir o valor contínuo pelo rótulo “HIGH” se for acima da média calculada e “LOW” se for abaixo ou igual à média.

A discretização temporal, ou definição dos *slots,* foi calculada através da conversão do atributo ‘Horário’ para segundos e dividindo o mesmo por 15. Neste caso, os valores de segundos e milésimos são desconsiderados. Esse cálculo permite que os registros sejam associados à intervalos de tempo discretos entre 0 a 95, ou seja, 96 slots.

As Figuras 14(A) representa uma amostra dos dados originais dos *datasets* e a Figura 14(B) representam o *dataset* após o pré-processamento.

(A)

```
==> hh107 <==
2012-07-20 13:05:08.136976 MA023 OFF
2012-07-20 13:05:08.281327 MA022 OFF
2012-07-20 13:05:08.411059 MA021 OFF
2012-07-20 13:05:08.739417 MA020 OFF
2012-07-20 13:05:09.009726 MA024 OFF

==> hh123 <==
2013-03-01 10:45:02.880093 T107 24
2013-03-01 10:45:02.948323 T104 25
2013-03-01 10:45:02.999974 LS018 0
2013-03-01 10:45:03.042639 BATP018 100
2013-03-01 10:45:03.109711 BATV018 9440

==> hh129 <==
2013-10-22 02:32:34.409036+00 T105 31
2013-10-22 02:32:34.560481+00 T104 28
2013-10-22 02:34:41.11752+00 M001 OFF
2013-10-22 02:34:41.122383+00 LS001 98
2013-10-22 02:34:41.127101+00 BATV001 9623

==> shib009 <==
2014-07-18 15:43:22.346740 DiningRoom OFF
2014-08-24 03:45:38.294681 Bathroom OFF
2014-08-24 03:46:09.032044 Bathroom OFF
2014-08-24 03:49:16.097031 Bathroom OFF
2014-08-24 03:49:39.643615 Hall OFF

==> tokyo <==
2008-01-02 08:00:00.596643 M44 OFF
2008-01-02 08:00:03.829399 M43 OFF
2008-01-02 08:00:08.109046 TO5 23.4375
2008-01-02 08:00:08.109046 TO3 25.5
2008-01-02 08:00:25.53859 TO3 25.5625
```

(B)

```
==> hh107 <==
slot,device,state
52,MA023,OFF
52,MA022,OFF
52,MA021,OFF
52,MA020,OFF

==> hh123 <==
slot,device,state
43,T107,LOW
43,T104,LOW
43,LS018,LOW
43,BATP018,HIGH

==> hh129 <==
slot,device,state
10,T105,HIGH
10,T104,HIGH
10,M001,OFF
10,LS001,HIGH

==> shib009 <==
slot,device,state
62,DiningRoom,OFF
15,Bathroom,OFF
15,Hall,OFF
15,Bedroom,OFF

==> tokyo <==
slot,device,state
32,M44,OFF
32,M43,OFF
32,TO5,HIGH
32,TO3,HIGH
```

Figura 14. (A) Registros originais; (B) Registros após o pré-processamento

É possível observar que após o pré-processamento os registros não exibem o atributo “Data”, isso se dá pelo fato de que cada checkpoint possui um arquivo independente que possui apenas os registros de um intervalo de tempo específico, definidos na Seção 6.1.

6.4. Execução

Conforme definido na Seção 6.1, cada dia da semana possui uma base de dados independente que possibilita a análise do padrão de uso de acordo com o dia da semana. Além disto, cada dia da semana está dividindo em *checkpoints* que agrupam os dados à cada intervalo de tempo (7, 14 ou 28 dias).

Durante cada *checkpoint* as seguintes etapas são executadas:

I. Preenchimento/Atualização da base de dados (Matriz de contadores);
II. Extração de padrão de mudanças de estados (Análise probabilística);
III. Criação da base de transações;
IV. Extração de Regras para o R-Rules (Regras de associação centralizada);
V. Extração de Regras seguindo o modelo de DMPSC (Regra de associação descentralizada);
VI. Comparação de Regras; e
VII. Transformação logarítmica na matriz de contadores.

Inicialmente (Etapa I) é gerada uma matriz de contadores para cada dispositivo identificado. Inicia-se então a leitura do *dataset* (pós-tratamento) permitindo o incremento dos contadores até que todos os registros, pertencentes do primeiro *checkpoint*, sejam registrados nas matrizes de contadores.

Durante a Etapa II é realizada a análise probabilísticas dos contadores para a identificação dos padrões de mudanças de todos os dispositivos. Uma vez obtido todos os padrões, é gerado um arquivo que contém todos os padrões em que cada linha é o conjunto de dispositivos/estados (*itemset*) mais frequentes em um determinado *slot*. Este arquivo é identificado como base de transações (Figura 15(A)) o qual será analisado pelo R-Rules. As regras obtidas desta análise são armazenadas em um segundo arquivo (Figura 15(B)) respeitando a ordem decrescente das métricas *support, lift* e *confidence* respectivamente.

(A)

```
Kitchen-OFF, Office-OFF
Kitchen-ON
LivingRoom-ON
LivingRoom-OFF
DiningRoom-ON, Kitchen-ON
DiningRoom-OFF, Kitchen-OFF
Bedroom-ON
Bedroom-OFF
```

(B)

```
rules,support,confidence,lift,count
{Office-OFF} => {Kitchen-OFF},0.125,1,4,1
{DiningRoom-ON} => {Kitchen-ON},0.125,1,4,1
{DiningRoom-OFF} => {Kitchen-OFF},0.125,1,4,1
```

Figura 15. (A) Base de transações; (B) Regras obtidas pelo R-Rules

Durante a etapa seguinte(III), os padrões de usos são analisados em pares para extração das regras conforme especificado no método proposto. Apenas a regra mais relevante para cada estado de cada dispositivos é armazenada.

Na etapa VI é feita a comparação entre as regras obtidas pelo DMPSC e pelo R-Rules. Esta, que consiste em verificar se as regras geradas pelo modelo descentralizado possuem a mesma relevância em relação as geradas pelo ambiente centralizado, obtém valores para suas métricas conforme as especificações na Seção 5.1 e finalmente é executada a transformação logarítmica na matriz de contadores dos dispositivos.

Uma vez finalizada as etapas, é possível avançar para o próximo *checkpoint* onde todo o processo se repete. Ressalta-se que a cada iteração as regras e os padrões de mudanças de estados são descartados, apenas as matrizes de contadores dos dispositivos após as transformações logarítmicas são preservadas e incrementadas de acordo com os novos registros para o novo *checkpoint*.

A Figura 16 apresenta um exemplo do relatório gerado durante a execução dos experimentos. Nela é possível observar os parâmetros de entradas (arquivo, intervalo de *scan*, tempo por *slot*, suporte mínimo, elevação mínima e confiança mínima) e os valores obtidos à cada *checkpoint* do dia da semana analisado. Ao final do experimento do dia, são apresentadas as taxas de *Hits, False Negative e Unmatched.*

```
File: tokyo
Number of Devices: 67
Scan Interval(per day): 28
Slot Interval(in min): 15
Num of Slots: 96
Minimum Support: 0.01
Minimum Lift: 1.1
Minimum Confidence: 0.9
---
Day of week: wed
Checkpoint: 1, States: 131, R-rules: 143, DMPSC-rules: 26, Checked: 26, Hits: 26, False Negative: 0, Unmatched: 0
Checkpoint: 2, States: 134, R-rules: 105, DMPSC-rules: 26, Checked: 26, Hits: 23, False Negative: 3, Unmatched: 0
Checkpoint: 3, States: 137, R-rules: 71, DMPSC-rules: 10, Checked: 10, Hits: 10, False Negative: 0, Unmatched: 0
Checkpoint: 4, States: 141, R-rules: 121, DMPSC-rules: 22, Checked: 22, Hits: 20, False Negative: 2, Unmatched: 0
Checkpoint: 5, States: 143, R-rules: 155, DMPSC-rules: 36, Checked: 36, Hits: 36, False Negative: 0, Unmatched: 0
Rates:Hits: 115(0.958333333333), False Negative: 5(0.0416666666667), Unmatched: 0(0.0)
```

Figura 16. Exemplo parcial de relatório gerado durante a comparação das regras geradas

Além das métricas já apresentadas, o relatório apresenta também a quantidade de estados (*States*) identificados, o número de regras identificadas pelo R-Rules e pelo DMPSC, além de destacar e o número de regras identificadas em ambas análises (*Checked*).

Ao final de todo o experimento para o *dataset* é apresentado um relatório final conforme apresentado na Figura 17.

```
---
Experiment Report:
Average Hits: 0.906976744186(429)
Average False Negative: 0.093023255814(44)
Average Unmatched: 0.0(0)
Number of comparisons: 23
---
```

Figura 17. Relatório de médias do experimento

Este relatório identifica a média geral das métricas de *Hits*, *False Negative*, *Unchecked,* além do número de comparações realizadas ao longo de todos os dias da semana.

6.5. Resultados Parciais

A Tabela 11 apresenta o resultado do pré-processamento de dados comparando a quantidade de registros originais e após o tratamento de dados no qual consistiu em discretizar e realizar a limpeza de dados.

Tabela 11. Resultado do pré-processamento dos *datasets*

DATASET	NÚMERO DE REGISTROS		REDUÇÃO
	ORIGINAIS	PÓS LIMPEZA E DISCRETIZAÇÃO	
hh107	3.369.689	2.811.279	16,57%
hh123	2.907.282	2.345.775	19,31%
hh129	12.303.984	56.523	99,54%
shib009	3.187.940	90.599	97,16%
tokyo	802.534	171.483	78,63%

A Tabela 12 apresenta o número de comparações realizadas durante todos os experimentos de acordo com o *dataset* e o intervalo de tempo para execuções das extrações. Tais comparações são executadas apenas se ambos métodos (R-Rules e o DMPSC) forem capazes de realizar extrações de regras.

Tabela 12. Número de comparações em cada experimento

DATASET	7 DIAS	14 DIAS	28 DIAS	TOTAL
hh107	371	189	98	665
hh123	588	294	147	1036
hh129	11	7	7	32
shib009	16	139	109	264
tokyo	35	19	23	84

Os valores obtidos da Tabela 13 representam as taxas médias de acerto (*hits*), falso negativo (f*alse negative*) e de regras não comparadas (*unm – unmatched*) obtidas durante a execução dos experimentos. As mesmas estão agrupadas de acordo com intervalo de extração e o dataset.

Tabela 13. Médias das taxas *Hits, False Negative (F.N) e Unmatched (UNM).*

DATASET	7 DIAS			14 DIAS			28 DIAS		
	HITS	F.N.	UNM	HITS	F.N.	UNM	HITS	F.N	UNM
hh107	0,8799	0,1201	-	0,8580	0,1420	-	0,9346	0,0616	0,0038
hh123	0,9380	0,0620	-	0,9187	0,0813	-	0,9623	0,0350	0,0027
hh129	0,5750	0,4250	-	0,5185	0,4815	-	0,5185	0,4815	-
shib009	1,0000	-	-	0,9852	0,0148	-	0,9867	0,0133	-
tokyo	0,8248	0,1752	-	0,8309	0,1691	-	0,9070	0,0930	-

Os valores representam a média das taxas de ocorrência de cada métrica durante as comparações apresentadas na Tabela 12, enquanto os símbolos “-” representam taxas iguais à 0.

6.6. Discussão dos resultados

Neste primeiro momento a discussão dos resultados analisa as métricas apresentadas nos relatórios gerados durante os experimentos tendo como foco principal as métricas de *Hits, False Negative* e *Unmatched*.

6.6.1. Taxa de Acertos (*Hits*) e Taxa de Falso Negativo (*False Negative*)

Na Figura 18 é apresentada a comparação entre as taxas de acertos para cada base de dados considerando os diferentes períodos de extração. Também é possível observar as médias de acertos por base de dados considerando todos os intervalos de extração.

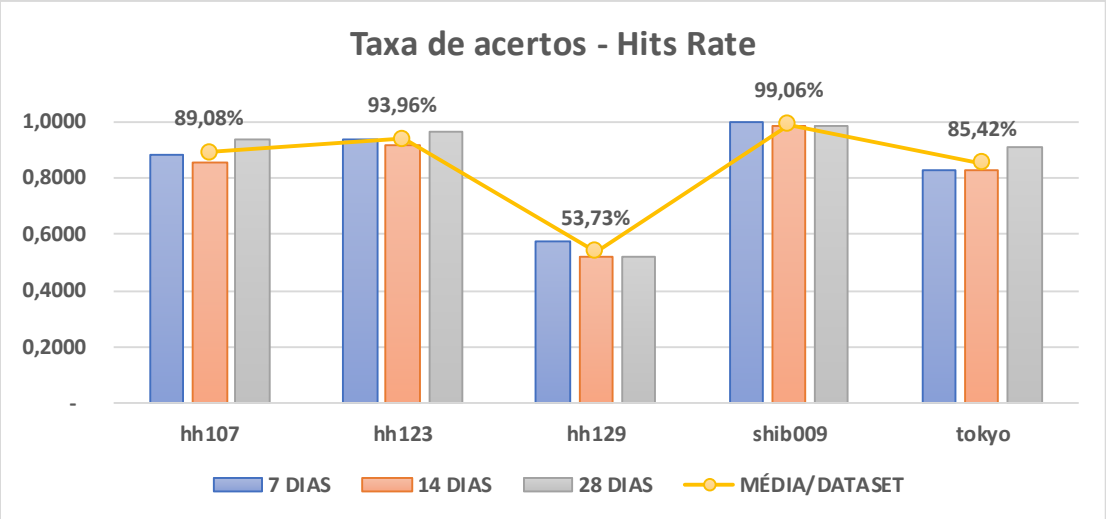

Figura 18. Taxas médias de similaridades por base de dados durante os experimentos.

Durante os experimentos para o *dataset hh129,* a média das taxas de acerto foi de 53,73%, o menor valor entre todos os experimentos. Embora seja o *dataset* mais volumoso, com mais de 12 milhões de registros, muitas destas informações foram descartadas durante o processo de discretização (ver Tabela 11) por não atenderem aos requisitos especificados no modelo, ou seja, muitos registros são repetidos ou com pequenas variações em seus valores que não representam uma mudança de estado. Já os registros satisfizeram as exigências do modelo apresentaram uma distribuição balanceada, na qual, para a maioria dos dispositivos os estados possuíam o mesmo número de registros, ou seja, não havia uma predominância de um estado nos *slots.* Concomitante a isto, muitos dispositivos que apresentaram um estado predominante, o fizeram em apenas um único *slot*, conforme apresentado na Figura 19.

```
M002-OFF, M004-OFF
T104-HIGH
M001-OFF, M002-OFF, M003-OFF, M004-OFF, LS004-LOW, M005-
OFF, M006-OFF, M007-OFF, M010-OFF, M012-OFF, M013-OFF,
M014-OFF, M015-OFF, M016-OFF, M017-OFF, M018-OFF, LS018-
LOW
LS014-LOW, LS015-LOW
T104-HIGH
T104-LOW
M006-ON
M006-OFF
M006-ON
M006-OFF, LS013-LOW
M010-ON
```

Figura 19. Base de transações correlacionado múltiplos estados em um mesmo *slot*.

Este agrupamento de dispositivos tem como consequência a geração de várias regras com métricas iguais para um mesmo antecedente. Isto pode ser observado na Figura 20 que apresenta uma amostra das regras extraídas pelo software R a partir da base de transações apresentadas na Figura 19.

```
rules,support,confidence,lift,count
{LS015-LOW} => {LS014-LOW},0.25,1,4,1
{LS014-LOW} => {LS015-LOW},0.25,1,4,1
{M018-OFF} => {M017-OFF},0.25,1,4,1
{M017-OFF} => {M018-OFF},0.25,1,4,1
{M018-OFF} => {M016-OFF},0.25,1,4,1
{M016-OFF} => {M018-OFF},0.25,1,4,1
{M018-OFF} => {M015-OFF},0.25,1,4,1
{M015-OFF} => {M018-OFF},0.25,1,4,1
{M018-OFF} => {M014-OFF},0.25,1,4,1
{M014-OFF} => {M018-OFF},0.25,1,4,1
{M018-OFF} => {M013-OFF},0.25,1,4,1
{M013-OFF} => {M018-OFF},0.25,1,4,1
{M018-OFF} => {M012-OFF},0.25,1,4,1
{M012-OFF} => {M018-OFF},0.25,1,4,1
{M018-OFF} => {M010-OFF},0.25,1,4,1
                ...
```

Figura 20. Amostra de regras obtidas pelo *software* R.

Na amostra é possível observar que há várias regras para o antecedente “M018-OFF” que possuem correlações com métricas iguais para diversos consequentes.

Diferente do exposto na Figura 20, o modelo proposto deseja extrair apenas a regra mais relevante para cada estado dos dispositivos, logo, para este cenário, o modelo escolhe aleatoriamente qualquer um dos consequentes que possuem métricas iguais.

A Figura 21 apresenta todas as regras extraídas para o mesmo conjunto de dados usando o método proposto nesta dissertação.

```
rule,support,lift,confidence,count
{M001-OFF} => {M002-OFF},0.010416666666666666,96.0,1.0,1
{M002-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M003-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M004-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{LS004-LOW} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M005-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M007-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M010-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M012-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M013-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M014-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{LS014-LOW} => {LS015-LOW},0.010416666666666666,96.0,1.0,1
{M015-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{LS015-LOW} => {LS014-LOW},0.010416666666666666,96.0,1.0,1
{M016-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M017-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{M018-OFF} => {M001-OFF},0.010416666666666666,96.0,1.0,1
{LS018-LOW} => {M001-OFF},0.010416666666666666,96.0,1.0,1
```

Figura 21. Regras geradas pelo DMPSC

Observa-se também que embora as regras possuam valores de métricas diferentes dos apresentados pelo modelo centralizado, as mesmas correlações são apresentadas.

Estas ocorrências são mensuradas por meio da Taxa de Falso Negativo (*False Negative* Rates) e representam a quantidade de regras que foram identificadas por ambos métodos (DMPSC e R-Rules) porém que não coincidiam em ser a mais relevante, sejam por métricas diferentes ou seja por escolha aleatória.

Tal comportamento também é apresentado para os experimentos nas demais bases de dados, porém com uma frequência menor que a apresentada para o *dataset hh129,* como é possível observar na Figura 22.

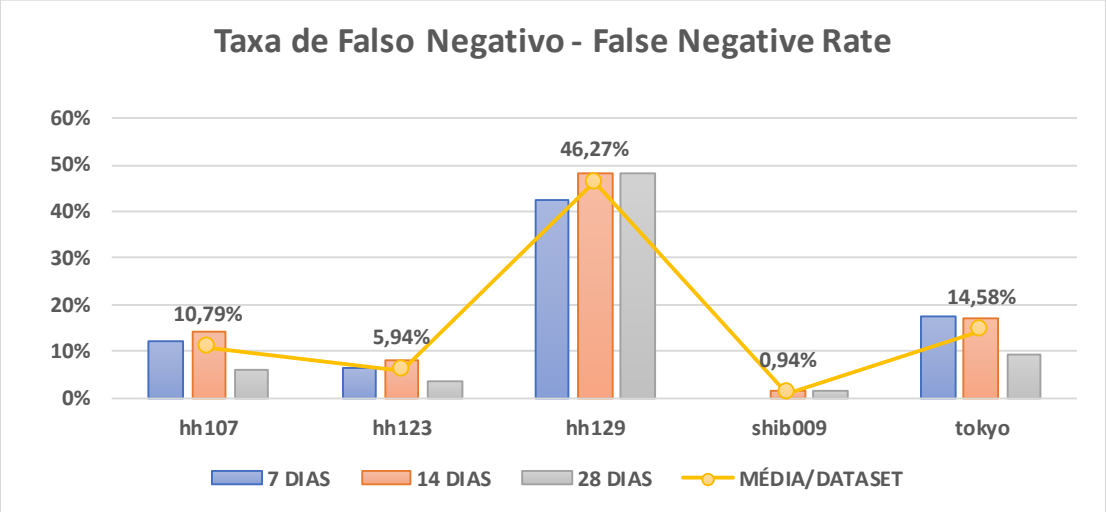

Figura 22. Taxa de falso negativo por *dataset* e intervalo de tempo.

As taxas de falso negativo, geralmente, são o complemento das taxas de acerto, quando isso ocorre, significa que todas do DMPSC também foram identificadas pelo R-Rules, mesmo que algumas não coincidam como a mais relevante para alguns antecedentes. Embora esta seja uma generalização, ambas não integralizam os resultados dos experimentos, para tal,

faz-se necessário avaliar os valores obtidos em relação à Taxa de Regras Não Comparadas na Seção 6.6.2.

6.6.2. Taxa de regras não comparadas (*Unmatched Rates*)

Uma particularidade foi notada durante a execução dos experimentos com os *datasets hh107* e *hh123* considerando o período de extração de 28 dias. Em ambos casos, o DMPSC identificou regras que não puderam ser comparadas pois não foram encontradas pelo modelo centralizado. Para os demais *datasets* não se aplicam tais particularidades.

Durante todo o experimento para o intervalo de 28 dias, a análise descentralizada do *dataset hh107* gerou 22 regras que não puderam ser comparadas, considerando um total de 5.812, uma taxa de aproximadamente de 0,38% das regras. Para os experimentos do *dataset hh123*, 11 de 4.055 regras não foram identificadas pelo R-Rules, o que equivale à taxa de 0,27% de todas as regras obtidas.

Para explorar estas particularidades, primeiramente foi analisado o relatório gerado durante a extração de regras para a Quinta-Feira (*Thuesday*) do *dataset hh107* no qual o DMPSC registrou 6 correlações singulares, sendo 2 para o *checkpoint* 5, 1 para o *checkpoint* 8, 2 para o *checkpoint* 11 e 1 regra para o *checkpoint* 14, conforme apresenta a Figura 23.

```
Day of week: thu
Checkpoint: 1, States: 133, R-rules: 524, DMPSC-rules: 38, Checked: 38, Hits: 31, False Negative: 7, Unmatched: 0
Checkpoint: 2, States: 160, R-rules: 773, DMPSC-rules: 60, Checked: 60, Hits: 57, False Negative: 3, Unmatched: 0
Checkpoint: 3, States: 163, R-rules: 801, DMPSC-rules: 60, Checked: 60, Hits: 60, False Negative: 0, Unmatched: 0
Checkpoint: 4, States: 173, R-rules: 902, DMPSC-rules: 56, Checked: 56, Hits: 53, False Negative: 3, Unmatched: 0
Checkpoint: 5, States: 175, R-rules: 486, DMPSC-rules: 44, Checked: 42, Hits: 42, False Negative: 0, Unmatched: 2
Checkpoint: 6, States: 175, R-rules: 373, DMPSC-rules: 52, Checked: 52, Hits: 50, False Negative: 2, Unmatched: 0
Checkpoint: 7, States: 176, R-rules: 381, DMPSC-rules: 57, Checked: 57, Hits: 55, False Negative: 2, Unmatched: 0
Checkpoint: 8, States: 177, R-rules: 555, DMPSC-rules: 62, Checked: 61, Hits: 61, False Negative: 0, Unmatched: 1
Checkpoint: 9, States: 177, R-rules: 638, DMPSC-rules: 54, Checked: 54, Hits: 51, False Negative: 3, Unmatched: 0
Checkpoint: 10, States: 178, R-rules: 859, DMPSC-rules: 57, Checked: 57, Hits: 54, False Negative: 3, Unmatched: 0
Checkpoint: 11, States: 178, R-rules: 702, DMPSC-rules: 58, Checked: 54, Hits: 54, False Negative: 2, Unmatched: 2
Checkpoint: 12, States: 178, R-rules: 727, DMPSC-rules: 56, Checked: 56, Hits: 56, False Negative: 0, Unmatched: 0
Checkpoint: 13, States: 178, R-rules: 1069, DMPSC-rules: 70, Checked: 70, Hits: 63, False Negative: 7, Unmatched: 0
Checkpoint: 14, States: 178, R-rules: 941, DMPSC-rules: 65, Checked: 64, Hits: 60, False Negative: 4, Unmatched: 1
Rates:Hits: 747(0.946768060837), False Negative: 36(0.0456273764259), Unmatched: 6(0.00760456273764)
```

Figura 23. Relatório das extrações do dataset hh107 na Quinta-Feira (thuesday) considerando intervalos 28 dias

As regras em questão são apresentadas na Figura 24 e estão agrupadas de acordo com o *checkpoint* destacando seu antecedente, consequente e suas métricas.

```
Checkpoint 05
LS017-HIGH -> ['LS021-LOW', 0.5104166666666666, 1.1094339622641507, 0.9245283018867924, 49]
LS004-LOW -> ['LS021-LOW', 0.5, 1.1076923076923078, 0.9230769230769231, 48]

Checkpoint 08
LS019-LOW -> ['LS021-LOW', 0.4583333333333333, 1.5563743551952836, 0.9565217391304347, 44]

Checkpoint 11
LS022-LOW -> ['LS019-LOW', 0.7395833333333334, 1.106134371957157, 0.9102564102564104, 71]
LS019-LOW -> ['LS015-HIGH', 0.75, 1.10751482134273337, 0.9113924050632912, 72]

Checkpoint 14
LS022-LOW -> ['LS019-LOW', 0.6145833333333334, 1.3831501831501831, 0.9076923076923077, 59]
```

Figura 24. Regras não comparadas para a quinta-feira do *dataset hh107* e intervalo de 28 dias.

Nos relatórios é possível observar que, pela análise descentralizada, todas as regras satisfazem as métricas mínimas, embora alguma estejam bem próximas de serem classificadas como inadequadas. À exemplo disto, tem-se a regra do *checkpoint 22* que possui a confiança de 90,07%, sendo o limiar mínimo de suporte igual à 90%. O mesmo se aplica para as demais regras, porém em relação a outras métricas.

Uma vez que tais métricas estão próximas ao limite foram executados experimentos específicos para tais *checkpoints* considerando métricas com limites mais flexíveis (*support*: 0.01, *lift*: 1 e *confidence*:0.8) de tal forma que fosse possível observar os valores obtidos pela análise centralizada nestes casos específicos. A Regras geradas considerando as novas métricas são apresentadas na Figura 25.

```
Checkpoint 05
{LS017-HIGH} => {LS021-LOW},0.515789473684211,0.924528301886792,1.09787735849057,49
{LS004-LOW} => {LS021-LOW},0.505263157894737,0.923076923076923,1.09615384615385,48

Checkpoint 08
{LS019-LOW} => {LS021-LOW},0.656716417910448,0.956521739130435,1.08621960206337,44

Checkpoint 11
{LS019-LOW} => {LS015-HIGH},0.782608695652174,0.911392405063291,1.06136837045345,72
{LS022-LOW} => {LS019-LOW},0.771739130434783,0.91025641025641,1.06004543979228,71

Checkpoint 14
{LS022-LOW} => {LS012-LOW},0.808219178082192,0.907692307692308,1.05177045177045,59
{LS022-LOW} => {LS019-LOW},0.808219178082192,0.907692307692308,1.05177045177045,59
```

Figura 25. Regras extraídas pelo modelo centralizado com limiar flexível das métricas.

Conforme o esperado, é possível observar na Figura 25 que a métrica *lift* possui valores inferiores às apresentadas pelo DMPSC (Figura 24). Tal diferença é gerada devido a forma que é calculado o suporte de um *itemset* nos modelos.

Durante a análise associativa em ambiente centralizado, o tamanho da base de transações varia de acordo com dispositivo que possui o padrão de mudanças de estados com maior volume de *slots* preenchidos, ou seja, pode assumir valores de 0 à |*T*| (número de colunas da matriz de contadores), enquanto que para análise descentralizada este valor sempre é igual à |T| uma vez que o processo iterativo trabalha apenas com informações locais, desconhecendo-se o padrão com maior número de *slots* preenchidos. Sendo o valor do *lift* de uma regra inversamente proporcional ao suporte do consequente e sendo o suporte do consequente inversamente proporcional tamanho da base, é possível afirmar que, ao considerar que o tamanho do *dataset* igual à |*T*|, o suporte do consequente será sempre o menor valor possível no intervalo $0 \leq x \leq |T|$. Consequentemente o *lift* calculado pelo DMPSC será sempre maior ou igual ao calculado pelo R-Rules já que seu denominador (suporte do consequente da regra) é o menor possível.

Em síntese, podemos definir esta particularidade do modelo como um reflexo de um grau mais elevado de sensibilidade do DMPSC em relação ao R-Rules. Tal sensibilidade permite destacar correlações que não são identificadas em uma arquitetura centralizada.

Capítulo 7

Considerações Finais

Neste capítulo são discutidas as considerações finais sobre esta dissertação e aborda os seguintes tópicos: Limitações da Proposta (Seção 7.1), Próximos Passos (Seção 7.2), Cronograma (Seção 7.3) e Publicações (Seção 7.4).

7.1. Limitações da Proposta

Embora os experimentos tenham apresentados resultados satisfatórios, foi possível observar algumas características do modelo que limitam seu uso e aplicação.

Predição de Estados: Embora seja possível realizar a predição de estados, esta se limita à dispositivos com apenas 2 estados, se for desconsiderado o intervalo de tempo correspondente a um *slot*. Por serem mutuamente excludentes, caso uma ação seja mais provável de ocorrer, sua probabilidade é equiparada ao estado contrário ao que o dispositivo irá assumir após a ação. Por exemplo, se em um dado *slot* há 80% de chances de ocorrer a ação “Desligar Lâmpada”, isso significa que em 80% das ocorrências ela estava “Ligada”, já que para transitar para “Desligada” é necessário estar em “Ligada”.

Discretização do tempo: Recomenda-se que a base de dados embarcada represente apenas um dia (24 horas) para que a análise associativa seja executada corretamente. Definir uma quantidade de slots para mais de 24 horas irá gerar correlações inválidas pois a correlação pode ocorrer ente 2 dias diferentes. Por exemplo: considere 2 dispositivos (DevA e DevB) que possuem 192 slots (48horas / 15 minutos), porém o DevA foi instalado no ambiente em uma terça-feira e o DevB em uma quarta-feira. Durante o processo de correlação os dados irão gerar correlações entre padrões de dias diferentes.

Múltiplas bases de dados: para o experimento executado, optou-se em criar uma base independente para cada dia da semana. Ressalta-se que para este cenário é necessário armazenar as regras de correlações para cada base de dados de forma independente, ou seja, as interações entre os dispositivos podem variar de acordo com o dia da semana. Embora estas características permitam que os dispositivos se adaptem melhor ao padrão de uso do usuário, o consumo de espaço de armazenamento dobra a cada base de dados criada.

Quantidade de dispositivos: por ser um processo iterativo, o tempo de execução da extração de correlação na rede é diretamente proporcional à quantidade de dispositivos que devem ser analisados. A redução do tempo de execução, neste caso, pode ser definindo grupos de *multicasts* diferentes para cada ambiente. Dessa forma os dispositivos que controlam um determinado ambiente interagem entre si, evitando a geração de correlações

indesejadas (ex.: “Abri Janela do Quarto” e “Ligar Irrigador do Jardim”) e reduzindo o tempo de execução da extração.

7.2. Próximos passos

Para dar continuidade ao desenvolvimento do método proposto, os próximos passos visam otimizar o processo de descoberta de correlações bem como definir uma métrica para identificar quando e se é necessário realizar o processo de extração de correlações.

Em um segundo momento serão realizados experimentos em cenário real de uso com o objetivo de observar a capacidade do método se adaptar às necessidades do usuário, capacidade de troca de informações e, finalmente, observar o comportamento colaborativo entre os dispositivos em ambientes distintos.

7.2.1. Dispositivos Inteligentes – Protótipos

O monitoramento/controle e o gerenciamento dos objetos serão realizados por meio de protótipos capazes que interagem de forma não invasiva com os objetos controlados. Tal protótipo possui uma arquitetura genérica que possibilita sua fácil adaptação para atuar em diversos cenários.

O protótipo (Figura 26) é composto por 2 componentes principais, um microcontrolador ATTINY 85 20 PU e um ESP8266S-01. Ambos se comunicam através de portas seriais e desempenham papéis diferentes dentro da arquitetura. Dispondo de 2 GPIO’s, sendo uma analógica e outra digital, é possível conectar o protótipo à sensores e atuadores, além suportar alimentação de 3.3V e 5V.

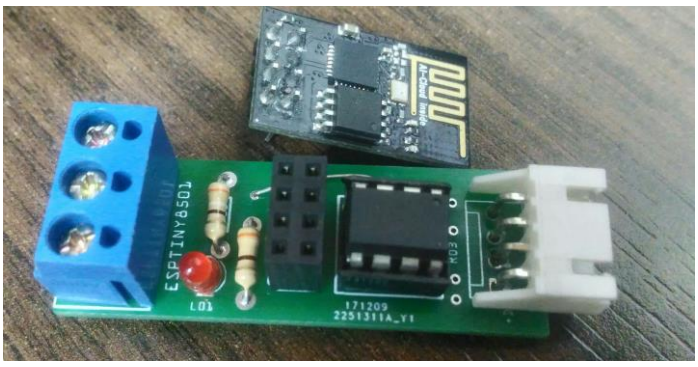

Figura 26. Protótipo para integração com objeto.

Ambos componentes implementam partes da arquitetura apresentada na Figura 12 sendo cada qual responsável em executar atividades diferentes. Enquanto o ESP8266 disponibiliza todos os recursos necessário para conectividade, armazenamento e processamento de dados, o ATTINY se adapta ao sensor/atuador conectado às GPIOs de tal forma que o processo de discretização dos valores é definida em seu *firmware*, ou seja, o ATTINY é responsável em ler

os valores reais e encaminhar ao ESP os rótulos que associam os intervalos aos estados dos sensores.

7.2.2. Cenários reais de uso

Para explorar a aplicação do método em diversos ambiente, foram definidos três cenários principais para sua avaliação:

- **Ambiente residencial (Experimento A).** Experimento será realizado em uma residência contendo uma família com 3 membros (2 adultos e 1 criança). A residência possui 4 ambientes divididos em banheiro, cozinha, sala e quarto. A Figura 27 apresenta a distribuição dos sensores no ambiente que irão monitorar portas, janelas, geladeira, luzes, ar condicionados, chuveiro, fogão, sensor de temperatura ambiente e de chuva.

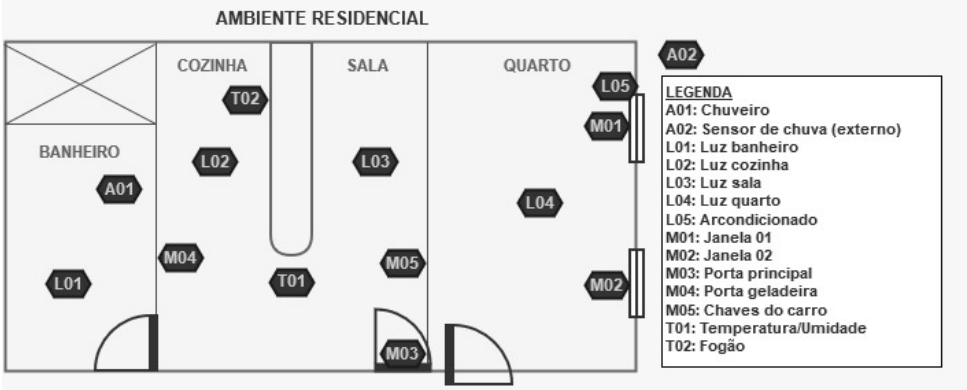

Figura 27. Distribuição de sensores em um ambiente residencial.

- **Ambiente profissional (Experimento B).** Experimento será realizado na sala de um professor vinculado ao Instituto de Computação da Universidade Federal do Amazonas. A Figura 28 apresenta a distribuição dos sensores no ambiente que irão monitorar a porta (M01), um ar-condicionado (L01), uma lâmpada (L02) e um sensor de temperatura (T01).

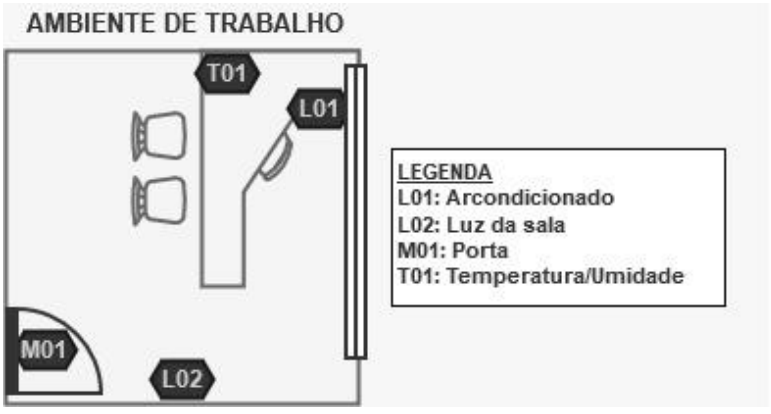

Figura 28. Distribuição de sensores em ambiente de trabalho.

- **Ambiente de estudo (Experimento C).** Experimento será realizado no Laboratório de Sistemas Embarcados no Bloco do Centro de Tecnologia da Informação e Comunicação (CTIC) na Universidade Federal do Amazonas. A Figura 29 apresenta a distribuição dos sensores no ambiente que irão monitorar as luzes (L01, L02), a porta (M01), o ar-condicionado (L03) e um sensor de temperatura (T01).

Figura 29. Distribuição dos sensores em ambiente de estudo.

7.3. Cronograma

A Tabela 14 apresenta o cronograma de atividades que serão desempenhadas durante o desenvolvimento deste estudo.

Tabela 14. Cronograma de atividades.

	2018							2019		
ETAPAS/MES	**JUN**	**JUL**	**AGO**	**SET**	**OUT**	**NOV**	**DEZ**	**JAN**	**FEV**	**MAR**
Otimização do modelo	X									
Experimento/Análise		X								
Preparação		X								
Experimento Cenário A		X	X							
Experimento Cenário B			X	X						
Experimento Cenário C				X	X					
Relatórios		X	X	X	X	X				
Publicações/Artigos					X	X	X			
Revisão da dissertação							X	X	X	
Dissertação										X

- **Otimização:** Inserir métricas e procedimentos que otimizem o modelo;
- **Experimento/Análise:** Executar experimentos simulados para comparação de resultados;
- **Preparação:** Desenvolvimento dos firmwares para os dispositivos inteligentes, confecção de hardwares e ajustes no ambiente do experimento;
- **Experimento A:** Execução do experimento em ambiente residencial;
- **Experimento B:** Execução do experimento em ambiente de trabalho;
- **Experimento C:** Execução do experimento em ambiente de acadêmico;
- **Relatórios:** Documentação das regras sugeridas, ativas e mapeamento de integração entre os dispositivos

- **Publicações/Artigos:** Desenvolvimento de textos para publicações em conferências e jornais de relevância na área
- **Revisão da Dissertação:** Realizar ajustes da banca (qualificação), inserir relatórios e resultados dos experimentos quanto a otimização e cenários reais de uso e acrescentar as publicações realizadas
- **Defesa:** Realizar defesa da dissertação

Referências

AGRAWAL R.; SRIKANT R. **Fast Algorithms for Mining Association Rules.** Proceeding of 20th VLDB Conference, Santiago, Chile, 1994

ALENCAR, Márcio A. da C*.* ***Distributed Mining of Pairs States Changes***. Disponível em: <https://github.com/macalencar/dmpsc>. Acesso em 20, junho, 2018

AL-FUQAHA, Ala; GUIZANI, Mohsen; MOHAMMADI, Mehdi; ALEDHARI, Mohammed; AYYASH, Moussa. **Internet of things: A survey enabling technologies, protocols and applications**. IEEE Communication Survey & Tutorials, vol. 17, n. 4, Fouth Quarter 2015.

ALI, S.H. **Miner for OACCR: Case of medical data analysis in knowledge discovery.** 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications - SETIT, 2012

BARR, Michael; MASSA, Anthony. **Programming Embedded Systems - With C and GNU Development Tools.** Second Edition. Sebastopol-CA: O'Reilly Media, 2009, 336 p.

BASILI, V. R.; CALDIERA, G.; ROMBACH, H. D. **The Experience Factory**. MARCINIAK, J. J. (Ed). In: Encyclopedia of Software Engineering, New York, John Wiley & Sons, 1994

BERGUER, Arnold S. **Embedded Systems Design: An introduction to Processes Tools & Techniques.** Taylor & Francis Ltda, 2001, 272 p.

BIOLCHINI, J.; MIAN, Paula G.; NATALI, Ana C. C.; TRAVASSOS, Guilerme, H. et al. **Systematic Review in Software Engineering. Relatório Técnico**, COPPE/UFRJ, 2005. Disponível em: <www.cin.ufpe.br/~in1037/leitura/systematicReviewSE-COPPE.pdf>. Acesso em: 12 abr. 2010.

BUYYA, R; DASTJERDI, Amir V. **Internet of things: Principles and Paradigms.** Cambidge-MA: Morgan Kaufmann Publishers, 2016, 354p ;

CAEIRO, Frederico. **Probabilidade e Estatística**. Faculdade de Ciências e Tecnologia Universidade Nova de Lisboa, Portugal**. Disponível em: <http://orium.pw/univ/lei/pe/SebentaPE_200910.pdf >. Acesso em 19 de janeiro, 2018.**

CAI, Zhipeng; BOURGEOIS, Anu; TONG, Weitian. **Guess Editorial: Special issue on Internet of Things.** Tsinghua Science and Technology, vol. 22, n. 4, Agosto, 2017.

CHEN, Yunliang; LI, Fangyuan; FAN, Junqing. **Mining association rules in big data with NGEP. Cluster Computing,** Ed. Springer, ver. 18, pg. 577-585, Junho, 2015.

CRANDALL, D. C. A; THOMAS, B; KRISHNAN. N. **CASAS: A smart home in a box**. IEEE Computer, 46(7):62-69, 2013.

EVANS, Dave. **The internet of things: How the next evolution of the internet is changing everything.** Cisco Internet Business Solution Group, Abril, 2011.

GONZALEZ, L.I.L.; AMFT, O. **Mining relations and physical grouping of building-embedded sensors and actuators.** IEEE International Conference on Pervasive Computing and Communications - PerCom, 2015

GRUENWALD, L.; CHOK, H.; ABOUKHAMIS, M. **Using data mining to estimate missing sensor data.** IEEE International Conference on Data Mining - ICDM, 2007

GUILLAME-BERT, M; CROWLEY, J.L. **Learning temporal association rules on symbolic time sequences**, in Proceedings of the 4th Asian Conference on Machine Learning, ACML 2012, Singapore, Singapore, November 4-6, 2012, 2012, pp. 159–174.

HAHSLER, Michael; CHELLUBOINA, Sudheer; HORNIK, Kurt; BUCHTA, Christian; **The arules R-package ecosystem: Analyzing interesting patterns from large transaction datasets**. Journal of Machine Learning Research Issue 12, 2011, p 1977--1981.

HAN, Jiawei; KAMBER, Micheline; PEI, Jian. **Data Mining: Concepts and Techniques.** Third Edition. Waltham-MA: Morgan Kaufmann Publishers, 2012, 703p ;

HEALTH, Steve. **Embedded Systems Design.** Second Edition. Burlington-MA: Newnes Elsevier, 2003, 430 p;

HEIERMAN, Edwin O. III; COOCK, Diane J. **Improving home automation by discovering regularly occurring device usage patterns.** Proceeding of the 3º IEEE International Conference on Data Mining, Novembro, 2003.

KARIMI-MAJD, A. M.; MAHOOTCHI, M. **A new data mining methodology for generating new service ideas.** Information Systems and e-Business Management, 2015

KARTHIK, K. **Key search and adaptation based on association rules for backward secrecy.** IEEE International Workshop on Information Forensics and Security - WIFS, 2015

KITCHENHAM, B. **Procedures for Performing Systematic Reviews. Joint Technical Report TR/SE-0401**. Software Engineering Group, Department of Computer Science, Keele University, Australia, 2004. Disponível em: <http://www.idi.ntnu.no/emner/empse/papers/kitchenham_2004.pdf>. Acesso em: 15 mar. 2010.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for Perfirming Systematic Literacture Reviewss in Software Engineering.** Ver. 2.3. Relatório Técnico, Eviente-Based Software Engineering (ESBSE) 2007. Disponível em:http://www.elsevier.com/inca/publications/misc/inf>. Acesso em: 15 mar. 2010.

LYNDEN, S. **Analysis of Semantic URLs to Support Automated Linking of Structured Data on the Web.** ACM International Conference Proceeding Series, 2017

MAFRA, S. N.; TRAVASSOS, Guilherme. H. **Estudos primários e secundários apoiando a busca por evidências em engenharia de softwares.** Relatório Técnico, PESC – COPPP/UFRJ, 2006. Disponível em: <http://www.cos.ufrj.br/uploadfiles/1149103120.pdf>. Acessado em 12 mar. 2010.

MAKOSHENKO, Denis; ENKOVICH, Ilya. **IoT Development: Discovering, Enabling, and Validation of a Real Life IoT Scenarios.** Second International Conference on Fog and Mobile Edge Computing - FMEC - 2017

McARTHUR, David P.; ENCHEVE, Sylvia; THORSEN, Inge. **Exploring the Determinants of Regional Unemployment Disparities in Small Data Sets.** International Regional Science Review, v. 35(4), p. 442-463, 2012

MORI, T.; TAKADA, A.; NOGUCHI, H.; HARADA, T.; SATO, T. **Behavior prediction based on daily-life record database in distributed sensing space.** IEEE/RSJ International Conference on Intelligent Robots and Systems - IROS, 2005

PAL, K.; ADEPU, S.; GOH, J. **Effectiveness of association rules mining for invariants generation in cyber-physical systems.** Proceedings of IEEE International Symposium on High Assurance Systems Engineering, 2017

PAUL, R.; GROZA, T.; HUNTER, J.; ZANKL, A. **Decision Support Methods for Finding Phenotype - Disorder Associations in the Bone Dysplasia Domain**, PLoS ONE, 2012

PULIAFITO, Carlo; ANASTASI, Giuseppelya. **Fog Computing for the Internet of Mobile Things: issue and challenges.** IEEE Internet Computing, Vol. 21, Issue 2, Março, 2017.

QIU, M.; JIA, Z.; XUC, C.; SHAO, Z.; LIU, Y.; SHA, E.H. M. **Loop scheduling to minimize cost with data mining and prefetching for heterogeneous DSP**. International Conference on Parallel and Distributed Computing and Systems, 2006

ROSE, Karen; ELDRIDGE, Scott; CHAPIN, Lyman. **The Internet Of Things: And Overview - Understanding the Issues and Challenges of a More Connected World.** Internet Society, Outubro, 2015.

SHANMUGANTHAN, S.; NARAYANAN, A.; MOHAMED, M.; IBRAHIM, R.; KHALID, H. **A hybrid approach to modelling the climate change effects on Malaysia's oil palm yield at the regional scale.** Advances in Intelligent Systems and Computing, 2014

SMITH, M.R.; WANG, X.; RANGAYYAN, R.M. **Evaluation of the sensitivity of a medical data-mining application to the number of elements in small databases**. Biomedical Signal Processing and Control, 2009

SOONG, T. T. **Fundamentals of Probability and Statistics for Engineers.** Ed. John Willey & Sons, 2004

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. **Introduction to data mining.** Ed. Pearson Addison Wesley, 2006.

VERHELST, Marian; MOONS, Bert. **Algorithmic and processor techniques bring deep learning to IoT and edge devices: Embedded Deep Neural Network Processing**. IEEE Solid-State Circuits Magazine, FALL, 2017

WANG, X.; CHEN, M.; CHEN, L. **Research of the optimization of a data mining algorithm based on an embedded data mining system.** Cybernetics and Information Technologies, 2013

XU, Y.; MA, Z.; CHEN, X.; LI, L.; DILLON, T.S. **Improving frequent patterns mining by LFP.** International Conference on Wireless Communications, Networking and Mobile Computing - WiCOM, 2008

ZAMBONI, A. B.; THOMMAZO, A. D.; HERNANDES, E. C. M.; FABBRI, S. C. P. F. (2010) **StArt Uma Ferramenta Computacional de Apoio à Revisão Sistemática**. In: Brazilian Conference on Software: Theory and Practice - Tools session. UFBA.