Extração Descentralizada de Regras de Associação em Bases de Dados Embarcadas de Dispositivos Internet das Coisas

Márcio Alencar, Raimundo Barreto,
Horácio Fernandes e Eduardo Souto
Instituto de Computação da UFAM
Manaus, Amazonas - BR 69067–005
{macalencar, rbarreto, esouto, horacio} @icomp.ufam.edu.br

Richard Pazzi
Ontario Tech University - OntarioTEch
Oshawa, Ontario - CA L1G 0C5
Email: richard.pazzi@uoit.ca

***Resumo*—Embora extração de conhecimento de bases de dados seja geralmente realizada em arquiteturas centralizadas, sua execução em cenário descentralizado é um desafio computacional importante, especialmente quando realizadas no contexto da Internet das Coisas onde há as rigorosas restrições de armazenamento e processamento nos dispositivos. Para contornar tais restrições, este artigo apresenta um método para minerar correlações implícitas entre as ações disponíveis nos dispositivos IoT através da análise associativa embarcada. Baseada nas variações das métricas *support*, *lift* e *confidence*, o método proposto identifica as correlações mais relevantes entre pares de ações de diferentes dispositivos e sugere-as ao usuário como forma de integração através de requisições HTTP. Os experimentos mostraram que, em média, as regras mais relevantes foram as mesmas em ambas arquiteturas para 99.75% dos casos. Além disso, o método proposto foi capaz de identificar correlações relevantes que não foram identificadas pela arquitetura centralizada, reforçando assim o entendimento positivo de que a análise do padrão de ações dos dispositivos é uma abordagem eficiente para prover um ambiente IoT altamente integrado e inteligente apesar das restrições existentes em cenários descentralizados.**

Palavras-Chaves: Internet das Coisas, Software Embarcado, Análise Associativa, Mineração de Dados Descentralizada

I. Introdução

A habilidade de prover inteligência e autonomia aos objetos na Internet das Coisas recai, principalmente, na necessidade de identificar padrões e correlações implícitas nestes/entre dispositivos. A arquitetura, centralizada ou descentralizada, sobre a qual estes ambientes inteligentes são implementados, está diretamente relacionada à técnica adotada para realizar o reconhecimento de padrões e correlações. Explorar tais técnicas nos permitem otimizar processo em diversos cenários que partem da automação residencial até complexas tarefas que reforcem a segurança em ambientes industriais.

De acordo com [1], [2] e [3], as redes neurais profundas podem se tornar um modelo popular e útil para reconhecimento, classificação e previsão de padrões em IoT. No entanto, devido ao alto custo computacional, seu uso em uma arquitetura totalmente descentralizada, onde cada objeto tem seu próprio controlador independente dos outros, requer dispositivos que possam lidar com o processo de aprendizado profundo embarcado, como computadores de placa única (e.g. *Raspberry Pi, Beagle Board, Rock64*). Essa abordagem pode resultar em desperdício de recursos e custos elevados, dependendo da quantidade de dispositivos. Por outro lado, uma arquitetura que requer o compartilhamento de recursos (arquitetura centralizada) apresenta uma característica delicada que é uma grande limitação para aplicações de baixo custo em IoT, a dependência do nó central em que ocorre o processo de aprendizagem. Tal método cria um único ponto de falha que pode afetar todos/ou vários os dispositivos ao mesmo tempo no caso de um problema no nó central. Uma segunda observação é a quantidade de recursos disponíveis naquele nó. Para lidar com a quantidade de dados gerados pelos dispositivos IoT, o nó central deve dispor de vários recursos de processamento e armazenamento para reter e analisar todas as informações geradas a partir dos dispositivos controlados, aumentando os custos de implementação e manutenção.

Para superar essas barreiras este artigo propões o método DARE (*Decentralized Association Rules Extraction*), o qual identifica correlações entre diversos bancos de dados embarcados em dispositivos IoT que compõem um ambiente inteligente. Nesta abordagem, cada dispositivo fornece todos os mecanismos para armazenar, processar, analisar e compartilhar os dados necessários para identificar correlações implícitas entre suas ações e as ações de outros dispositivos no mesmo ambiente, mas, diferente da abordagem usual de mineração de dados, o método proposto quebra o processamento em pequenos passos para que possa ser feito dentro do dispositivo IoT, garantindo sua autonomia. Além disto, este abordagem introduz um esquema para economizar recursos de processamento e armazenamento de dados sem perder a capacidade de identificar correlações relevantes.

Nas seções seguintes iremos discutir a teoria da análise associativa (Seção II) e os trabalhos relacionados (Seção III). Seguidos da apresentação do método proposto e suas especificações (Seção IV). Posteriormente são apresentadas a avaliação (Seção V), os resultados dos experimentos (Seção VI) e conclusões (Seção VII).

II. Análise Associativa

A análise associativa é a descoberta de regras que exibem as condições atributo-valor que geralmente ocorrem juntas em um determinado conjunto de dados (*dataset*) [1], [4]–[6]. Este *dataset* é formado por transações que são compostas por múltiplos itens independentes e não repetidos. Um exemplo simplificado pode ser aplicado em um *dataset* de um supermercado que registra as compras dos clientes, onde a correlação (regra) pode ser expressa através de uma inferência que sugere que frequentemente (métricas) os consumidores que compram café (antecedente) também compram leite (consequente). A relevância desta inferência é dada pelas métricas que indicam o quão frequente (*support*), o quão dependente (*lift*) e o quão confiável (*confidence*) esta correlação é.

A. Métricas

Geralmente, não limitando-se à isso, uma regra de associação é considerada interessante se ela satisfizer um valor mínimo de *support*, a qual é uma métrica que reflete o quão útil a regra é. Além desta, há outra métrica, chamada *confidence*, que define a garantia de uma regra descoberta. Ambas são formalmente representadas por:

$$support(A \Rightarrow B) = P(A \cup B) = \frac{freq(AB)}{|D|} \quad (1)$$

$$confidence(A \Rightarrow B) = \frac{support(A \Rightarrow B)}{P(A)} = \frac{freq(AB)}{freq(A)} \quad (2)$$

A Equação (1) representa a probabilidade de ambos itens (A e B) aparecerem juntos em D, e a Equação (2) representa o quão frequente B é, dada todas as transações em D que possuem o item A. Além destas, uma terceira métrica, chamada *lift* é definida por:

$$lift(A \Rightarrow B) = \frac{confidence(A \Rightarrow B)}{support(B)} = \frac{freq(AB) \times |D|}{freq(A) \times freq(B)} \quad (3)$$

Esta métrica define que a ocorrência *A* é independente da ocorrência de *B* se $P(A \cup B)$ for igual a $P(A)P(B)$, caso contrário *A* e *B* são dependentes e correlacionados.

Diferente das métricas *confidence* e *support*, as quais tem seus valores dentro do intervalo de 0 à 1, a métrica *lift* define que valores menores que 1 indicam que A e B são inversamente correlacionados ($A \wedge \neg B$); valores iguais à 1, A e B são independentes e não estão correlacionados; ou, para valores maiores que 1, A e B são diretamente correlacionados.

Geralmente os algoritmos de regras de associação buscam identificar o maior conjunto de itens frequentes em um *dataset*. Para isso há várias abordagens como as Regras de Associação *Apriori* [7], *FP-Growth* [8], and *Constraints-Based Mining* [9]. Diferente desses, o DARE busca identificar regras que indiquem uma forte dependência do antecessor e do consequente, sendo ambos presentes com uma alta frequência no *dataset* e compostos de apenas um item cada.

III. Trabalhos relacionados

Esta seção discute estudos que utilizam técnicas de análise associativa para correlacionar dispositivos distribuídos e/ou extrair correlações interessantes de *datasets* pequenos (com pouco volume de dados).

Uma abordagem distribuída, chamada $DPmining$ [10], extrai conhecimento de diversos dispositivos em uma rede de sensores sem fio. Esta abordagem faz uso de *clusters* de dispositivos no qual um nó central (controlador do cluster) gerencia as partições dos *datasets* que serão processadas por um ou mais dispositivos pertencentes ao cluster.

O método proposto em [5], chamado TEREDA, extrai a ordem das atividades, seu tempo de início e duração através da aglomeração (*clustering*) do tempo de inicio das atividades e correlaciona-os ao seu tempo de duração. Além disso, através ao algoritmos de análise associativa *FP-Growth*, a atividade atual se correlaciona com a próxima atividade mais provável, baseada em teu intervalo de tempo.

A pesquisa de [6] resultou em um preditor, baseado nos padrão de estados dos dispositivos, das probabilidade de uma falha no sistema por meio do algoritmo de *FP-Growth*.

O trabalho apresentado por [11] explora a sensibilidade e a confiabilidade das regras extraídas de um *dataset* com poucos registros, que correlaciona diagnósticos clínicos e laboratoriais à mamografias. As regras foram avaliadas pelo método *Effron's Bootstrap*, o qual produz vários modelos a partir de amostras do *dataset*. Quanto maior o número de modelos parecidos, mais confiável é a regra.

A meta dos estudos apresentados por [12] e [13] foi identificar os padrões de uso de um ambiente inteligente. Entretanto, o primeiro busca identificar os padrões de consumo de energia dos dispositivos, enquanto o segundo identifica o padrão de estados do dispositivos. Através de técnicas de agregação e mineração de regras de associação, foi possível concluir que, embora cada dispositivo possua um padrão único de uso, alguns dispositivos compartilham de características similares que revelam a correlação entre eles. Além disso, foi possível classificar as atividades do usuário agrupando tais similaridades.

Os estudos conduzidos por [14] explora a mineração de regras de associação (ARM) em *dataset* pequenos para identificar correlações entre as taxas de desemprego e fatores socioeconômicos no Sudoeste da Noruega. As correlações foram validadas usando técnica de *Formal Concept Analysis* (FCA) a qual, inesperadamente, identificaram as mesma correlações que o ARM, indicando que as maiores taxas estão localizadas nos centros econômicos que, intuitivamente, possuem a menor probabilidade de altas taxas de desempregos.

A proposta de [15] ameniza o impacto do custo da busca em análises associativas reduzindo o número de itens frequentes e excluindo registros duplicados. Além disso, outra otimização proposta pelos autores é a compressão de dados a qual, considerando valores binários para indicar os ausência (bit 0) e a presença (bit 1) dos itens em um *dataset*, pode representar uma transação como uma *string* em que cada caractere representa um item indicando sua presença ou não na transação;

Buscando explorar as correlações entre grupos de sensores e atuadores em um prédio, o método proposto por [16] demonstra que tais correlações estão limitadas a um ambiente específico em um dado intervalo de tempo. O método proposto, chamado de *Weighted Transitive Clustering*, é baseado em correlações temporais entre as mudanças de estados dos dispositivos de um mesmo ambiente.

Os estudos conduzidos por [17] identificou possíveis ataques a uma planta de estação de tratamento de água através da geração de invariantes nos dados gerados pelos sensores. Os experimentos consistiram em identificar e correlacionar tais invariantes entre os 51 sensores por meio do algoritmo de Regra de Associação *Apriori* [7].

As evidências apresentadas nos estudos de [5], [6], [12]–[17] demonstram que é possível extrair correlações interessantes ao analisarmos os padrões de mudança de estados dos dispositivos, invés de seus padrões de uso. Esta pequena diferença reduz muito o volume de dados gerados que deverão ser analisados, satisfazendo um requisito essencial em sistemas que não dispõem de muitos recursos. Além disto, o processamento de *clustering* para agregação de registros, apresentado em [5], [6], [11], [16], e [10], aumenta o custo computacional durante a etapa de pré-processamento, mas otimiza o processamento de dados durante a análise associativa por meio da redução do número de candidatos.

As características e técnicas avaliadas nos trabalhos relacionados nos ajudaram a propor alternativas que resolvem os principais problemas de nossa abordagem tais como: (i) registro de valores em intervalos de tempos discretos como método de pre-processamento para agrupamento de valores; (ii) uso de análise probabilística para determinar qual ação possui a maior probabilidade de ocorrer em um dados intervalo discreto de tempo; e (iii) a distribuição probabilística considerando todos os intervalos de tempo discretos, que facilita a identificação de similaridades entre os dispositivos. Nas seções seguintes serão apresentadas como estas características e técnicas serão aplicadas ao método proposto neste artigo.

IV. Método Proposto

O esquema proposto neste artigo, DARE, é um método colaborativo no qual cada dispositivos deverá comparar seu próprio padrão aos de outros para identificar as correlações mais relevantes entre as ações do dispositivo local e as ações do dispositivo remoto. Este mecanismo é baseado no algoritmo de Mineração de Regras de Associação *Apriori* [7] porém o cálculo de suas métricas apresentam uma ligeira modificação que será apresentada nesta seção.

O dispositivo deve realizar periodicamente comparações, em seu ambiente embarcado, e gerar regras que são aplicadas apenas para si. Isto é, cada dispositivo terá seu próprio conjunto de regras, as quais permitirão a sincronização do estado de um dispositivo remoto (consequente da regra) baseada em uma simples ação de entrada no dispositivo de origem (antecedente da regra) se ambas ações satisfizerem os padrões de ambos dispositivos no intervalo de tempo corrente.

Para melhor entendimento do método proposto, é necessário esclarecer os seguinte itens: o comportamento esperado dos dispositivos (Seção IV-A), armazenamento de dados (Seção IV-B), e o processo de mineração (Seção IV-C).

A. Estados e Ações

No método proposto, cada dispositivo possui dois conjuntos: $S = \{s_1, s_2, \ldots, s_i\}$ como um conjunto finito de i itens que representam os possíveis estados do dispositivo, e $A = \{a_1, a_2, \ldots, a_i\}$ como um conjunto de i ações disponíveis (e.g.: "lâmpada ligada" e "lâmpada desligada") que permitem transitar entre os estados em S (e.g.: "ligar" e "desligar").

Considerando que cada dispositivo age de forma independente, os mesmos devem prover todos os recursos necessários para tratar os estímulos físicos (sinais e interrupções) e/ou estímulos lógicos (requisições HTTP)

B. Base de dados embarcada e Padrão de ações

Assumindo que $T = \{t_1, t_2, \ldots, t_j\}$ é um conjunto finito de j intervalos discretos de tempo (*slots*), é possível definir uma base de dados embarcada como uma matriz $M_{ij} = A \times T$ (ver M_{ij} na Tabela I) onde cada elemento c_{xy} é um contador para cada ação $a_x \in A$ no *slot* $t_y \in T$. Esses contadores irão incrementar à cada estímulos recebido que gere uma mudança de estado no dispositivo (padrão de ações).

A extração do padrão de ações de M_{ij} é realizada por uma transformação logarítmica (ver M_{ij}' na Tabela I) em todos os contadores para reduzir o impacto de informações antigas e excluir registros não usuais (i.e.: *outliers*). Esta transformação é definida por: $c_{xy} \leftarrow \log_{(|A|)} c_{xy} \mid 1 \leq x \leq |A| \ \wedge \ 1 \leq y \leq |T|, \forall c_{xy} \in M_{ij}$. Caso o resultado da transformação seja menor que 1, o contador assumirá o valor 0, evitando que em transformações futuras sejam gerados valores negativos.

Após a transformação, é possível extrair um padrão de ações confiável da seguinte forma: Sendo $C_y = \{c_{1y}, \ldots, c_{iy}\}$ um conjunto contendo todos os contadores da coluna y em M_{ij}, e uma função $max\_action(C_y, A)$ a qual retorna a ação (do conjunto *A*) correspondente ao maior valor obtido em C_y (ou *null* em caso de não predominância de apenas uma ação), então é possível criar um conjunto $P = \{(p_1, \ldots, p_j) \in A \mid \forall p_y \leftarrow max\_action(C_y, A)\}$, onde $1 \leq y \leq |T|$ que representa o padrão de ações do dispositivo (ver P_l na Tabela I).

Este esquema de armazenamento reduz a quantidade de dados que devem ser pré-processados para realizar a análise associativa embarcada. Além disso, é importante esclarecer que todos os dispositivos devem assumir o mesmo número de *slots* ($|T|$) para possibilitar a criação de uma base de transação (D) através do agrupamento das ações de um mesmo *slot*, formando uma única transação. Tal procedimento é apresentado na Tabela I onde o Padrão local (P_l) é combinado com um padrão remoto (Pr) para gerar a base de transação (D). Esta premissa também é refletida nas Equações (1) e (3) onde o valor de $|D|$ sempre será igual ao número máximo de *slots* ($|T|$). Esta modificação permite que as regras extraídas localmente, a partir de apenas um par de padrões, possuam métricas válidas globalmente, mesmo desconhecendo os padrões de todos os demais dispositivos.

Tabela I: Exemplos de armazenamento de dados embarcado, transformações de valores, padrões de ações e base de transações

SLOTS (T)→		1	2	3	4	5	6	7	8	9	10
$\mathbf{M_{ij}}$	**LIGAR**	2	1	4	9	25	13	4	0	1	4
	DESLIGAR	5	3	4	6	34	17	2	0	0	5
$\mathbf{M_{ij}}$'	**LIGAR**	1	0	2	**3.16**	4.64	3.7	**2**	0	0	2
	DESLIGAR	**2.32**	**1.58**	2	2.58	**5.08**	**4.08**	1	0	0	**2.32**
	$\mathbf{P_l}$	DESLIGAR	DESLIGAR	-	LIGAR	DESLIGAR	DESLIGAR	LIGAR	-	-	DESLIGAR
	$\mathbf{P_r}$	ABRIR	ABRIR	FECHAR	-	-	ABRIR	-	-	FECHAR	ABRIR
	D	DESLIGAR, ABRIR	DESLIGAR, ABRIR	FECHAR	LIGAR	DESLIGAR	DESLIGAR, ABRIR	LIGAR	-	FECHAR	DESLIGAR, ABRIR

$\mathbf{M_{ij}}$: Base de dados embarcada
$\mathbf{M_{ij}}$' : Base de dados transformada
$\mathbf{P_l}$: Padrão de ações (da base M_{ij}')
$\mathbf{P_r}$: Padrão de ações (de um dispositivo remoto)
D : Bases de transações (P_l + P_r)

C. Extração de correlações

O processo de extração de correlações ocorre individualmente em cada dispositivo em intervalos de tempo fixos (chamados de *checkpoints*), de forma que é possível se adaptar ao padrão de interação dos usuários. Para isso, um dispositivo deve conhecer todos os demais (dispositivos alvos) que possuam um padrão para compartilhar. Portanto cada dispositivo deve ingressar previamente em um grupo de *multicast* específico, permitindo-os criar uma lista de alvos por meio de um pacote *echo request/response* de *multicast* [18]. Também é possível atribuir diferentes grupos de *multicasts* para os dispositivos agrupando-os por diferentes características (por exemplo, localização, funcionalidade, diferentes intervalos de *slot*) e reduzindo a quantidade de iterações durante mineração.

Vale ressaltar que o método proposto desconsidera aspectos relacionados ao roteamento de pacotes, *firewalls* e configurações de ISP (*Internet Service Provider*). Para isso, o administrador da redes deve fornecer todos os recursos para garantir que todos os dispositivos possam se comunicar.

Uma vez que o dispositivo obtenha a lista de alvos, um processo iterativo, ilustrado na Figura 1, é iniciado:

Passo I: O dispositivo identifica seu próprio padrão de ações (P_l: Padrão Local) a partir da base de dados embarcada (M_{ij}), conforme descrito na Seção IV-B e ilustrado na Tabela I;

Passo II: O *Coletor* solicita o padrão de ações do (próximo) destino na lista de alvos. Este, que recebe a solicitação, realiza o Passo I em si mesmo respondendo a requisição enviando seu padrão de ações, representados por P_r (Padrão Remoto).

Passo III: O *Fusor* recebe ambos padrões (P_l e P_r) e gera a base de transações (D) unindo as ações de ambos padrões para cada *slot* da seguinte forma: $D = \{(p_{l1}, p_{r1}), \ldots, (p_{lj}, p_{rj})\}$ onde $p_{ly} \in P_l$, $p_{ry} \in P_r$, e $1 \leq y \leq |T|$;.

Passo IV: Finalmente, a regra de associação extrai correlação mais relevante em D para cada ação do dispositivo local. Então as métricas da regra atual são comparadas com as previamente obtidas e armazenadas na Base de Correlações, as quais podem ser substituídas, anexadas ou ignoradas, de acordo com os valores de *support*, *lift* e *confidence* (nesta ordem específica).

Este processo se repete até que todos os dispositivos alvo sejam comparados e as regras mais relevantes sejam armazenadas na Base de Correlações do dispositivo que está executando a extração.

Embora seja recomendado que para cada ação do dispositivo local seja associada uma única regra que o correlacione a outra ação de um dispositivo remoto é possível que a mesma possua mais de uma correlação, sendo um critério de desenvolvimento e disponibilidade de recursos.

V. Avaliação do Método Proposto

A avaliação consistiu em identificar o quão similar são as regras extraídas ao analisar os mesmos conjuntos de dados pelas abordagens descentralizada (DARE) e centralizada (Regra de Associação *Apriori*). A extração de dados centralizada foi

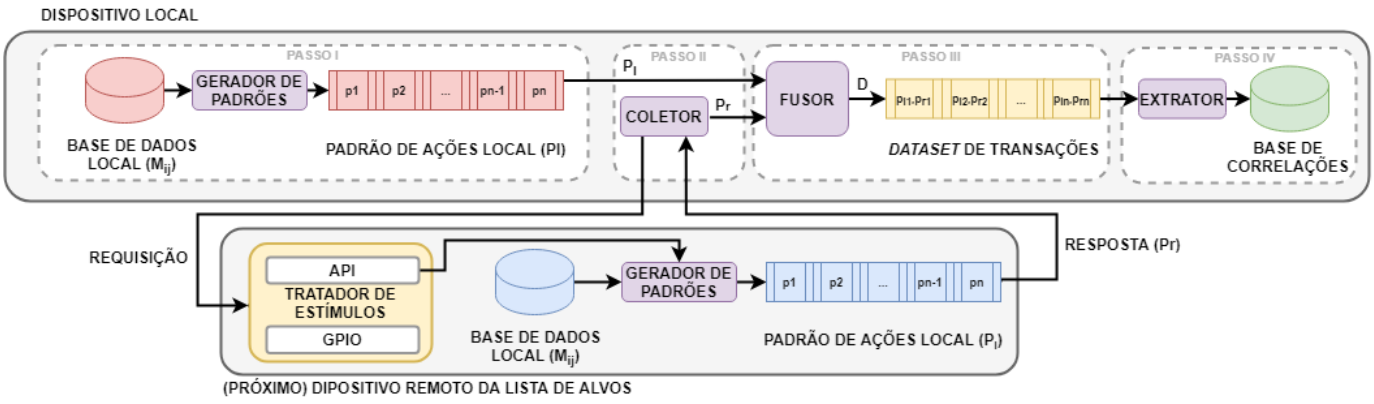

Figura 1: Passos do método proposto

realizada através do software R [19] com o auxílio da biblioteca *aRules* [20], que implementa as Regras de Associação *Apriori* [7] e o DARE foi implementado em Python 3 para execução dos experimentos.

Foram avaliados cinco *datasets* públicos (WSU CASAS [21]) considerando diferentes intervalos entre os *checkpoints* para extração as regras. Para cada *checkpoint* ambos métodos eram executados diferenciando-se na formação da base de transações onde o DARE extraiu as regras analisando várias bases de transações formadas por diferentes pares de padrões de ações enquanto o *aRules* analisou apenas uma base de transação formada por todos os pares de ações dos dispositivos. Este processo se repetiu para cada *checkpoint* até o fim dos registros válidos para cada *dataset*. Como métricas, definiu-se duas variáveis que iriam registrar as ocorrências durante as comparações das regras, sendo *hit* caso fossem iguai e *miss* caso fossem diferentes.

A. Parâmetros da avaliação

Os parâmetros para a matriz de contadores e limites mínimos das metricas das análise associativa são:

- Cada *slot* agrega 15 minutos ($|T|$ = 96);
- *Support* $\geq$ 1% (1 *slot* preenchido);
- *Lift* $\geq$ 1.1 (correlações diretas: $A \wedge B$);
- *Confidence* $\geq$ 90% ;

Considerou-se também a geração de *checkpoints* nos seguintes intervalos de tempo:

- **Intervalo I**: diariamente;
- **Intervalo II**: semanas alternadas;
- **Intervalo III**: uma semana sim e três semanas não;

Esses parâmetros foram os mesmos para todos os conjuntos de dados em todos os experimentos.

VI. Resultados

A Tabela II apresenta número de dispositivos e quantidade de registros antes e depois da limpeza e a taxa de redução. O processo de limpeza ignora os registros que não representam uma mudança de estado dos dispositivos, conforme especificado na Seção IV-B, e transforma os valores contínuos em intervalos discretos com base na média dos valores registrados.

Tabela II: Dados e resultados do pré-processamento

DATASET	# DISP.	REGISTROS	USADOS	REDUÇÃO
hh107	110	3,369,689	2,811,279	16.57%
hh123	88	2,907,282	2,345,775	19.31%
hh129	86	12,303,984	56,523	99.54%
shib009	8	3,187,940	90,599	97.16%
tokyo	67	802,534	171,483	78.63%

Este processo resultou em uma redução massiva nos registros de dados para os *datasets* `hh129` e `shib009`, mais precisamente 99,54% e 97,16%, respectivamente. No *dataset* `tokyo`, a redução foi de 78,63% do seu conteúdo original. Os outros dois *datasets*, `hh107` e `hh123`, tiveram reduções menores, 16,57% e 19,31%, respectivamente. Essas reduções são reflexos do grande número de dispositivos que registraram valores contínuos (e.g. sensores de temperatura). Considerando que esses valores foram discretizados, muitos registros não expressaram uma mudança real nos estados, se caracterizando como informações redundantes.

A Tabela III mostra os resultados dos experimentos para todos *datasets*. Nela estão contidos os números de regras identificadas como as mais relevantes em ambas as análises (*acertos*) e os números de regras identificadas apenas pelo DARE (*erros*). Além disso, existem as taxas médias para cada *dataset* e as taxas médias para todo o experimento.

Tabela III: Resultados Experimentais

DATASET	ACERTOS/ERRROS			TX MÉDIA (%)	
	I	II	III	HITS	MISSES
hh107	3,656/-	3,493/-	5,769/43	99.67	0.33
hh123	1,952/-	2,373/-	4,040/15	99.82	0.18
hh129	80/-	54/-	54/-	100	-
shib009	16/-	135/-	75/-	100	-
tokyo	508/-	337/-	473/-	100	-
TOTAL	**23,015 / 58**			**99.75**	**0.25**

Todos os experimentos para os *datasets* `hh129`, `shib009` e `tokyo` obtiveram 100% de concordância. Em outras palavras, todas as regras identificadas pelo DARE também foram as mais relevantes nas análise centralizada.

Exceções ocorreram nos *datasets* (`hh107` e `hh123`) onde, apesar de haver concordância integral nas regras obtidas para o intervalo I e II, para o intervalo III, o *aRules* discorda do *DARE* em 43 regras em `hh107` e 15 regras em `hh123`. Essas regras não foram consideradas relevantes pela análise centralizada, uma vez que não atendiam aos limites mínimos da métrica. Avaliando os motivos de tal comportamento, foi possível identificar um fator comum para todos os casos. Para fins didáticos, a Tabela IV apresenta uma amostra dessas regras não identificadas (*aRules Permissivo*) em comparação às regras identificadas pelo DARE.

Todas as três regras tiveram o mesmo consequente, e suas frequências na base de transação são expressas pela coluna "contagem". Ligeiras diferenças nas métricas de *support* e *lift* foram observadas em ambas as análises. A explicação para esta diferença é que, na análise centralizada, todos os padrões são reunidos em uma única transação e seu tamanho ($|D|$) pode assumir qualquer valor entre 0 a 96, pois alguns *slots* podem estar vazios caso não haja uma ação mais provável entre todos os padrões do dispositivo (ver Tabela I slot 8). Além disso, a abordagem centralizada utiliza as Equações (1) e (3), que são diretamente dependentes de $|D|$, fazendo com que as métricas de suporte e confiança assumam valores menores que os obtidos na análise descentralizada, que sempre assume que $|D| = |T|$ nas mesmas Equações (ver Seção IV-C).

Com os experimentos é possível afirmar que tal modificação permite que o DARE seja mais sensível às regras que podem ter valores próximos aos limites das métricas, conforme exposto na Tabela IV. A regra $\#2$ está presente em 71 dos 96 registros e foi considerada uma correlação interessante pelo DARE, enquanto o *aRules* a ignorou, pois o tamanho da base de transação assumiu um valor inferior a $|T|$ durante a análise centralizada.

Tabela IV: Comparação das métricas do DARE com as métricas do *aRules*)

#	Antecedente	Consequente	DARE (Misses)			*aRules* (Permissivo)			Contagem
			Support	Confidence	Lift	Support	Confidence	Lift	
1	LS023 (HIGH)	LS021(LOW)	0.5729	0.9821	1.1359	0.5978	0.9821	1.0886	55
2	LS019 (LOW)	LS021(LOW)	0.7395	0.9861	1.1405	0.7717	0.9861	1.0930	71
3	LS004 (HIGH)	LS021(LOW)	0.6250	0.9836	1.1376	0.6521	0.9836	1.0902	60

Todos os resultados experimentais, código-fonte, imagens, regras extraídas e os links para acesso aos conjuntos de dados estão disponíveis em [22].

VII. Conclusão

O foco do DARE é fornecer um mecanismo embarcado que permita que cada dispositivo extraia conhecimentos distribuídos a partir de um ambiente inteligente, mas possuindo recursos limitados para armazenar, gerenciar e processar. Esse mecanismo correlaciona pares de ações do dispositivo para oferecer aos usuários um conjunto de sugestões de integração inteligente entre as ações dos dispositivos.

Este trabalho reproduz (com 99,75% de concordância) um conhecido algoritmo de mineração de dados centralizado (*Apriori*), porém, utilizando uma abordagem descentralizada, onde a mineração de regras é realizada em ambiente embarcado, correlacionando as ações de um dispositivo às de outros quando satisfeitos os critérios mínimos de *suport*, *lift* e *confidence*. Além disso, modificação no cálculos das métricas de *support* e *lift* permitiram identificar regras relevantes não identificadas pela abordagem centralizadas.

Apesar dos bons resultados, o DARE apresenta algumas limitações que devem ser exploradas em trabalhos futuros, tais como: (i) limitação da dimensionalidade do conjunto de dados, que cresce proporcionalmente ao número de ações/estados/*slots* disponíveis; (ii) explorar aplicações multi-domínio; (iii) criar protótipos para executar experimentos do mundo real; e (iv) avaliar a experiência do usuário relacionada a sugestões para integração de dispositivos.

Agradecimentos

Esta pesquisa, conforme previsto no Art. 48 do decreto nº 6.008/2006, foi parcialmente financiada pela Samsung Electrônica da Amazônia Ltda, nos termos da Lei Federal nº 8.387/1991, convênio nº 003/2019, firmado com o ICOMP/UFAM. Reconhecemos o apoio concedido pela *Ontario Tech University (UOIT)*; Fundação de Amparo à Pesquisa do Estado do Amazonas (FAPEAM), bolsa de doutorado e Edital Universal 002/2018; *Natural Sciences and Engineering Research Council of Canada (NSERC)*; e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Referências

[1] F. Chen, P. Deng, J. Wan, D. Zhang, A. Vasilakos, and X. Rong, "Data mining for the internet of things: Literature review and challenges," *International Journal of Distributed Sensor Networks*, 2015.

[2] M. Verhelst and B. Moons, "Embedded deep neural network processing: Algorithmic and processor techniques bring deep learning to iot and edge devices," *IEEE Solid-State Circuits Magazine*, vol. 9, no. 4, pp. 55–65, 2017.

[3] M. Z. Alom, T. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. Nasrin, M. Hasan, B. Essen, A. Awwal, and V. Asari, "A state-of-the-art survey on deep learning theory and architectures," *Electronics*, vol. 8, p. 292, 03 2019.

[4] L. Li, Q. Li, Y. Wu, Y. Ou, and D. Chen, "Mining association rules based on deep pruning strategies," *Wireless Personal Communications*, vol. 102, pp. 2157–2181, Oct 2018.

[5] E. Nazerfard, "Temporal features and relations discovery of activities from sensor data," *Journal of Ambient Intelligence and Humanized Computing*, May 2018.

[6] V. Kireev, A. Guseva, P. Bochkaryov, I. Kuznetsov, and S. Filippov, "Association rules mining for predictive analytics in iot cloud system," in *Biologically Inspired Cognitive Architectures* (A. V. Samsonovich, ed.), (Cham), pp. 107–112, Springer, 2019.

[7] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. of 20th Intl. Conf. on VLDB*, pp. 487–499, 1994.

[8] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining frequent patterns without candidate generation: A frequent-pattern tree approach," *Data Mining and Knowledge Discovery*, vol. 8, pp. 53–87, Jan 2004.

[9] J.-F. Boulicaut and B. Jeudy, *Constraint-Based Data Mining*, pp. 399–416. Boston, MA: Springer US, 2005.

[10] M. Z. A. Bhuiyan and J. Wu, "Event detection through differential pattern mining in internet of things," in *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 109–117, Oct 2016.

[11] M. Smith, X. Wang, and R. Rangayyan, "Evaluation of the sensitivity of a medical data-mining application to the number of elements in small databases," *Biomedical Signal Processing and Control*, vol. 4, no. 3, pp. 262–268, 2009.

[12] Y.-C. Chen, Y.-L. Ko, and W.-C. Peng, "An intelligent system for mining usage patterns from appliance data in smart home environment," in *Conference on Technologies and Applications of Artificial Intelligence(TAAI)*, pp. 319–322, 2012.

[13] E. Heierman and D. Cook, "Improving home automation by discovering regularly occurring device usage patterns," in *IEEE International Conference on Data Mining, ICDM*, pp. 537–540, 2003.

[14] D. McArthur, S. Encheva, and I. Thorsen, "Exploring the Determinants of Regional Unemployment Disparities in Small Data Sets," *International Regional Science Review*, vol. 35, no. 4, pp. 442–463, 2012.

[15] X. Wang, M. Chen, and L. Chen, "Research of the optimization of a data mining algorithm based on an embedded data mining system," *Cybernetics and Information Technologies*, vol. 13, no. SPECIALISSUE, pp. 5–17, 2013.

[16] L. Gonzalez and O. Amft, "Mining relations and physical grouping of building-embedded sensors and actuators," in *2015 IEEE International Conference on Pervasive Computing and Communications, PerCom 2015*, pp. 1–10, 2015.

[17] K. Pal, S. Adepu, and J. Goh, "Effectiveness of association rules mining for invariants generation in cyber-physical systems," in *Proceedings of IEEE International Symposium on High Assurance Systems Engineering*, pp. 124–127, 2017.

[18] S. Venaas, "Multicast Ping Protocol." RFC 6450, Dec. 2011.

[19] R Development Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[20] M. Hahsler, S. Chelluboina, K. Hornik, and C. Buchta, "The arules r-package ecosystem: Analyzing interesting patterns from large transaction datasets," *Journal of Machine Learning Research*, vol. 12, pp. 1977–1981, 2011.

[21] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer (Long Beach Calif)*, vol. 46, p. 10.1109/MC.2012.328, Jul 2013.

[22] M. Alencar, R. Barreto, H. Oliveira, R. Pazzi, and E. Souto, "eMbedded Associative Knowledge Extraction - MAKE." https://github.com/macalencar/make, 2018.