DARE: A decentralized association rules extraction scheme for embedded data sets in distributed IoT devices

Márcio Alencar, Raimundo Barreto, Horácio Fernandes, Eduardo Souto and Richard Pazzi

Abstract
In the context of smart home, it is very important to identify usage patterns of Internet of things (IoT) devices. Finding these patterns and using them for decision-making can provide ease, comfort, practicality, and autonomy when executing daily activities. Performing knowledge extraction in a decentralized approach is a computational challenge considering the tight storage and processing constraints of IoT devices, unlike deep learning, which demands a massive amount of data, memory, and processing capability. This article describes a method for mining implicit correlations among the actions of IoT devices through embedded associative analysis. Based on support, confidence, and lift metrics, our proposed method identifies the most relevant correlations between a pair of actions of different IoT devices and suggests the integration between them through hypertext transfer protocol requests. We have compared our proposed method with a centralized method. Experimental results show that the most relevant rules for both methods are the same in 99.75% of cases. Moreover, our proposed method was able to identify relevant correlations that were not identified by the centralized one. Thus, we show that associative analysis of IoT device state change is efficient to provide an intelligent and highly integrated IoT platform while avoiding the single point of failure problem.

Keywords
Internet of things, knowledge discovery, embedded software, association analysis, decentralized data mining

Introduction
The Internet of things (IoT) is composed of everyday objects with embedded and distributed systems that allow their access and control via the Internet. When combined with intelligence and machine learning techniques, IoT devices can understand their surroundings and take appropriate, automated actions, allowing them to be used in some exciting scenarios ranging from simple residential automation to complex industrial safety tasks.

The ability to provide intelligence and autonomy to IoT devices relies mainly on the need to identify patterns and implicit correlations from such devices. The architecture, centralized or decentralized, in which the intelligent environments are implemented is directly related to the pattern recognition technique adopted.

According to Chen et al. and Verhelst and Moons, the deep neural networks can become a popular and helpful model for pattern recognition, classification,
and prediction in IoT. However, due to the high computational cost, its use in a fully decentralized architecture, where each object has its own controller independent from the others, requires devices that can handle an embedded deep learning process, such as single-board computers (e.g. Raspberry Pi, Beagle Board, and Rock64). This approach can result in a waste of resources with high costs, depending on the number of devices. However, an architecture that requires sharing resources from a single device (centralized architecture) presents a delicate characteristic that is a major limitation for low-cost applications in IoT, namely, the central node dependency in which the learning process occurs. Such a method creates a single point of failure that can affect all of the devices at the same time in the case of a problem on the central node. A second observation is the amount of resources available in that node. In order to handle the amount of data generated by IoT devices, the central node must dispose of several resources for both processing and storage to retain and analyze all of the generated information from the controlled devices, increasing both implementation and maintenance costs.

In order to overcome these barriers, we propose a method that performs association knowledge extraction from several decentralized databases embedded in IoT devices that compose a smart environment. Each device provides all mechanisms to store, process, analyze, and share all data needed to identify implicit correlations between its actions and the actions of other devices in the same environment, but, different from usual data mining approach, the proposed method breaks the processing in small steps so it can be done inside of IoT device, guaranteeing its autonomy.

Driven by the challenge to obtain globally interesting correlations based on local association analysis, the proposed method, called decentralized association rules extraction (DARE), presents an approach that identifies the strongest correlations between a pair of devices, which share the same embedded storage parameters, and uses them to create an intelligent and integrated environment that models the users’ behavior. The observed correlations could be used as triggers to synchronize the states of multiple devices in a chain reaction and adapt the environment during the user’s activities, providing ease and comfort in their daily activities. Furthermore, this approach introduces an embedded storage and data processing scheme to save space without losing the capability to extract knowledge using the constrained resources from distributed IoT devices.

This approach could be used in several multi-domain applications such as traffic control systems in smart cities to avoid traffic jams or for setting up fast lane corridors for emergency services such as fire brigades and ambulances. As for applications in a single-domain environment such as smart buildings/houses, it is possible to synchronize the states of different devices, for example, power off the air conditioner (AC) when the office’s lights switch off at 18:00 o’clock, turn on a computer when an employee registers its entrance in the building, or even closing the houses windows when it starts raining.

The remainder of the article is organized as follows. In section “Association analysis,” this theory is discussed. In section “Related work,” we show how association analysis has been used in the literature to perform extractions in a decentralized environment with small data sets. Next, in section “Proposed method,” we describe the proposed architecture and its components followed by performance evaluation in section “Rule’s extraction evaluation.” Finally, section “Conclusion” presents conclusions and future research directions.

**Association analysis**

An association analysis is the discovery of rules that exhibit attribute-value conditions that often occur together in a given data set.\(^1\)\(^3\)\(^5\) This data set is formed by transactions that are composed of multiple items. A rule could be expressed as \( A \Rightarrow B \)\[^{metrics}\] where \( A \) defines the premise of rule (antecedent), \( B \) is the conclusion of rule (consequent), and \( metrics \) are the values that quantify the quality of this inference, in other words, the metrics express how relevant is this rule.

A simple example of this association analysis could be applied to a supermarket database where the rules correlate groups of items that are often shown together. An instance of this correlation (rule) could be expressed as an inference that suggests that usually (metrics) the customers who buy coffee (antecedent) also buy milk (consequent). The relevance of this inference is given by the metrics that indicate how frequent (support), how dependent (lift), and how reliable (confidence) this inference is.

**Metrics**

Typically, association rules are considered interesting if they satisfy the minimum support thresholds, which is an interesting measure of a rule that reflects its usefulness. Also, there is another metric called confidence, which defines the assurance of discovered rules. Assuming \( support(A) \) as the probability \((P)\) of an item \( A \) appears in a given data set \((D)\) of \(|D|\) registers and \( support(B) = P(B) \) as well, then, a rule that correlates \( A \) and \( B \) has both metrics formally represented as...
The lift metric states that the occurrence of an item set \( A \) is independent from occurrence of an item set \( B \) if \( P(\{A \cup B\}) = P(A)P(B) \), otherwise \( A \) and \( B \) are dependent and correlated.

Different from confidence and support, which have their range of values from 0% to 1% (or 0%–100%), lift defines that values less than 1 indicate that \( A \) and \( B \) are inversely correlated (\( A \land \neg B \)); if lift equals 1, then \( A \) and \( B \) are independent and are not correlated; or if the lift was greater than 1, \( A \) and \( B \) are directly correlated.

Typically, an association analysis aims at identifying the largest and most frequent item set in the data set. For that, there are several association analysis algorithms such as Apriori,FP-Growth, and Constraint-Based Mining. Different from these, our proposed method aims at identifying the strongest correlations between two items. In other words, we are looking for a rule that strongly correlates the antecedent and the consequent formed by 1-item set each. Assuming that each device’s state is an item in a transaction data set, through these metrics, it is possible to identify how the devices correlate to each other and how a single state’s change could affect the state of other devices.

### Related work

This section presents studies that use association analysis techniques to correlate distributed devices and/or extract interesting correlations from small data sets. It also considers studies that perform extractions by grouping registers to reduce the dimensionality of data sets as a mechanism to reduce the processing workload during the association analysis. These issues represent essential features of our proposed approach (small data sets, embedded data mining, and storage/processing restrictions). This section is based on a systematic review available in Alencar et al. The work presented by Smith et al. assessed the sensitivity and reliability of rules obtained from a small database that correlates clinical diagnoses and laboratory evaluations from a mammography data set. The rules were evaluated by Effron’s bootstrap method, which consists of creating several models from a database sample and comparing these models. The smaller the number of different models, the more reliable the rule. Even though it is an interesting technique, creating many models and comparing them demand a considerable amount of processing and memory space.

The work presented by Chen et al. and Heierman and Cook was to identify the usage patterns in an intelligent environment. However, the first identifies the device’s power consumption patterns, and the second identifies a device’s state pattern. Through aggregation techniques and association rule mining (ARM), it was possible to conclude that, although each device has a unique usage pattern, some devices have similar characteristics that reveal a correlation among them. Also, it was possible to classify the user’s activity by clustering these similarities. However, both methods cannot predict the number of clusters created, and some interesting correlations may be lost if the clusters belong to slightly different time intervals.

The work conducted by McArthur et al. explores the ARM in a small database to identify correlations between the unemployment rate and socioeconomic factors in southwestern Norway. To validate these correlations, the formal concept analysis (FCA) technique was applied, which is a principle-based method of deriving a conceptual hierarchy or formal ontology, from a
collection of objects and their properties. Unexpectedly, both techniques (ARM and FCA) identified a correlation between the attributes of unemployed citizens and the higher unemployment rate regions. The result expresses that the same rules created by the mining method were expressed by relevant correlations between properties in FCA and shows that high unemployment is not occurring in rural or peripheral areas, but is occurring in the economic centers which, intuitively, have lower probabilities of having high unemployment rates. To achieve these results, the author analyzed a small but dense data set with a high computational cost, which is not recommended for constrained devices.

The proposal of Wang et al. reduces the impacts on the searching cost in the association analysis by reducing the frequent item sets and removing duplicate records. Moreover, another optimization proposed by the authors is data compression where, considering binary values to represent the absence (i.e. bit 0) and presence (i.e. bit 1) of a particular item, a transaction can be represented by a string containing a list of items present and its index in a checklist. Although this compression mechanism reduces the storage space needed to represent the data, this method considers a centralized transaction data set, which could be a problem in decentralized IoT environments.

The proposal of Wang et al. reduces the impacts on the searching cost in the association analysis by reducing the frequent item sets and removing duplicate records. Moreover, another optimization proposed by the authors is data compression where, considering binary values to represent the absence (i.e. bit 0) and presence (i.e. bit 1) of a particular item, a transaction can be represented by a string containing a list of items present and its index in a checklist. Although this compression mechanism reduces the storage space needed to represent the data, this method considers a centralized transaction data set, which could be a problem in decentralized IoT environments.

Seeking to explore the correlation between groups of sensors and actuators in a building, the method proposed by Gonzalez and Amft proves their hypothesis that there is a correlation between the sensor’s state changes from a specific environment in a given space of time. The proposed grouping of variables, from the paper called Weighted Transitive Clustering, is based on timing correlations that exist between the sensors' state changes that monitor the same environment. Thus, the grouped variables have a strong correlation during their state changes. Otherwise, variables belonging to other environments are correlated. Therefore, it is possible to infer which variables are related to the same space and can be grouped based on the rules of relationships. Similar to Wang et al., this method demands access to all data set information. Besides that, considering a timing correlation, it may create an invalid set of correlations assuming devices from different environments have a similar pattern of activity.

The study conducted by Pal et al. identified possible attacks at a water treatment plant through the generation of invariants. The experiment consisted of identifying correlated invariants among the 51 sensors with the Apriori association rule algorithm. Such a study provides evidence of association rule analysis efficiency to correlate sensor states in a cyber-physical environment. The result shows that it was able to identify these invariant generations during an attack on system. Though finding correlations between sensors from cyber-physical during an attack, it was necessary to analyze a considerable amount of data to identify them as a threat.

The pieces of evidence exposed by the literature show that it is possible to extract interesting information from a device’s change patterns instead of usage patterns. This slight difference reduces the amount of data that must be stored and analyzed, which satisfies an essential requirement in constrained environments. Also, the clustering data process, presented by the literature, certainly increases the computational cost due to the pre-processing but optimizes the association analysis reducing the number of candidates. The approaches presented by McArthur et al. and Wang et al. expose more compelling evidence by extracting relevant correlations in a small but dense data set.

Table 1 compares the related work based on some interesting features that could be useful to the goal of this article. From them, we could also extract novel techniques such as (1) to register all data in a discrete embedded data set, allowing us to avoid clustering operations in raw data and also acts as a pre-processing by grouping the registers that belong to the same discrete time interval; (2) the use of probabilistic analysis to determine which actions have the highest frequency in these discrete intervals; and (3) the probabilistic

| Feature/study | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | DARE |
|---------------|---|---|---|---|---|---|---|---|---|----|------|
| Association algorithm | FP-Growth | FP-Growth | DP mining | Apriori | Apriori | Apriori | Apriori | Apriori | Apriori | Apriori |
| Small data set | X | X | X | X | X | X | X | X | X | X |
| Clustering processing | X | X | X | X | X | X | X | X | X | X |
| Distributed data set | X | X | X | X | X | X | X | X | X | X |

DARE: decentralized association rules extraction.

*Preprocessing data to create clusters

Study reference — 1: Nazerfard, 2: Kireev et al., 3: Bhuiyan and Wu, 4: Smith et al., 5: Chen et al., 6: Heierman and Cook, 7: McArthur et al., 8: Wang et al., 9: Gonzalez and Amft, 10: Pal et al.
distribution along all discrete intervals, which makes it easy to identify similarities between devices, which is attractive to our proposed method. In the following sections, it is possible to identify how these characteristics and features were applied.

Proposed method

The DARE scheme for local and remote embedded data sets in distributed IoT devices is a collaborative mechanism in which each device should compare its pattern to the other devices’ patterns to identify the strongest correlations between the device’s actions and the remote devices’ actions. This mechanism is based on the Apriori ARM algorithm, but its goal and metrics are different as shown in this section.

A device must perform these embedded comparisons periodically, and the generated rules are only applied to this specific device, that is, each device has its own set of correlation rules. These rules allow the synchronization of a remote device’s state (rule’s consequent) based on a single input action in the source device (rule’s antecedent) if these actions satisfy both device’s patterns at the current time slot.

The proposed method is described in more detail in the following sections, which present all necessary specifications to provide a comprehension of devices expected behaviors (section “States and actions”), data storage (section “Embedded database and the pattern of changes”), and the mining process (section “Correlation extraction”).

States and actions

In our proposed method, each device has two well-defined sets: $S = \{s_1, s_2, ..., s_j\}$ as a finite set of $i$ items which represent the possible states that a device has and $A = \{a_1, a_2, ..., a_i\}$ as set of $i$ actions available to transition between the $S$ states where each state represents a device’s interaction with the environment (e.g. “lamp on” and “lamp off”) and each action represents the physical or logical stimulus to change the device’s state (e.g. “turn on” and “turn off”).

Considering that each device acts independently, they must provide all the resources needed to handle the physical stimulus (signals/interruptions) and/or logical stimulus (hypertext transfer protocol (HTTP) requests).

Embedded database and the pattern of changes

The proposed method also defines a specific way to store the raw data to mitigate the lack of resources in IoT devices. Unlike the usual data mining and machine learning approach, DARE focuses on registering and analyzing the actions that stimulate a state change (pattern of actions), instead of creating multiple repeated registers for mapping the usage of devices along a period (pattern of usage). This difference reduces the amount of data that should be stored and analyzed during knowledge extraction.

Assuming $T = \{t_1, t_2, ..., t_j\}$ is a finite set of $j$ discrete time interval, it is possible to define an embedded database as a matrix of counters $M_{ij} = A \times T$ (see $M_{ij}$ in Table 2) where each element $c_{xy}$ is a counter for each action $a_x \in A$ at slot $t_y \in T$.

To extract the pattern of action from $M_{ij}$, it is necessary to perform a logarithmic transformation (see $M'_{ij}$ in Table 2) in all counters to reduce the impact of older information and to exclude unusual register (i.e. outliers). This transformation is defined by $c_{xy} \leftarrow \log_{10}(|a|) c_{xy}, 1 \leq x \leq |A| \land 1 \leq y \leq |T|, \forall c_{xy} \in M_{ij}$. Also, if the transformation values were lower than 1, the counter assumes a zero value, otherwise it assumes the value of logarithmic operation so that it is possible to avoid negative values in further transformations.

Once the transformation is performed, it is possible to extract a reliable pattern of actions as follow: let $C_r = \{c_{1r}, ..., c_{ir}\}$ be a set containing all counters from column $y$ in $M_{ij}$, and a function max action($C_r, A$) which returns the action (from the set $A$) correspondent to the greatest value from $C_r$ (or nil in case non-predominance of a single action), then it is possible to create a set

| Table 2. Illustration of an embedded data set ($M_{ij}$), transformed data set ($M'_{ij}$) and the pattern of actions. |
|------------------------------------|---|---|---|---|---|---|---|---|---|---|
| Slots ($T$) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| $M_{ij}$ | on | 2 | 1 | 4 | 9 | 25 | 13 | 4 | 0 | 1 | 4 |
| | off | 5 | 3 | 4 | 6 | 34 | 17 | 2 | 0 | 0 | 5 |
| $M'_{ij}$ | on | 1 | 0 | 2 | 3.16 | 4.64 | 3.7 | 2 | 0 | 0 | 2 |
| | off | 2.32 | 1.58 | 2 | 2.58 | 5.08 | 4.08 | 1 | 0 | 0 | 2.32 |
| $P$ | off | on | off | on | off | on | off | on | off | on | off |
| $P_r$ | open | open | close | open | off | open | on | close | open |
| $D$ | off, open | off, open | close | on | off, open | open | on | close | off, open |

$M_i$: embedded database; $P$: pattern of actions (from $M'_i$); $D$: transaction data set ($P$ and $P_r$); $M'_i$: transformed database; $P_r$: pattern of actions (from a remote device).
Table 3. Transaction data sets.

| Slots | 1 | 2 | \( |T| - 1 | |T| |
|-------|---|---|---|---|---|
| \(p_1\) | on | off | ... | \(\sim\) | off |
| \(p_2\) | \(\sim\) | open | ... | \(\sim\) | open |
| \(p_3\) | high | \(\sim\) | ... | low | \(\sim\) |
| \(D_{12}\) | on | off, open | ... | \(\sim\) | off, open |
| \(D_{13}\) | on, high | off | ... | \(\sim\) | low | off |

\[ P = \{(p_1, ..., p_r) \in A | \forall p_r \rightarrow \text{max action}(C_r, A)\}, \text{ where } 1 \leq y \leq |T|, \text{ which represents the device’s pattern of actions (see } P \text{ in Table 2).} \]

This storage organization reduces the amount of data that should be preprocessed to perform the embedded association analysis. Combining a pair of action patterns from different devices into a transaction data set \((D)\), we can extract interesting correlations between the actions from both devices.

It is important to clarify that all devices must assume the same number of slots \((|T|)\) to make it possible to create a transaction data set \((D)\) by grouping the actions from the same slot to form a single transaction, as presented in Table 3.

As this process combines the patterns in pairs \((D_{12} = P_1P_2 \text{ and } D_{13} = P_1P_3)\), it could create transaction data sets with different sizes if any empty slots exist in both patterns, as illustrated in \(|T| - 1\) from \(D_{12}\). During the association analysis, the extracted rules could not be fairly compared since the support (equation (1)) and lift (equation (3)) metrics were directly dependent on the data set size. To circumvent that, both support and lift metrics were slightly modified for the purpose of this article by assuming that all transactions data sets have the highest number of slots filled as possible in \(P\), which equals to \(|T|\). This modification allows performing a fair comparison of rules obtained from two transactions data sets with different sizes, without previously knowing the patterns of all devices.

### Correlation extraction

The process of correlation extraction occurs individually in each device in fixed time intervals (called checkpoints), so it is possible to adapt to the users’ pattern of interactions. For that, a device must know all other devices (targets) that have a pattern to share. Therefore, each device must previously join a specific default multicast group, shared by all devices, which allows a device to create a list of targets through a multicast echo request/response. Different from broadcast, the multicast protocol allows sending a message to specific nodes instead of sending the message to every host in the broadcast domain. This also allows grouping a set of devices based on their characteristics (e.g. location, functionality, and different slot intervals). Our approach does not address problems related to routing, firewalls, and intrusion prevention system (IPS) configurations. For that, the network administrator must provide all resources to make sure that all devices can reach each other.

Once the device obtains the list of targets, an iterative process, as illustrated in Figure 1, begins.

- **Step I:** The device identifies its own pattern of state changes \((P_l: \text{local pattern})\) from an embedded database \((M_p)\), as described in section “Embedded database and the pattern of changes” and illustrated in Table 2.
- **Step II:** The pattern collector component requests the pattern of actions from the (next) target in the list. The current target receives the request (through its application programming interface (API)), performs Step I on itself, and then replies its patterns, which is represented as \(P_r: \text{remote pattern})\).
- **Step III:** The fuser component receives both patterns \((P_l \text{ and } P_r)\) and creates a set of transactions \((D)\) by merging the action from both patterns in each slot as follows: \(D = \{(p_{11}, p_{12}), ..., (p_{y1}, p_{y2})\}\), where \(p_{y} \in P_l, p_{y} \in P_r\), and \(1 \leq y \leq |T|\).
- **Step IV:** Finally, the association analysis extracts the strongest correlated pairs of actions from \(D\) for each action from local set \(A\). Then, the rules from the current interaction are compared to the previously stored rules in correlation data set which may replace, be appended, or discarded according to their relevance by support, lift, and confidence metrics (in that specific order).

This process repeats until all target devices are compared and the most relevant rules are stored in the correlation data set if it is the case.

### Device’s architecture

This section describes a device architecture, all of its components and how they interact with the environment and with each other. Figure 2 depicts the proposed architecture.

As previously specified (section “States and actions”), each device is responsible for managing a single object from the environment (e.g. a temperature sensor, a water pump, an alarm, a relay, an AC, and a window). Therefore, it does not depend on any other device. This aspect helps to reduce the amount of storage and processing capability of IoT devices. Also, it controls the state space explosion by simplifying the analysis combining only single states.

Each device must be connected to a local network, joined to a shared multicast group address, and provide
an API to make all of its actions accessible (as described in section “States and actions”) to control the object states. Also, through this API, it is possible to set some devices configurations, such as the multicast address, number of slots in the data set, extractions interval, and enable/disable rules in the correlations data set.

The correlated actions, identified by the association analysis, act as a triggered reaction that synchronizes the states from a pair of devices if it satisfies their patterns of actions. As an illustrative scenario, assume that an AC identifies an interesting correlation between its action “Turn On” with the action “Close” from a specific room’s window (RW), then the rule would be

\[
\text{AC} / \text{C0 TurnON} \rightarrow \text{RW} / \text{C0 Close}
\]

Based on both devices’ patterns of actions, it is possible to use this rule to identify when the AC receives an input “Turn On” to send a request to RW perform the action “Close” if both were the most likely action to occur at the current slot time.

To obtain such behavior, each device must implement the components presented in Figure 2, which works as follows:

- **Stimulus handler** validates whether the input stimulus from the environment is logical, such as requests or messages or whether they are physical, such as interruptions on the microcontroller general-purpose input/outputs (GPIOs). If it is a valid input, it forwards a signal to the correspondent component to perform a change in the device’s state, sharing the device’s information (patterns, states, id, and others) or fire/enable/disable a correlation rule. It may also implement additional features such as user interface, general configurations, and custom functionality.

- **State controller** changes the representational device state and, in case of an actuator, sends a correspondent signal to GPIOs to change the module state (e.g. sends a “high” signal to the pin where the LED is connected). Once the action is performed, this component consults the embedded database to check the most probable action in the current slot and increases the input action counter. If the input action most likely occurs at the current slot time, it forwards a signal to Stimulus Handler to send a logical
stimulus due to an integration rule for this action (if one exists).

- Recommender manages the rules stored in the correlation data set. It forwards an HTTP request stimulating a correlation action (rule’s consequent) to the correlated device and allows the users to enable, disable, or discard integration rules. Also, after the extraction process, this component suggests the strongest correlations found to the user;
- Correlations data set stores the rules that correlate a local action to the actions from other devices. It also stores the rule’s metrics and the correlated device’s address (e.g. IP, URL).
- Database stores the matrix of counters as specified in section “Embedded database and the pattern of changes”.
- Pattern generator performs a probabilistic analysis on the embedded database to generate the device state change patterns. This component can share these patterns through the API or send them to the fuser component.
- Pattern collector identifies other devices that belong to the same multicast group and performs an interactive process to retrieve the state change pattern of each device in the target list.
- Fuser performs data fusion of local and remote patterns (created by pattern generator) to create a transaction data set. In other words, it merges the most probable actions slot-by-slot from both patterns.
- Extractor performs the association analysis on the transaction data set to identify the strength of action correlations of both devices and, additionally, updates the rules in the correlation data set.

This architecture must be the same for all devices, and the implementation decisions must share the same parameters to perform reliable knowledge extraction from the distributed data set.

**Rule’s extraction evaluation**

The methodology employed to assess the proposed method performance is based on comparative analysis between DARE and Apriori’s association rules using real smart environments data sets. The evaluation consists in comparing the rules extracted from both approaches to evaluate the reliability of DARE when dealing with the identification of correlations based only on local decisions, rather than global decisions as it is made in Apriori algorithm.

Since our performance evaluation focuses on the method, the hardware specifications of IoT devices and the networking parameters are not taken into account at this moment. The evaluation compared the rules extracted by the Apriori’s ARM to the rules extracted by DARE. All experiments were running in a computer equipped with an Intel Core i7 (3.4 GHz) processor, 8 GB of RAM, and Python 3 installed on a Linux distribution (Debian).

Multiple experiments were executed for different public data sets considering different intervals between the checkpoints. All data sets were preprocessed to remove duplicated registers, discretize the continuous values, and collect the registers that represented the device’s state changes.

The statistical software R was used to perform the centralized analysis with the help of the aRules library, which implements the Apriori association rules. For each checkpoint, all devices’ patterns were gathered to create a unique transaction data set, which was then analyzed by aRules. This allowed us to obtain all rules from all devices in a single Apriori’s algorithm execution, and it was also possible to store these rules (and their metrics) in a file (arules.log). The correlations identified by DARE were also stored together in a file (mrules.log) to facilitate the comparisons between the rules. Both files store the rules in descending order by support, lift, and confidence metrics.

The comparisons consisted of identifying whether the correlations in the mrules.log file were also the most relevant correlations in the arules.log. In this case, a metric called hit was incremented, otherwise a second metric (miss) was incremented.

**Database description**

Due to the lack of public industrial benchmarks, five different data sets from WSU CASAS were used to evaluate the proposed method. These data sets contain raw data with several sensors such as battery levels, magnetic doors sensor, light switches, lights sensors, infrared motions sensors, and temperature sensors. Table 4 shows the number of devices, number of days, and the number of registers on each data set.

These data sets were selected to perform a stress test on the proposed method in different scenarios regarding the number of devices, the number of registers, and the number of days that should be analyzed.

**Evaluation parameters**

Some parameters were predefined in the evaluations in order to mimic a real smart environment as close as possible.
All devices created a database (matrix of counters) for each day of the week (Monday to Sunday). Each slot comprehends a 15-min interval, so each matrix has 96 columns ($|T|$). Minimum support threshold: 1% (one slot filled). Minimum confidence threshold: 90%. Minimum lift threshold: 1.1 (only positive correlations).

The experiment was executed for each data set considering the following intervals:

- Interval I: the association analysis was executed day by day.
- Interval II: the association analysis was executed in alternate weeks, that is, one week yes and the other week no.
- Interval III: the association analysis was executed one week yes and three weeks no.

These parameters were the same for all data sets in all experiments.

**Results**

Table 5 shows the results of data cleaning before the execution of the algorithm. This process ignores records that do not represent a devices state change, as specified in section “Embedded database and the pattern of changes” and transforms continuous values into discrete intervals based on the range of recorded values. This process resulted in a massive reduction in the data registers for hh129 and shib009 data sets, more precisely 99.54% and 97.16%, respectively. In tokyo data sets, the reduction was 78.63% of its original content. The two other data sets, hh107 and hh123, had smaller reductions of 16.57% and 19.31%, respectively.

Table 6 shows the experimental results for all data sets. It shows the number of rules identified as the most relevant in both analyses (hits) and the number of rules identified only by DARE (misses). Moreover, there are the average rates by each data set and the average rates for the entire experiment.

All experiments for data sets hh129, shib009, and tokyo obtained 100% of agreement. In other words, all rules identified by DARE were also the most relevant in a centralized analysis.

Exceptions occurred in two data sets (hh107 and hh123) for Interval III. The aRules disagrees with DARE in 43 rules from hh107 data set and 15 rules from hh123. These rules were not considered relevant by the centralized analysis once they did not satisfy the minimum metric thresholds. In the experiments for Interval I and Interval II, all rules matched as the most relevant.

The aRules identified 23,015 rules and DARE identified 23,073 rules, namely, 58 more rules. This represents 0.25% of all correlations identified. The average accuracy of DARE reaches 99.75% of similarity compared to the rules created by a centralized analysis.

All experiments’ results are available in Alencar et al. This repository also contains the source code, images, rules extracted, and the data sets links for further consulting.

**Discussion**

This section is organized into two parts: (1) pre-processing analysis and (2) comparison between the obtained rules. The first consists of exploring the results presented in Table 5, and the second discusses the results from Table 6.

**Pre-processing analysis**

By analyzing the pre-processing step, it was possible to identify a massive reduction in data sets, especially in hh129, shib009, and tokyo. This reduction occurred due to the high number of devices that registered continuum values (e.g. a temperature sensor). Considering that these values had to be discretized, many registers did not express a real change of discretized states. An instance of that is the temperature sensor T105 from...
the hh129 data set, which has a range of registered values from 17.5°C to 33°C, and its average value is 25.25°C. In this article, we assume that each value greater or equal to the average was labeled as “high,” and the lower values were labeled as “low.” Analyzing the raw data set, one can see that the first value lower than the average occurs after 26 registers. In other words, the first 26 registered values were greater than or equal to 25.25°C. As they did not represent a real change (“high” to “low”), all these registers were counted as a single “high” in the embedded database (\(M_{ij}\)).

The same behavior repeats for multiple devices in all data sets. In this case, the number of devices that register continuum values directly affects the reduction rate. However, the range from continuum registered values in hh107 and hh123 were smaller, which implies a smaller number of discrete labels by devices. In addition, there are also fewer devices that registered continuum values. This behavior affects directly the number of rules extracted from the environment when we put aside the number of rules created (Figure 3(a)) and the amount of useful registers in data set (Figure 3(b))

A particularity was noted during the execution of the experiments with data sets hh107 and hh123 in regards to extraction Interval III. In both cases, DARE (decentralized) identified rules that could not be identified by aRules in R software (centralized). The distribution of rules that fits this particularity can be seen in Figure 4.

After analyzing the experiment’s report, it was possible to identify that some missing rules had metric values close to the minimum threshold. To check that particularity, in each experiment that contained missed rates, a new centralized analysis was performed considering more permissive metrics (metrics thresholds with lower values than the proposed in parameters), which allowed the centralized analysis to identify the missing (non-

---

### Table 6. Experiments’ results.

| Data set |   |   |   |   |   |   |
|----------|---|---|---|---|---|---|
|          | I | II | III | Hits | Misses | Average |
| hh107    | 3656/- | 3493/- | 5769/43 | 99.67 | 0.33 |
| hh123    | 1952/- | 2373/- | 4040/15 | 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** |
relevant) rules. A sample of these comparisons can be found in Table 7.

All three rules had the same consequence, and their frequencies in the transaction data set are expressed by column “count.” Slight differences in support and lift metrics were noted for both analyses. The explanation for this difference is that in the centralized analysis all patterns are gathered (as in Step III during the mining process) into a single transaction data set and its size ($|D|$) may assume any value from 0 to 96, since some slots may be empty in case there is no most probable action among all device’s patterns. Furthermore, the centralized approach uses equations (1) and (3), which are directly dependent on $|D|$, leading the support and confidence metrics to assume values lower than obtained in the decentralized analysis, which always assume that $|D| = |T|$ in the same Equations (see in section “Device’s architecture”).

This modification allows DARE to be more sensitive to rules that may have values near the metrics thresholds, as exposed in Table 7. The rule $LS019 \rightarrow low \Rightarrow LS21 \rightarrow low$ is present in 71 of 96 registers and was considered an interesting correlation by the DARE while the $aRules$ ignored it since the transaction data set size assumed a value lower than $|T|$ during the centralized analysis.

### Rules visualization

Figure 5 shows a directed graph that illustrates all correlations found from data set hh123 considering the extraction interval III (28 days) for Thursday at checkpoint 006. Each node represents a device’s state and each arrow a correlation to another device’s state (e.g. $MA006\rightarrow T107\rightarrow high$). For didactic purposes, Figure 5 ignores the temporal aspects to validate the correlation, which means the correlations may not coexist simultaneously once they are valid only for a specifics slots (see section “Device’s architecture”).

This illustration details that multiple devices could find correlations to the same consequent, as it happens to $LS012\rightarrow high$ (consequent) and $LS003\rightarrow low$, $MA011\rightarrow off$, $M001\rightarrow off$, $LS015\rightarrow high$, $LS007\rightarrow high$, and $LS016\rightarrow high$ (antecedents). In this case, if these rules coexist at the same slot and one of the antecedent changes the consequent’s state to satisfy its rule, all other devices will have its own rules satisfied as well.

The same condition must be considered to create a chain reaction as illustrated by the rules $LS006\rightarrow low \Rightarrow LS014\rightarrow high$, $LS014\rightarrow high \Rightarrow LS015\rightarrow high$, and $LS015\rightarrow high \Rightarrow LS012\rightarrow high$. If the device $LS006\rightarrow low$ fires a request to change $LS014$’s state to high, the consequent also changes the $LS015$’s state to high which fires its own request to stimulating the $LS012$ to change its state to high either. It is important to note that this approach is free of loops once a device could not fire a trigger and receive a stimulus to change to another state (different than the one that fired the trigger) at the same slot. This means that devices are not able to accept two states as the most probable state at the same slot.

### Application and emerging technologies compatibility

Extracting knowledge from embedded and decentralized data sets is an unusual approach for data mining. Our proposed solution, DARE, is flexible enough to
adapt to emerging technologies since it works in the application layer. However, there are some aspects that could be improved to make it work in other application domains.

This section describes the application of DARE considering emerging technologies and new paradigms such as massive machine-type communications (mMTC) and slicing networks specifications, the web of thing (WoT) models for standardization of objects in the WoT, and edge computing with device’s virtualization.

**MMTC**

The mMTC is a new paradigm for 5G networks focused on providing connectivity to a huge number of IoT devices. It means that mMTC devices may be connected to a given base station but, at a given time, only a random subset of them becomes active and attempt to send their data. Naturally, this approach would not allow allocating a priori resource to individual devices, instead, it is necessary to provide resources that can be shared through random access. In 5G networks, heterogeneous services are allowed to coexist within the same network architecture by means of network slicing, which is a method to allocate network computing, storage, and communication resources among the active services with the aim of guaranteeing their isolation and high performance levels. These slices could group devices by specific characteristics such as functionality, time interval to access the resources, rate of sent packets, and so on.

Both characteristics have behaviors that are similar to our proposed DARE specifications. The first assumes that, sporadically, a random and variable number of devices have access to the media to send their data. This behavior has some similarities to Step II of DARE (Figure 1), in which a mining device performs iterative requests to other devices considering their pattern of usage. DARE was designed to perform comparisons in pairs, thus, assuming that more than one device will share the resource to send data, it is possible to adapt the component pattern collector to identify which devices are simultaneously active so that it would be possible to gather the remote pattern and identify the strongest correlations.

The second feature, network slicing, allows to group multiples devices into a network segment based on specific characteristics, which is similar to the multicast group, specified in section “Correlation extraction.” This feature decreases the number of devices that must be consulted during the mining process and creates a limited scope of interactions based on these characteristics such as functionality, geographic range, data set dimensions, and others.

**WoT**

A second interesting paradigm that may coexist with mMTC and DARE is the WoT, which assumes that each device acts as a web service, through virtualization or not, providing the resources that allow to control some physical object in real world.

The WoT model proposes the basis of a common model to describe the virtual counterpart of physical objects in the WoT. It defines a model and a Web API for Things to be followed by anyone wanting to create a product, device, service, or application for the WoT. While the WoT requirements are described using the HTTP and WebSockets protocols using JavaScript Object Notation (JSON) since they are well supported, the WoT model can also be applied to describe the models and interactions of things supported via other RESTful protocols such as constrained application protocol (CoAP).

DARE also specifies a sub-component from Stimulus Handler (section “Device’s architecture”) that allows both changing the device’s states and configuring other device’s features. Since there is no specification for this API, the WoT models could fill this gap since protocols, best practices, resources, and data models, up to the semantic level, are able to define descriptions and extensions to that applications. This model would standardize all devices APIs offering a set of well-defined parameters to make these devices compatible with both new and legacy systems.

**Edge computing**

Even as a decentralized approach, the DARE architecture must provide all resources a means of controlling objects through HTTP requests (section “States and actions”), but not only that it allows to specify configuration parameters and device’s information as well (section “Device’s architecture”). This behavior allows creating a virtual model of each device in a centralized node to optimize the correlation extraction through parallel processing using edge computing, for instance.

The study conducted by Femminella et al. specifies the minimum requirements of this proposition, which is modeled using Hadoop (https://hadoop.apache.org/), a framework that allows the distributed processing of large data sets across clusters of computers using simple programming models. It is worth noting that the device’s virtualization in a centralized node would neither affect embedded mining storage nor the device’s capability to interact with each other directly. This would act as a parallel approach to ratified the embedded and decentralized processing to obtain other interesting correlations or to use a parallel knowledge’s extraction from nodes.
Conclusion

The main focus of DARE is to provide an embedded mechanism to allow IoT devices to extract knowledge from an intelligent environment that has limited resources for storage, management, and processing. This mechanism correlates pairs of device’s actions to offer a set of intelligent integration suggestions via HTTP requests to users. These suggestions, when accepted, allow the devices to control each other based on their patterns of actions, which may stimulate automatic state changes between correlated devices.

The proposed method deals with pairs of devices and only analyzes the transactions that contain useful information, while a centralized environment unnecessarily considers correlations between actions from the same devices.

This work reproduces (with 99.75% of agreement) a well-known centralized data mining algorithm (Apriori) using a decentralized approach that processes an embedded association analysis that correlates a pair of items. Also, the use of a static size data set allows for the identification of relevant correlations not found in a centralized analysis.

This article shows that DARE is an alternative for implementing a decentralized and highly integrated environment based on embedded knowledge extraction to correlate devices in a smart environment and to adapt itself to the users’ behavior patterns. The proposed method avoids the single point of failure problem since it adopts a decentralized association analysis.

Although it proves to be an efficient methodology, DARE presents some limitations that should be explored in future works, such as (1) data set dimensionality, since the embedded data set grows proportionally to the number of actions/states available, which could cause a high storage consumption; (2) discretized time interval, since the device has a high number of slots \([T]\), collecting remote patterns may fail once the device cannot handle both patterns (local and remote) on memory in order to perform the association analysis; (3) predictions, since the embedded database registers the action in discrete time slots (15 min, for instance) this does not mean that the action is active in this entire interval, but it may be activated at any time in this interval; (4) the mining processing only considers item sets with size two to reduce the amount of data that must be analyzed by the mining device (device which is performing DARE); and (5) explore a multi-domain application of the proposed method based on interoperability standards such as WoT model and manufacturer usage description.

Finally, as for future research, we intend to advance this work by creating prototypes to execute real-world experiments and evaluate the user experience related to suggestions for device integration.

Acknowledgements

This research, according to Article 48 of Decree No. 6.008/2006, was funded by Samsung Electronics of Amazonia Ltda, under the terms of Federal Law No. 8.387/1991, through Agreement No. 003, signed with Institute of Computing of the Federal University of Amazonas (ICOMP/UFAM). The authors also acknowledge the support granted by Amazon State Research Support Foundation (FAPEAM) through project No. 122/2018 (UNIVERSAL) and the Ontario Tech University (UOIT). This work was partially funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), by Discovery Grants, and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001. Also the Federal University of Amazonas and the Institute of Computing by the Graduate Program in Informatics PPGI.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Márcio Alencar [5] https://orcid.org/0000-0002-6070-9682

References

1. Chen F, Deng P, Wan J, et al. Data mining for the Internet of things: literature review and challenges. Int J Distrib Sensor Netw 2015; 11: 431047.
2. Verhelst M and Moons B. Embedded deep neural network processing: algorithmic and processor techniques bring deep learning to IoT and edge devices. IEEE Solid-State Circuit Mag 2017; 9(4): 55–65.
3. Li L, Li Q, Wu Y, et al. Mining association rules based on deep pruning strategies. Wirsel Pers Commun 102(3): 2157–2181.
4. Nazerfard E. Temporal features and relations discovery of activities from sensor data. J Ambient Intel Human Comput 2018; 2018: 149.
5. Kireev VS, Guseva AI, Bochkaryov PV, et al. Association rules mining for predictive analytics in IoT cloud system. In: Samsonovich AV (ed.) Biologically inspired cognitive architectures 2018. Cham: Springer, 2018, pp.107–112.
6. Agrawal R and Srikant R. Fast algorithms for mining association rules. In: Proceedings of the 20th international conference on very large data bases, Santiago de Chile,
7. Han J, Pei J, Yin Y, et al. Mining frequent patterns without candidate generation: a frequent-pattern tree approach. *Data Min Knowl Discov* 2004; 8(1): 53–87.
8. Boulicaut JF, Jeudy B, Maimon O, et al. Constraint-based data mining. Boston, MA: Springer, 2005.
9. Alencar M, Barreto R, Oliveira H, et al. eMbedded Associative Knowledge Extraction—MAKE, 2018, https://github.com/macalencar/make
10. Bhuiyan MZA and Wu J. Event detection through differential pattern mining in Internet of things. In: 2016 IEEE 13th international conference on mobile ad hoc and sensor systems (MASS), Brasilia, 10–13 October 2016, pp.109–117. New York: IEEE.
11. Smith M, Wang X and Rangayyan R. Evaluation of the sensitivity of a medical data-mining application to the number of elements in small databases. *Biomed Signal Pr Control* 2009; 4(3): 262–268.
12. Chen YC, Ko YL and Peng WC. An intelligent system for mining usage patterns from appliance data in smart home environment. In: *Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, Tainan, Taiwan, 16–18 November 2012. New York: IEEE.
13. Heierman E and Cook D. Improving home automation by discovering regularly occurring device usage patterns. In: *Third IEEE international conference on data mining*, Melbourne, FL, 22 November 2003. New York: IEEE.
14. McArthur D, Encheva S and Thorsen I. Exploring the determinants of regional unemployment disparities in small data sets. *Int Regional Sci Rev* 2012; 35(4): 442–463.
15. Wang X, Chen M and Chen L. Research of the optimization of a data mining algorithm based on an embedded data mining system. *Cybernet Inform Tech* 2013; 13: 5–17.
16. Gonzalez L and Amft O. Mining relations and physical grouping of building-embedded sensors and actuators. In: *2015 IEEE international conference on pervasive computing and communications*, PerCom 2015, St. Louis, MO, 23–27 March 2015. New York: IEEE.
17. Pal K, Adepu S and Goh J. Effectiveness of association rules mining for invariants generation in cyber-physical systems. In: Proceedings of IEEE international symposium on high assurance systems engineering, Singapore, 12–14 January 2017. New York: IEEE.
18. Venaas S. Multicast Ping Protocol. *RFC 6450*, 2011, https://rfc-editor.org/rfc/rfc6450.txt
19. R Development Core Team. R: a language and environment for statistical computing. *R Foundation for Statistical Computing*, Vienna, https://www.gbif.org/tool/81287/r-a-language-and-environment-for-statistical-computing
20. Hahsler M, Chelluboina S, Hornik K, et al. The Arules R-package ecosystem: analyzing interesting patterns from large transaction datasets. *J Mach Learn Res* 2011; 12: 1977–1981.
21. Cook DJ, Crandall AS, Thomas BL, et al. Casas: a smart home in a box. *Computer* 2013; 46(7): 62–69.
22. Bockelmann C, Pratas N, Nikopour H, et al. Massive machine-type communications in 5G: physical and MAC-layer solutions. *IEEE Commun Mag* 2016; 54: 59–65.
23. Trifa V, Guinard D and Carrera D. Web of thing model, https://www.w3.org/Submission/wot-model/ (accessed 6 November 2019).
24. Popovski P, Trillingsgaard KF, Simeone O, et al. 5G wireless network slicing for eMBB, URLLC, and mMTC: a communication-theoretic view. *IEEE Access* 2018; 6: 55765–55779.
25. Femminella M, Pergolesi M and Reali G. Performance evaluation of edge cloud computing system for big data applications. In: *2016 5th IEEE international conference on cloud networking (Cloudnet)*, Pisa, 3–5 October 2016, pp.170–175. New York: IEEE.
26. Lear E, Dorms R and Romascanu D. Manufacturer usage description specification. *RFC 8520*, IETF, 2019, https://www.rfc-editor.org/rfc/rfc8520.txt