Uncovering data stream behaviour of automated analytical tasks in edge computing

Lilian Hernandez* and Monica Wachowicz

University of New Brunswick,
Fredericton, NB, Canada
Email: lhernand@unb.ca
Email: lilianhrg@gmail.com
Email: monicaw@unb.ca
*Corresponding author

Robert Barton

Cisco Canada,
Vancouver, Canada
Email: robbarto@cisco.com

Marc Breissinger

Cisco USA,
Washington, DC, USA
Email: mbreissi@cisco.com

Abstract: Massive volumes of data streams are expected to be generated by the internet of things (IoT). Due to their dispersed and mobile nature, they need to be processed using automated analytical tasks. The research challenge is to uncover whether the data streams, which are being generated by billions of IoT devices, actually conform to a data flow that is required to perform streaming analytics. In this paper, we propose process discovery and conformance checking techniques of process mining in order to expose the flow dependency of IoT data streams between automated analytical tasks running at the edge of a network. Towards this end, we have developed a Petri Net model to ensure the optimal execution of analytical tasks by finding path deviations, bottlenecks, and parallelism. A real-world scenario in smart transit is used to evaluate the full advantage of our proposed model. Uncovering the actual behaviour of data flows from IoT devices to edge nodes has allowed us to detect discrepancies that have a negative impact on the performance of automated analytical tasks.

Keywords: streaming analytics; process mining; Petri Net; smart transit; internet of things; IoT; edge computing.

Reference to this paper should be made as follows: Hernandez, L., Wachowicz, M., Barton, R. and Breissinger, M. (2020) 'Uncovering data stream behaviour of automated analytical tasks in edge computing', *Int. J. Big Data Intelligence*, Vol. 7, No. 1, pp.15–28.

Biographical notes: Lilian Hernandez is a Post-Doctorate at the People in Motion Lab. Her research interests include big data analytics, process mining and the internet of things.

Monica Wachowicz is a Full Professor at the University of New Brunswick and the Director of the People in Motion Lab. She works at the intersection of streaming analytics for analysing massive IoT data in search of valuable information in real-time; and automated analytical frameworks for a world in which 'intelligence' will be embedded in virtually everything around us.

Robert Barton graduated from the University of British Columbia with a degree in Engineering Physics, and is a registered professional engineer. He holds dual CCIEs in Routing and Switching and Security, and was also the first CCDE in Canada. Currently, he is working on SmartGrid network technologies, including smart meter and intelligent substation design.

Marc Breissinger graduated from Yale University and he is currently the Executive Vice President of the Cisco Information Server. The group offers data federation, warehouse extensions, virtualisation layers, cloud data, integration, and composite monitors, as well as offers consulting, education, and training services.

1 Introduction

Streaming analytics is a process that consists of well-defined automated tasks designed to retrieve, manage and analyse data streams that are generated by a large number of IoT devices. These devices operate as providers and consumers of data related to a specific application, usually supporting point-to-point communications that are happening in real-time. They are a global network of sensors, actuators, smartphones, vehicles, appliances, wearables, and any other type of device that is converted into a ‘connected thing’ for unlocking new services that can be applied in smart transit (Mastroianni et al., 2017; Cao and Wachowicz, 2018; Hernandez et al., 2019; Tang et al., 2015), smart homes (Zhang et al., 2012; Zheng et al., 2012), manufacturing industries (Garrido-Hidalgo et al., 2017; Fernández-Caramés and Fraga-Lamas; McNeil, 2018), farming (Jeon et al., 2018; Ruengittinun et al., 2017), utility mining (Lee and Kim, 2015; Gcaba and Dlodlo, 2016; Ramesh et al., 2017), oil and gas plants (Sharma et al., 2017; Badalotti et al., 2018), to mention a few.

The analytical tasks are usually required to be automated for handling the fast data flow generated from moving data streams from IoT devices to edge nodes and later to a cloud platform. These tasks are also dependent on one another for receiving the data streams from IoT devices as an input which in turn will produce new information as part of its output. In particular, the dependencies among tasks are considered dynamic because they are executed as soon as the data streams arrive according to an event time window. Previous research work has been focused on scalability and communication issues due to the heterogeneity of technologies and the pitfalls in moving data streams from the IoT devices directly to a cloud platform (Atzori et al., 2010; Xie et al., 2013; Tolosana-Calasanz et al., 2014; Bendoukha et al., 2013). In contrast, very little is known about the actual behaviour of the data streams when executing automated analytical tasks at the edge nodes of a network. The rationale of edge computing is that automated analytical tasks should be performed closer to IoT devices in order to reduce network latency and the risk of infringing privacy rights.

Process mining (PM) techniques provide a unique prospect to compare the expected behaviour against the actual behaviour of IoT data streams across automated tasks (van der Aalst, 2011). Automated analytical tasks are essentially modelled as a process discovery which is based on modelling the expected tasks and observing their actual behaviour that emerges from executing them. Diagnosing discrepancies such as path deviations can help us unfolding path deviations caused by the data streams that have followed different paths to those expected to occur (i.e., conformance checking). Bottlenecks can also impact the speed in which the data streams flow, causing that the tasks involved in the bottleneck to experience higher processing time than expected, and as a result, triggering a delay in their execution.

Traditional PM techniques have been previously used to model task behaviour, but they failed to consider the association between a data flow and the execution of a task that depends on this dataflow (Adam et al., 1998). It is of paramount importance to model the expected behaviour of IoT data streams during the execution of automated analytical tasks in edge computing. Logical specifications are needed to reflect what actually happens to the IoT data streams arriving at a large number of edge nodes. PM techniques are promising to model the behaviour of IoT data streams by extracting knowledge from event logs available for a real-world scenario (van der Aalst, 2011; van der Aalst, 2014). Based on an event log, a process model can be constructed for capturing the behaviour that emerges from this log.

Previous research work has proposed a variety of PM models such as marked graphs (MG) (Reisig, 2013), signal transition graphs (STG) (Workcraft.org, https://workcraft.org/tutorial/modelling/stg/start), Petri-nets (PN) (Adam et al., 1998), temporal constrain Petri Nets (TCPN) (Gonzalez del Foyo and Silva, 2008), predicated Petri Nets (PPN) (Adam et al., 1998), and matrix vector transition net (MVTN) (Spiteri Staines, 2016). In this research paper, we propose to develop a PN model for analysing IoT data streams during the process of performing automated tasks at an edge node. A PN model was selected mainly because an MG model is a restricted graph approach that does not allow any modelling choices or data stream variations at all. It is only recommended in cases whose transitions are very simple. Although an STG model can be considered as similar to a PN model, it has a major limitation since STG models usually omit transitions boxes and divides tasks into inputs, outputs, and internal. Other examples such as the TCPN, PPN and MVTN models also follow the PN principles but then again, they have modelling constraints on temporal behaviour and abstract data types that hamper their use to capture the relationships among automated tasks running at an edge node.

APN model is a bipartite directed graph which provides a generic approach that should be sufficient to represent any process discovery of billions of IoT data flows and perform conformance checking of automated tasks running at different edge nodes. In particular, a PN model with data (DPN) offers a logical specification that can provide a basis for accurate conformance checking that can enable us to foster higher confidence levels in the correctness of the execution of the tasks. This is vital for streaming analytics because it will allow us to diagnose if there is a change between the observed behaviour recorded in an event log and the planned behaviour of the algorithm developed to perform an automated task. Moreover, a DPN model has the advantage of monitoring the flow dependency of the IoT data streams between tasks and their temporal relationships since the event logs can be generated on the fly.

Our research challenge consists of processing a vast volume of data streams continuously coming at high velocity from a large number of IoT devices, but also making sure that the behaviour of these data streams conform to the constraints of automated tasks. Towards this challenge, our scientific contributions are as follows:

- Our research work is a first step towards understanding the actual behaviour of IoT data streams and its impact on the performance of automated tasks running at edge nodes. PM techniques have a potential to help us to identify inconsistent discrepancies in IoT data stream such as path deviations, bottlenecks, and parallelism.
- Our proposed DPN model outlines the importance of control-flow alignment for IoT data streams. Previous research work on DPN models has been focused on the task itself to which an event log refers to, overlooking the actual data flow taking place to execute this task, in particular with automated tasks in edge computing.
- A smart transit scenario is used to validate a new application for DPN models. Smart transit in the cities is expected to generate billions of IoT data streams. Our DPN model provides a unique approach to verify the execution of automated analytical tasks. To the best of our knowledge, DPN models have not been applied in the context of IoT and a smart transit scenario before.

This paper is organised as follows: Section 2 introduces the preliminary concepts related to PN models, event logs, alignments, and streaming analytics. The related work is discussed in Section 3, and Section 4 introduces our real-world scenario where IoT data streams are generated by a smart transit system. Section 5 explains our proposedDPN model and Section 6 describes its process mining discovery. Section 7 contains a description of the implementation steps and obtained results. Finally, Section 8 concludes this research, and shares our future research work.

2 Preliminaries

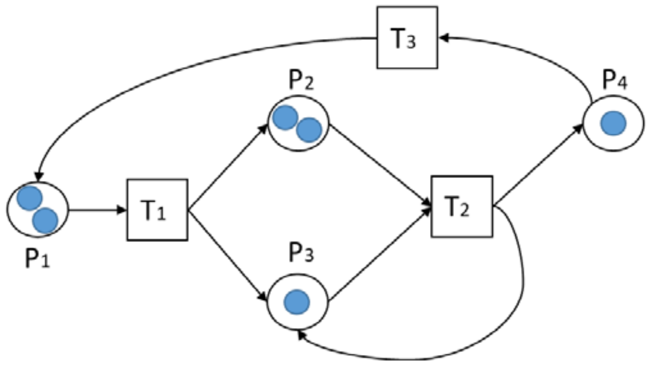
This section introduces basic concepts related to Petri Nets previously defined in van der Aalst (2011), Leoni and van der Aalst (2013), Adriansyah et al., 2011; Prabhakara et al., 2010) that are relevant for streaming analytics (Xie et al., 2013; van der Aalst, 2011; Adriansyah et al., 2011; Al Ridhawi et al., 2017).

Definition 1: Petri Net

A PN is a tuple (P, T, F, m_0) where P is a finite set of places; T is a finite set of transitions; F is a flow relation where $F \subseteq (P \times T) \cup (T \times P)$, and m_0 is an initial marking representing the initial distribution of tokens (Leoni and van der Aalst, 2013). One place may contain tokens that flow to other places by executing a transition (i.e., an action). Figure 1 provides a representation of a PN and the flow relations between places.

We state that a transition T_1 is enabled in m_1 at place P_1 and that its firing produces the successor marking m_2 at place P_2 and m_3 at place P_3. Following the flow relations, the transition T_2 is enabled in m_2 and m_3, and its firing produces the successor m_4 and m_3. Meanwhile transitions consume and produce tokens, places represent the resources that are needed to be available before a transition is triggered as well as states that need to be met before a transition can be carried out.

Figure 1 An example of a Petri Net diagram (created in 1939) (see online version for colours)



Definition 2: Petri Net with data

A data Petri Net (DPN) is a PN in which tokens carry data. It can handle data variables, allocate resources, define time constraints and perform read/write actions.

A DPN is a tuple DPN = (P, T, F, V, U, R, W, G) where P is a finite set of places; T is a finite set of transitions; F is a flow relation where $F \subseteq (P \times T) \cup (T \times P)$, V is a finite set of variables; U is a function that defines the values admissible for each variable $v \in V$, R is a function $R \in T \to 2\ V$ that labels each transition with the set of variables that it must read; W is a function $W \in T \to 2\ V$ that labels each transition with the set of variables that it must write; and a data dependent guard $G \in T \to GV$ that verifies whether all the input places are marked before a transition is triggered. A guard can be any Boolean expression over V using logical operators such as conjunction ($\wedge$), disjunction ($\vee$), and negation ($\neg$).

Definition 3: IoT data streams

IoT data streams are unbounded set of tuples that are transported as discrete data packages of varying sizes at periodic intervals of time (i.e., event time windows). Each tuple may contain several attributes $T_1 = (S_1, x_1, y_1, t_1)$ where S_1 is a fixed number of measurements generated by an IoT device, and (x_1, y_1) is the geographical location of an IoT device at the time t_1 which is when the measurements were generated.

Definition 5: streaming analytics

Streaming analytics is a network of automated tasks t_1; t_2, …, t_n, that are related to each other based on their order of execution. Each task consists of a set of actions that belong to an algorithm used to execute such a task. Tuples flow from one task to another, and different tasks require a variety of computational resources that will determine their processing time. In this paper, we introduce streaming analytics as a DPN model that is a generic-model used for modelling, ordering, and analysing the behaviour of IoT data streams during the execution of automated tasks at edge nodes.

In our DPN model, the bi-partite graph consists of:

- Place nodes that are the required state of a tuple and the computational resources which are needed to trigger a transition.
- Transition nodes represent an action that is needed in order to execute an automated task. Multiple transitions can refer to the same automated task or different automated tasks.
- A flow relation is the continuously transport of tuples (i.e., tokens) from one transition to another at periodic intervals of time. Event time windows are created since tuples may arrive out-of-order of their timestamp.
- Tokens are the tuples (i.e., IoT data streams).
- Variables are the attributes of a tuple that was generated by an IoT device.
- Initial Marking is continuously being updated since the number of tuples inside an event window may vary.
- Final Marking is unknown. It will usually be defined when an IoT device stops sending data to an edge node.

Definition 6: process discovery

It is a learning process that relates a modelled behaviour of a Petri Net and an observed behaviour recorded on an event log L. The events in the event log L must be related to transitions in the model and can be represented by a pair (a, φ) consisting of an action to execute an automated task and a value assignment φ associated with cases (i.e., process instances).

Definition 7: data-aware conformance checking

It is the process of diagnosing and quantifying discrepancies between modelled behaviour and observed behaviour. It requires an alignment of an event log L and the DPN model in such a way each single trace $\sigma \in L$ and the DPN model. This means that conformance checking seeks to match the cases inside an event log with the planned behaviour of the automated tasks of the DPN model. A DPN is aligned if every trace in the event log can be mirrored somehow by the model.

3 Related work

Previous research work has already shown the important role of applying PM techniques for discovering behaviour patterns with the aim of improving the way to process data (Mastroianni et al., 2017; Zhang et al., 2012; Zheng et al., 2012; Xie et al., 2013; Tolosana-Calasanz et al., 2014; Bendoukha et al., 2013; Leoni and van der Aalst, 2013; Adriansyah et al., 2011; Al Ridhawi et al., 2017; van der Aalst et al., 2013; Jagadeesh et al., 2013; Kapitanova et al., 2011; Petri et al., 2017; Caesarita et al., 2017; Pulsanong et al., 2017; Appice and Malerba, 2016). In particular, Leoni and van der Aalst (2013) were pioneering in presenting a data-aware process discovery technique for applying a DPN model using real-life event logs obtained from hospitals and mobile phone carriers. In their research, they point out the importance of using real-world event logs to discover data-flow patterns that can be applied to improve the way to analyse process behaviours. Process cubes have also been proposed for modelling a set of events as individual cells of a process cube structure (van der Aalst et al., 2013). A multidimensional PM is developed based on online analytical processing (OLAP) queries which are defined according to different dimensions of events. The WABO1 event log containing 20 dimensions that is publicly available was used to illustrate the temporal distribution of events (Lohmann et al., 2013).

Regarding the data quality contained into the event logs, Adriansyah et al. (2011) proposes a conformance checking approach that deals with identifying unobservable actions in event logs that might lead to false-negative patterns in data management systems. Jagadeesh et al. (2013) provides a summary which identifies ten categories of data quality issues in PM including event granularity, case heterogeneity, voluminous data, timestamp issues, missing data, ambiguity between events, process flexibility, noisy data, mashed process, and scoping. More research is needed to address data quality issues in event logs, in particular with IoT data which is usually noisy and incomplete, making it more challenging to generate reliable event logs.

From a data streaming perspective, Kapitanova et al. (2011) proposed the MEDAL formal specification language based on combining features from stochastic, timed, and coloured PNs, to model and analyse stream queries in terms of workload and query cost. Using simulated event logs, a snapshot of the data streams is created for each query statement, even when there are new data streams arriving in the system. The simulation results are obtained from the PN simulator Yasper (van Hee et al., 2006).

From a PM perspective, Al Ridhawi et al. (2017) use PN models to generate event logs containing actions needed for mobile edge node cooperation, compare them, and find the one that produces the minimal cost in terms of latency and path stability. Bioinformatics analytics based on PN models has also been proposed in the literature to show the trade-off between the cost of storing intermediate data and the computing costs incurred in regenerating this data using cloud resources (Xie et al., 2013). They are also used as modelling methods for understanding the dynamic resource allocation in the cloud with the target of assuring quality of service and throughput (Tolosana-Calasanz et al., 2014).

Control-flow analysis in the cloud has been explored to coordinate actions of a group of distributed resources within a cloud infrastructure. Bendoukha et al. (2013) explores a PN model for modelling resource sharing for edge computing applications. Petri et al. (2017) describe potential PN models for micro data centres to be deployed at edge nodes. The tested scenarios are healthcare, vehicle-to-vehicle (V2V), and vehicle-to-interface (V2I) communications.

Few attempts could be found on applying DPN models on streaming analytics. And there is even fewer attempts to integrate DPN models and IoT in general. Mastroianni et al. (2017) apply a DPN model to a simulated event log generated from a simulated set of IoT devices being carried out by pedestrians or deployed in vehicles moving on a smart street. The PM is designed to improve our understanding on how to tackle scalability and network issues considering to important features of the IoT devices, such as mobility and geo-distribution aspects. Their resulting patterns are showing the trade-off between scalability and latency to improve Quality of Services. No attention was given to the impact of data flows on the performance of automated analytical tasks.

Regarding to specialised tool kits that support PM approaches, Caesarita et al. (2017) utilise the alpha and heuristic miner algorithms to identify bottlenecks and frauds of business processes. The authors take full advantage of ProM6 (Verbbek et al., 2012), that is as a PM tool kit, to compare the modelling performance among several types of mine algorithms. Complementary, the inductive miner algorithm (Pulsanong et al., 2017) is used to find out the most optimised path, in which the implementations are done over Disco Fluxicon Co. (van der Aalst, 2011) tool kit. These previous research works validate the potential of PM tools for analysing real-world event logs generated by healthcare and online business applications.

Finally, Appice and Malerba (2016) propose a multiple view clustering solution to reduce/clean spaghetti-like PM models. This approach aims to unveil the problems that arise when an event log is examined under several perspectives, such as the control-flow perspective (ordering of actions), the organisational perspective (organisation of resources), the trace perspective (frequency of actions), and the performance perspective (time processing).

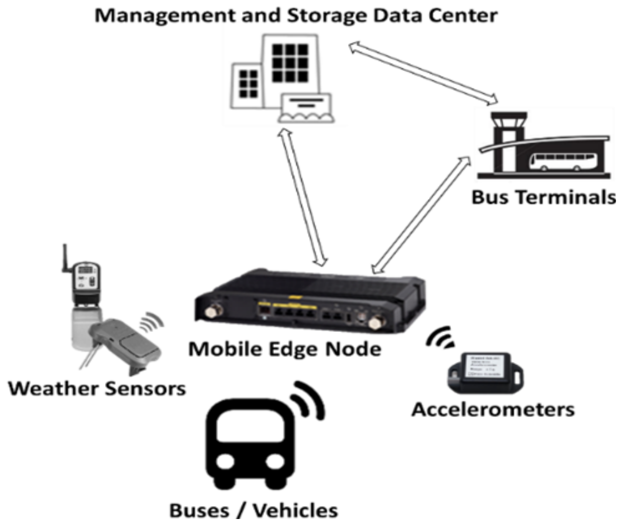
4 The smart transit scenario

In our smart transit scenario, every vehicle of a transit network is equipped with an IoT device which sends one tuple every five seconds to an edge node installed inside a bus (Figure 2). Event time windows are used in order to create more accurate event logs, even if the tuples arrive out-of-order of their timestamp.

In edge computing, automated tasks are performed near IoT devices. The main reason is to take the advantage of edge nodes as distributed resources in charge of carrying out many automated tasks. In our smart transit scenario, four automated tasks have been selected: data ingestion, data cleaning, data contextualisation, and data aggregation.

The data streams that reach an edge node are unbounded tuples having 17 attributes containing information as shown in Table 1. Once the tuples arrive at the edge node, the automated analytical tasks are triggered without human intervention. This has been achieved by developing algorithms for each analytical task (i.e., data ingestion, data cleaning, data contextualisation, and data aggregation). The edge nodes have no control over the order in which a tuple arrives within an event time window.

Figure 2 Overview of our smart transit scenario



One expected stream behaviour in our smart transit scenario is when a single input tuple is processed by one task, and after the execution of this task, an updated output tuple is generated. The tasks that present this type of stream behaviour include the data ingestion task which consists of flowing all the raw tuples from the IoT devices to an edge node. The tuples being generated by the IoT devices of the transit system might use different event time windows, having time granularities of each minute, hour, week, or month depending on the mobility context. All the tuples that arrive at an edge node are kept in memory.

Another task that presents this linear behaviour is the data cleaning task which is triggered as soon as the raw tuples arrive at an edge node, and it aims to remove errors and inconsistencies. Ensuring data quality for a high volume of tuples is a nontrivial step since IoT devices usually produce noisy data. Once the data cleaning task is finished, many tuples might have been deleted and as a result, only the cleaned tuples will be ready to flow to the data contextualisation task.

The data contextualisation task aims to perform semantic enrichment by adding new attributes to each cleaned tuple accordingly to a bus trip. Two new attributes (i.e., move and stop) are added to the original tuples to give information about if a vehicle is moving or not moving during a particular trip. The moves and stops are computed using the distance between two consecutive locations of a moving bus along a trip. We consider that if the distance between two consecutive locations is larger than 15m, the bus is moving (i.e., move), otherwise the bus is not moving (i.e., stop) (Cao and Wachowicz, 2018).

Table 1 Transit feed of the smart transit scenario

ID	*Attribute name*	*Description*
1	vlr_id	The ID of the data point in the vehicle location report table.
2	route_id_vlr	The route ID in the vehicle location report table.
3	route_name	The name of the route.
4	route_id_rta	The route ID in the route in the route transit authority table.
5	route_nickname	The abbreviation of the route.
6	trip_id_br	The trip Id in the route table.
7	transit_authority_service_time_id	Transit authority service time ID.
8	trip_id_tta	Transit authority trip ID.
9	trip_start	Start time of the trip.
10	trip_finish	Finish time of the trip.
11	vehicle_id_vab	Vehicle ID.
12	vehicle_id_vlr	Vehicle ID in the vehicle location report table.
13	vehicle_id_vlr_ta	Descriptive name of the bus.
14	bdescription	Bus description.
15	lat	Latitude.
16	lng	Longitude.
17	timestamp	Timestamp of the data point.

Table 2 The transitions in our DPN model

Transition	*Actions*	*Automated task*	*Tuple state*
A	Receiving tuples	Data ingestion	Raw
B	Normalising tuples	Data cleaning	Cleaned
C	Eliminating tuples		
D	Grouping tuples	Data contextualisation	Contextualised
E	Sorting tuples		
F	Computing moves/stops		
G	Computing total number of stops per trip	Data aggregation	Aggregated
H	Computing total number of moves per trip		
I	Computing actual duration of trips		

Another stream behaviour found in our smart transit scenario is when a set of ordered tuples are processed by a task at once, and as a result, a single new tuple is generated. One example includes the data aggregation task. This task can offer information that may have a wide impact on the observed behaviour by summarising particular patterns that can generate global mobility patterns of the entire transit network. For instance, the data aggregation task consists of summarising the information contained in all tuples that belong to the same trip. The move/stop information can be used to identify the behaviour of the trips (i.e., group of tuples), instead of the behaviour of a single tuple. It can also provide new insights about which bus stops are most used and which others are not. Finally, the data aggregation task can also provide information about the total trip duration considering the real stream behaviour of the transit system as a whole system. Once the automated taskswere performed at the edge nodes, the historical outputs are sent to a cloud platform. This prevents semantically incorrect results in case of backpressure or delays due to failure recovery.

5 The proposed DPN model

Our DPN model is a bipartite graph consisting of two sets of nodes: places and transitions. The flow relation between the nodes are defined from a place to a transition or from a transition to a place. Table 2 summarises the transitions that have been used to model the expected data stream behaviour.

Figure 3 Overview of our DPN model for the smart transit scenario (see online version for colours)

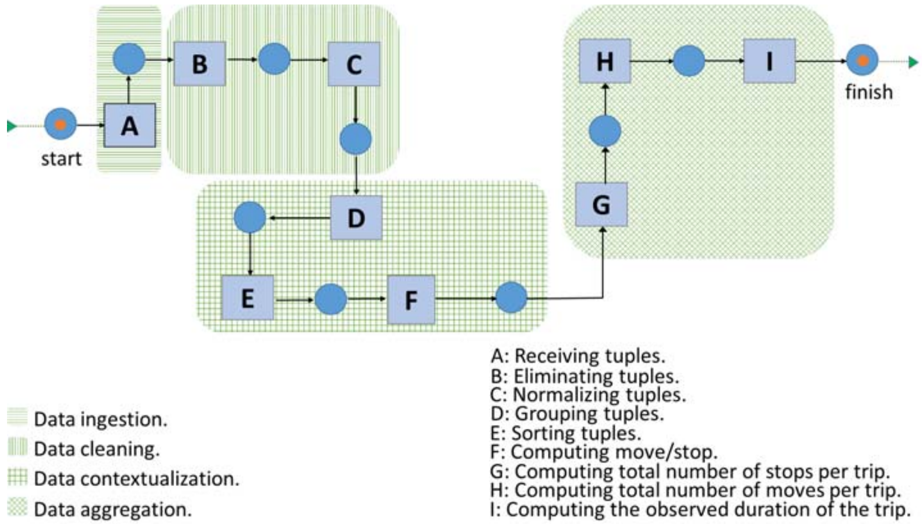


Figure 3 illustrates our DPN model using a workflow net for our smart transit scenario. The transitions (squared nodes) consist of a sequence of actions that should occur when an automated task is being executed by running an algorithm. Transitions also represent the changes that occur when tuples flow from one automated task to another. In our model, tuples being generated by IoT devices have four different states ranging from raw state, and cleaned state to a contextualised state and aggregated state. Places (circle nodes) are used to represent the resources available at the edge nodes. In our model, the resources are the algorithms developed for the execution of the tasks. They consist of Python scripts, which are automatically running in an edge node. The sequence of places and transitions is based on the occurrence order among the events in which the actions are triggered any time a tuple arrives in order to execute an automated task. Once an edge node has received a tuple, the transition *A: receiving tuples* is triggered, and it will run continuously until all tuples of an event time window are processed. The same will happen to all other transitions *B: normalising tuples; C: eliminating tuples; D: grouping tuples, E: sorting tuples, and F: computing moves/stops.* These transitions are again triggered when a new set of tuples arrived at a place belonging to a consecutive event time window. In contrast, the transitions *G: computing total number of stops per trip, H: computing total number of moves per trip, and finally I: computing actual duration of trips* require that their place nodes make sure that all tuples pertaining to a bus trip have been contextualised (i.e., contextualised state) before any transition is triggered.

The proposed DPN model is based on the following modelling constraints:

- A tuple is a high-quality tuple if it assures two requirements. The first requirement is accomplished if a tuple has successfully followed the path (i.e., trace) that was envisaged for every transition. The second requirement is fulfilled if the attribute values within the tuples are complete and are not missing. Otherwise a tuple is considered as an anomaly.
- A trip is a good trip if 80% the tuples belonging to the same trip successfully follow the stream data flow of all transitions. Otherwise, a trip is considered as an anomaly.
- A route is a good route if 80% of the tuples belonging to the same route has successfully followed the stream data flow of all transitions. Otherwise, a route is considered an anomaly.

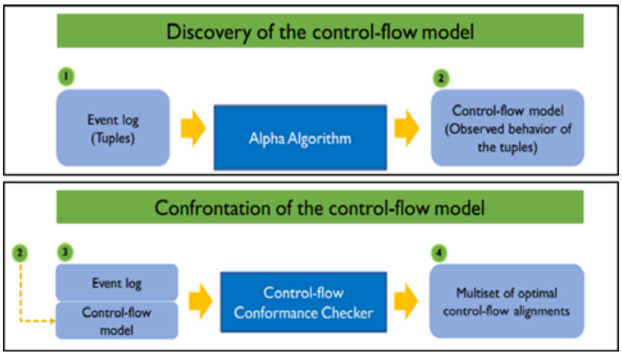
In our Smart Transit Scenario, a good trip and a good route also indicate that their tuples provide reliable and complete information about such a trip and route. This plays an important role in the quality assurance of the outputs of the automated tasks.

Our DPN model is a generic model that can allow us to capture the stream behaviour of the traces in the event log, from the time a tuple arrives at a transition running at an edge node until the tuples are ready to be sent to the cloud. It can also allow us to capture billions of traces that might not been expected. Finally, process mining can help us to match the transitions and places, the data contained in them, and the patterns built by modelling the sequence of transitions within an automated task. Therefore, the PM approach is discussed in detail in next section.

6 How IoT data streams are actually flowing between automated tasks?

Our main goal is to extract patterns about path deviations, detect the presence of bottlenecks, point out process constrains, and find out whether the flow has had an impact on the performance of the tasks. Each raw tuple is modelled as a case in the event log. The updated and new tuples are also modelled as a case that took place according to the different tuple states as previously described in Table 2. Each case within the event log refers to a transition triggered to perform an automated task, and it is related to a specific trip that occurs at a particular timestamp. The cases belonging to the same trip are seen as one trace. And finally, each trace describes the life-cycle of a specific trip in terms of the executed automated tasks.

Figure 4 Process mining overview (see online version for colours)



In our DPN model, the execution of automated tasks is known as a control-flow. The PM technique being used here was first proposed by Leoni and van der Aalst (2013), and it consists of four steps. Figure 4 illustrates the four steps that can be described as one of the following:

Step 1 Create an event log.

Step 2 Compute the control-flow of the discovered DPN.

Step 3 Combine the control-flow model from the previous step and the event log.

Step 4 Verify whether the DPN is an accurate representation of the control-flow, according to the real behaviours observed in the event log.

The event log creation (step 1) refers to creating a CSV file formatted as a common table. Every row in the table represents a case within the event log, whereas the columns are the variables that describe these cases; such as tuple identification number, action name, timestamp, resource, and others. Among the all the variables, only three of them are mandatory for the process mining. These are the tuple identification number, the transition name, and the timestamp columns. The tuple identification number is a unique number that every tuple receives (i.e., case id). It is used to make reference to a specific tuple while it is flowing through the different automated tasks. The transition name has information about the actual action taken to execute a particular transition (i.e., receiving tuples, eliminating tuples, normalising tuples, etc.). Finally, the timestamp sets the date and time in which the transitions take place. The computation of the control-flow (step 2) consists of making use of available algorithms, such as alpha algorithm, to draw a visual version of the observed stream behaviour within the event log.

Step 3 combines the control-flow model obtained in Step 2 with the original event log. Four quality dimensions are used for comparing the model and the event log. These dimensions are fitness, simplicity, precision and generalisation (van der Aalst, 2011). Fitness means the level of freedom that the model allows for representing most of the behaviour seen in the event log (van der Aalst, 2011; van der Aalst, 2014). Simplicity refers to the level of complexity infer from the model built. The precision and generalisation dimensions are other important aspects of process mining because they provide information about how a model is over fitting the data (i.e., extremely general) or underfitting the data (i.e., extremely precise). Finally, step 4 applies the conformance checking approach that is available as a plug-in of ProM6 (Verbbek et al., 2012). The implementation for the smart transit scenario is explained in detail in the following section.

7 Implementation and discussion of results

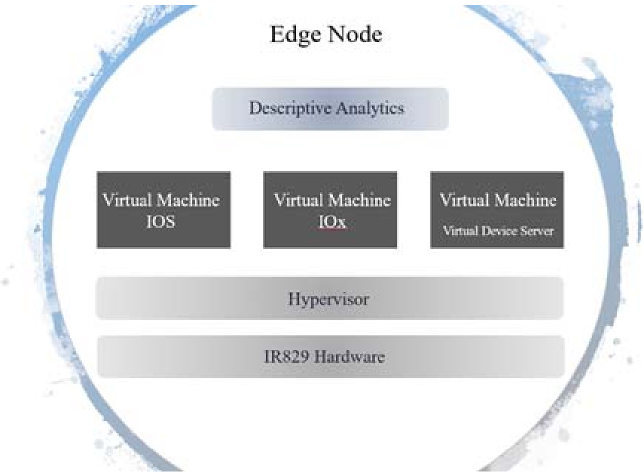
The IoT data streams were real-time transit feeds of route 51 provided by the CODIAC Transit System of the City of Moncton, New Brunswick. An edge node was deployed inside three bus vehicles of the route 51. The edge nodes were IR829GW edge nodes which belong to the Cisco 829 series and they were selected because they can support a wide variety of applications including fleet management, mass transit, and remote asset monitoring. They also offer five main advantages as described as one the following (Data Sheet: Cisco 809 Industrial Integrated Services Routers, 2017):

- compact form and size
- multimode compatibility with diverse WAN technologies
- high scalability implementation
- resistance to humidity as well as a wide temperature range (–40°C to +60°C and type-tested at +85°C for 16 hours)
- routers are shock and vibration resistant
- edge computing capabilities, where computing tools can be installed as micro services.

This edge node handles all routing, switching and networking traffic using IOx operating system running on a virtual machine that uses Linux Yocto. The IoT data streams are fetched using the Gateway Management Module (GMM) and Data Control Module (DCM) which are the main modules of the Cisco Kinetic platform. This platform is a scalable, open system, adaptable for a variety of IoT applications that is used to extract, compute, and move the IoT data streams (Figure 5).

The transit feeds were transported from the devices to the edge nodes using the available 3G network. The four automated tasks were running using a script code written in Python version 2.7.14 at the edge node due to its available light libraries and high-level general-purpose programming (Cao and Wachowicz, 2018). The automated tasks have generated five outputs as CSV files. These tasks consist of nine transitions ranging from *A: receiving tuples* to *I: computing the observed duration of the trip*. See Table 2 for an overview of the transitions.

Figure 5 Edge node architecture (see online version for colours)



For creating an event log of the observed stream behaviour that belongs to the data ingestion, data cleaning, and data contextualisation tasks (i.e., *A: receiving tuples, B: eliminating tuples, C: normalising tuples, D: grouping tuples, E: sorting tuples, and F: computing move/stops*), we have combined the content of the first three CSV output files. The remaining CSV output files were merged to build another event log of the observed stream behaviour that belongs to the aggregation task (i.e., *G: computing total number of stops per trip, H: computing total number of moves per trip, and I: computing the observed duration of the trip*). The first event log file size exceeded 1 million cases, and the second log file size was in the range of 150,000 cases.

It is important to point out that since the IoT data streams have tuples with 17 variables, only three main variables have been standardised for the process mining (Figure 6). They are Case ID, transition name, and time for the event logs. In our DPN model, the Case ID is the Tuple ID. Depending on the granularity, and/or the context level we want to analyse, the Case ID may be exchanged by the Route ID or the Trip ID attribute. The action name is represented by variables that describe the analytical task being performed. The time column is the timestamp.

We have used ProM6 (Verbbek et al., 2012) and Disco Fluxicon Co. (Rozinat, 2018) for the implementation. The ProM6 toolkit is a generic open-source framework with a high scalability. It has processing capabilities that allows us to work with event log files containing more than one million cases, assuring a processing time of milliseconds. Disco Fluxicon Co does not have restrictions of file sizes neither. It allows files with more than one million cases. However, some limitations of these tools are related to the available plug-ins (i.e., algorithms for modelling the event log files), the restrictions to configure them, and the type of output they delivered (Verbeek, 2012; Verbeek et al., 2012). For example, the file format required by the ProM6 and Disco Fluxicon Co tools is not the CSV data format. The Disco Fluxicon Co outputs, with a xes.gz file format, represent the ProM6 inputs that are necessary to start with the process mining performed by ProM6. Therefore, some additional steps have been performed to generate the event logs having the appropriate format. These steps were carried out manually, therefore we cannot assure that the event log does not contain human errors which can have a negative impact on the accuracy of observed behaviours. More research is needed to study how errors can be detected in event logs. Figure 7 illustrates the observed behaviour patterns that belong to the data ingestion, data cleaning, and data contextualisation tasks. It reveals the occurrence of a path deviation during the data flow between the data ingestion and data cleaning tasks. In fact, the expected behaviour was that all cases after being normalised would have been grouped according to their respective routes. However, there were 22,329 cases that have been directly flown from the transition *C: normalising tuples* to the ending marking point of our DPN model, and after sent to the cloud without being contextualised and aggregated. This path deviation could have happened, either by unknown conditions in the algorithm. Further an in-depth analysis is required to determine the main reasons behind this observed stream behaviour.

Figure 6 Event log composition (see online version for colours)

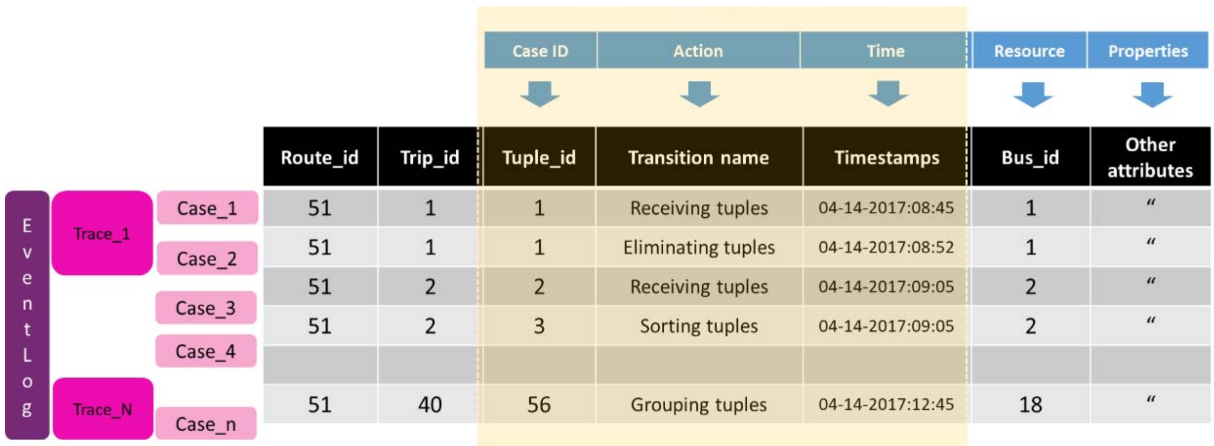
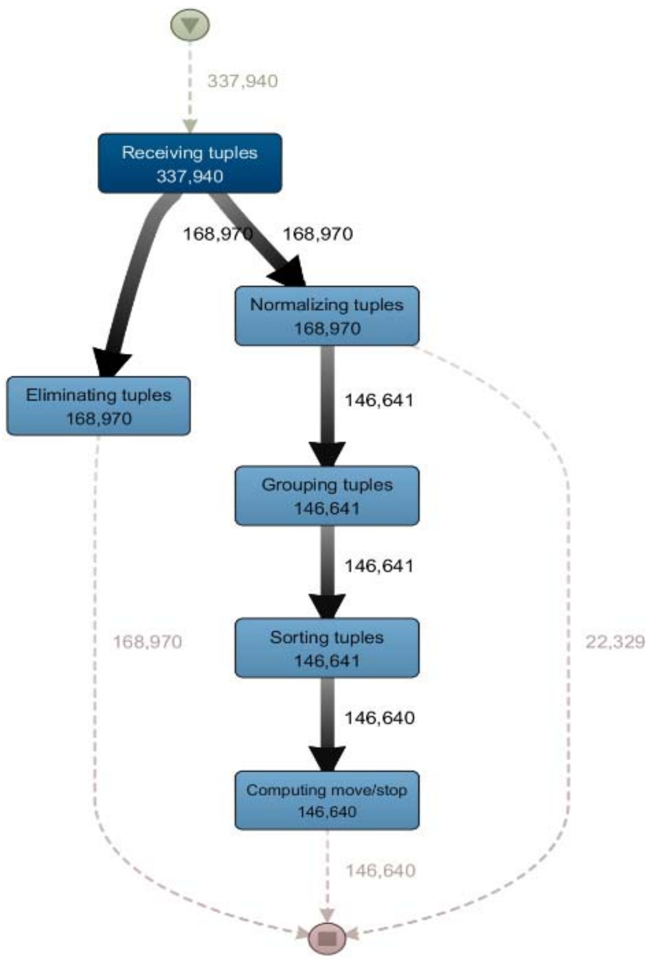


Figure 7 Process mining results (see online version for colours)



Another path deviation has occurred during the transition *B: eliminating tuples* of the cleaning task. In this case, half of the original number of cases were eliminated, which was much higher than expected. Thus, an investigation was carried out to identify the causes of such a behaviour. It was found that due to network failures, the tuples generated by the IoT devices have been duplicated and many missing values were also found. In particular, IoT data streams are usually noisy and incomplete, making them more challenging to conform to an expected stream behaviour.

Figure 8 reveals a parallelism among three transitions that have occurred during the data aggregation task. This kind of stream behaviour was not expected since these transitions were modelled as a sequence of actions. This behaviour has also exposed an important logical problem with the algorithm used for the execution of the aggregation task. For example, the total trip time was computed for only 352 trips for bus line 51, since more cases were classified as stops than moves. We were able to infer that more than 57.8% of cases were classified as stops during the aggregation task, whereas 42.2% were classified as moves. The output of this automated tasks has generated 145,288 cases which were later sent to the cloud. There is a high probability that a significant number of trips were not computed during the aggregation tasks.

Figure 9 illustrates four snapshots of the stream data flows that were observed during the execution of the transitions. In total, eight bottlenecks have emerged during the execution of five transitions. The first bottleneck took place due to a delay of the input cases for the execution of the transition *C: normalising tuples* [Figure 9(a)]. This indicates that the time spent to execute this transition was significant longer than the time used for the event windows. The second bottleneck was a consequence of the first bottleneck since the following transition *D: grouping tuples* experienced a delay in its input cases as well as in its output cases [Figure 9(b)]. This cascade effect was further observed in the next transition *E: sorting tuples* as shown in Figure 9(c). And even during the execution of the parallel transitions of our aggregation task, the bottleneck has persisted to occur as shown in Figure 9(d). These process mining results demonstrate the important role of DPN models in identifying bottlenecks issues with algorithms that are being developed for IoT data streams. Ideally, an automated analytical workflow should support flexible data rates to make sure any relevant tuple has arrived at an automated task at the right time.

Figure 8 Process mining results for the three transitions of the aggregation task (see online version for colours)

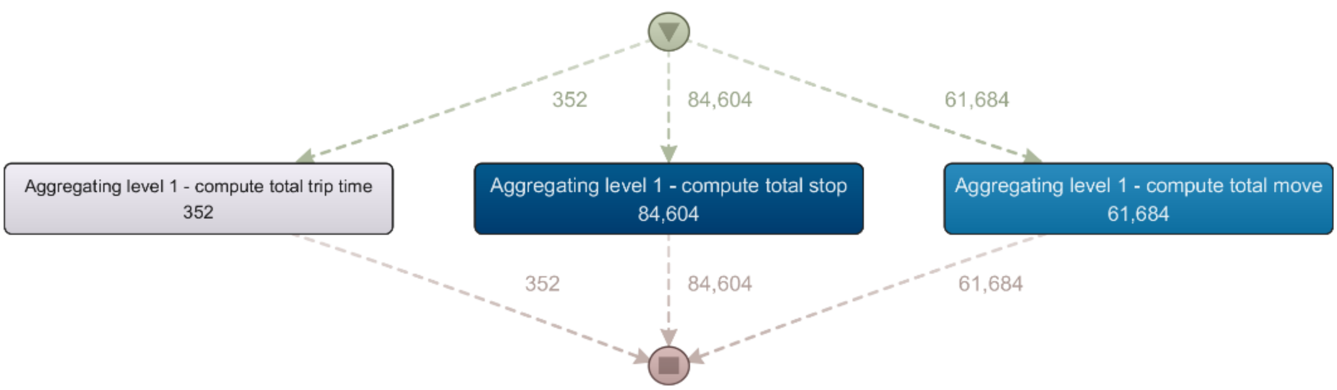


Figure 9 Bottlenecks observed during the execution of the transitions (see online version for colours)

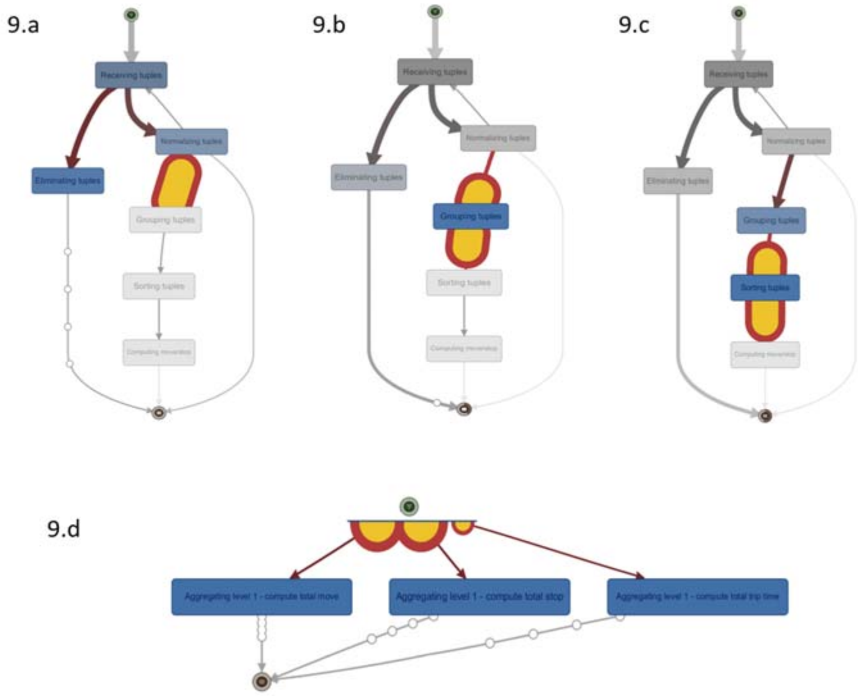


Figure 10 Misalignments observed during the execution of the transitions (see online version for colours)

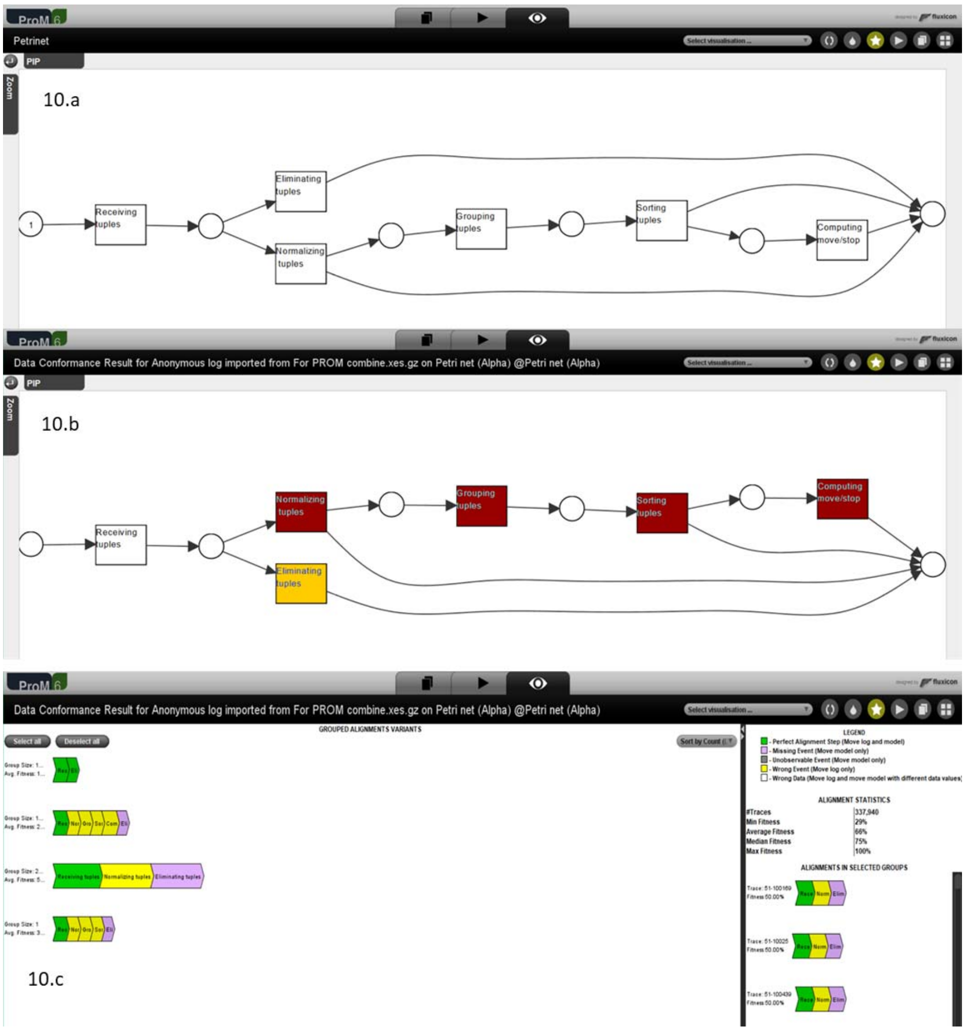
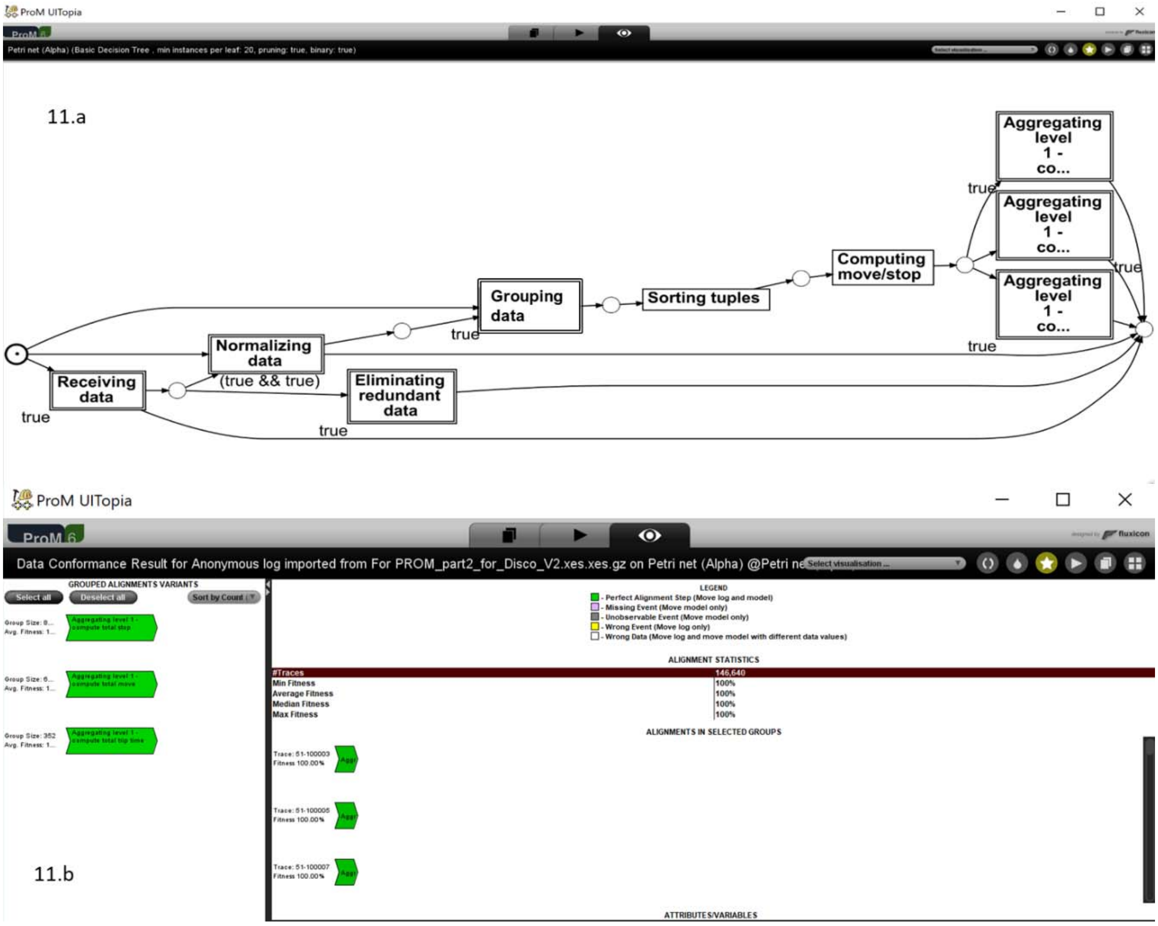


Figure 11 Conformance-checking results observed during the execution of the transitions (see online version for colours)



The total number of traces registered was 337,940 [Figure 10(a)]. The misalignments observed in our DPN model were exceptionally revealing since the transitions of receiving and eliminating tuples were aligned, meanwhile the remaining transitions were misaligned [red transitions in Figure 10(b)]. In general, misalignments are expected to occur while complex automated analytical tasks are being executed. The minimal alignment percentage was 29%, whereas the average and median alignment percentages were 66% and 75%, respectively as shown in Figure 10(c). We believe that these misalignments might have occurred due to the cascade bottleneck effect that has taken place.

More research is needed to identify the causes for misalignments in IoT data streams. Additionally, in order to understand the behavioural patterns of cases and their deviations, we suggest that the observing deviation on a specific transition of an automated task could be only a true deviation if the percentage of alignment for that specific action is above 75% (i.e., the median alignment percentage). Therefore, our assumption is that if the percentage of alignment is below of such a limit, the behavioural deviation of cases will be the result of misalignments in the transitions, rather than to be a true deviation in the patterns of the cases. Thus, an important part of a PM for streaming analytics is to align as much as possible all transitions of automated tasks.

Finally, Figure 11 shows that all transitions in the process discovery that were misaligned have successfully passed a conformance checking. In this case, the function of the conformance checking was used to determine that the alignments were successfully performed since the minimal, average, median, and maximal fitness percentages have reached 100%.

8 Conclusions and future research

A real-world smart transit scenario was used to explore process mining for uncovering the actual behaviour of IoT data streams when executing streaming analytics in edge computing. The results from this learning process have uncovered the limitations of computational resources and algorithms designed for the execution of automated analytical tasks. We have observed two main path deviations that have indicated that 6.6% of the total number of cases followed a different path from the one expected. On the other hand, the constraint of one single transition has actually eliminated 50% of the raw tuples generated from the IoT devices. This is an unacceptable data flow behaviour in streaming analytics, mainly because it hinders the extraction of valuable information from the internet of things (IoT). This behaviour has emerged due to poor connectivity conditions between IoT devices and edge nodes in our smart transit scenario.

Several bottlenecks patterns were further observed during the process mining. They have occurred during the transitions nodes *C: normalising tuples, D: grouping tuples* and *E: sorting tuples*. The algorithms were modified in order to avoid these bottlenecks in the future. We also seek to create new event logs that will include other transitions in a smart transit scenario. Specially, it would be important to have a stringer control of the tasks in which the bottlenecks have occurred. Finally, we would like also to explore the potential use of DPNs in other real-world scenarios.

Regarding the scalability of the tools used, there is a trade-off between the size of event logs and the new insights emerging from them. Even though there are no size restrictions for an event log, a time-consuming processing step is required in order to generate an event log from IoT data streams. The reformatting of the original tuples into the ProM format is not a straightforward step. Large volume of event logs will play a role in discovering more patterns and unknown behaviours in edge computing.

References

Adam, N.R., Atluri, V. and Huang, W.K. (1998) 'Modeling and analysis of workflows using Petri Net', *Journal of Intelligent Information Systems*, Vol. 10, No. 2, pp.131–158.

Adriansyah, A., vaDongen, B.F. and van der Aalst, W.M.P. (2011) 'Conformance checking using cost-based fitness analysis', *Proceedings of IEEE International Conference on Enterprise Distributed Object Computing*, Helsinki, Finland.

Al Ridhawi, I., Kotb, Y. and Al Ridhawi, Y. (2017) 'Workflow-net based service composition using mobile edge nodes', *Journal of IEEE Open Access, volume 5, Special Section on Mobile Edge Computing*, pp.23719–23735, DOI: 10.1109/ACCESS.2017.2766068.

Appice, A. and Malerba, D. (2016) 'A co-training strategy for multiple view clustering in process mining', *Journal of IEEE Transactions on Services Computing*, Vol. 9, No. 6, pp.832–845, DOI: 10.1109/TSC.2015.2430327.

Atzori, L., Iera, A. and Morabito, G. (2010) 'The internet of things: a survey', *Computer Networks*, Vol. 54, No. 15, pp.2787–2805.

Badalotti, F., Espindola, J.C., Schwanke, D., Petry, G., Nieto, C., Bove, M. and Barbosa, A. (2018) 'RFID technology as a life cycle management tool in the liquefied petroleum gas industry', *Journal & Magazines of IEEE Latin America Transactions*, Vol. 16, No. 2, pp.391–397, DOI: 10.1109/TLA.2018.8327391.

Bendoukha, S., Moldt, D. and Wagner, T. (2013) 'Enabling cooperation in an inter-cloud environment: an agent-based approach', *Proceedings of IEEE 24th International Workshop on Database and Expert System Applications*, pp.217–221, Los Alamitos, CA, DOI: 10.1109/DEXA.2013.27.

Caesarita, Y., Sarno, R. and Sungkono, K.R. (2017) 'Identifying bottlenecks and fraud of business process using alpha ++ and heuristic miner algorithms (case study: CV. WicaksanaArtha)', *Proceedings of IEEE 11th International Conference on Information & Communication Technology and System (ICTS)*, pp.143–148, Surabaya, DOI: 10.1109/ICTS.2017.8265660.

Cao, H., and Wachowicz, M. The design of an IoT-GIS platform for performing automated analytical tasks. Computers, Environment and Urban Systems, 74, 23-40, 2019. DOI: 10.1016/j.compenvurbsys.2018.11.004

Data Sheet: Cisco 809 Industrial Integrated Services Routers (2017) C78-734980-03 05/17, Cisco USA, pp.1–17.

Fernández-Caramés, T.M. and Fraga-Lamas, P. (2018) 'A review on human-centered IoT-connected smart labels for the Industry 4.0', *Journal of IEEE Open Access (Early Access)*, pp.1–19, DOI: 10.1109/ACCESS.2018.2833501.

Garrido-Hidalgo, C., Hortelano, D., Roda-Sanchez, L., Olivares, T., Ruiz, M.C. and Lopez, V. (2017) 'IoT heterogeneous mesh network for human-in-the-loop challenges towards a social and sustainable Industry 4.0', *Journal of IEEE Open Access Class Files (Early Access)*, Vol. 14, No. 8, pp.1–20, DOI: 10.1109/ACCESS.2018.2836677.

Gcaba, E. and Dlodlo, N. (2016) 'The internet of things for South African tourism', *Proceedings of IEEE International Conference on the IST-Africa Week*, Durban, South Africa, DOI: 10.1109/ISTAFRICA.2016.7530573.

Gonzalez del Foyo, P.M. and Silva, J.R. (2008) 'Using time petri nets for modeling and verification of timed constrained workflow systems', *Proceedings of ABCM Symposium Series in Mechanics*, Vol. 3, pp.471–478, ABCM, Sao Paulo, Brazil.

Hernandez, L., Cao, H. and Wachowicz, M. (2017) 'Implementing an edge-fog-cloud architecture for stream data management', *Proceedings of the 2017 IEEE International Conference on Fog World Congress*, Santa Clara, California, USA.

Jagadeesh, R.P., Bose, C., Mans, R.S. and van der Aalst, W.M.P. (2013) 'Wanna improve process mining results? It's high time we consider data quality issues seriously', *Proceeding of IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*, Singapore.

Jeon, Y.J., Eun An, K., Lee, S.W. and Seo, D. (2018) 'Improved durability of soil humidity sensor for agricultural IoT environments', *Proceedings of IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, USA, DOI: 10.1109/ICCE.2018.8326223.

Kapitanova, K., Son, S.H., Woochul, K. and Won-Tae, K. (2011) 'Modeling and analyzing real-time data streams', *Proceeding of 14th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC)*, Newport Beach, CA, USA, pp.91–98, DOI: 10.1109/ISORC.2011.21.

Lee, Y.J. and Kim, E.K. (2015) 'Smart device based power generation facility management system in smart grid', *Proceedings of 17th IEEE International Conference on Advanced Communication Technology (ICACT)*, Seoul, South Korea, DOI: 10.1109/ICACT.2015.7224919.

Leoni, M. and van der Aalst, W.M.P. (2013) 'Data-aware process mining: discovering decisions in processes using alignment', *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pp.1454–1461, Portugal.

Lohmann, N., Song, M. and Wohed, P. (2013) 'Business process and management workshops', *Proceedings of the BPM International Workshop 2013*, Vol. 127 of LNBIP, Springer, pp.72–76.

Mastroianni, C., Cesario, E. and Giordano, A. (2017) 'Balancing speedup and accuracy in smart city parallel applications', *Proceedings of Euro-Par 2016: Parallel Processing Workshops*, Springer International Publishing AG, Lecture Notes in Computer Science (LNCS) 10104, pp.224–235, DOI: 10.1007/978-3-319-58943-5_18.

McNeil, P. (2018) 'Secure internet of things deployment in the cement industry: guidance for plant managers', *Proceedings of IEEE Industry Applications Magazine*, Vol. 24, No. 1, pp.14–23, DOI: 10.1109/MIAS.2017.2739833.

Petri, I., Rana, O., Bignell, J., Nepal, S. and Auluck, N. (2017) 'Incentivizing resource sharing in edge computing applications', *Proceedings of International Conference on the Economics of Grids, Clouds, Systems, and Services*, Springer International Publishing AG, Lecture Notes in Computer Science (LNCS) 10537, pp.204–215.

Prabhakara, R., Bose, J.C. and van der Aalst, W.M.P. (2010) 'Trace alignment in process mining: opportunities for process diagnostics', *Proceedings of 8th International Conference, and Business Process Management Publications*, Hoboken NJ, USA, DOI: 10.1007/978-3-642-15618-2_17.

Pulsanong, W., Porouhan, P. and Tumswadi, S. (2017) 'Using inductive miner to find the most optimized path of workflow process', *Proceedings of IEEE fifteenth International Conference on ICT and Knowledge Engineering*. pp.1–5, Bangkok, DOI: 10.1109/ICTKE.2017.8259633.

Ramesh, M.V., Nibi, K.V., Kurup, A., Mohan, R., Aiswarya, A., Arsha, A. and Sarang, P.R. (2017) 'Water quality monitoring and waste management using IoT', *Proceedings of IEEE International Conference on Global Humanitarian Technology Conference (GHTC)*, San Jose, CA, USA, DOI: 10.1109/GHTC.2017.8239311.

Reisig, W. (2013) *Marked Graphs*. *In: Understanding Petri Nets*, Springer, pp.163–167, Berlin, Heidelberg, DOI: https://doi.org/10.1007/978-3-642-33278-4_18.

Rozinat, A. (2018) *Disco User's Guide*, pp.1–10, Disco Fluxicon Co.

Ruengittinun, S., Phongsamsuan, S. and Sureeratanakorn, P. (2017) 'Applied internet of thing for smart hydroponic farming ecosystem (HFE)', *Proceedings of 10th IEEE International Conference on Ubi-media Computing and Workshops (Ubi-Media)*, Thailand, DOI: 10.1109/ UMEDIA.2017.8074148.

Sharma, Y., Mathew, M. and Yanamandram, K. (2017) 'Enhancement of the biogas system application using solar photovoltaic and IoT based automation', *Proceedings of IEEE International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, Kannur, India, DOI: 10.1109/ICICICT1.2017.8342676.

Spiteri Staines, A. (2016) 'Modelling simple network graphs using matrix vector transitions net', *International Journal of Computers and Communications*, Vol. 10, pp.11–17, ISSN: 2074-1294.

Tang, B., Chen, Z., Hefferman, G., Wei, T., He, H. and Yang, Q. (2015) 'A hierarchical distributed fog computing architecture for big data analysis in smart cities', *Proceedings of the ASE Big Data & Social Informatics*, ACM.

Tolosana-Calasanz, R., Diaz-Montes, J., Rana, O. and Parashar, M. (2014) 'Extending cometcloud to process dynamic data streams on heterogeneous infrastructures', *Proceedings of IEEE International Conference on Cloud and Autonomic Computing*, pp.196–205, London, DOI: 10.1109/ICCAC. 2014.22.

van der Aalst, W.M.P. (2013) 'Process cubes: slicing, dicing, rolling up, and drilling down event data for process mining', *Asia Pacific Conference on Business Process Management (AP-BPM 2013), Volume 159 of Lecture Notes in Business Information Processing*, pp.1–22, Springer-Verlag, Berlin.

van der Aalst, W.M.P. (2014) *Extracting Event Data from Databases to Unleash Process Mining*, BPM Center Report BPM-1410, BPMcenter.org.

van der Aalst, W.M.P.(2011) *Process Mining – Discovery, Conformance and Enhancement of Business Processes*, Springer-Verlag, Berlin, Heidelberg.

van Hee, K., Oanea, O., Post, R., Somers, L. and van der Werf, J.M. (206) 'Yasper. A tool for workflow modeling and analysis', *Proceedings of the Sixth International Conference on Application of Concurrency to System Design (ACSD 2006)*, IEEE Computer Society, Washington, DC, pp.279–282.

Verbbek, H.M.W., Buijs, J.C.A.m., van Dongen, B.F. and van der Aalst, W.M.P. (2012) 'XES, XESame, and ProM 6', *Proceedings of Business Process Management Workshops*, Vol. 99 of LNBIP, Springer Verlag.

Verbeek, H.M.W. (2012) 'ProM6 Tutorial', *Proceedings of promtools.org*.

Verbeek, H.M.W., Buijs, J.C.A.M., van Dongen, B.F. and van der Aalst, W.M.P. (2012) 'ProM6: the process mining toolkit', *Proceedings of ceur-ws.org*, Vol. 615, No. 13, pp.1–6, Eindhoven University of Technology, The Netherlands.

Workcraft.org, *Modeling with Signal Transition Graphs: Distributed Mutual Exclusion*, Workcraft Tutorial in Modelling [online] https://workcraft.org/tutorial/modelling/ stg/start (accessed 2 October 2019).

Xie, Z., Han, L. and Baldock, R. (2013) 'Augmented petri net cost model for optimisation of large bioinformatics workflows using cloud', *Proceedings of IEEE European Modelling Symposium (EMS)*, Manchester, pp.201–205, DOI: 10.1109/EMS.2013.35.

Zhang, X., Zheng, H. and Liu, Y. (2012) 'A petri net based context-aware workflow system for smart home', *Proceedings of IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, pp.2336–2342, Shanghai, DOI: 10.1109/ IPDPSW.2012.287.

Zheng, H., Wang, Y. and Zhong, W. (2012) 'Analysis of the digital home wireless meter reading interface execution process', *Proceedings of the IEEE Second International Conference on Cloud and Green Computing*, pp.614–620, Xiangtan, DOI:10.1109/CGC.2012.107.