IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 17, 2019, accepted May 20, 2019, date of publication May 28, 2019, date of current version June 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919514

Analytics Everywhere: Generating Insights From the Internet of Things

HUNG CAO[1], MONICA WACHOWICZ[1], CHIARA RENSO[2], AND EMANUELE CARLINI[2]
[1]People in Motion Lab, University of New Brunswick, Fredericton, NB E3B 5A3, Canada
[2]HPC Lab, ISTI-CNR, 56127 Pisa, Italy

Corresponding author: Hung Cao (hcao3@unb.ca)

This work was supported in part by the NSERC/Cisco Industrial Research Chair under Grant IRCPJ 488403-14.

ABSTRACT The Internet of Things is expected to generate an unprecedented number of unbounded data streams that will produce a paradigm shift when it comes to data analytics. We are moving away from performing analytics in a public or private cloud to performing analytics locally at the fog and edge resources. In this paper, we propose a network of tasks utilizing edge, fog, and cloud computing that are designed to support an Analytics Everywhere framework. The aim is to integrate a variety of computational resources and analytical capabilities according to a data life-cycle. We demonstrate the proposed framework using an application in smart transit.

INDEX TERMS Descriptive analytics, diagnostic analytics, predictive analytics, edge computing, fog computing, cloud computing, Internet of Things.

I. INTRODUCTION

Across the Internet of Things (IoT), transferring data from sensors to remote data centers is currently not efficient from a performance perspective due to the limitation on bandwidth and the high latency. In fact, the technological gap between the computational resources in the spectrum between an IoT sensor and the cloud is closing rapidly, especially with the advent of edge and fog devices that can support federated multi-tasking computation [1], [2] and virtualization [3]. In addition, an important requirement of IoT applications is related to privacy and confidentiality [4]. Keeping sensitive data closer to their sources may potentially reduce the risk of infringing privacy rights and breaking confidentiality.

Two phases can be distinguished in the evolution of IoT. The first phase has focused on the proliferation of sensors, protocols, and architectures where the main research challenges were related to network connectivity, IoT platforms, and sensor configurations. A second phase is gradually taking place where the core research challenges are shifting from physical infrastructures to analytical capabilities that are being developed according to the requirements of IoT applications [5].

In this paper we introduce the concept of "Analytics Everywhere" as a conceptual framework that facilitates building computational resources that are needed to support data analytics for IoT applications. We advocate that supporting the new generation of IoT applications is more than just moving computation from the cloud to the edge/fog nodes in a straightforward way. Instead, it requires an "Analytics Everywhere" framework in which computational resources are designed and work as a whole toward the completion of a network of analytical tasks. This embeds the concept of data streams moving around distributed computational resources (i.e. cloud, fog, and edge nodes) that provide storage and processing power for the execution of a network of tasks in such a way that a graph, sparse, and low-rank structure between the tasks is known *a priori*.

The research challenge is three-fold. First, there is a need to rethink how previous analytical algorithms have been independently developed. They must now be integrated in a network structure, in a way that makes explicit the dependency between the same tasks belonging to different algorithms as well as different tasks belonging to the same algorithms. This network structure will require a mathematical formulation such as Directed Acyclic Graphs (DAG), Petri-Nets, and WF-nets. Research work has been done in the past years on the mapping of DAG nodes onto computational resources, as for example in [6], [7]. Second, a mapping between analytical capabilities and computational resources for running the analytical tasks must be defined, taking into account the variety of data life-cycles of IoT applications. In this case, analytical

The associate editor coordinating the review of this manuscript and approving it for publication was Chin-Feng Lai.

capabilities can be described as being descriptive, diagnostic, and predictive. However, it is still unknown what type of behaviour data streams exhibit during the data-life cycles of IoT applications. Finally, an overall orchestration of the computational resources (i.e. edge, fog and cloud nodes) must be accomplished in order to guarantee a smooth execution of a variety of analytical tasks.

The contribution of this paper can be summarized as follows:

- We propose an Analytical Everywhere framework that integrates computational resources needed for a seamless execution of a network of analytical tasks having automated analytical capabilities, generating useful and high level information in a timely way.
- We demonstrate that a single computational resource (e.g. cloud) is not sufficient to support all analytical capabilities that are needed for IoT applications, considering computing power, data stream management, storage and networking capabilities.
- We discuss the challenges and how an Analytics Everywhere framework can be designed to perform descriptive, diagnostic, and predictive analytical tasks.
- We validate our Analytics Everywhere framework using a transit experiment by highlighting the pitfalls and discussing our experience.

The remainder of this paper is organized as follows. In Section II, we reviewed different IoT enabling technologies and the data analytics that have been previously implemented using cloud/fog/edge computing. In Section III, the Analytics Everywhere framework is presented, including the components of resource capability, analytical capability, and data life-cycle. Section IV is dedicated to building an Analytics Everywhere architecture. Section V describes in detail the experiment of implementing our framework for a smart transit scenario and discusses the results. Section VI concludes the paper and discusses further research.

II. RELATED WORK

It is indisputable that IoT sensors will produce a large amount of high-speed streamed and heterogeneous data that poses many challenges to performing management, processing, and analytical tasks within an acceptable time [8].

A. IOT ENABLING TECHNOLOGIES

Al-Fuquha et al. [9] provide an overview of IoT enabling technologies that can offer automation, data aggregation, and protocol adaptation using different IoT sensors. Overall, four main technologies can be identified in IoT: cloud, fog, edge, and communication technologies.

1) CLOUD COMPUTING

Cloud Computing has dominated the infrastructure and processing architectures developed to support Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) models during the last decade, leading to a trend of Everything as a Service (XaaS) [10]. By providing on-demand processing services with high availability and rapid elasticity through a selection of cloud architectures (e.g. Private, Public, Community, and Hybrid Cloud), previous research has pointed out that IoT devices can benefit from the virtually unlimited resources of the cloud, which compensates for their limitations in storage and computing capabilities ([11]–[13]). As a result, most of the architectures used to monitor ([14], [15]), optimize [16], and analyze [17] IoT data streams have been developed based on cloud computing.

However, cloud computing has shown limitations in supporting the short response time needed for processing the high data rates generated by IoT devices. Several open sources for processing IoT data streams such as Apache Storm [18] or Apache Spark [19] have been proposed in the literature but they still present major drawbacks due to the geographic distribution, large-scale, and latency-sensitive characteristics of IoT applications ([20], [21]). It is worth noting that transporting the data streams to the cloud can still generate bottlenecks. While data storage density and computing power have increased 10^{18} and 10^{15} times respectively, the broadband capability has increased only 10^{4} times over the last 20 years [11]. Pushing the processing closer to IoT devices has emerged as an alternative solution, and edge and fog computing have been proposed as alternative IoT enabling technologies ([20]–[23]).

2) EDGE COMPUTING

According to Shi et al. [24], *edge computing* refers to *"the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services."* The rationale behind edge computing is that 45% of IoT data will be processed and analyzed at the edge of the network in the future [24]. Recently, Harth et al. [25] have attempted to alleviate the network burden of transporting IoT data to the cloud by locally applying aggregation analytics at the edge, and sacrificing the analytical capability power due to the constraints of edge resources. A sliding window was applied to execute a simple linear classification to infer the context vectors (n-dimension row vector of contextual parameters such as temperature, sound, and humidity) within a specific tolerance threshold. Then, an aggregation analytics task including distributive, algebraic, and holistic functions was triggered if the errors of the inferred context vectors were lower than the threshold. Otherwise, the smoothing algorithm reconstructed the context vectors before executing the aggregation analytics task.

3) FOG COMPUTING

Fog computing was first introduced by Cisco as a bridge between the edge and cloud resources [26]. Other technologies having a similar concept were also proposed in the literature such as cloudlet [27] and mobile cloud computing [28] as well as mobile edge computing [29]. Lee et al. [30] proposed an online computational caching framework to minimize

the latency by storing and reusing intermediate computation results using fog nodes. Moreover, near realtime analytics was demonstrated in a seismic case study and realtime analytics was also achieved in an ambient noise imaging case study where a fog computing middle-ware architecture was developed for distributed cooperative analytics [31]. Other scenarios have been envisaged to apply fog computing, including Augmented Reality (AR), realtime Video Analytics, Mobile Big Data Analytics [32], Smart Grid, Smart Traffic Lights and Connected Vehicles [23], Decentralized Smart Building Control, Wireless Sensors and Actuators Networks [33]. Unfortunately, none of these scenarios have been implemented so far.

4) COMMUNICATION TECHNOLOGIES

Advances in communication technology play a vital role in bolstering the current growth of IoT. The proliferation of IoT sensors/devices is partially thanks to the advancements in wireless communication technologies including Wireless Local Area Network (WLAN), Wireless Personal Area Network (WPAN), and Low-Power Wide Area Network (LPWAN) [34]. While WLAN/WPAN provide a short range connectivity (about 1-100 metres) to support device-to-device (D2D) communication with a high data rate, LPWAN does not require much power, nor bandwidth to operate and provides long range connectivity (up to 50 kilometres) [35]. Some typical communication technologies of WLAN/WPAN including Radio-frequency Identification (RFID) [36], Bluetooth Low Energy 4.0 [37], Zigbee [38], and Wi-Fi (IEEE 802.11) are applied in different IoT applications such as Smart Tourism [39], Smart Home [40], Connected Health [41]. LPWAN technologies including unlicensed (e.g. SigFox, LoRa [42]) and licensed (i.e. NB-IoT [43]) spectrum band are promising in terms of lowering power consumption, and cost, and increasing reliability and range [44].

Cellular technologies that offer reliable broadband communication have had a certain role in shaping the IoT applications in the past, and they are expected to play an important role in the future. We have witnessed the growth of several generations of cellular networks from 2G and 2.5G which were designed to support voice services with an extension of small amount of data transmission, to 3G and 4G LTE that were capable of offering a wide coverage area, high security, and a dedicated spectrum allocation [45]. Although cellular technologies are not fit for all IoT applications, since they require very high operational cost and power consumption, they have shown to be suitable for specific scenarios such as connected cars or fleet management [46]. In particular, the next-generation, 5G, is expected to provide extreme mobile broadband (xMBB), massive machine-type communications (mMTC), and ultra-reliable machine-type communications (uMTC) and is positioned to be the future communication technology for IoT applications that require ultra-low latency [47], [48].

B. DATA ANALYTICS FOR IOT

Table 1 provides an overview of the type of analytical capability that has been implemented using cloud/fog/edge resources for different IoT applications. Most of the research efforts have been focused on descriptive analytics, and in particular, using edge computing resources to support near realtime/realtime analytics. The variety of IoT devices requires analyzing heterogeneous data "on the fly" and storing these data using various storage technologies. Very few studies found in the literature propose diagnostics and predictive analytics and were usually implemented in the cloud. To the best of our knowledge, our proposed *"Analytics Everywhere"* framework is the first research effort to combine different analytical capabilities in such a way that data streams can be transported and analyzed using the edge, fog, and cloud resources. These resources are inter-dependent and should be jointly developed to support IoT applications.

III. ANALYTICS EVERYWHERE FRAMEWORK

This section describes our Analytics Everywhere framework to support the development of new data life-cycles and facilitate the building of effective resource and analytics capabilities for IoT applications. The three main components are as follows:

- Resource capability: This component consists of distributed computational nodes (i.e. cloud, fog, and edge nodes) that provide I/O, storage, computation and processing power for the execution of a network of analytical tasks;
- Analytical capability: This component describes the best practice methods/algorithms for the execution of a network of analytical tasks that can meet the requirements of IoT applications;
- Data life-cycle: This component describes the changes that data streams go through during the automated execution of a network of analytical tasks.

A. RESOURCE CAPABILITY

An Analytics Everywhere framework is required to integrate resource capabilities taking into account one of the following aspects:

- Vicinity: This dimension describes how geographically close a compute node is to the source of data in order to execute a network of analytical tasks in that particular node. This dimension plays an important role in supporting IoT applications since compute nodes can be static (i.e. deployed inside a building) or mobile (e.g. deployed in a car), and their proximity to IoT devices, which are usually widespread geographically and mobile, will require integrated resource capabilities.
- Reachability: This dimension represents how easy it is to reach a compute node via a network. Typically, if a compute node is connected to the Internet with a fixed IP address, this can be considered a highly reachable resource, as opposed to a node connected using a private network and behind a Network Address

TABLE 1. Overview of the analytical capabilities and their cloud/fog/edge resources for IoT applications.

Resource Capability	IoT devices	Analytical Capability	Applications	Ref.
Cloud	RFID Tags, BLE	Class Association Rule Mining using Sub-group Discovery	Anticipatory Ubiquitous Computing	[49]
Cloud	WiFi, BLE	Clustering and Aggregating, Naïve Bayes	Location/Future Movement Prediction	[50]
Cloud	Spatial-Temporal Data, GPS, Camera, Environmental Sensors	Clustering (DBSCAN), Querying	Moving Object Map Analytics (MOMA), Contextual Spatial-Temporal Analytics	[51]
Cloud	GPS, Rain Gauge Data, Road Incident Report, Social Media	Descriptive (Statistical) and Predictive (Addictive Model, Kernel, SVM)	Urban Trajectory Data Analytics System	[52]
Edge + Cloud	BLE	Descriptive (Statistical)	O/D Transportation Planning	[53]
Edge + Cloud	RFID Tags	Descriptive (Statistical)	RFID Ecosystem for management, IoT applications	[54]
Edge + Cloud	Sensors, Traffic Lights	Diagnostic (Virtual Representation and Data enrichment)	Virtual Object (VO) model to enrich context information with Cognitive Internet of Things	[55]
Edge	Phone Camera	Event Detection	Pedestrian Safety Detection (Offline Training/Online Detection)	[56]
Edge	Sensors, RFID	Descriptive (Statistical)	Proposed the Smart Object framework to encapsulate RFID, sensor, Internet-based data	[57]
Edge	Wearable Sensors, GPS Receivers, Laptop, Smartphone	Descriptive (survey, threshold analysis, self-report)	wearable system which can learn context-dependent personal preferences	[58]
Edge	Wifi Signal, GPS	Descriptive (Quantitative Analysis, Statistics)	Wireless monitoring system that can track pedestrian and passenger behaviors	[59]
Fog	R1+ Seismograph Nodes	Descriptive (Onboard cooperative processing)	A fog computing middleware for distributed cooperative data analytics for the seismic and ambient noise imaging case studies	[31]
Fog	Augmented Reality, Virtual Reality data	Descriptive (online computational caching algorithm)	A computational caching frame-work in a fog network to minimize the transmission latency and computational latency by storing and reusing intermediate computation results	[30]
Fog	Sensors tagged on animals	Descriptive (Statistical)	Analyzing animal behaviors and monitoring animal's health	[60]
Fog	Wearable Sensors	Diagnostics (K-mean Clustering)	Clustering on clinical speech data obtained from patients with Parkinson's disease	[61]

Translation (NAT). In the case of IoT applications, the heterogeneity of IoT devices combined with the predominance of wireless access and short range networks will require an always-on reachability.

- In-memory and storage: This aspect describes how much data in a compute node should be kept in memory or be stored as a single ordinary disk file or in a database. The IoT data streams are expected to stay in-memory for a limited period of time as needed by an analytical task, and this decision will also depend on the data rate and data latency of the compute nodes. The data rate varies from high rates of data collected at the edge to a low rate of aggregated and cleaned data arriving at the cloud. The latency is clearly very low at the edge due to the proximity to the IoT devices and increases as we move to the cloud.
- Computation: This dimension describes how much processing power is available at a compute node for performing a network of analytical tasks. A proper modeling taking into account the IoT application requirements can help in driving the decision about which computational resource to use in executing the analytical tasks.
- Standardization: This dimension represents the strongest challenge yet to be met in the implementation of Analytics Everywhere frameworks. The IoT standards range from network protocols and data-aggregation standards to security and privacy.

These dimensions play an important role in designing an Analytics Everywhere framework as shown in Figure 1. While computation and memory capabilities can increase as the analytical tasks are run from the edge to the cloud, reachability must be always available to an analytical task. Reachability is a critical dimension that requires analytical tasks to return well-timed and synchronized results, which demand a rapid increase in computational resources. Because fog nodes are intermediary gateways that seamlessly integrate edge and cloud resources, they can eliminate resource

contention in the compute nodes and the communication links. In contrast, edge nodes can facilitate the necessary scaling of IoT applications because of their proximity to the IoT devices, making them an important computational resource for supporting near or realtime data analytics. However, the lack of adoption of standards in edge resources and IoT devices is currently hampering the implementation of Analytics Everywhere frameworks for IoT applications.

B. ANALYTICAL CAPABILITY

In Analytics Everywhere frameworks, analytical capabilities can be described as being *descriptive, diagnostic,* and *predictive*. In general, descriptive analytics aims to summarize a given dataset, which can be either a representation of the entire population or a sample of it. While descriptive analytics can provide some key metrics and measures that might reveal *"What is happening in the real-world?"*, the diagnostic analytics aims to provide some insight to answer the question *"Why is it happening?"*. The findings of descriptive and diagnostic analytics can be utilized in predictive analytics to build prediction models for predicting tendencies, clusters and exceptions, and future trends. Based on the insights obtained from predictive analytics we can answer *"What will happen?"*.

Four major types of methods can be used to support descriptive analytics: *frequency measurement, central tendency measurement, dispersion or variation measurement,* and *position measurement*. Although descriptive analytics can be performed at the edge, fog, and cloud, we anticipate that it will be more often executed at the edge. This is due to its proximity to IoT sensors, and also because (i) raw data are usually small in volume at the edge, and (ii) raw data can be subject to IoT application requirements that prevent data from being moved to a cloud due to privacy concerns.

Diagnostic analytics can be executed close to or far from an IoT sensor, depending on where it is more feasible to install relatively powerful computational resources. Diagnostic analytical tasks are usually supported by several algorithms such as DBSCAN [62] and Affinity Propagation Clustering [63], which are executed to uncover hidden insights, patterns from contextualized data. Fog and cloud resources can be used to perform diagnostic analytics since they provide more powerful computation, storage, and accelerator resources than edge nodes. They can improve the accuracy and reduce the computational complexity of the diagnostic process by performing automated tasks in near realtime or periodically.

Predictive analytics requires on-demand processing services with high availability and rapid elasticity through the virtually unlimited resources of the cloud. New insights can be achieved by applying prediction algorithms such as Random Forest, Hidden Markov Model (HMM), and Neural Networks. Auto-scaling, scheduling, and monitoring services can also be used to handle the data streams received from the edge and fog nodes. The analytical tasks use a massive amount of historical IoT data that need to be processed according to the nature of IoT applications.

The overall network of tasks of our Analytics Everywhere framework is represented as a Petri-Net model in order to ensure the optimal conceptualization and execution of analytical tasks by avoiding path deviations, bottlenecks, and parallelism. For example, bottlenecks directly impact the speed at which the data streams flow, causing the tasks involved in the bottleneck to experience higher processing time than expected, and as a result, causing a delay in the execution of a network of analytical tasks. Petri-Nets can not only detect bottlenecks, but it can also help us unfolding their causes. In the case of path deviations, our Petri-Net model allows us to detect the data streams that have followed different paths to those expected to occur within a network of analytical tasks. However, our Petri-Net model is not further discussed in this paper since it is out of the scope of this research work.

C. DATA LIFE-CYCLE

In our Analytics Everywhere framework, the data life-cycle consists of five data abstractions that are used to describe the data input and output of an analytical task. They are raw, aggregated, contextualized, transformed, and extracted data. The actual data-life cycle process will depend on the sequence of the analytical tasks designed to support an IoT application. We expect that different IoT applications will require specific data life-cycle processes, but will have similar data abstractions.

Definition 1 (Raw Data): The data streams $\mathbb{D}$ generated by IoT devices can be defined as a sequence of tuples $T_i \subseteq (T_1, \ldots, T_n)$ that contain a set of attributes such as:

$$T_i = (S_i, x_i, y_i, t_i)$$

where

- S_i*: is a set of attributes (i.e. measurements) obtained from an IoT device;*
- x_i, y_i*: is the geographical location of an IoT device;*
- t_i*: is the timestamp t when a measurement has occurred.*

These tuples represent the raw data in a data life-cycle and their main characteristics have been previously outlined by [64] as one of the following:

- They are potentially unbounded in size and they are transported using data packages according to a priori known time window.
- Each tuple in a data package arrives online. When the tuples are transported in batches, they are gathered in discrete packages at periodic intervals of time. An effective process begins by prioritizing routing data packages to a platform.
- There is no control over the order in which a tuple arrives within a data package or across data streams; and the probability distribution of the unknown data generation process may change over time due to its non-stationary state.
- It is not feasible to locally store a stream in its entirety since the local resources are normally limited.

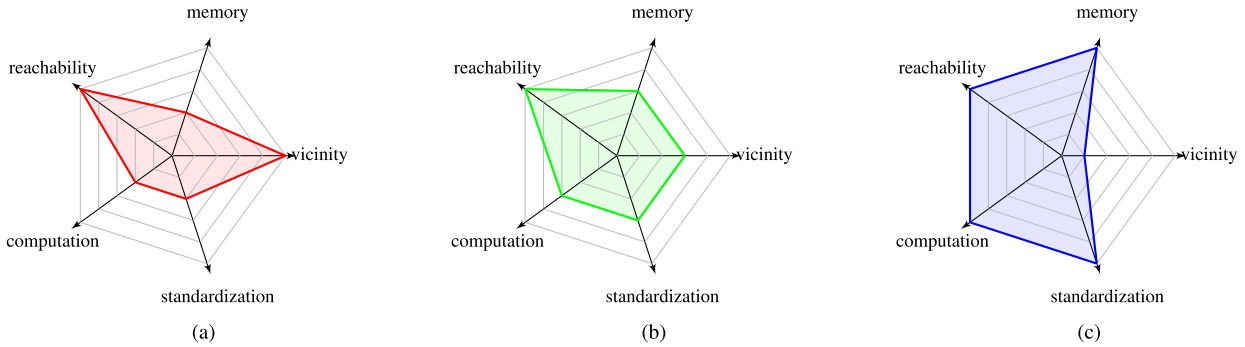


FIGURE 1. The main dimensions of resource capabilities. (a) Edge. (b) Fog. (c) Cloud.

This means that data tuples are active and stay only for a limited time period in memory locally.

Definition 2 (Aggregated Data): is defined as a set of new data tuples Q that are created by an aggregation operation Φ executed on a selected attribute (or a set of selected attributes) of a set of original data tuples T.

$$\begin{cases} \forall T_i \in (T_1, T_2, \ldots, T_n) : T_i = (S_i, x_i, y_i, t_i) \\ \mathbb{D} = (T_1, \ldots, T_n) \xrightarrow[\text{on attribute } S]{\Phi} \widehat{\mathbb{D}} = (Q_1, \ldots, Q_m) \\ \forall Q_j \in (Q_1, \ldots, Q_m) : Q_j = (Agg\_value_1, Agg\_value_2, \ldots) \end{cases}$$

Aggregation is a mathematical operation (e.g. sum, average, count, minimum) that takes multiple attributes of many tuples and returns a single value. However, some challenges still remain and they are associated with how to determine the granularity level that is needed by an analytical task and how the data output should be structured to avoid overly aggregating the data. For example, Analytics Everywhere frameworks depend on the time granularity being used at a compute node, which can be a priori defined (e.g. every day, every month) or can be event-based where the time granularity is defined by when an event occurs. Moreover, the heterogeneity of IoT devices brings a variety of granularity relationships among compute nodes within an Analytics Everywhere framework. Bettini et al. [65] described them as being groups into, finer than, shift equivalent, groups periodically into. The challenge is to design an Analytical Everywhere framework that can handle these relationships meanwhile the tuples are being aggregated at different compute nodes.

Definition 3 (Contextualized Data): is defined as a set of new data tuples P that are created throughout the contextualization process using contextualization operation Ψ to add new attributes to the original data tuples T.

$$\begin{cases} \forall T_i \in (T_1, T_2, \ldots, T_n) : T_i = (S_i, x_i, y_i, t_i) \\ \mathbb{D} = (T_1, \ldots, T_n) \xrightarrow{\Psi} \overline{\mathbb{D}} = (P_1, \ldots, P_n) \\ \forall P_i \in (P_1, \ldots, P_n) : P_i = (S_i, x_i, y_i, t_i, Context_1, Context_2, \ldots) \end{cases}$$

Contextualization is the most complex step in a data life-cycle that is performed to enrich the tuples using high level concepts accordingly to a particular IoT application. It is crucial in transforming meaningless tuples generated by IoT devices into semantically enriched data that are needed as an input to analytical tasks. New attributes are added to each tuple that can actually represent a context that characterizes a situation and the surroundings of IoT devices.

Definition 4 (Transformed Data): is defined as a set of new data tuples K that are created by a transformation operation Υ executed on a selected attribute (or a set of selected attributes) of a set of original data tuples T.

$$\begin{cases} \forall T_i \in (T_1, T_2, \ldots, T_n) : T_i = (S_i, x_i, y_i, t_i) \\ \mathbb{D} = (T_1, \ldots, T_n) \xrightarrow{\Upsilon} \ddot{\mathbb{D}} = (K_1, \ldots, K_n) \\ \forall K_i \in (K_1, \ldots, K_n) : K_i = (Trans\_value_1, Trans\_value_2, \ldots) \end{cases}$$

Transformation refers to the replacement of an attribute by a function since there is a need to change the scale of an attribute or standardize the values of this attribute that belongs to a tuple. In Analytics Everywhere frameworks, transformation plays an important role in using categories or bins to incrementally create new attributes that can help to advance the analytical tasks.

Definition 5 (Extracted Data): is defined as a subset of data tuples that are extracted from a set of original data tuples T using extraction (filtering) operation Ω; or a set of data tuples L that are created by an extraction (filtering) operation Ω executed on a selected attribute (or a set of selected attributes) of a set of original data tuples T.

$$\begin{cases} \forall T_i \in (T_1, T_2, \ldots, T_n) : T_i = (S_i, x_i, y_i, t_i) \\ \mathbb{D} = (T_1, \ldots, T_n) \xrightarrow[\text{on attributes } (S|x|y|t)]{\Omega} \mathbb{D}'' = (L_1, \ldots, L_n) \\ \forall L_i \in (L_1, \ldots, L_n) : L_i = (att_1, att_2, \ldots), \quad \forall att \subset (S, x, y, t) \end{cases}$$

D. DATA LIFE-CYCLES IN RELATION TO RESOURCE AND ANALYTICAL CAPABILITIES

Determining how to map different analytical capabilities with the most appropriate computing resources based on a data life-cycle of an IoT application is far from being a trivial endeavour since several aspects must be taken into account. Not all analytical tasks can run on all compute nodes due to the complexity of learning paradigms that currently exist such as deep learning, on-line learning, local learning, and anticipatory learning, to mention a few. Moreover, it is important to point out that an Analytics Everywhere framework will

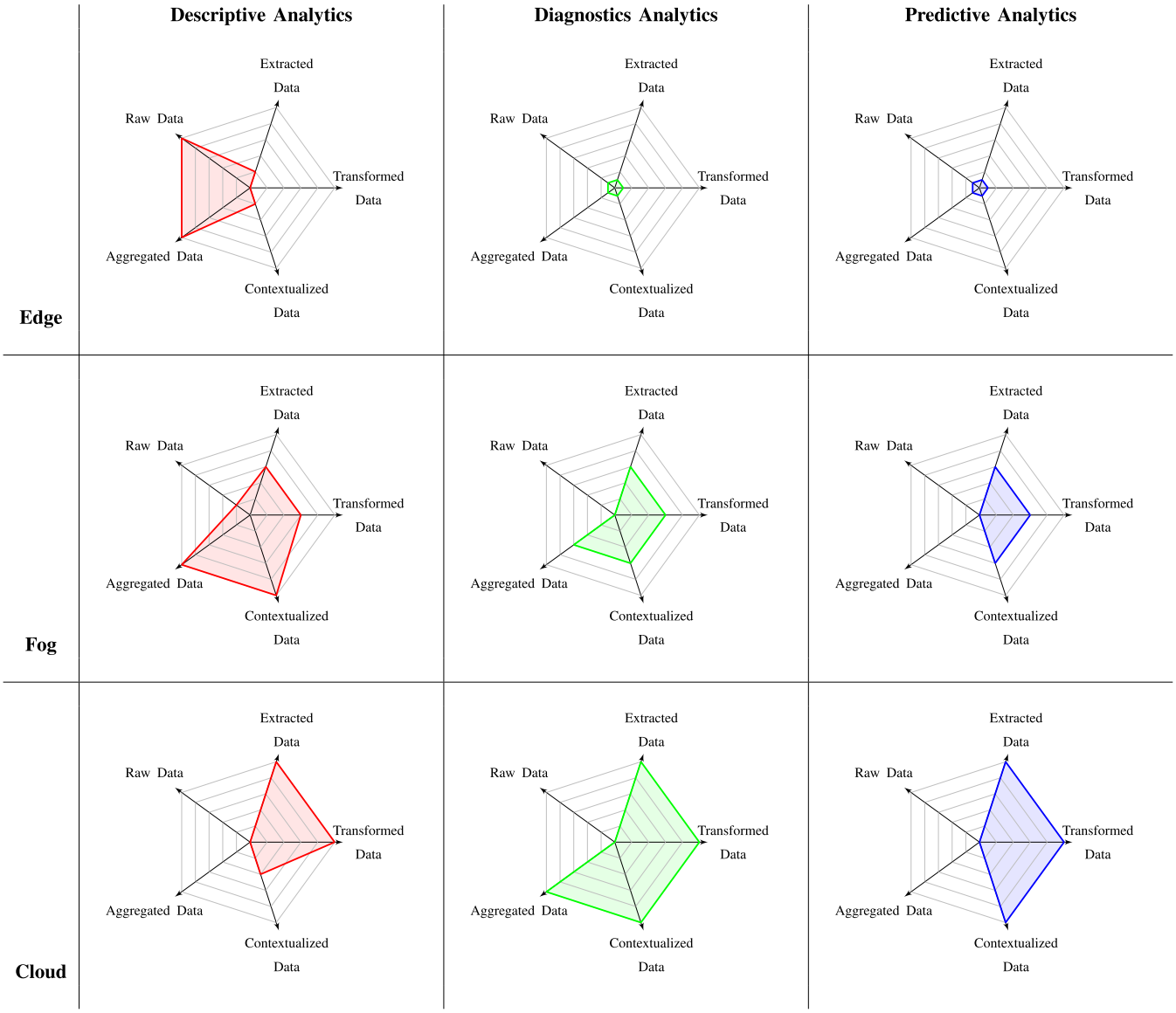


FIGURE 2. The matrix of data life cycle in relation to the analytical and resource capabilities.

have limitations and real-world IoT applications will play an important role in providing empirical evidence to validate and improve such a framework.

In Figure 2 we provide an overview of our proposed Analytics Everywhere framework, where each cell of the grid represents the expected data life-cycle according to analytical and resource capabilities. Overall, descriptive analytics at the edge will be more likely to handle raw data and aggregated data; while diagnostic and predictive analytics will be impracticable at the edge. By comparison, descriptive analytics in the fog will require data contextualization tasks that will support further extraction and transformation of data in the cloud.

On the one hand, fog resources are aimed at scaling up the processing power of edge nodes since larger data sets will be aggregated, contextualized, and transformed as needed for the descriptive, diagnostic, or predictive analytical tasks. On the other hand, the data life-cycles in the cloud are dependent on the type of data analytics that is required by an IoT application. Fog resources are not expected to replace the cloud. In fact, predictive analytics in the cloud will deal with contextualized, transformed and extracted data as well. We also can observe how data aggregation will play a significant role in diagnostic analytical tasks.

One example of these permutations includes IoT applications where analytical tasks are expected to be running at edge and fog resources since network and cloud connections are not available. For example, only 1 percent of data from an oil rig with 30,000 sensors is currently being analyzed for anomaly detection and control rather than optimization and

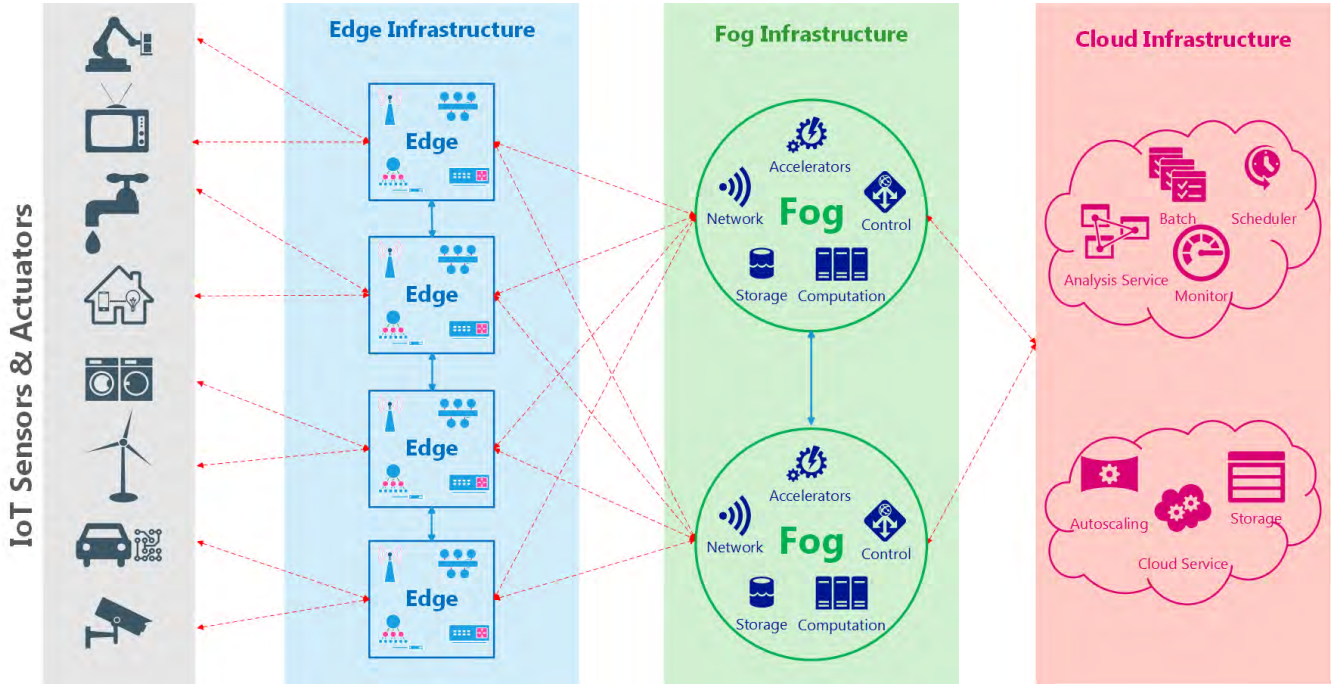


FIGURE 3. The proposed edge-fog-cloud architecture.

prediction [66]. Other IoT applications in smart buildings and smart mobility will typically require different permutations at all three resource levels (edge, fog, and cloud). The transit application we discuss later in this paper is a typical example of this case.

IV. ANALYTICS EVERYWHERE ARCHITECTURE

We propose an architecture in which any analytical capability is mapped into and executed by a distributed resource architecture composed of a hierarchy of resources available at the edge, the fog, and the cloud. The proposed architecture is illustrated in Figure 3. The aim is to support analytical tasks using a combination of different computation resources available at the edge nodes, the fog nodes and the cloud in order to provide meaningful information, actionable insights, and knowledge anytime and anywhere.

This section describes a general design guidance to implement an Analytical Everywhere framework. It consists of the following main components: networking, storage, computation/accelerators, controller/feedback, and data stream management/monitoring.

A. NETWORKING

It is very important to choose the right networking technology for supporting a variety of IoT sensors. Therefore, network standards, topology, and protocols should be considered carefully. Network developers need to consider various networking characteristics including throughput, fault tolerance, data rate, frequency band, power consumption per bit, number of nodes (hops) per network, and nominal range. In order to balance the evaluations of these networking characteristics, a network topology is vital to outline the connections between the elements in the network (i.e.: IoT sensors/devices, hub, gateways, edge nodes, fog nodes).

It is important to point out that due to the nature of our Analytics Everywhere framework, a comprehensive management of the entire network topology is required including wired and wireless, and seeking access and data transfer from the edge to core network elements. The networking connection between sensors and edge nodes can support many types of connections (i.e.: Wi-Fi 802.11 a/b/g/n, LoRaWAN, Zigbee, 2G/3G/LTE Cellular) for rapid retrieval of tuples from the IoT devices themselves as well as a broadcasting service in which a forever loop of event time windows can be applied. One main requirement for implementing an Analytics Everywhere framework is to be able to guarantee that any unbounded size of raw generated tuples can be always transported independently from the type of an IoT device being used.

Once the management of the entire network topology is known, the appropriate communication protocols need to be selected. Figure 4 summarizes the most popular networking protocols and communication layers that are currently available. The protocol stack is described from low physical layers to high abstracted application layers.

The protocol selection will rely on the requirements related to what type of IoT devices are going to be used, how much realtime or near realtime versus batch processing is required, and what type of resource capabilities are available in the network. In other words, a one-protocol-fits-all approaches cannot be applied when implementing Analytics Everywhere frameworks.

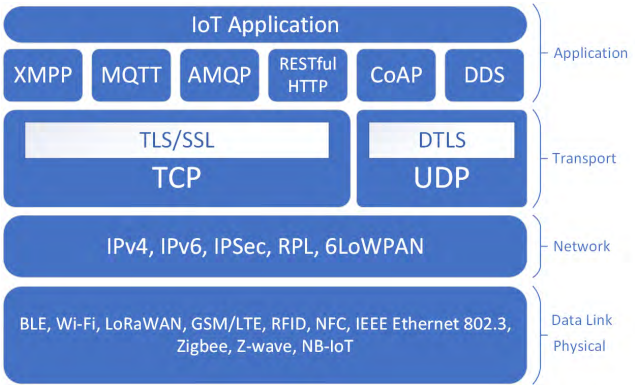


FIGURE 4. Current networking protocols supported by the analytics everywhere framework.

B. STORAGE

The second component of our system architecture that needs careful evaluation is the storage space. Indeed, the raw data tuples are constantly being generated by the IoT devices, transmitted over the network, and accumulated gradually over time. To find an optimal solution to storing the data is a non-trivial task when designing a system architecture for an Analytics Everywhere framework. The main design solutions are related to the following questions: *(1) which type of storage method should be applied? (2) where should the data be stored? (3) when is there a need to store data? (4) how can high availability be provided?*

The general guidelines are as follows:

- *In memory vs disk storage:* The mission-critical data tuples (hot data) that need to be accessed frequently by the analytical tasks should be stored in ways offering fast retrieval and updates. Therefore, they should be kept in memory of the computational nodes, while less urgently accessed data (cold data) can be stored in a database, on disk, or in data files. Edge nodes in particular should be used to store in-memory data only.
- *Small vs medium vs large data:* Edge nodes are normally lightweight with low storage capabilities, while nodes at the fog have higher storage capability, and nodes in the cloud have the highest storage capability. Therefore, small, medium, and large data can be stored at the edge, fog, and cloud, respectively.
- *Nodes federation:* It is necessary to provide fault tolerance and high availability for data storage in our system architecture. All the computational nodes (at the edge, fog, and cloud) in the network can be used to aggregate and interconnect their storage environment as a unique place where data can be partitioned into many copy blocks and distributed everywhere in the IoT network.

C. COMPUTATION/ACCELERATORS

The computational nodes are usually deployed covering a large geographical area and they can be static (i.e. a fog node deployed inside a building) or dynamic (e.g. an edge node deployed in a car). The core hardware of the computational nodes could be one or the combination of several processing units such as Graphics Processing Units (GPUs), Central Processing Units (CPUs), Accelerated Processing Units (APUs), Application Specific Integrated Chips (ASICs), Field Programmable Gate Arrays (FPGAs), and System-on-Chip (SoC) accelerators. These computational devices can handle tasks either in independent style or in parallel, concurrent, or distributed styles. In this paper, three main types of shared resources based on the geo-distribution (at the edge, the fog, and the cloud) can be used to determine the type of computational nodes that are needed for the analytical tasks.

From the acceleration of data processing perspective, the processing power of the computational nodes at the edge, fog, and cloud are sorted from low to medium to high. Therefore, nodes at the edge (static or dynamic) should be used to implement analytical algorithms for performing lightweight tasks such as descriptive analytics (in local scale) in order to generate new insights about the IoT device behaviors such as communication problems and low battery. Many IoT devices are expected to be connected to one or more edge nodes. However, high performance processing capabilities at the edge are prohibitive and may cause computational resource contention. Therefore, the accelerators at the fog can handle the heavier analytical tasks including descriptive (in regional scale) or diagnostic to reveal the patterns such as anomalies in the system. The highest computational capability in the cloud allows the nodes to handle the heaviest analytical tasks such as descriptive (in global scale), diagnostic (in long-term diagnosing), or predictive to forecast future changes in the system.

D. CONTROLLER/FEEDBACK

The controller/feedback is an important component in this architecture. Once the analytical results of different analytical capabilities on the compute nodes at different places (edge, fog, cloud) are achieved, the actions of the IoT system need to be guided to optimize or adapt with the new change, new situation, new environment. Therefore, the feedback, which is a relevant result of the analytical capabilities, is pushed back from any computational nodes to order users or IoT actuators to take immediate actions. The controller/feedback can be real time, near real time or batch processing time depending on the place where it is computed. The criteria to choose the ramification (real time vs near real time vs batch processing time) of feedback is closely tied to the requirements of the application. For example, real time feedback detects anomalies in the operational behavior of the device at the edge, or abnormal behavior in a traveling object's movement detected at the fog or the cloud.

E. DATA STREAM MANAGEMENT/MONITORING

In the Analytics Everywhere framework, there are two main options to select a data stream management engine: horizontal and vertical. The option chosen depends on the requirements of the application. Horizontal deployment means that the main components of a data stream

management engine are horizontally deployed across remote nodes. Some examples include the open-source platforms such as Apache Flink, Apache Samza, Apache Apex, Apache Storm, Apache Spark Streaming[1]. In contrast, vertical deployment not only expands their services to the edge but also scales the data stream management components to the nodes close to the IoT devices. This latter deployment is a new trend so there are not many unique options available. However, some platforms can be considered such as Cisco Kinetic, IBM Watson IoT Platform Edge, Microsoft Azure IoT Edge, or Apache Edgent[2].

Streaming management can be either *stateful* or *stateless* depending on the analytical requirements of an IoT application. Stateless streaming management treats each event independently and creates the output only depending on the data tuples of that event. As an example, we can use a filtering operation to filter an incoming data stream of a transit network by a field (i.e.: busID) and write the filtered messages to their own stream. In contrast, stateful streaming management combines different events together and creates the output based on multiple data tuples taken from those events. A good example of this is counting the number of stops made at bus stations at which all buses in the transit network pull over during a day. Moreover, developers can also specify a reliability mode or management semantics that guarantee it will provide for IoT data streaming across the entirety of the application architecture. It is worth noting that the guarantee is not only at the protocol level but it also can apply to the data stream management platforms. There are three main approaches as follows:

- At most once: At most once is a euphemism for there being no correctness guarantees that data tuples in a stream are guaranteed to be handled at most once by all streaming operators in the application. In other words, in the event of a failure, no additional attempts are made to re-handle these data tuples.
- At least once: At least once means that data tuples in a stream are guaranteed to be handled at least once by all operators in the application. If the failure happens, additional attempts will be made to re-handle these data tuples. This approach may cause unnecessary duplication of data tuples in the streams.
- Exactly once: Exactly once means that data tuples are guaranteed to be handled exactly the same as it would be in the failure-free scenario, even in the event of various failures.

V. PUBLIC TRANSIT SCENARIO

A. OVERVIEW OF THE CODIAC TRANSPO SERVICE

Public transport authorities must understand the performance of transit services to develop strategies for better transportation decision-making policies. Traditional solutions either failed to find the answers or have been too expensive to be widely deployed. Our Analytics Everywhere framework can provide automated analytical capabilities that rely on the most appropriate computing resources. Moreover, the outcomes of our Analytics Everywhere framework can not only serve a transit authority, but it can also support a variety of user groups such as bus drivers and passengers who are seeking new insights to optimize their decisions and adjust their behaviors. For example, bus drivers might be interested in knowing how their driving performance has been for the last week while passengers would be interested in how frequently the services are delivered on-time.

In this section, we present the CODIAC Transpo as a public transit scenario to evaluate our proposed Analytics Everywhere framework. CODIAC Transpo serves the area of Greater Moncton, Canada[3]. Annually, CODIAC Transpo provides more than 2.3 million rides to transit users from Moncton, Dieppe and Riverview Area. The transit network currently operates 30 bus routes from Monday to Saturday, some of which have additional evening and Sunday services. Aiming to assist CODIAC Transpo in providing a safe, reliable, and professional transit service for passengers, we selected the following analytical capabilities:

- Descriptive Analytics: What is currently happening with the bus services in the CODIAC Transpo network?
- Diagnostic Analytics: Why have abnormal phenomena (e.g. congested, service interrupted, or normal events) happened to a bus service?
- Predictive Analytics: What will likely to happen to a bus service in the near future?

The CODIAC Transpo scenario can be described as each moving bus in the transit network generating realtime transit data feeds which are fetched by a mobile edge node installed directly in each bus. Here, descriptive analytical tasks are running while the bus moves around a city. Once the analytical results are locally generated at the edge, they provide actionable information about what is happening to a moving bus. There are several transit hubs around the city where passengers and cargo are exchanged. At the transit hubs, the fog nodes are deployed to collect the cleaned data streams and the descriptive analytic results from different edge nodes whenever the buses gather there. At the fog resources, automated diagnostic analytic tasks are applied to understand why any abnormal phenomena have happened. Finally, a private cloud infrastructure is deployed in the transit headquarters aiming to summarize and handle the data streams from all the buses in the transit network. Figure 5 illustrates the scenario developed for the CODIAC Transpo network.

1) THE TRANSIT FEEDS

In this scenario, each bus is equipped with a mobile edge node that receives streaming transit feeds every 5 seconds containing the GPS position and telemetry data from sensors

[1]https://flink.apache.org, http://samza.apache.org/, https://apex.apache.org/, https://storm.apache.org/, https://spark.apache.org/streaming/

[2]https://www.cisco.com/c/en/us/solutions/internet-of-things/iot-kinetic.html, https://console.bluemix.net/docs/services/IoT/edge/WIoTP_edge.html, https://azure.microsoft.com/en-ca/services/iot-edge/, https://edgent.apache.org/

[3]http://www.codiactranspo.ca/

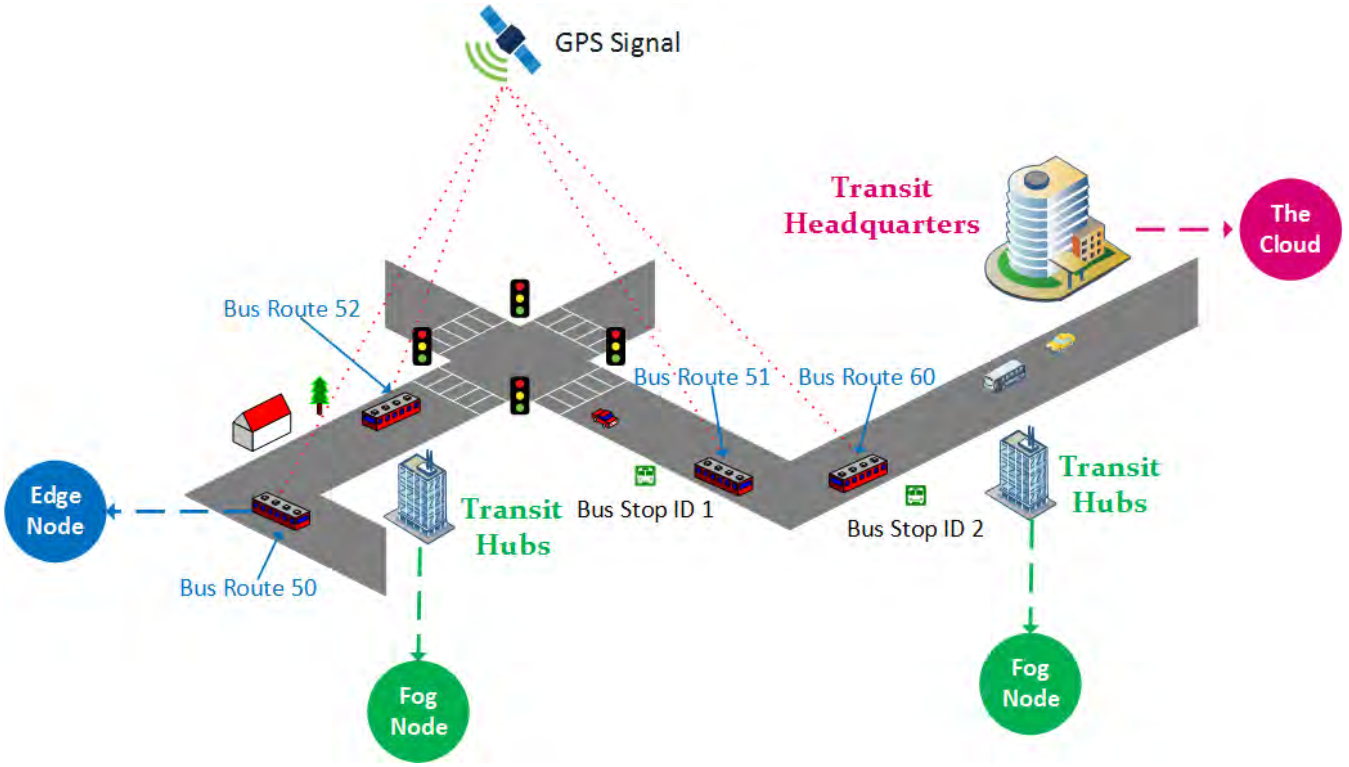


FIGURE 5. The CODIAC transpo scenario.

installed in the bus. These transit data feeds consist of a sequence $T_1, \ldots, T_n$ of out-of-order tuples containing attributes in the format:

$$T_i = (S_i, x_i, y_i, t_i) \tag{1}$$

where

S_i: is a set of attributes containing telemetry data such as the bus route identifier, the bus route number, the vehicle identifier, the trip identifier, the start time of a trip, and the end time of a trip. In this scenario we have a total of 17 attributes belonging to a tuple and they are listed in Table 2;

x_i, y_i, t_i: are the geographical coordinates x_i, y_i of the device at the sampling time t_i.

The bus route 51 was selected for evaluating our Analytics Everywhere framework because it has the highest trip density during a day. We have used 168,970 data tuples retrieved during a period of one week from 02/14/2017 to 02/20/2017. According to the transit schedule, there were 66 bus trips operating each day from Monday to Saturday and 23 bus trips on Sunday. As scheduled, each trip can take approximately 45 minutes.

2) ANALYTICAL CAPABILITIES

The descriptive analytics are expected to reveal schedule adherence patterns which can be used by transit operators to adjust their operations such as route optimization, schedule modification, or bus maintenance. The diagnostic analytics also provide new insights that can assist bus drivers to change their driving behaviors to improve their scheduled adherence to the services. Finally, predictive analytics offer global insights on the whole transit network such as predicting trip behavior. Table 3 provides an overview of analytical capabilities and their corresponding techniques that have been implemented for the CODIAC Transpo scenario.

TABLE 2. The 17 attributes of the transit data feed.

ID	Attribute Name	Description
1.	vlr_id	The data point ID in the vehicle location report table.
2.	route_id_vlr	The route ID in the vehicle location report table.
3.	route_name	The route name.
4.	RouteID	The route ID in the route transit authority table.
5.	route_nickname	The abbreviation of the route.
6.	trip_id_br	The trip ID in the bid 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	End 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.

3) DATA LIFE-CYCLE

It consists of two cycles:

- Raw data arriving at an edge node, aggregated data are transported from the edge nodes to a fog node, and

TABLE 3. Analytical capabilities of the CODIAC transpo scenario.

Analytical Capability	Techniques	IoT application	Target Group of Users
Descriptive	- Statistics	- Schedule adherence	Transit Operators
Diagnostic	- Affinity Propagation Clustering	- Abnormalities detection	Bus Drivers
Predictive	- Random Forest	- Trip behaviors prediction	Passengers

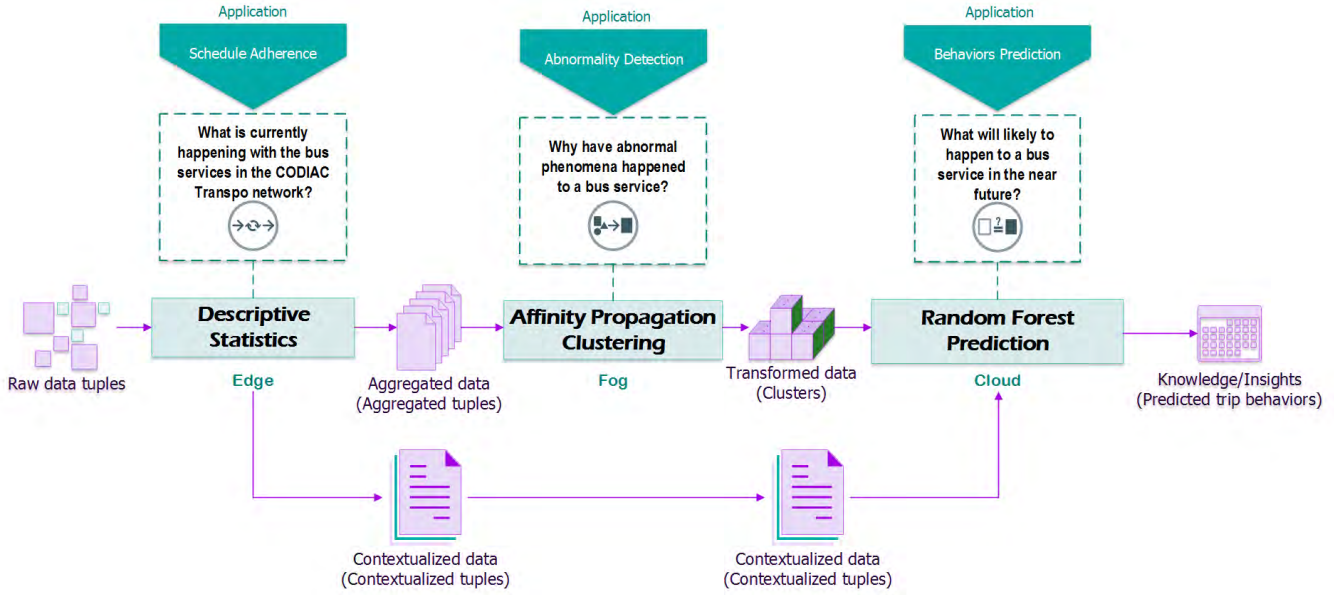


FIGURE 6. The knowledge/insights lifecycle from our public transit scenario.

transformed data are transported from the fog nodes to the cloud.

- Contextualized data are transported from the edge nodes directly to the cloud.

Figure 6 illustrates the data life-cycle implemented for the CODIAC Transpo scenario. The raw data tuples are generated every 5 seconds and the high volume of tuples, belonging to each sliding time window, is kept in-memory until it is transported to the fog node. The raw data tuples from the first time window are cleaned and pre-processed to remove errors, redundancies, and inconsistencies; the same tasks are performed for the next time windows in a sequential manner. The data tuples collected for the bus route trips were then contextualized at the mobile edge node to determine whether a bus is moving or stationary. These tuples have been further processed and analyzed at the edge using multiple descriptive statistical functions. From analytical results at the edge, the aggregated data were computed and passed through the fog for further diagnostic analytic tasks while the contextualized tuples were continuously sent to the cloud for prediction analytic tasks.

Every 6 hours, all aggregated data were scheduled to arrive at the fog node. Here, we ran the affinity propagation clustering algorithm over the aggregated data to transform them into clusters that can reveal abnormal trip behavior. Then, all transformed data (clusters) were also sent to the cloud for prediction analytic tasks.

The cloud receives the contextualized data tuples continuously being pushed from all the edge nodes as well as the transformed data resulting from the diagnostic analytical nodes. Both data sources (contextualized data tuples and transformed data) were used as input data of our random forest predicting model to predict the trip behaviors.

B. ANALYTICS EVERYWHERE ARCHITECTURE

The system architecture is shown in Figure 7. For the data ingestion, an http POST, Wi-Fi and a 3G connection were used for rapid tuples retrieval from the IoT devices themselves as well as a broadcasting service in which a forever loop of event time windows can be applied. At the edge, the Cisco IR829 Industrial Integrated Services Router was used as a mobile edge node and was installed inside a bus. The router has an Intel Atom Processor C2308 (1M Cache, 1.25 GHz) Dual Core X86 64bit, 2GB DDR3 memory and Wi-Fi connection. This edge node handles all traffic routing, switching, and networking using an IOx operating system, running on a virtual machine that uses Linux Yocto [67]. To collect the raw data tuples, Gateway Management Module (GMM) and Data Control Module (DCM), which are the integral parts of the Cisco Kinetic platform, were deployed on

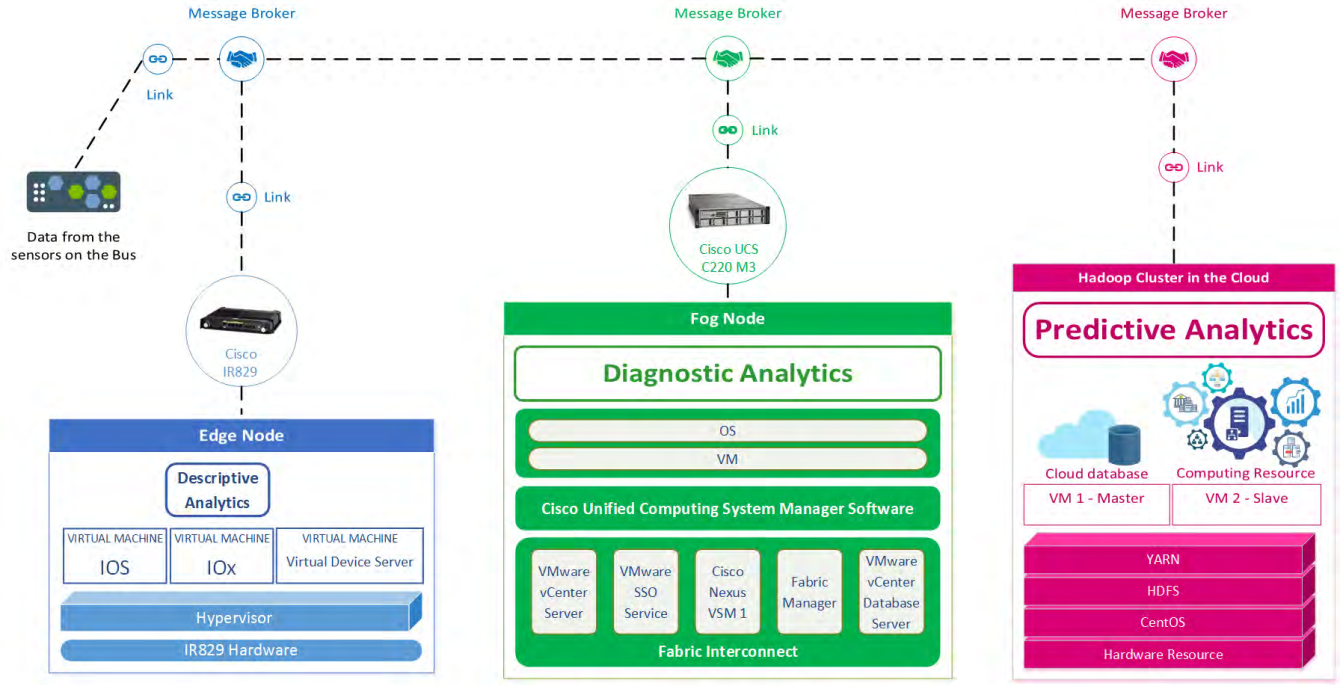


FIGURE 7. The analytics everywhere architecture implemented for our public transit scenario.

top of this mobile edge node. The Cisco Kinetic platform is a scalable, open system, and is adaptable for a variety of IoT applications. It can be used to extract, synchronize, compute, and move the data tuples to the right applications at the right time [64]. A Message Broker was established at the edge to move the data from the edge to fog.

The fog node was implemented using the Cisco UCS 240 modular with a two rack-unit (2RU) server and 2 Intel Xeon processor E5-2600 CPUs, 24 double-data-rate 4 (DDR4) dual in-line memory (DIMMs) of up to 2400 MHz speeds, 6 PCI Express (PCIe) Generation 3 slots, and 12 large-form factor hard drives. It is managed by the Cisco Unified Computing System Manager Software. The fog node can host a virtual machine where an operating system can be run.

The cloud cluster is supported by Compute Canada which provides an IaaS where we have created and allocated cloud resources such as VMs, Servers, Storage, Load Balancers, IP addresses. Our cloud capabilities include a maximum of 5 Instances, 40 VCPUs, 150GB RAM, 2 Floating IPs, 5TB Volume Storage. In the cloud, we have the capability to handle the global geo-distribution of data (the whole transit network) and we have enough computing resources to perform complex analytical tasks. All necessary data needed for different analytical tasks are stored and are available in the cloud. The Hadoop ecosystem, in particular Apache HBase, Apache Zookeeper have been deployed in the cloud.

C. DESCRIPTIVE ANALYTICS

A contextualize function was implemented to interpret the status of a bus. The GPS coordinates were sent to the edge node every 5 seconds. A fixed distance value between two consecutive GPS positions of the bus was used for determining stops and moves. This value was empirically determined for the CODIAC Transpo network as being 15 meters. When the distance between the previous point and the current point is more than 15 meters, the bus is moving; therefore the current point is tagged as a move. In contrast, when the distance is less than 15 meters, the current point is tagged as a stop.

Additionally, a temporal aggregation function was used to compute (i) the actual time duration of a trip using the timestamps of the origin and destination points of each trip; (ii) the total number of stops during a trip; and (iii) the total number of moves during a trip. In summary, five data fields (Trip Id, Date, Start_Time, Move_Status, Stop_Status, Finish_Time) were used for the temporal computations. The following function was used to implement this step:

$$f(m,s,t) = \begin{cases} M = \sum_{i=1}^{n} m_i & \text{if } m_i \neq 0 \\ S = \sum_{i=1}^{n} s_i & \text{if } s_i \neq 0 \\ \Delta(t) = T_D - T_O \end{cases}$$

where

M, S: are the total number of moves and stops, respectively.

m_i, s_i: are the move and stop status in each tuple.

i = 1..n: is the index of the tuple in the data stream.

$\Delta(t)$: is the total time length of the trip.

T_D, T_O: are the timestamps of the destination and origin tuple.

Next, we computed the average trip time in the morning (5AM-12PM), afternoon (1PM-6PM), and evening

(7PM-12AM). The average of the total number of moves and stops was computed for the different times of the day (i.e. morning, afternoon, evening) using the following function:

$$g(m,s,t) = \begin{cases} \overline{M} = \dfrac{\sum_{i=1}^{n} M_i}{n} \\ \overline{S} = \dfrac{\sum_{i=1}^{n} S_i}{n} \\ \overline{T} = \dfrac{\sum_{i=1}^{n} \Delta(t)_i}{n} \end{cases}$$

where

M_i, S_i, $\Delta(t)_i$: are the total moves, total stops, and total length of time for each trip.

n: is the number of trips during a period of time (morning, afternoon, evening).

D. DIAGNOSTIC ANALYTICS

The goal was to demonstrate how it is possible to diagnose the causes of abnormalities, such as the interruption of services in near realtime. The affinity propagation clustering algorithm [63] was selected to detect clusters. First, this algorithm automatically classified the clusters without prior knowledge about the number of clusters. Second, it can allow for non-metric dissimilarities. Therefore, we can handle non-metric space in our aggregated data. Also, the affinity propagation clustering algorithm is deterministic over runs. The main idea behind this algorithm was to use a graph-based approach to let all data points collectively vote on their preferred âĂŸexemplars', which are identified as those most representative of others. It is worth noting that implementing the affinity propagation clustering algorithm is a typical option of many options that we can choose for diagnostic analytics.

Algorithm 1 describes our implementation of the aggregated data pulled from the edge every 6 hours; its purpose is to discover any outliers that may indicate abnormal events (i.e.: traffic congestion). The input of this algorithm is a set of aggregated data points in which each data point contains 5 features *(TripID* $\langle Id_i \rangle$*, Start Time* $\langle St_i \rangle$*, Total_Move* $\langle \overline{M}_i \rangle$*, Total_Stop* $\langle \overline{S}_i \rangle$*, Total Trip Time* $\langle \overline{T}_i \rangle$*)* obtained from the edge computation after the end of each bus trip. The two most important features, Total_Move $\langle \overline{M}_i \rangle$ and Total_Stop $\langle \overline{S}_i \rangle$, are used as input for the clustering algorithm. At the end of this implementation process, the output will contain a set of original aggregated data points plus the cluster labels $\langle \widehat{C}_i \rangle$, which represent the aggregated information related to each trip, and a cluster that this set of data points belong to.

E. PREDICTIVE ANALYTICS

We have used Random Forest (RF) to build a predictive model based on the performance benchmark carried out by [68]. Random Forest is an ensemble learning algorithm that can be used both for classification and regression problems by combining many small, weak decision trees in parallel to form a single, strong predictive model [69]. Figure 8 depicts the predictive model showing a number of decision trees that

Algorithm 1: Clustering Algorithm Using Affinity Propagation Approach

Data: Set of $U = (U_1, U_2, U_3, \ldots)$ such that $U_i = (Id_i, St_i, \overline{M}_i, \overline{S}_i, \overline{T}_i)$ is the aggregated data point

Result: $Q = (Q_1, Q_2, \ldots)$ such that $Q_i = (Id_i, St_i, \overline{M}_i, \overline{S}_i, \overline{T}_i, \widehat{C}_i)$ in which $\widehat{C} = (\widehat{C}_1, \ldots, \widehat{C}_n)$, $\widehat{C}_j = argmax[a(j,k) + r(j,k)]$

Initialize: The Similarity Matrix S $\forall j,k : s(j,k) = 0$; The Availability Matrix A $\forall j,k : a(j,k) = 0$; The Responsibility Matrix R $\forall j,k : r(j,k) = 0$;

Function `AP_Clustering(U)`**:**

  Compute Matrix S: $\forall j,k : s(j,k) \leftarrow -||V_j - V_k||^2$ where $V_j = (\overline{M}_j, \overline{S}_j)$ extracted from U_j; $V_k = (\overline{M}_k, \overline{S}_k)$ extracted from U_k;

  repeat

    Update Matrix R:

$$\forall j,k : r(j,k) \leftarrow s(j,k) - \max_{k':k' \neq k} \{a(j,k') + s(j,k')\}$$

    Update Matrix A:

$$\forall j,k : \begin{cases} a(j,k) \leftarrow \min\{0, r(k,k) + \sum_{j':j' \notin \{j,k\}} \max\{0, r(j',k)\} \\ a(k,k) \leftarrow \sum_{j' \neq k} \max(0, r(j',k)) \end{cases}$$

    Cluster assignments:

$$\widehat{C} = (\widehat{C}_1, \ldots, \widehat{C}_n), \widehat{C}_j = argmax[a(j,k) + r(j,k)]$$

  until *The Responsibility R and Availability Matrix A converge*;

  $Q = U \bowtie \widehat{C}$

  return Q;

were created during the training phase. Each decision tree contains a random subset of the most relevant features. When a new data tuple comes to the prediction model, it is predicted through each decision tree and returns the target class label. A majority-voting function was utilized to vote the majority target class label and predict the label.

Algorithm 2 provides details for the purpose of predicting trip behavior such as abnormal/normal events. The algorithm inputs are the clustering data pulled from the fog and the contextualized tuples received from the edge. The clustering data are a set Q of data points in which each data point contains 7 features *(TripID* $\langle Id_i \rangle$*, Start Time* $\langle St_i \rangle$*, Total_Move* $\langle \overline{M}_i \rangle$*, Total_Stop* $\langle \overline{S}_i \rangle$*, Total Trip Time* $\langle \overline{T}_i \rangle$*, Cluster Label* $\langle \widehat{C}_i \rangle$*, Behavior Label* $\langle Behavior_i \rangle$*)*. Meanwhile, the contextualized tuples belong to a set T' in which each tuple contains 17 features of the original tuple plus the new context feature.

The first step of Algorithm 2 is to merge the two datasets together to form a unique dataset that can be used for the predictive model. For this purpose, the contextualized data

Algorithm 2: Predicting algorithm using Random Forest

Data: Set of $T' = (T'_1, T'_2, ..)$ such that $T'_i = (S_i, x_i, y_i, t_i, context_i)$ is the contextualized tuples; Set of $Q = (Q_1, Q_2, ...)$ such that $Q_i = (Id_i, St_i, \overline{M}_i, \overline{S}_i, \overline{T}_i, \widehat{C}_i, Behavior_i)$ is clustering data

Result: Prediction model P

Function `Merge_Dataset` (T', Q) **:**
  $G = T' \bowtie Q$ using TripID and Start Time $\langle Id_i, St_i \rangle$;
  `/* Left outer join 2 datasets */`
  $G = G.delete(\langle \overline{M}_i, \overline{S}_i, \overline{T}_i, \widehat{C}_i \rangle) = (G_1, G_2, ...)$ such that $G_i = S_i, x_i, y_i, t_i, context_i, Behavior_i)$;
  return G;

Function `Handle_Class_Imbalance` $(G,$ *Method* $)$ **:**
  switch *the value of Method* **do**
    case *1* Upsample the minority class;
    case *2* Downsample the majority class;
    otherwise Synthesize new minority class;
  endsw
  K-fold Cross-Validation $(G) \rightarrow$ Training set (G') and Testing set (G'');
  return G', G'';

Initialize: Set number of small tree $Forest = int\_value$; Get number of features $F = Random\_number(2 : max\_no\_feature(G'))$;

Function `Build_Tree` (G', F) **:**
  At each node:
    $f \leftarrow$ randomly select subset of Feature F;
    Split on best feature in f;
  return *Small_Tree*;

Function `Random_Forest` (G', F) **:**
  $P \leftarrow \emptyset$
  foreach $Tree_i \subseteq Forest$ **do**
    $\widehat{G}' \leftarrow$ A bootstrap sample from G'
    $p_i \leftarrow Build\_Tree(\widehat{G}', F)$
    $P \leftarrow P \cup p_i$
  end
  return P;

tuples need to be indexed according to whether they have normal or abnormal behavior, based on the label provided by the clustering dataset. Therefore, we executed a left outer join operation on these datasets to form a new unique dataset. Then, we only keep the Behavior Label on this new dataset and eliminate the other features *(TripID* $\langle Id_i \rangle$, *Start Time* $\langle St_i \rangle$, *Total_Move* $\langle \overline{M}_i \rangle$, *Total_Stop* $\langle \overline{S}_i \rangle$, *Total Trip Time* $\langle \overline{T}_i \rangle$, *Cluster Label* $\langle \widehat{C}_i \rangle$*)* in order to avoid the impact on the predicted result since these other features are directly correlated to the Behavior Label.

Next, we handled another problem due to the data being outnumbered by normal behaviors with few instances of abnormal behaviors. This might cause bias towards the normal behaviors. Therefore, we used several solutions to balance the dataset; we used some methods such as upsampling the minority class (abnormal behaviors), downsampling the majority class (normal behaviors), or synthesizing a new minority class (abnormal behaviors) based on the existing samples. Then we applied cross validation procedure on the new dataset (training set G', testing set G'') to avoid overfitting or selection bias problems.

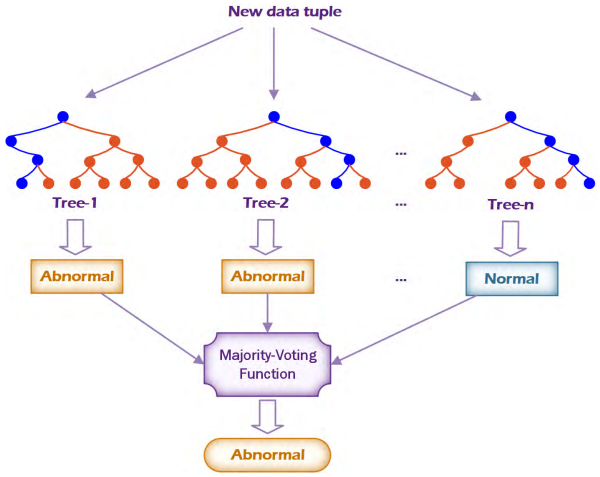


FIGURE 8. Random forest model with majority voting.

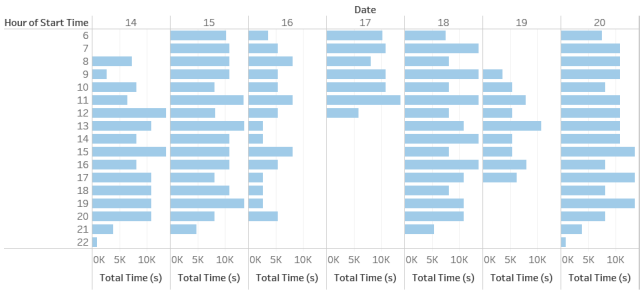


FIGURE 9. The distribution of the hourly trip times for each day of the week.

Once the class imbalance problem is handled, a predictive model is built based on the Random Forest approach: (i) A random number of decision trees are built in parallel. (ii) Each tree in the forest is built using a subset of features of the training set G' (the features are selected randomly among 17 features plus the context feature). (iii) Then, a bootstrap number of training samples from the training set G' are selected to form each tree in the forest. (iv) Finally, all the trees are combined together to form a single predictive model (see Algorithm 2).

F. RESULTS AND DISCUSSION

1) DESCRIPTIVE ANALYTICAL RESULTS AT THE EDGE

Fig. 9 illustrates the existence of several missing trips that have been detected in realtime. The buses did not run on February 14th at 6 AM to 7 AM; and there were no trips at 10 PM on the 15th, 16th, 18th. Moreover, missing trips have

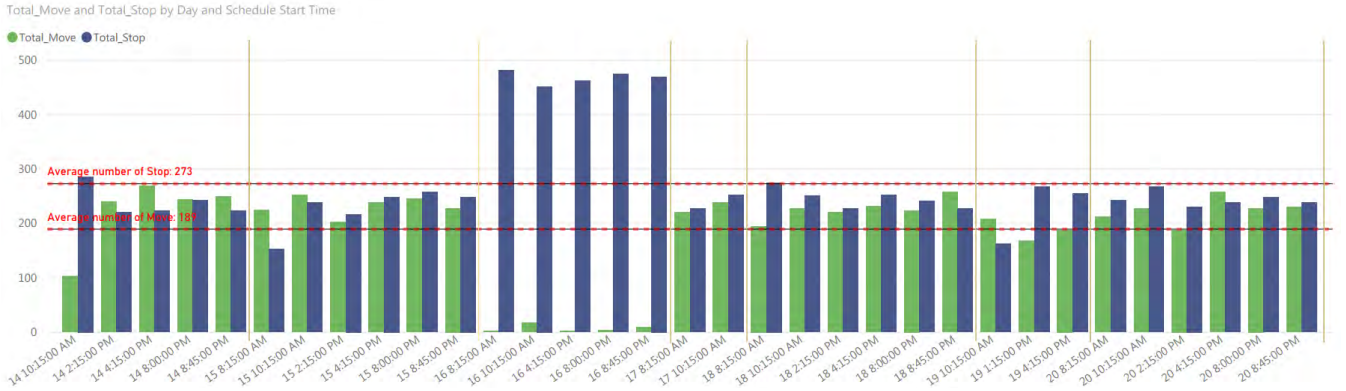


FIGURE 10. The comparison between the total number of stops and moves at different times during a week of observation.

also occurred on the 17th after 12 PM, on the 19th early in the morning (6 AM and 7 AM), and in the evening (6 PM to 10 PM). This is relevant information since it can generate warnings to the transit managers as well as passengers about the current state of the network at the trip level.

Moreover, computing the total trip time in realtime can provide relevant information to the transit manager about the abnormalities occurring with the bus service. For example, Figure 9 shows the total trip times from February 14th to February 20th. On February 14th, the shortest trip took 897 seconds (at 10 PM of the start time), meanwhile the longest trip took 13,468 seconds (at 12 PM of the start time). The weather conditions were fair on that day, making such an information relevant as a feedback to the transit manager in order to identify the actual cause of these disruptions on the bus service. In contrast, on February 16th the bus service was erratic due to a snowstorm as shown by the different values of the total trips. This information is relevant as a feedback to be provided to the passengers in such a way that they would be able to make a decision to take a bus or to search for another mode of transportation.

To assess the mobility patterns of bus route 51 during the week, we selected 2 trips in the morning, 2 trips in the afternoon, and 2 trips in the evening, with each pair of trips starting at the same time in order to plot the total number of moves and total number of stops and compare the trips (see Fig. 10). By comparing these two aggregation numbers of each trip during an operating date, we can find which trip is congested/unblocked based on pace behavior by reasonably assuming that the higher number of Stops will cause a congested trip. Fig. 10 indicates that bus route 51 is a busy route based on the fact that the average number of Stops (273) in a trip is higher than Moves (189).

2) DIAGNOSTICS ANALYTICAL RESULTS AT THE FOG

Figure 11 illustrates the results obtained from running the clustering algorithm on the aggregated data. As we can notice in this figure, there are a total of 24 clusters found from 419 trips accumulated from a week of data in this experiment. Most of them - which are located in the blue diamond box

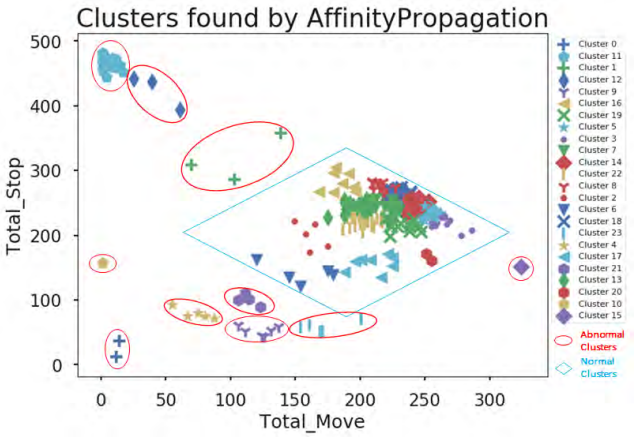


FIGURE 11. Overview of the clusters that were computed at the fog node.

(see Fig. 11) - adhered to the schedule, having ordinary pace behaviors. Therefore, they were labelled as the normal trips based on the identification of the transit managers. However, there were also some trips containing anomalous behaviors. For example, when the total number of Moves is outnumbered by the total number of Stops, this means that the total trip time is much shorter than usual. Hence, these trips were identified as the abnormal trips (shown as red circle of clusters in Fig. 11).

After the clustering algorithm produced its results, a new data feature representing the behavior label (normal/abnormal) was added to the clustering dataset. Therefore we have now a dataset with 7 features *(TripID* $\langle Id_i \rangle$, *Start Time* $\langle St_i \rangle$, *Total_Move* $\langle \overline{M}_i \rangle$, *Total_Stop* $\langle \overline{S}_i \rangle$, *Total Trip Time* $\langle \overline{T}_i \rangle$, *Cluster Label* $\langle \widehat{C}_i \rangle$, *Behavior Label* $\langle Behavior_i \rangle$*)*). This clustering dataset was finally transmitted to our cloud environment for further predictive analytics.

3) PREDICTIVE ANALYTICAL RESULTS IN THE CLOUD

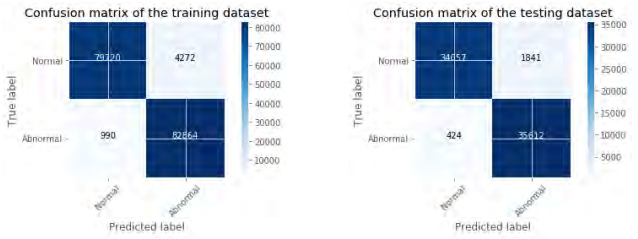
We evaluated our predictive model using 10-fold cross validation. There were a total of 239,780 tuples used to build this model, of which 2/3 are used for the training while the 1/3 remaining tuples are used for the testing. We then

TABLE 4. The evaluation of our prediction model.

	Accuracy	Precision	Recall	F1 Score	AUC	Support
Training Set	0.9686	0.9510	0.9882	0.9692	0.9687	167846
Testing Set	0.9685	0.9508	0.9882	0.9692	0.9685	71934

computed the average accuracy of the model. Table 4 shows the several main evaluation metrics such as accuracy, precision, recall, F1 score, and Area under the ROC Curve (AUC) on both training and testing datasets. In comparison, the accuracy of both sets is very similar, accounting for 96,86% (training set) and 96.85% (testing set). Similarly, the precision score of the training set is not very different from the one of the testing set (95.10% vs 95.08%). Also, while the recall and F1 score are the same, the AUC differed by only 0.02% on both sets.

Figure 12 illustrates the confusion matrices on both sets. As can be seen, the type I and type II errors on both sets are very low, while the predicted condition positive and predicted condition negative values remain very high.



(a) (b)

FIGURE 12. Confusion matrices.

We also studied to find the importance of each feature that affects the predictive results of this model. Therefore, we visualized the importance score of each feature in the training set. Figure 13 indicates some important points to improve our model. First, the latitude, longitude, and the timestamp of a tuple are the 3 most important features that highly influence the predictive results in our model. Second, the first 4 features *(RouteID, route_id_vlr, route_name, route_nickname)* in Figure 13 are almost unimportant to our predictive model. Therefore, they can be removed during the training phase to improve our predictive results since keeping them can introduce some noise in our model.

To evaluate how the accuracy of the prediction model changes as a function of the training set size, we have plotted the accuracy curve as shown in Figure 14. This plot indicates that, not surprisingly, when training data samples increase, the accuracy of our predictive model increases.

Moreover, Figure 15 shows the area under the ROC curve to measure the quality of our predictive model. As can be seen, our predictive model has a very high AUC score (0.97) indicating that it performs well as a general measure of predictive accuracy.

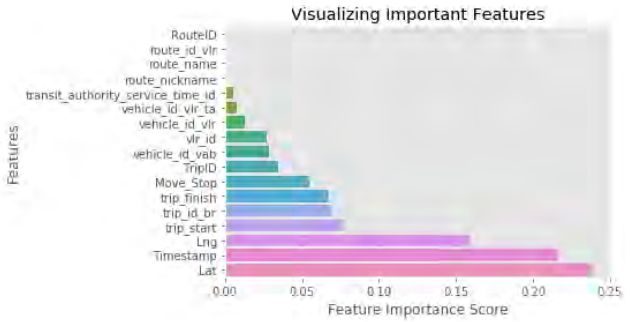


FIGURE 13. List of the most influential attributes in the prediction model.

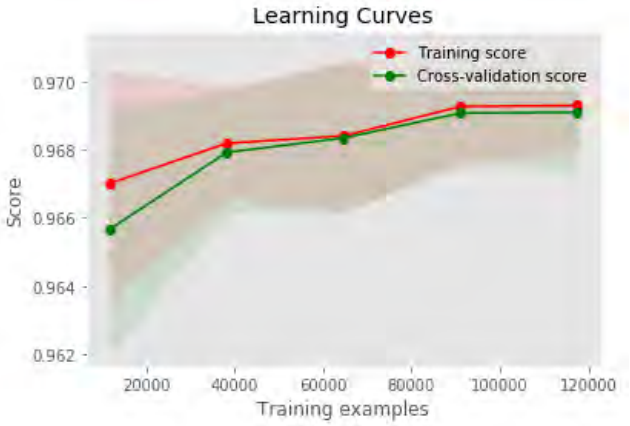


FIGURE 14. Accuracy of the prediction based on number of training items.

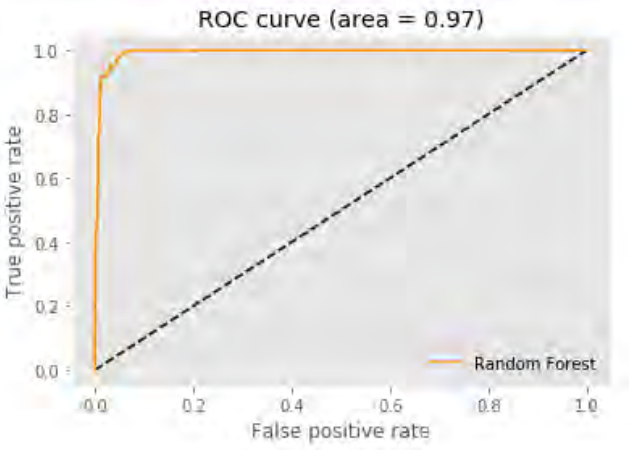
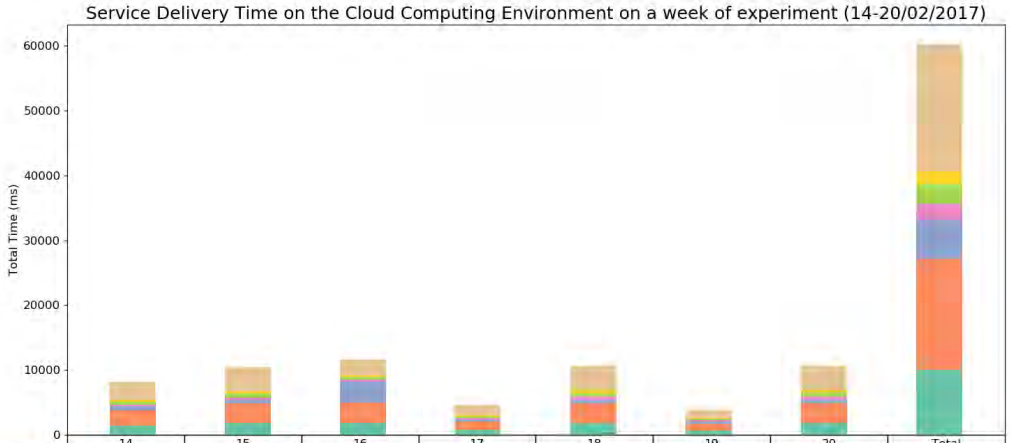


FIGURE 15. Area under the ROC curve of our predictive model.

At the end of the computation in the cloud, the predicted values become the historical feedback for the transit managers, bus drivers, and passengers in order to understand how efficient the bus service is at the transit network level during a long period of time. In this experiment we have only used the data generated by one bus route as an example; however, the predictive model can be applied to the whole transit network. It is also worth noting that our model can



	14	15	16	17	18	19	20	Total
A3: Computing Stop/Move	2842.9	3752.2	2512.0	1609.7	3811.8	1304.3	3797.6	19630.4
A2: Sorting	280.8	370.6	248.1	159.0	376.5	128.8	375.1	1939.0
A1: Extracting Value	431.7	569.8	381.5	244.4	578.9	198.1	576.7	2981.0
P3: Normalizing Missing Value	278.5	347.1	363.0	232.9	551.5	188.7	549.4	2511.0
P2: Remove Duplicated Data	618.9	566.9	3251.0	268.8	510.6	317.0	546.8	6080.0
P1: Eliminate Redundant Data	2380.9	3058.3	3112.0	1316.2	3072.0	1104.5	3076.0	17120.0
I: Ingesting Data Time	1393.6	1790.2	1821.6	770.5	1798.2	646.5	1800.5	10021.0

FIGURE 16. Performance results based on service delivery time.

continuously retrain and update itself with the new datasets that are consecutively sent to the cloud and will be used to offer better predictive results.

4) DISCUSSION

We can evaluate the performance of this proposal using the Service Delivery Time (SDT) metric. SDT is computed as

$$SDT = T_I + \sum_{i=1}^{n} T_{P_i} + \sum_{i=1}^{n} T_{A_i} + T_F$$

where

- T_I: Total time the data streams are ingested in the system
- T_{P_i}: The processing time of the task i^{th} in the system
- T_{A_i}: The analytical time of the task i^{th} in the system
- T_F: The feedback time that the system emits the actionable insights to the users or devices.

Figure 16 illustrates the detailed performance during a week of experiments of 7 tasks to delivery the service in the cloud. They include the ingestion time I, processing time P *(P1: Eliminating Redundant Data, P2: Removing Duplicated Data, P3: Normalizing Missing Value)*, analytical time A *(A1: Extracting Value, A2: Sorting, A3: Computing Stop/Move)*. At the current stage, we have not reached the level of fully computing the feedback time yet, but we could assume that the feedback time will take $\delta(t)$ (ms). Therefore, the service delivery time on our cloud computing environment can be computed by $SDT = T_I + \sum_{i=1}^{3} T_{P_i} + \sum_{i=1}^{3} T_{A_i} + \delta(t)$.

From our experience, it is not worth gathering all the data streams to the cloud then processing and analyzing them in batch since (1) A massive number of data tuples contain errors and inconsist information; almost half of the tuples used in our implementation [70] were deleted. In fact, processing time in Figure 16 accounts for about 40% of service delivery time in the cloud. (2) With such a large amount of unnecessary data arriving in our system, there is a burden on our system in terms of energy consumption, bandwidth contention, and maintenance cost. Therefore, our new Analytics Everywhere framework is a fresh step forward to tackle these issues. Although further empirical experiments at the edge and the fog need to be done in the near future, it is expected that the data ingestion time T_I will be less than shown in Figure 16 because we will move some processing and analytical tasks close to the data source. Also, the data processing time is expected to be reduce as well as the new feedback time $\delta(t') < \delta(t)$ since the data processing and analytical tasks happen close to the data source instead of being sent to the cloud.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents an Analytics Everywhere framework in the context of a composite architectural paradigm that includes edge, fog, and cloud resources for analyzing data streams generated from the Internet of Things. The framework aims to facilitate the design of IoT applications, bringing together in the same conceptual framework the computational capabilities of resources and analytical tasks, taking into account the characteristics of data life-cycles. The framework is based on the idea that IoT applications are convenient to push the computation toward the edge while trying to keep most of the data as close as possible to where it originated. This presents immediate advantages that would be favourable for today's IoT applications. It can support data privacy to a certain extent, reduce the cost to transfer large amounts of data to data centers, and make it possible to transmit feedback quickly to a variety of users. In contrast, it creates

data management issues ranging from data governance, data heterogeneity, to data integrity.

We have applied the proposed framework on an actual real-world scenario for the management of a public transit. Our lesson learned is that if any of the edge/fog/cloud resources of the system architecture are considered in isolation, they would not be able to manage the IoT application, without compromising on functionalities or performance. Still, using a combination of edge, fog, and cloud resources requires careful coordination and a precise allocation of analytical capabilities. That is why the a-priori mapping between analytical capabilities with the appropriate computation resources should be set up by a developer; we do not expect that a user will take this role. Failing to achieve this mapping will have a negative impact on the performance and accuracy of the analytics performed. More research work is needed to determine this impact on over fitting our analytical models.

Despite the fact that PaaS/IaaS models are still an open issue in edge/fog/cloud computing environments in an IoT ecosystem, our prototype has outlined the interchanging major components as being resource capability.

For future research work, we plan to extend the framework by considering security, latency, fault tolerance, and privacy requirements of IoT applications. Regarding the IoT application, we plan to increase the requirements in the cloud resources by adding a data visualization component, such as Kibana or Grafana. Our current prototype is not capable of accommodating dynamic task sharing, but this is definitely our next step. It is important to point out that our Analytical Everywhere framework does not need to be modified to support dynamic task sharing since it relies on the assumption that tasks should be a priori allocated, exploiting the different resources, regardless of workload balancing. Finally, more research is needed to understand the balance between supervised versus unsupervised learning for future reinforcement and federated learning.

ACKNOWLEDGMENTS

The authors would like to thank the Codiac Transpo for providing the transit data. They also appreciate the feedback provided by three anonymous reviewers on the previous version of this manuscript. The authors would like to thank Compute Canada for providing them with the cloud resources.

REFERENCES

[1] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4424–4434.

[2] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečný, S. Mazzocchi, H. B. McMahan, T. Van Overveldt, D. Petrou, D. Ramage, and J. Roselander "Towards federated learning at scale: System design," in *Proc. 2nd SysML Conf.*, 2019, pp. 1–15.

[3] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate IoT edge computing with lightweight virtualization," *IEEE Netw.*, vol. 32, no. 1, pp. 102–111, Jan./Feb. 2018.

[4] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Comput. Netw.*, vol. 76, pp. 146–164, Jan. 2015.

[5] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa, and I. Yaqoob, "Big IoT data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[6] D. Kliazovich, J. E. Pecero, A. Tchernykh, P. Bouvry, S. U. Khan, and A. Y. Zomaya, "CA-DAG: Modeling communication-aware applications for scheduling in cloud computing," *J. Grid Comput.*, vol. 14, no. 1, pp. 23–39, Mar. 2016.

[7] G. F. Anastasi, E. Carlini, M. Coppola, and P. Dazzi, "QoS-aware genetic cloud brokering," *Future Gener. Comput. Syst.*, vol. 75, pp. 1–13, Oct. 2017.

[8] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Netw. Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014.

[9] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[10] P. Banerjee, R. Friedrich, C. Bash, P. Goldsack, B. Huberman, J. Manley, C. Patel, P. Ranganathan, and A. Veitch, "Everything as a service: Powering the new information economy," *Computer*, vol. 44, no. 3, pp. 36–43, Mar. 2011.

[11] A. Botta, W. de Donato, V. Persico, and A. Pescapé, "On the integration of cloud computing and Internet of Things," in *Proc. Int. Conf. Future Internet Things Cloud (FiCloud)*, 2014, pp. 23–30.

[12] M. Díaz, C. Martín, and B. Rubio, "State-of-the-art, challenges, and open issues in the integration of Internet of Things and cloud computing," *J. Netw. Comput. Appl.*, vol. 67, pp. 99–117, May 2016.

[13] B. B. R. Rao, P. Saluia, N. Sharma, A. Mittal, and S. V. Sharma, "Cloud computing for Internet of Things & sensing based applications," in *Proc. 6th Int. Conf. Sens. Technol. (ICST)*, Dec. 2012, pp. 374–380.

[14] J. A. Galache, T. Yonezawa, L. Gurgen, D. Pavia, M. Grella, and H. Maeomichi, "ClouT: Leveraging cloud computing techniques for improving management of massive IoT data," in *Proc. IEEE 7th Int. Conf. Service-Oriented Comput. Appl.*, Nov. 2014, pp. 324–327.

[15] W. Ren, Y. Ren, M.-E. Wu, and C.-J. Lee, "A robust and flexible access control scheme for cloud-IoT paradigm with application to remote mobile medical monitoring," in *Proc. 3rd Int. Conf. Robot, Vis. Signal Process. (RVSP)*, Nov. 2015, pp. 130–133.

[16] Y. Zhang, H. Wang, and Y. Xie, "An intelligent hybrid model for power flow optimization in the cloud-IoT electrical distribution network," *Cluster Comput.*, pp. 1–10, Oct. 2017. doi: 10.1007/s10586-017-1270-0.

[17] A. Mukherjee, H. S. Paul, S. Dey, and A. Banerjee, "ANGELS for distributed analytics in IoT," in *Proc. IEEE World Forum Internet Things (WF-IoT)*, Mar. 2014, pp. 565–570.

[18] P. Karunaratne, S. Karunasekera, and A. Harwood, "Distributed stream clustering using micro-clusters on Apache Storm," *J. Parallel Distrib. Comput.*, vol. 108, pp. 74–84, Oct. 2017.

[19] M. Zaharia *et al.*, "Apache spark: A unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.

[20] P. Patel, M. I. Ali, and A. Sheth, "On using the intelligent edge for IoT analytics," *IEEE Intell. Syst.*, vol. 32, no. 5, pp. 64–69, Sep. 2017.

[21] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the Internet of Things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, Dec. 2016.

[22] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.

[23] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: A platform for Internet of Things and analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*. Cham, Switzerland: Springer, 2014, pp. 169–186.

[24] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[25] N. Harth, K. Delakouridis, and C. Anagnostopoulos, "Convey intelligence to edge aggregation analytics," in *Studies in Computational Intelligence*, vol. 715. Cham, Switzerland: Springer, 2018, pp. 25–44.

[26] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Ed. MCC Workshop Mobile Cloud Comput.*, 2012, pp. 13–16.

[27] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Comput.*, vol. 8, no. 4, pp. 14–23, Oct./Dec. 2009.

[28] A. N. Khan, M. M. Kiah, S. U. Khan, and S. A. Madani, "Towards secure mobile cloud computing: A survey," *Future Gener. Comput. Syst.*, vol. 29, no. 5, pp. 1278–1299, 2013.
[29] Nokia and Intel. (2014). *Increasing Mobile Operators Value Proposition With Edge Computing*. Accessed: Nov. 15, 2017. [Online]. Available: https://www.intel.co.id/content /dam/www/public/us/en/documents/technology-briefs /edge-computing-tech-brief.pdf
[30] G. Lee, W. Saad, and M. Bennis, "Online optimization for low-latency computational caching in fog networks," in *Proc. IEEE Fog World Congr. (FWC)*, Oct./Nov. 2017, pp. 1–6.
[31] J. Clemente, M. Valero, J. Mohammadpour, X. Li, and W. Song, "Fog computing middleware for distributed cooperative data analytics," in *Proc. IEEE Fog World Congr. (FWC)*, Oct./Nov. 2017, pp. 1–6.
[32] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, 2014, pp. 1–8.
[33] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, 2015, pp. 37–42.
[34] Y. Liao, E. de Freitas Rocha Loures, and F. Deschamps, "Industrial Internet of Things: A systematic literature review and insights," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4515–4525, Dec. 2018.
[35] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 855–873, 2nd Quart., 2017.
[36] D. P. Rose, M. E. Ratterman, D. K. Griffin, L. Hou, N. Kelley-Loughnane, R. R. Naik, J. A. Hagen, I. Papautsky, and J. C. Heikenfeld, "Adhesive RFID sensor patch for monitoring of sweat electrolytes," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 6, pp. 1457–1465, Jun. 2015.
[37] J.-H. Huh and K. Seo, "An indoor location-based control system using Bluetooth beacons for IoT systems," *Sensors*, vol. 17, no. 12, p. 2917, 2017.
[38] C. Wang, T. Jiang, and Q. Zhang, Eds., *ZigBee Network Protocols and Applications*. New York, NY, USA: Auerbach Publications, 2014. doi: 10.1201/b16619.
[39] S. Cha, M. P. Ruiz, M. Wachowicz, L. H. Tran, H. Cao, and I. Maduako, "The role of an IoT platform in the design of real-time recommender systems," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 448–453.
[40] R. Aburukba, A. R. Al-Ali, N. Kandil, and D. AbuDamis, "Configurable ZigBee-based control system for people with multiple disabilities in smart homes," in *Proc. Int. Conf. Ind. Inform. Comput. Syst. (CIICS)*, 2016, pp. 1–5.
[41] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," *Future Gener. Comput. Syst.*, vol. 78, pp. 641–658, Jan. 2018.
[42] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, 2018.
[43] W. Yang, M. Wang, J. Zhang, J. Zou, M. Hua, T. Xia, and X. You, "Narrowband wireless access for low-power massive Internet of Things: A bandwidth perspective," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 138–145, Jun. 2017.
[44] R. Sharan Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Exp.*, vol. 3, no. 1, pp. 14–21, Mar. 2017.
[45] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5G networks for the Internet of Things: Communication technologies and challenges," *IEEE Access*, vol. 6, pp. 3619–3647, 2018.
[46] J.-F. van Dam, N. Bißmeyer, C. Zimmermann, and K. Eckert, "Security in hybrid vehicular communication based on ITS G5, LTE-V, and mobile edge computing," in *Fahrerassistenzsysteme*, T. Bertram, Ed. Wiesbaden, Germany: Springer, 2019, pp. 80–91. doi: 10.1007/978-3-658-23751-6_8.
[47] R. Molina-Masegosa and J. Gozalvez, "LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications," *IEEE Veh. Technol. Mag.*, vol. 12, no. 4, pp. 30–39, Dec. 2017.
[48] S. Li, L. Da Xu, and S. Zhao, "5G Internet of Things: A survey," *J. Ind. Inf. Integr.*, vol. 10, pp. 1–9, Jun. 2018.
[49] M. Atzmueller, B. Fries, and N. Hayat, "Sensing, processing and analytics: Augmenting the Ubicon platform for anticipatory ubiquitous computing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput., Adjunct*, 2016, pp. 1239–1246.
[50] K. Nahrstedt, H. Li, P. Nguyen, S. Chang, and L. Vu, "Internet of mobile things: Mobility-driven challenges, designs and implementations," in *Proc. IEEE 1st Int. Conf. Internet-Things Design Implement. (IoTDI)*, Apr. 2016, pp. 25–36.
[51] W. Sun, J. Zhu, N. Duan, P. Gao, G. Q. Hu, W. S. Dong, Z. H. Wang, X. Zhang, P. Ji, and C. Y. Ma, "Moving object map analytics: A framework enabling contextual spatial-temporal analytics of Internet of Things applications," in *Proc. IEEE Int. Conf. Service Oper. Logistics, Inform. (SOLI)*, Jul. 2016, pp. 101–106.
[52] M. R. Vieira, L. Barbosa, M. Kormáksson, and B. Zadrozny, "Usapiens: A system for urban trajectory data analytics," in *Proc. 16th IEEE Int. Conf. Mobile Data Manage. (MDM)*, vol. 1, Jun. 2015, pp. 255–262.
[53] L. F. Herrera-Quintero, K. Banse, J. Vega-Alfonso, and A. Venegas-Sanchez, "Smart ITS sensor for the transportation planning using the IoT and Bigdata approaches to produce ITS cloud services," in *Proc. 8th Euro Amer. Conf. Telematics Inf. Syst. (EATIS)*, 2016, pp. 1–7.
[54] E. Welbourne, L. Battle, G. Cole, K. Gould, K. Rector, S. Raymer, M. Balazinska, and G. Borriello, "Building the Internet of Things using RFID: The RFID ecosystem experience," *IEEE Internet Comput.*, vol. 13, no. 3, pp. 48–55, May/Jun. 2009.
[55] A. Somov, C. Dupont, and R. Giaffreda, "Supporting smart-city mobility with cognitive Internet of Things," in *Proc. Future Netw. Mobile Summit*, 2013, pp. 1–10.
[56] T. Wang, G. Cardone, A. Corradi, L. Torresani, and A. T. Campbell, "WalkSafe: A pedestrian safety app for mobile phone users who walk and talk while crossing roads," in *Proc. 12th Workshop Mobile Comput. Syst. Appl.*, 2012, Art. no. 5.
[57] T. S. López, D. C. Ranasinghe, M. Harrison, and D. McFarlane, "Adding sense to the Internet of Things," *Pers. Ubiquitous Comput.*, vol. 16, no. 3, pp. 291–308, 2012.
[58] A. Krause, A. Smailagic, and D. P. Siewiorek, "Context-aware mobile computing: Learning context- dependent personal preferences from a wearable sensor array," *IEEE Trans. Mobile Comput.*, vol. 5, no. 2, pp. 113–127, Feb. 2006.
[59] B. Qi, L. Kang, and S. Banerjee, "A vehicle-based edge computing platform for transit and human mobility analytics," in *Proc. 2nd ACM/IEEE Symp. Edge Comput.*, 2017, Art. no. 1.
[60] M. Taneja, J. Byabazaire, A. Davy, and C. Olariu, "Fog assisted application support for animal behaviour analysis and health monitoring in dairy farming," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 819–824.
[61] D. Borthakur, H. Dubey, N. Constant, L. Mahler, and K. Mankodiya, "Smart fog: Fog computing framework for unsupervised clustering analytics in wearable Internet of Things," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2017, pp. 472–476.
[62] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, no. 34, 1996, pp. 226–231.
[63] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
[64] L. Hernandez, H. Cao, and M. Wachowicz, "Implementing an edge-fog-cloud architecture for stream data management," in *Proc. IEEE Fog World Congr. (FWC)*, Oct./Nov. 2017, pp. 1–6.
[65] C. Bettini, C. E. Dyreson, W. S. Evans, R. T. Snodgrass, and X. S. Wang, "A glossary of time granularity concepts," in *Temporal Databases: Research and Practice*. Berlin, Germany: Springer, 1998, pp. 406–413.
[66] J. Manyika, "The Internet of Things: Mapping the value beyond the value," McKinsey Global Inst., San Francisco, CA, USA, Tech. Rep., 2015.
[67] H. Cao, M. Wachowicz, and S. Cha, "Developing an edge computing platform for real-time descriptive analytics," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 4546–4554.
[68] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *The J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.
[69] G. Biau, "Analysis of a random forests model," *J. Mach. Learn. Res.*, vol. 13, pp. 1063–1095, Apr. 2012.
[70] H. Cao and M. Wachowicz, "The design of an IoT-GIS platform for performing automated analytical tasks," *Comput., Environ. Urban Syst.*, vol. 74, pp. 23–40, Mar. 2019.

HUNG CAO received the B.Eng. degree in computer engineering from the University of Information Technology, Vietnam National University, Ho Chi Minh City, Vietnam, in 2011, the M.Sc. degree in computer science from the University College Dublin, Ireland, in 2015, and the Diploma degree in university teaching from the University of New Brunswick, Canada, in 2018, where he is currently pursuing the Ph.D. degree with the People in Motion Laboratory (PIML). From 2011 to 2014, he was with the University of Information Technology as a Lecturer of computer science. At PIML, he is working as a Data Scientist. He has been involved in different research projects (MITACS, NSERC Engage projects) collaborating with different companies, including Cisco, Rimot, The Black Arcs, Codiac Transpo, and so on to develop working prototypes that could be used as a template for the company's products in the future. His research interests include big data analytics, the Internet of Things, machine learning, cloud computing, edge computing, and fog computing.

MONICA WACHOWICZ is currently a Full Professor and the NSERC/Cisco Industrial Research Chair in big data analytics with the University of New Brunswick, Canada. She is also the Director of the People in Motion Laboratory, a center of expertise in the application of Internet of Mobile Things (IoMT) to smart cities. Her research interests include fog computing, machine learning on graphs, mobility analytics, stream data management, and the IoMT applications. She works at the intersection of (1) Streaming Analytics for analyzing massive IoMT data in search of valuable spatio-temporal patterns in real-time; and (2) Art, Cartography, and Representations of mobility for making the maps of the future which will be culturally and linguistically designed to provide a greater ''sense of people'' in motion. She is a Founding Member of the IEEE Big Data Initiative and the *International Journal of Big Data Intelligence*. Her pioneering work in multidisciplinary teams from government, industry, and research organizations is fostering the next generation of data scientists for innovation.

CHIARA RENSO received the M.Sc. and Ph.D. degrees in computer science from the University of Pisa, in 1992 and 1998, respectively. She is currently a Researcher with the HPC Laboratory, ISTI-CNR, Italy, where she is involved in trajectory data mining and semantic trajectories. She has authored over 100 peer-reviewed publications. She is a Co-Editor of the book *Mobility Data: Modeling, Management, and Understanding* (Cambridge Press, 2013).

EMANUELE CARLINI received the Ph.D. degree in computer science and engineering from IMT Lucca, in 2012. He is currently a Researcher with the HPC Lab, ISTI Institute of CNR, Italy. His research interests include cloud computing, peer-to-peer applications, and graph analysis. He is currently co-responsible of Matrice, a project in collaboration with the Italian health service.