Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

A Scalable, Research Oriented, Generic, Sensor Data Platform

Joseph Rafferty[1], Jonathan Synnott[1], Chris Nugent[1], Andrew Ennis[1], Philip Catherwood[2], Ian McChesney[1], Ian Cleland[1] and Sally McClean[3]

[1] School of Computing and Mathematics, Ulster University, Northern Ireland, UK, BT37 0QB
[2] School of Engineering, Ulster University, Northern Ireland, UK, BT37 0QB
[3] School of Computing, Ulster University, Northern Ireland, BT52 1SA

Corresponding author: Joseph Rafferty (e-mail: j.rafferty@ulster.ac.uk).

This work was supported by Invest Northern Ireland through the Competence Centre Programs Grant RD0513853 – Connected Health Innovation Centre and the BT Ireland Innovation Centre.

ABSTRACT Research interests spanning numerous domains increasingly rely upon computational systems which can store and process a large *volume* of *variable* data that is stored at high *velocity* – representing a big data problem. This is particularly notable within the domain of ubiquitous and pervasive computing. This domain increasingly relies on storage and retrieval of sensor data to enable outcomes such as predictive analytics and activity recognition. Several current big data platforms exist; however, they have a range of deficiencies including lack of generic interoperability with agnostic sensors and an absence of features supporting academic research. Due to these deficiencies a custom, research oriented, high performance, big data platform was devised and implemented. This platform is called *SensorCentral* and is presented within this manuscript. SensorCentral provides a framework which enables interoperability with a large range of agnostic sensor devices whilst simultaneously providing features which support research. Research supporting features include; facility to define experiments, ability to annotate experimental instances via purpose-built mobile applications, integrated machine learning functionality, facility to export data sets, rule-based classification and an extensible platform. The flagship implementation of this platform has been in operation for over 28 months within a University research group and has been successfully integrated with a range of sensors from a variety of manufacturers. This implementation currently stores over 850 million records and has been central to several research and industrial projects. Future work will integrate this platform into the Open Data Initiative enabling collaboration with the international community of researchers.

INDEX TERMS Data analysis, Data storage systems, Database systems, Internet of Things, Machine learning, Sensor systems, Wireless sensor networks, LoRa, Open Data Initiative, Research tools

I. INTRODUCTION

Large volumes of data are increasingly becoming central to a variety of research interests. Notably the domain of ubiquitous and pervasive computing is reliant on data generated from sensing elements [1]–[3]. Research interests in these domains include: activity recognition, sensor-based supported safety solutions, environmental monitoring and enabling industry 4.0. Storing, processing, exploiting and presenting such sensor data is a big data problem. Big data problems carry three key characteristics, which are summarized as the three V's [4], [5]. These three V's are:

- Variety: the data to be stored varies greatly
- Volume: a large quantity of data is present
- Velocity: data records are stored at a high sample rate

Typically, the research interests in these domains incorporate a range of sensor device types from a range of vendors. These heterogenous devices produce data dictated by what they sense. This represents the *variety* characteristic in data generated. Current sensor solutions can generate a great *volume* of data which must be adequately catered for. In addition, sensor data is ideally sampled at the highest possible rate in order to provide a more valuable data set for research efforts – this represents the high *velocity* aspect of this problem.

This big data problem is best illustrated by discussing two research efforts within these domains. Specifically, the studies described in [6], [7]. These studies have produced behavioral and fall detection services that achieve their goals via environmentally deployed thermal vison sensors. These thermal vison sensors perceive the world through a low-resolution grid of emissive thermal readings. In their current configuration, these sensors employ a sample rate of 10Hz. Each sampled thermal frame is on average 4 kilobytes in size. Given the frame size and the sample rate, a single thermal vison sensor can generate 144 megabytes of data an hour and approximately 3.5 gigabytes a day.

In order to provide adequate coverage of a domicile, the intended deployment environment, multiple sensors are required – compounding the problem. In addition, these solutions require other sensor records, such as those representing annotations, to be stored. Furthermore, processed sensor data, describing the persons detected in the scene need to be stored in conjunction with the raw thermal frames and annotation data. Additionally, during the development of these solutions sensor data from a smart floor was used as a ground truth for models and computer vison processing modules. As such, this solution incorporates a *variety* of sensors which produce a large *volume* of data at high velocity.

A number of solutions to store, exploit and process such data exists, however, they are not research oriented and so have some limitations. As determined by the requirements of the research group which originated this platform, research-oriented solutions require several features including:

1. agnostic sensor integration
2. the ability to define experiments, related information and researchers
3. the ability record instances of experiments
4. annotation interfaces for experiments
5. the ability to export records from experiments and experimental instances
6. integration of machine learning functionality
7. the ability to forward sensor data to independent processes
8. a flexible, extensible platform with a modular interface

A sensor data platform was devised and developed to address these deficiencies. This platform is called *SensorCentral* and aims to offer features and functions which will aid research efforts.

The remainder of this paper adopts the following structure: related works are explored in Section II, the developed platform is presented in Section III, some current use cases are presented in Section IV, and Section V provides concludes the paper and presents some planned future work.

II. RELATED WORK

A multitude of solutions facilitating storing and querying sensor data at the big data scale exist. However, they have some technical and functional deficiencies. Notably, the majority of solutions have little or no support for research-oriented functionality [5], [8]–[19]

Specifically, no platform adequately supports the eight research-oriented features that were identified by the candidate researchers and enumerated within the introduction Section.

Beebotte [15] is a cloud-based platform that supports storage and querying of IoT/sensor data. The platform operates on Amazon Web Services, offering redundant and scalable hosting. Communication is supported though REpresentational State Transfer (REST) [20], Message Queue Telemetry Transport interfaces (MQTT) [21], [22] and WebSockets [23], [24]. This platform doesn't provide any extensive analysis functionality. Additionally, it requires a commercial license, which may not be ideal for use by researchers in all cases.

Bonomi *et al.* [16] proposed a 'fog computing' approach supporting storing, querying and processing sensor data. This approach supports scalable storage in addition to integration of real-time analytics. Although showing promise, this approach stores data within silos. Employing such information silos greatly reduces the ability to query across the entire data set.

Cecchinel *et al.* [17] produced an architecture to store a large quantity of sensor data. This approach incorporates heterogenous sensors which produce data at a high velocity. Data can be accessed via a REST interface by consumer applications. The core data storage component of the is based upon a document-oriented database, MongoDB. The platform has promise but a number of deficiencies related to research oriented functionality and carries a potential performance bottleneck due to its reliance of MongoDB [25]–[29] when considering sensor/time series data only.

Cheng *et al*. [18] produced a sensor data platform, named CiDAP, that was designed to support realization of smart cities. Specifically, the candidate test smart city has a population of over 180,000 people and contains more than 15,000 sensors. The core data storage component incorporated a document database, CouchDB, and the Hadoop platform. Notably, the authors did not consider incorporation of any Time-Series DataBase (TSDB), potentially reducing the scalability of their approach [25]–[29]. The system, however, is a proven platform that has been successfully deployed. Deficiencies include a lack of research-oriented features.

Kx for Sensors [19] is a commercial sensor data platform with origins in managing data from the stock market. Kx offers a scalable, agnostic, solution that incorporates visualization functions, distributed queries, analytics and incorporation of machine learning components. The core storage engine of Kx for Sensors is a time-series database, kdb+. As emphasized by previous evaluations [25]–[29] and considering the design goals of TSDBs, this is an appropriate choice for storing this type of data. Kx does not sufficiently support research-oriented features and has limitations related to commercial licensing.

Lee *et al.* [9] produced a sensor data platform which was focused on storing data related to railway systems and related infrastructure. This platform provides real time analysis of this data to offer services such as predictive maintenance and asset tracking. Although this is called a "universal sensor platform", this name is a misnomer as it only supports a set of specific sensors and does not offer agnostic function. Furthermore, there is limited discussion of the storage strategy or its scalability as a big data platform.

openHAB [10] is an open source sensor platform which offers agnostic device integration. It was initially devised as a solution to converge differing home automation standards and technologies. openHAB supports integration of a large array of sensors, actuation interfaces, protocols and online services. Whilst primarily focused on automation and control, it supports persistence of sensor data within a variety of storage engines, including TSDBs. Although this platform supports an extensive array of sensors/technologies and is extremely mature, it has some deficiencies, particularly when applied to research activities.

Sowe *et al.* [12] proposed a platform to enable storage of a large quantity of sensor data from a variety of heterogenous devices. The core data storage component of this platform is based upon a document-oriented database and a relational database, MongoDB and MySQL respectively. The technologies incorporated into the data storage component have, however, do not offer adequate performance or scalability when processing large quantities of sensor data [25]–[29].

Thingspeak [11] is a cloud-based platform that supports storage and querying of IoT/sensor data. This platform provides agnostic integration with sensors and communicates via REST or MQTT. This platform incorporates a MatLAB based component facilitating analysis and presentation of data. This component is a reduced functionality, web based, implementation of MatLAB. These MatLAB scripts can be scheduled to provide some automated analysis. It is notable that this analysis function is relatively high latency. Finally, this platform is a commercial pursuit and so may not be suited for research efforts.

Notably, the evaluated sensor data platforms do not adequately support the desired research-oriented features identified by the requirements of the candidate research group. In particular, Kx, CiDAP and openHAB lack the ability to support definition of experiments, ability to forward datasets to independent process and lack of tools/ability to annotate datasets.

In order to address the deficiencies, a custom solution was produced. This solution has built upon the knowledge provided by previous solutions [5], [8]–[19], [25]–[29] to provide a sensor agnostic, research oriented and scalable platform. This platform is called *SensorCentral* and is detailed in Section III.

III. A SCALABLE, RESEARCH ORIENTED, GENERIC, SENSOR DATA PLATFORM

The developed, scalable, generic, sensor data platform supports integration with diverse range of heterogenous sensors produced by a variety of manufacturers. Additionally, the platform integrates the eight research-oriented features indicated by the requirements of the intended research group. This platform supports a modular, web-based interface which supports sensor data visualization in addition to data and research management. Finally, this platform has been developed to offer scalable high performance incorporating proven, open-source, technologies.

Discussion of a range of supported sensors is presented in Subsection A. The approach taken to support for generic sensors is detailed in Subsection B. The architecture of the platform, and modular interface, is presented and discussed in Subsection C. The integration and availability of research-oriented features is presented in Subsection D.

A. CURRENTLY SUPPORTED SENSORS

Currently, this platform has been integrated with over 20 classes of device produced by over 20 different manufacturers. These sensors communicate over an array of communications protocols including Bluetooth, custom Radio Frequency (RF), LoRaWAN, Wi-Fi, Ethernet, IEEE 802.15.4 and Z-wave. A subset of currently supported sensors is presented in Table I. Notably, this list reflects the sensors that have been fully integrated into the platform, therefore it is not exhaustive and can be expanded.

TABLE I
A SUBSET OF THE SENSORS SUPPORTED BY SENSORCENTRAL

Sensor Class	Manufacturer	Communication Protocol
Accelerometer	Bosch Microchip Sun Microsystems Texas Instruments	I^2C with Wi-Fi LoRaWAN IEEE 802.15.4 Bluetooth
Air Quality	Elsys	LoRaWAN
Analogue Voltage	Adeunis	LoRaWAN
Bluetooth Beacon	Various	Bluetooth
Contact Sensor	Everspring Nexa Tynetec	Z-Wave Custom RF (433MHz) Custom RF (169MHz)
GPS Location	Adeunis RF GlobalSat Ulster University	LoRaWAN LoRaWAN Wi-Fi/4G (via App)
Humidity	Adeunis RF Microchip Texas Instruments Ulster University	LoRaWAN LoRaWAN Bluetooth Wi-Fi
Inertial Measurement Unit	Slever Technologies	Bluetooth/ USB
Light Intensity Meters	Sun Microsystems Texas Instruments	IEEE 802.15.4 Bluetooth
Magnetometer	Bosch Texas Instruments	Wi-Fi/Bluetooth Bluetooth
NFC Tags	Various	Wi-Fi/4G/Ethernet
Passive Infra-Red Motion Sensors	Belkin Elsys Nexa	Wi-Fi LoRaWAN Custom RF (433MHz)
Power Usage Monitor	Belkin NKE Watteco	Wi-Fi LoRaWAN

Push button	FLIC	Bluetooth
Smart Floor	Future-Shape GMBH	Custom RF (868MHz)
Sound Pressure	Ulster University	Wi-Fi/ Bluetooth
Temperature: Ambient / Immersive	Adeunis RF	LoRaWAN
	Microchip	LoRaWAN
	Sun Microsystems	IEEE 802.15.4
	Texas Instruments	Bluetooth
	Ulster University	Wi-Fi
Thermal Vision	Heimann GMBH	Ethernet
	IOTTech	USB
	Ulster University	Wi-Fi & Bluetooth

Other supported sensors include blood pressure monitors, pulse oximeters, water leak detectors, smart watches and weight/body fat scales.

Notably, two generic sensor connectors have been developed to *seamlessly* support integration of all sensors deployed to two complementary platforms. These platforms are the things connected network [30] and the RaZberry z-wave server [31].

The things connected is a UK wide, IoT communications network. This uses LoRaWAN technology to communicate with IoT devices on a wide range (regional/national) scale. SensorCentral has a native integration endpoint which enables *all* and *any* sensors deployed to this national network to automatically store data within SensorCentral. Integration of these sensors with SensorCentral is a seamless process which requires no additional effort beyond what is normally required to enroll a device on the things connected network. LoRaWAN devices generate a low quantity of data on a per device level, however, the base stations support tens of thousands of devices, therefore introducing an aggregate effect wherein a large volume of variable data is generated at a high velocity.

The RaZberry z-wave server is a listener for heterogenous sensors which communicate locally using the Z-wave protocol. A connector has been written for SensorCentral to automatically relay *all* and *any* data from sensors that are enrolled to a RaZberry instance.

The approach for generic sensor support that enables support of this range of sensors is presented in the following subsection.

B. ENABLING GENERIC SENSOR SUPPORT

A key feature of this platform is its ability to support a wide range of sensor types produced by a variety of manufacturers, as presented in the previous Subsection.

This generic sensor support is facilitated through two key architectural decisions, these are presented below:

1. a strategy to assign globally unique sensor IDs was devised and incorporated
2. a generic sensor record format was adopted incorporating schema on read principles

A strategy to derive globally unique sensor IDs ensures that steams of sensor data do not erroneously contain values from unexpected sources, specifically other sensors.

Typically, sensor manufacturers provide ‘unique’ identifiers for sensors they produce. However, due to lack of global coordination, these identifiers may conflict with those assigned by other sensor manufacturers. It is feasible that manufacturer X could produce a contact switch sensor with an identifier of 000001 and manufacturer Y could also produce a contact switch sensor with that same ID. If these sensors were both deployed to an environment, there would be no way to discern data that they generate based upon the manufacturer assigned ‘unique’ identifier alone.

To address this limitation and cater for potential conflicts, a derived global identifier would need to be produced. It is assumed that identifiers are unique within specific classes of sensors produced by a manufacturer. Therefore, it is possible to leverage this assumption to create a derived Universally Unique IDentifier (UUID).

These UUIDs would extend the ‘unique’ identifier provided by the sensor manufacturer by appending sensor class and manufacturer identifiers. For example, a thermal vision sensor produced by Heimann GMBH has the UUID of t0097ff000758_1_2. In this example the manufacturer assigned sensor ID is t0097ff000758, the sensor class is 1 indicating a thermal vison sensor and the sensor manufacturer is 2 indicating Heimann GMBH.

Typically, such UUIDs are generated through sensor listener software. These sensor listeners enroll sensors to the SensorCentral platform by producing a sensor metadata record. After sensors are enrolled these listeners then proceed to relay sensor data.

Generally, these listeners read sensor data and convert it to the sensor record format which is used by SensorCentral. Typically, this transmits sensor metadata and sensor data via REST through a JavaScript Object Notation (JSON) formatted message. The sensor metadata used to enroll and represent enrolled sensors is presented in Table II below.

TABLE II
THE FORMAT OF SENSOR METADATA RECORDS WITHIN THE PLATFORM

Value	Data type	Description
associatedEnv	64-bit Integer	Optional: A value indicating the associated environment, this is a pointer to the ID of a record within the associated environments roster.
deviceMfg	64-bit Integer	A value indicating the associated manufacturer, this is a pointer to the ID of record within the manufacturers roster.
exampleData	String	Optional: Some example data, this provides a reference to end users.
forwardParamsList	Array of objects	Optional: This is an array of forwarding rules. These indicate a target system to forward sensor data to. Each rule details transmission protocols, authentication options and destination parameters.
label	String	Recommended: This is a relatable label identifying a sensor, such as “Front door”
location	String	Recommended: This is a relatable label identifying the location of a sensor, such as “Apartment 23 – BT6 A92”

relatedMime Resources	String	Optional: This is a field which indicates if the sensor provides some standard mime media types as output, such as audio/x-mpeg-3.
sensorClass	64-bit Integer	A value indicating the associated sensor class, this is a pointer to the ID of a record within the sensor class roster.
sensorID	String	The manufacturer provided sensor ID.
UUID	String	The globally unique UUID of the sensor.

Generally, such metadata records are transmitted and manipulated as JSON representations. The JSON representation of the metadata for a contact sensor stored within this system is presented in Figure 1. This sensor has its associated environment set to a "do not associate" record (-1) and has no example data, related mime type or forwarding rule.

```
{
    "associatedEnv":-1,
    "deviceMfg":10,
    "exampleData":null,
    "forwardParamsList":[],
    "label":"J27/Kitchen Door",
    "location":"16J27 - Jordanstown",
    "relatedMimeResources":"",
    "sensorClass":3,
    "sensorID":"19804566",
    "UUID":"19804566_3_10"
}
```

FIGURE 1. A sensor metadata record, represented in JSON, as consumed and produced by the *SensorCentral* platform.

In addition to the metadata, samples generated from sensors are stored. The sensor data format used within the platform is presented in Table III below.

TABLE III
THE FORMAT OF SENSOR DATA STORED WITHIN THE PLATFORM

Value	Data type	Description
blobJson	String	An escaped string containing a JSON based representation of the sensor data. This is typically used when sensor data is non-binary. This is to be processed on a Schema on Read basis.
deviceMfg	64-bit Integer	A value indicating the associated manufacturer, this is a pointer to the ID a of record within the manufacturers roster.
eventCode	64-bit Integer	An enumeration indicating the state of the sensor. For simple binary sensors, this may be 0 or 1, indicating off or on. For more complex sensors this may be 101, indicating that the blobJson should be referred to.
sensorClass	64-bit Integer	A value indicating the associated sensor class, this is a pointer to the ID of a record within the sensor class roster.
sensorUUID	String	The manufacturer provided sensor ID.
timeStamp	Float	The "UNIX-time" based timestamp of the sensor reading. This is a high-resolution value in *nanoseconds*.
uID	String	The globally unique UUID of the sensor.

It is notable that this approach integrates a schema on read strategy for complex sensor data. This is a common approach leveraged in big data systems [5], [32] and is exemplified by the "data lake" approach to big data storage [14], [33].

Generally, such sensor data records are transmitted and manipulated as JSON representations. The JSON representation of the sensor data stored within this platform is presented in Figure 2. This sensor record is a sample generated from a power usage monitor. This sensor generates complex data and so uses event code 101 which by convention indicates read the *blobJson* element. The *blobJson* element, in this case, encapsulates the power related metrics, such as: current state, IP address and friendly name supplied for voice-based assistant, such as an Amazon Echo.

```
{
    "blobJson":
     "{ 'serialNumber':'221649K1200190',
        'currentPower': 0,
        'ipAddress': '192.168.0.101',
        'todayKWH': 0.057829501156589996,
        'todayOnTime': 0,
        'todayStandbyTime': 0,
        'friendlyName': 'Switch',
        'currentState': 0}",
    "deviceMfg":11,
     "eventCode": 101,
     "sensorClass": 4,
     "sensorUUID":"221649K1200190",
     "timeStamp":1.486063588902671E9,
     "uID": "221649K1200190_4_11"
}
```

FIGURE 2. A sensor data record of a power usage monitor, represented in JSON, as consumed and produced by the *SensorCentral* platform.

The generic sensor data records and metadata records are stored and presented by the SensorCentral platform. This platform provides a high performance, low-latency, scalable storage engine based upon proven open-source technologies. Additionally, this platform provides a modern, modular web interface supporting management and visualization. Further Information on this platform is presented in Subsection C.

C. SENSORCENTRAL PLATFORM ARCHITECTURE

Central to the design of this platform is a scalable, high-performance, low-latency storage engine. This storage engine incorporates two proven and open-source database systems - MongoDB [34] and InfluxDB [35].

These databases were chosen following a performance evaluation process. This process compared several databases including: Apache Cassandra, Apache HBase, InfluxDB, MongoDB, Microsoft SQL, Oracle Database and Oracle MySQL. During this evaluation, the Hadoop platform was not directly considered as it did not prioritize low-latency operation and introduces a complex, heavyweight, distributed solutions beyond what is required for this platform [36]–[39]. Additionally, other databases that aren't optimized for performance or are unsuited to storage of big sensor data, such as semantic stores and graph databases, were not considered [40]–[42].

The set of database systems were subject to performance testing. This testing focused on how quickly sensor metrics could be generated from raw sensor data.

The sensor metric generation process used a Java program which integrated with each database. When testing each database *only* the associated connection logic was changed. The construction of these database connectors adhered to best practices for each platform, as detailed within developer documents.

The raw sensor data was a standard set used across all testing. This data was generated from two thermal vision sensors deployed to simulated kitchen and living room environments. These sensors were configured to sample the environment at a rate of 10Hz and, across two days, over three million samples were captured.

These raw sensor data records were loaded into each type of database and a standard metric generation request was applied. The database was hosted and accessed locally, reducing network transmission overhead and related uncertainty/variability.

In testing, InfluxDB had shown to have the best average performance and MySQL was shown to have the worst average performance. When integrated with InfluxDB the metric generation took less than 10 seconds. In comparison, when integrated with MySQL, the process took longer than 23 minutes to perform this standard task.

InfluxDB is a TSDB, a type of database optimized for storage and retrieval of data that uses timestamps as an index, such as stock trading records or sensor data [43] . The timestamp index is unique and in order to support a large volume of data it is *high resolution*. The indexed timestamp within InfluxDB has an accuracy of nanoseconds. It is infeasible for there to be a collision between records with an index of such high resolution. This class of database is designed to handle a high *velocity* of sequential read and write operations in a high-volume manner. Such TSDB systems are generally limited by throughput of Input/Output interfaces on their hosts opposed to computational or memory-based limitations.

InfluxDB enables arbitrary data to be stored for each stored record. Additionally, InfluxDB clustered operation therefore enabling scalable operation [44]. These characteristics of InfluxDB, and TSDBs in general, make it suited to storage of sensor data in a purpose built big data platform.

TSDB systems are highly sequential and are not optimized to support random insertion, deletion or modification of records stored. Considering this limitation an independent database is needed to store and manipulate non-sensor data. Such non-sensor data includes such as sensor metadata, user profiles, API keys and experimental metadata. As such, another database would be required to support these types of records within the platform [44].

To store these other records, MongoDB was selected. This database incorporates the document paradigm [27] where records are modelled as documents that may be created, updated, read and deleted. In MongoDB the documents are stored in the BSON format [45] and represented by JSON. MongoDB has been developed to be scalable and so is suited for use within this platform. MongoDB was the third best performing database within the environment, however, it was chosen due to its proven scalability and incorporation of a storage model which is suitable for this application. This scalability has been proven and widely accepted in recent years, however, early iterations of this platform did not scale well especially when write operations were required.

InfluxDB and MongoDB are open source and thus don't require licensing fees to use. Additionally, this reduces risk associated with being dependent on a vendor which may cease support for the database or surreptitiously change the terms of service. Additionally, the royalty free nature of these databases enables scalability without any financial considerations. However, licensing fees may be paid for advanced support and tools [34], [35].

This storage engine was subsequently integrated into the overall *SensorCentral* platform. The architecture of this platform is presented in Figure 3.

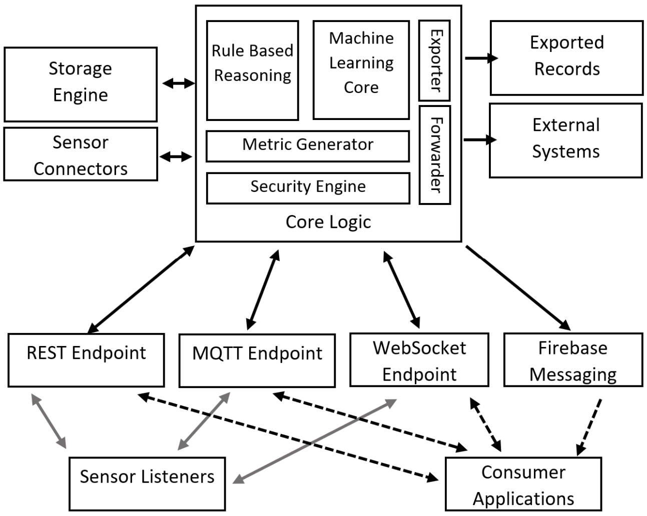


FIGURE 3. The architecture of the SensorCentral platform.

In addition to the storage engine, the SensorCentral platform has a number of notable components, described below.

The storage engine is connected to a core logic component. This core logic contains a number of elements including a security manager, metric generation routines, a rule-based reasoning engine, a machine learning core and record exporters and forwarders.

The security manager is used to provide authentication/verification and offer cryptographic services.

Authentication services include:

1. challenge/handshake authentication
2. API key provision and verification (512-bit)
3. user profile management
4. verification of API/login token rights

Cryptographic services are offered by the bouncy castle library [46]. This primarily provides SensorCentral with a high-quality pseudo random number generator and suite of cryptographic functions.

Server-side metric generation logic is present enabling consumer applications to offset their processing to a SensorCentral node, reducing network traffic and reducing processing time due to the benefits of data locality.

A rule-based engine may be used to classify windows of metrics. Rules support a number of logical operators including: *greater than, less than and equal to.* Additionally, negations of these operators are supported. Rules can be specified using a web interface [7] or through a Java API.

Machine learning services are provided within the SensorCentral platform using the Neuroph framework [47]. This framework offers implementations of algorithms which are compatible with those present within the Weka tool [48] thus enabling prototyping of compatible and predictable solutions using a graphical interface.

Additionally, this core logic offers sensor forwarding and exporting functionality. Sensor forwarding logic enables live records for nominated sensor data groups/records to be forward to specified external systems. This currently supports forwarding this data via REST calls or WebSockets. The ability to forward such sensor records enables SensorCentral to function as a router of sensor data. This facilitates independent systems to be produced without affecting other efforts, reducing the risk of compromising other systems and removing a potential route to produce information silos.

Exporting functionality enables full records of experiments and associated sensor data to be exported into single documents/datasets. Currently SensorCentral can produce a single JSON document containing all associated sensor and experimental data. The ability to directly export such datasets to the Open Data Initiative (ODI) [49] is being actively investigated and is under development.

Finally, this core logic is also available as a Java library. This Java library enables external solutions to integrate with a SensorCentral instance and process data/exploit data within without using shared resources on hosted instances.

This core logic is coupled with three endpoints which enable integration with other components, such as sensors, web interfaces and mobile applications. These endpoints are REST based, MQTT based and WebSocket based.

The REST based endpoint is the primary web service endpoint which enables applications to interact with the platform, these applications include web apps, sensor listeners, mobile applications and other consumer software. This endpoint is based upon stateless Java EE technology and so offers a scalable operation. In addition, endpoints/connectors for MQTT and WebSockets are presented. MQTT is a pub-sub based interface for consumer applications and integration with sensor listeners. WebSockets offer a further interface for consumer applications and sensor listeners. Also, the core logic contains libraries and code, to integrate push messaging services for mobile and desktop applications through the cross-platform Firebase Cloud messaging platform [50].

Sensor listeners read low level sensor data from sensor devices, convert the data into the JSON format required by SensorCentral and subsequently transfer the JSON representation to SensorCentral.

Akin to sensor listeners are sensor data connectors. Sensor data connectors facilitate integration with other sensor platforms and networks. Currently two of these connectors exist, one integrates all sensor data from RaZberry servers and the other integrates the things connected network.

The RaZberry connector enables seamless integration with Z-Wave based sensors. This connector operates by relaying all and any data from sensors that are enrolled to a RaZberry instance.

The things connected network connector seamlessly integrates sensor data from the UK-wide things connected LoRaWAN IoT network. Current coverage of this network in the Northern Ireland region is presented in Figure 4 where each pushpin represents a LoRaWAN base station with up to 25km range.

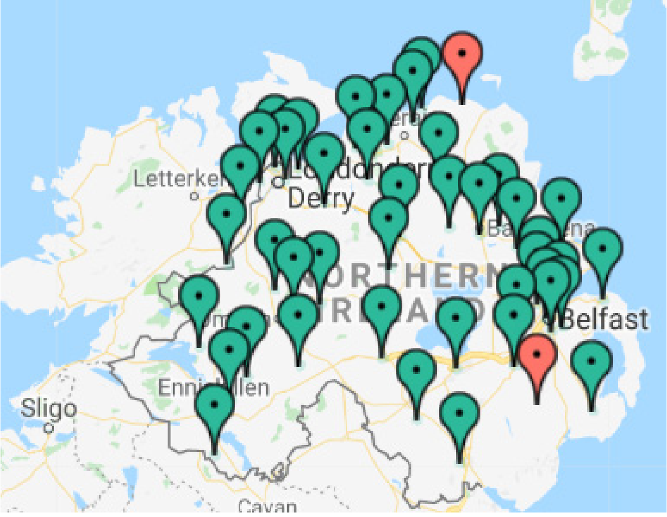


FIGURE 4. A regional deployment of the UK-wide things connected IoT network. Each pushpin represents a base station with an interaction range of up to 25km. SensorCentral offers seamless integration with devices on this network. Base stations in green are active, those in red are under maintenance.

A modular web application exists to manage SensorCentral functionality, this includes: management of sensors, visualization of sensor data, management user access, management of API keys, management experimental setup and experimental instances. This web app has been developed using modern technologies, specifically AngularJS [51] and the bootstrap [52] front-end frameworks.

The developed web app is extensible and modular enabling researchers and developers to produce modules which can be easily integrated into an overall platform. This extensibility and modularity also offers the capability to adapt an existing web app into a smaller module – enabling rapid production of a purpose-built interface with reduced complexity. Such a purpose-built interface may allow reuse of software to produce a dedicated web app for third parties to use.

This modular web interface can be delivered from servers other than the REST endpoint, due to a relaxation of Cross-Origin Resource Sharing [53] restrictions on the REST endpoint. This can be reenabled for marginally stronger security at the expense of greater convenience and flexibility.

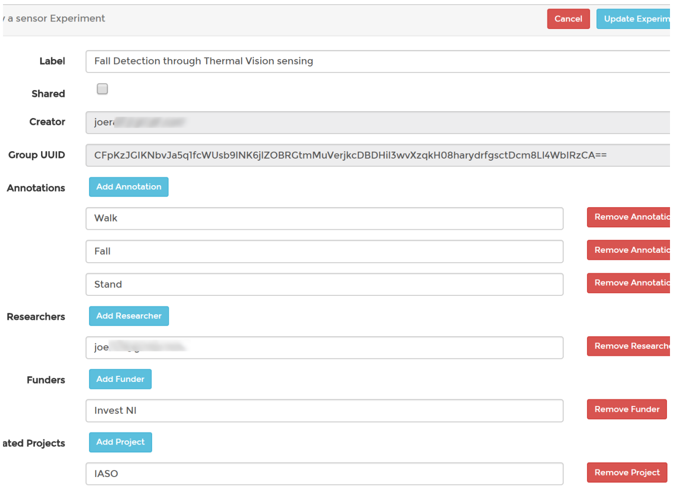
In addition to a scalable, modular platform, SensorCentral offers a number of research-oriented features. The integration and availability of research-oriented features is presented in Subsection D.

D. SENSORCENTRAL FEATURES SUPPORTING RESEARCH ACTIVITIES

SensorCentral has a number of features which can ease research efforts. These features satisfy the 8 requirements that were previously outlined in this manuscript.

Requirement 1, agnostic sensor integration, has been satisfied as detailed previously and within Subsection A and B of this Section.

Requirement 2 has been satisfied through the ability to define metadata related to experiments within the standard SensorCentral web app. This interface is shown within Figure 5.



/ a sensor Experiment

Cancel | Update Experim

Label: Fall Detection through Thermal Vision sensing

Shared: □

Creator: joer[redacted]

Group UUID: CFpKzJGIKNbvJa5q1fcWUsb9INK6jIZOBRGtmMuVerjkcDBDHil3wvXzqkH08harydrfgsctDcm8Ll4WbIRzCA==

Annotations: Add Annotation

Walk — Remove Annotatic

Fall — Remove Annotatic

Stand — Remove Annotatic

Researchers: Add Researcher

joe[redacted] — Remove Researche

Funders: Add Funder

Invest NI — Remove Funder

ated Projects: Add Project

IASO — Remove Project

FIGURE 5. The experiment definition interface offered by the standard SensorCentral web app.

This interface allows researchers to provide labels for experiments, enabling control of sharing data within research groups, specification of annotations to be consumed by the experiment manager mobile app and web app, association of logical sensor groupings, definition of researchers, specification of funders and specification of associated projects.

Once defined, experiments may be used as templates for experimental instances. These instances are managed by either the web app or experiment manager mobile app. These instances clone the experiment metadata on creation and store the date and time of an instance being performed. These may be shared with other SensorCentral users, if desired by the researcher this satisfies requirement 3.

In addition to providing management of experimental instances, the experiment manager mobile application enables annotation of these instances through the labels defined within the experimental setup. The interface of this app is shown in Figure 6. Notably, the SensorCentral experiment manager app was developed using the cross platform Ionic framework thus supporting most modern smart device platforms.

In addition, a separate NFC annotation app exists to support intuitive annotation. In this method of annotation, NFC tags are affixed to an environment. Once deployed the researcher configures them with an associated annotation – initializing the tag. Once initialized users/researchers with the app installed may simply tap a smart device to the tags in order to generate and store an annotation. NFC based annotation is further detailed in [54].

All annotations are time synchronized to the SensorCentral instance easing an aspect of dataset annotation. These annotation features satisfy requirement 4.

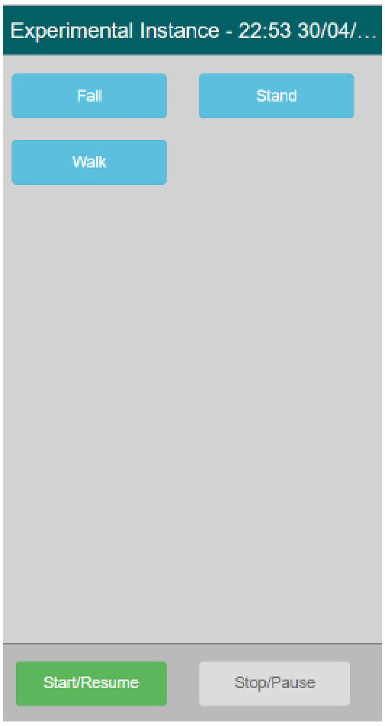


FIGURE 6. The developed SensorCentral experiment manger app showing the annotation interface.

This platform supports exporting data from experimental instances into a single JSON file, facilitating simplified sharing of sensor data with the research community. This format is currently bespoke export, however, integration of the eXtensible Event Stream format and integration with the ODI are actively being explored and pursued [49]. This satisfies requirement 5.

Machine learning functionality is currently offered through integration of the Neuroph library, as detailed previously satisfying requirement 6. However, future work will investigate integration with the TensorFlow platform to benefit from its optimized algorithms and rapidly expanding capabilities [55].

SensorCentral natively offers the ability to forward sensor data to independent processes and systems, easing prototyping – satisfying requirement 7. Forwarding of such sensor data may be reduced via filters. These filters may reduce forwarded data to that of logical units, on a sensor, experimental or sensor grouping level. In addition to reducing sensor data to that of logical units, time-based windows for these logical units may be specified. Also, a window may be specified to filter data to the obtain the most recent data generated by such logical units. In addition to supporting a reduction of/filtering data when forwarding, the platform enables these filters to be applied when accessing data such as when querying via a REST endpoint or leveraging the Java based library.

SensorCentral is an extensible and modular platform offering rapid integration with sensors, supporting other integration with other sensor platforms and enabling modular interfaces – satisfying the eighth requirement.

Additionally, SensorCentral enables easier sensor deployment and configuration though use of supplemental NFC tags that can be enrolled with a sensors UUID in order to modify related parameters, such as location and label.

Finally. SensorCentral is a mature project which has been used within a number of different projects. These are explored in Section IV.

IV. CURRENT USE CASES

This platform has been in place for over 28 months and holds over 850 million records. This is currently central to over 13 projects. Some of these are detailed in this Section.

One study [56] has offered a platform to model sensor placements within environments. This model is then used to simulate data generation. An extension to this has integrated SensorCentral into its visualization engine, providing a real time representation of the state of a sensor as determined by real sensor data from deployed devices.

An ongoing study has used this platform to produce a healthcare solution which monitors the egress of at risk care home residents via wearable Bluetooth beacons. This has been deployed to a real, residential care, environment and has shown promising results wherein the solution has accurately identified egress activities. This solution, called SafeBeacon, is the subject of an upcoming publication. This solution has been n. Leveraging the SensorCentral platform reduced time to producing a solution to a number of days instead of the otherwise projected weeks.

Two studies have used this platform to monitor inhabitants of an environment with thermal vision sensors in order to monitor and classify a number of behaviors of interest such as wandering in Alzheimer's Disease sufferers, Meltdown behavior in Autism Spectrum Disorder sufferers and Sedentary behavior within a work place environment [7], [57].

A recent study [58] used this platform in conjunction with Thermal Vision sensing to determine Gait speed of individuals within an environment to determine wellbeing metrics. This is particularly beneficial when evaluating progression of aging related illnesses.

An additional ongoing study has integrated instances of SensorCentral in a commercial emergency services safety assurance project [59]. This study uses a reduced complexity edition of SensorCentral that is deployed to a single board computer therefore supporting a physically portable solution.

A recent project [60] has used this platform within a multi agent system in order to enable research related to identification interleaved activities of daily living from simple sensors.

In all cases, use of this platform has greatly decreased development time and has provided portability of projects and solutions between environments and solutions.

V. CONCLUDING REMARKS

This work has produced a big data platform designed to enable storage and exploitation of sensor data. This platform has a number of research-oriented features intended to reduce overheads and increase the speed of research activities. These research-oriented features include the ability to define experiments, tools to enable swift annotation of data sets, and machine learning services.

This platform has been integrated into a number of projects, studies and solutions. In each of these cases the solution enabled researchers to rapidly integrate with masses of sensor data and develop solutions.

This sensor data platform is generic and has been shown to integrate with over 20 classes of sensor devices which were produced by over 20 manufacturers. These sensors communicate to the platform using at least 10 different protocols. Notably, two connectors with a wide range of scope have been produced. These connectors can integrate any device connected to a national LoRaWAN network and any device on a Z-Wave compatible instance. This integration is provided with minimal additional effort. The support for devices that these connectors offer is innumerable due to their wide remit.

Notably, this platform is central to a number of research interests within its development environment, Ulster University. However, efforts are underway to make this available to other research groups and universities. Notably, Universidad de Jaén has made efforts in adopting this platform within their research activities [58]

In addition, this platform has been installed to servers which will provide its functionality within residential care giving environments. These servers host a virialized image of an implementation of this platform. Additionally, this solution has been licensed to a commercial entity to support a solution which is based upon it [7]. This instance has been deployed to a dedicated platform.

Further to impacting and benefiting activities within the originating environment, this solution has can assist a broader community and commercial entities. Research communities can benefit from a common platform integrating sensor agnostic and research-oriented functions. In addition to the previously described benefits derived from sensor agnostic function and research-oriented functions. Beyond these functions, further research-oriented benefits are afforded by the platform. These benefits include production of a common platform and a collaborative community.

Providing a common platform would enable research to be portable across research groups. Such portability would enable better collaboration within the research community. Additionally, targeting a common platform enables researchers to share sensor listener software to integrate sensor devices. Sharing such sensor listener software enables communities to reduce the overall effort required to integrate new sensors via code sharing or collaboration. Currently a number of such efforts are shared in an open source fashion on GitHub. The majority of these listeners are operation across a number of research groups and commercial entities.

Future works will aim to integrate this platform with the TensorFlow platform thereby enabling state of the art scalable machine learning services to be leveraged.

Integration with TensorFlow will enable the SensorCentral platform to leverage advances in a dedicated machine learning platform which is being actively developed by Google, who are at the time of writing a world leader in machine learning research and application.

The SensorCentral platform additionally supports storage of multimedia data via an Amazon S3 compatible storage platform facilitated by the open source CEPH platform. CEPH offers a distributed object storage solution, facilitating resilient storage of large and varied multimedia data. Currently, this functionality has been integrated into a single project, upon evaluation of performance this will be reflected upon in future manuscripts.

Further integration between this project and the ODI will be explored to enable seamless sharing of experimental datasets to the international community.

ACKNOWLEDGMENT

Invest Northern Ireland is acknowledged for supporting this project under the Competence Centre Programs Grant RD0513853 – Connected Health Innovation Centre. Additionally, British Telecom and Invest NI are acknowledged for supporting this project under the BT Ireland Innovation Centre (BTIIC).

REFERENCES

[1] D. López-de-Ipiña, L. Chen, A. Jara, E. Mannens, and Y. Li, “Internet of Things, Linked Data, and Citizen Participation as Enablers of Smarter Cities,” *Int. J. Distrib. Sens. Networks*, vol. 12, no. 5, p. 2595847, May 2016.

[2] A. J. Brush, J. Hong, and J. Scott, “Pervasive Computing Moves in,” *IEEE Pervasive Comput.*, vol. 15, no. 2, pp. 14–15, 2016.

[3] D. Patterson, H. Kautz, D. Fox, and L. Liao, *Pervasive computing in the home and community*. CRC Press, 2007.

[4] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, “Big Data: Issues and Challenges Moving Forward,” in *2013 46th Hawaii International Conference on System Sciences*, 2013, pp. 995–1004.

[5] S. Sagiroglu and D. Sinanc, “Big data: A review,” in *2013 International Conference on Collaboration Technologies and Systems (CTS)*, 2013, pp. 42–47.

[6] J. Rafferty, J. Synnott, C. Nugent, G. Morrison, and E. Tamburini, *Fall detection through thermal vision sensing*, vol. 10070 LNCS. 2016.

[7] J. Rafferty, J. Synnott, and C. Nugent, “A Hybrid Rule and Machine Learning Based Generic Alerting Platform for Smart Environments. Engineering in Medicine and Biology Society,” in *Engineering in Medicine and Biology Society (EMBC), 2016 38th Annual International Conference of the IEEE*, 2016.

[8] C. C. Aggarwal, *Managing and mining sensor data*. Springer, 2013.

[9] T. Lee and M. Tso, “A universal sensor data platform modelled for realtime asset condition surveillance and big data analytics for railway systems: Developing a ‘Smart Railway’ mastermind for the betterment of reliability, availability, maintainbility and safety of railway s,” in *2016 IEEE SENSORS*, 2016, pp. 1–3.

[10] T. openHAB Foundation, “openHAB.” [Online]. Available: https://www.openhab.org/.

[11] “IoT Analytics - ThingSpeak.” [Online]. Available: https://thingspeak.com/. [Accessed: 24-May-2017].

[12] S. K. Sowe, T. Kimata, M. Dong, and K. Zettsu, “Managing Heterogeneous Sensor Data on a Big Data Platform: IoT Services for Data-Intensive Science,” in *2014 IEEE 38th International Computer Software and Applications Conference Workshops*, 2014, pp. 295–300.

[13] M. Chen, S. Mao, and Y. Liu, “Big Data: A Survey,” *Mob. Networks Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014.

[14] H. Cai, B. Xu, L. Jiang, and A. V. Vasilakos, “IoT-Based Big Data Storage Systems in Cloud Computing: Perspectives and Challenges,” *IEEE Internet Things J.*, vol. 4, no. 1, pp. 75–87, 2017.

[15] “Beebotte.” [Online]. Available: https://beebotte.com/. [Accessed: 24-May-2017].

[16] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, “Fog

Computing: A Platform for Internet of Things and Analytics.”

[17] C. Cecchinel, M. Jimenez, S. Mosser, and M. Riveill, “An Architecture to Support the Collection of Big Data in the Internet of Things,” in *2014 IEEE World Congress on Services*, 2014, pp. 442–449.

[18] B. Cheng, S. Longo, F. Cirillo, M. Bauer, and E. Kovacs, “Building a Big Data Platform for Smart Cities: Experience and Lessons from Santander,” in *2015 IEEE International Congress on Big Data*, 2015, pp. 592–599.

[19] First Derivatives PLC, “Kx.” [Online]. Available: https://kx.com/.

[20] X. Feng, J. Shen, and Y. Fan, “REST: An alternative to RPC for Web services architecture,” in *Future Information Networks, 2009. ICFIN 2009. First International Conference on*, 2009, pp. 7–10.

[21] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, “MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks,” in *Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on*, 2008, pp. 791–798.

[22] A. Banks and R. Gupta, “MQTT Version 3.1. 1,” *OASIS Stand.*, vol. 29, 2014.

[23] D. G. Puranik, D. C. Feiock, and J. H. Hill, “Real-time monitoring using AJAX and WebSockets,” in *Engineering of Computer Based Systems (ECBS), 2013 20th IEEE International Conference and Workshops on the*, 2013, pp. 110–118.

[24] Internet Engineering Task Force (IETF), “RFC6455: The WebSocket Protocol,” 2011.

[25] T. Rabl, S. Gómez-Villamor, M. Sadoghi, V. Muntés-Mulero, H.-A. Jacobsen, and S. Mankovskii, “Solving big data challenges for enterprise application performance management,” *Proc. VLDB Endow.*, vol. 5, no. 12, pp. 1724–1735, 2012.

[26] S. Madden, “From databases to big data,” *IEEE Internet Comput.*, vol. 16, no. 3, pp. 4–6, 2012.

[27] J. Han, E. Haihong, G. Le, and J. Du, “Survey on NoSQL database,” in *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, 2011, pp. 363–366.

[28] I. Andreev, “Advanced Open IoT Platform for Prevention and Early Detection of Forest Fires,” in *World Conference on Information Systems and Technologies*, 2018, pp. 319–329.

[29] O. Almootassem, S. H. Husain, D. Parthipan, and Q. H. Mahmoud, “A Cloud-based Service for Real-Time Performance Evaluation of NoSQL Databases,” *arXiv Prepr. arXiv1705.08317*, 2017.

[30] T. Connected, “Things Connected Northern Ireland.” [Online]. Available: https://www.thingsconnected.net/.

[31] “RaZberry z-wave server.” [Online]. Available: https://z-wave.me/products/razberry/.

[32] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, “Big Data technologies: A survey,” *J. King Saud Univ. Inf. Sci.*, 2017.

[33] R. Ramakrishnan *et al.*, “Azure data lake store: a hyperscale distributed file service for big data analytics,” in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 51–63.

[34] “MongoDB for GIANT Ideas | MongoDB.” [Online]. Available: https://www.mongodb.com/.

[35] “InfluxData (InfluxDB) - Open Source Time Series Database for Monitoring Metrics and Events.” [Online]. Available: https://www.influxdata.com/.

[36] S. K. J. Basha, P. A. Kumar, and S. G. Babu, “Storage and Processing Speed for Knowledge from Enhanced Cloud Computing With Hadoop Frame Work: A Survey,” 2016.

[37] J. Shafer, S. Rixner, and A. L. Cox, “The hadoop distributed filesystem: Balancing portability and performance,” in *Performance Analysis of Systems & Software (ISPASS), 2010 IEEE International Symposium on*, 2010, pp. 122–133.

[38] H. M. Makrani, S. Tabatabaei, S. Rafatirad, and H. Homayoun, “Understanding the role of memory subsystem on performance and energy-efficiency of Hadoop applications,” in *Green and Sustainable Computing Conference (IGSC), 2017 Eighth International*, 2017, pp. 1–6.

[39] Z. Li, H. Shen, J. Denton, and W. Ligon, “Comparing application performance on HPC-based Hadoop platforms with local storage and dedicated storage,” in *Big Data (Big Data), 2016 IEEE International Conference on*, 2016, pp. 233–242.

[40] F. Holzschuher and R. Peinl, “Querying a graph database – language selection and performance considerations,” *J. Comput. Syst. Sci.*, vol. 82, no. 1, pp. 45–68, 2016.

[41] A. Morari *et al.*, “Scaling semantic graph databases in size and performance,” *IEEE Micro*, vol. 34, no. 4, pp. 16–26, 2014.

[42] L. T. Martin, F. O. F. Peña, A. A. L. Mederos, and J. Nummenmaa, “An empirical performance evaluation of a semantic-based data retrieving process from RDBs & RDF data storages,” *Maskana*, vol. 7, no. Supl., pp. 23–34, 2017.

[43] A. Bader, “Comparison of time series databases,” Diploma Thesis, Institute of Parallel and Distributed Systems, University of Stuttgart, 2016.

[44] B. Leighton, S. J. D. Cox, N. J. Car, M. P. Stenson, J. Vleeshouwer, and J. Hodge, “A Best of Both Worlds Approach to Complex, Efficient, Time Series Data Delivery,” Springer, Cham, 2015, pp. 371–379.

[45] “BSON Specification.” [Online]. Available: http://bsonspec.org/.

[46] “The legion of the bouncy castle, bouncy castle crypto apis.” .

[47] “Neuroph.” [Online]. Available: http://neuroph.sourceforge.net/.

[48] E. Frank, M. A. Hall, and I. H. Witten, “The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques,” *Morgan Kaufmann*, vol. Fourth Edi, 2016.

[49] I. McChesney, C. Nugent, J. Rafferty, and J. Synnott, *Exploring an open data initiative ontology for shareable smart environment experimental datasets*, vol. 10586 LNCS. 2017.

[50] Google, “Firebase Cloud Messaging.”

[51] “AngularJS — Superheroic JavaScript MVW Framework.” [Online]. Available: https://angularjs.org/. [Accessed: 23-May-2017].

[52] “Bootstrap.” [Online]. Available: https://getbootstrap.com/.

[53] “Cross-Origin Resource Sharing (CORS).” [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS.

[54] J. Rafferty, J. Synnott, C. Nugent, G. Morrison, and E. Tamburini, “NFC Based Dataset Annotation Within a

Behavioral Alerting Platform," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017, pp. 146–151.

[55] M. Abadi *et al.*, "TensorFlow: A System for Large-Scale Machine Learning.," in *OSDI*, 2016, vol. 16, pp. 265–283.

[56] J. Synnott *et al.*, "Environment Simulation for the Promotion of the Open Data Initiative," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2016, pp. 1–6.

[57] J. Synnott, J. Rafferty, and C. D. Nugent, "Detection of workplace sedentary behavior using thermal sensors," in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2016, vol. 2016–Octob.

[58] J. Medina-quero, C. Shewell, I. Cleland, J. Rafferty, C. Nugent, and M. Espinilla, "Computer Vision-Based Gait Velocity from Non-Obtrusive Thermal Vision Sensors," *PerHealth'18 - 3rd IEEE PerCom Work. Pervasive Heal. Technol.*, pp. 522–527, 2018.

[59] "Slever Locator Services." [Online]. Available: http://sleverlocatorservices.co.uk/.

[60] C. Orr, C. Nugent, H. Wang, and H. Zheng, "A Multi Agent Approach to Facilitate the Identification of Interleaved Activities," in *Proceedings of the 2018 International Conference on Digital Health*, 2018, pp. 126–130.

Joseph Rafferty (M'14) received the B.Eng. degree in Computer Science from Queens University Belfast, the M.Sc. degree in Computing from Ulster University and the PhD degree in Computer Science from Ulster University. He is currently a Lecturer within the School of computing, Ulster University. His research interests include intention recognition, smart environments, agent-based systems, connected health, sensor technology, planning and intelligent systems.

Jonathan Synnott received the B.Sc. degree in Computing Science from Ulster University and the Ph.D. degree in Computing Science from Ulster University. He is currently a Lecturer in Data Analytics within the School of Computing, Ulster University. His research involves working closely with care providers to develop novel solutions within the domains of smart environments, connected health, sensor technology, and data analytics.

Chris D. Nugent (M'96) received the B.Eng. degree in electronic systems and the D.Phil. degree in biomedical engineering both from Ulster University, U.K. He is currently a Professor of biomedical engineering with the School of Computing, Ulster University. His research addresses intelligent data analysis for Smart Environments and the design and evaluation of pervasive and mobile solutions within the context of ambient assisted living.

Andrew Ennis received the B.Sc. degree in Computing Science, specializing in mobile technology and is currently studying his PhD at Ulster University, U.K. He is currently a Research Associate at the School of Computing, Ulster University. His research interests include connected health, smart environments, sensor technology, semantic enrichment and geospatial information.

Philip A. Catherwood received the B.Eng. Hons. degree in engineering ('97) and the M.Sc. degree in electronics ('01) from the University of Ulster, U.K., and the Ph.D. degree in electrical and electronic engineering ('11) from Queen's University of Belfast, U.K. He worked in industry over an 11-year period developing bespoke scientific measurement equipment and high-speed optical communications devices. His research explores Internet of Things networks, wearable wireless medical devices, and indoor radio channel modelling, His technical contribution to the Telecoms industry was acknowledged through two prestigious industrial recognition awards.

Ian McChesney received the BSc degree in Computing Science and D.Phil. degree in Software Engineering both from the University of Ulster, U.K. He is currently a Senior Lecturer in the School of Computing, Ulster University. His research interests include Software Engineering, Smart Environments and Computer Science Education.

Ian Cleland received the B.Sc. degree in Biomedical Engineering and the PhD degree from Ulster University, U.K. He is currently a Lecturer in Data Analytics, within the School of Computing at Ulster University. His research focuses on the development and evaluation of novel healthcare technologies that incorporate concepts from pervasive computing, biomedical engineering and behavioral science.

Sally McClean (M'00) received the M.A. degree in mathematics from Oxford University, Oxford, U.K., the M.Sc. degree in mathematical statistics and operational research from Cardiff University, Cardiff, U.K., and the Ph.D. degree in mathematics (stochastic modeling) from the University of Ulster, Coleraine, U.K., in 1970, 1971, 1976, respectively. She is currently a Professor of mathematics at Ulster University. She is the Leader of the Information and Communications Engineering Research Group at Ulster University's Computer Science Research Institute.