

## RESEARCH ARTICLE

# SmartHerd management: A microservices-based fog computing–assisted IoT platform towards data-driven smart dairy farming

Mohit Taneja<sup>1,2</sup>  | Nikita Jalodia<sup>1,2</sup> | John Byabazaire<sup>1,2</sup> | Alan Davy<sup>1,2</sup> | Cristian Olariu<sup>3</sup>

<sup>1</sup>Emerging Networks Laboratory, Telecommunications Software and Systems Group, Department of Computing and Mathematics, School of Science and Computing, Waterford Institute of Technology, Waterford, Ireland

<sup>2</sup>CONNECT - Centre for Future Networks and Communications, Dublin, Ireland

<sup>3</sup>Innovation Exchange, IBM Ireland, Dublin, Ireland

## Correspondence

Mohit Taneja, Emerging Networks Laboratory, Telecommunications Software and Systems Group, Department of Computing and Mathematics, School of Science and Computing, Waterford Institute of Technology, Waterford, Ireland.

Email: mtaneja@tssg.org

## Funding information

Science Foundation Ireland (SFI); European Regional Development Fund, Grant/Award Number: 13/RC/2077; CISCO Research Gift Fund

## Summary

Internet of Things (IoT), fog computing, cloud computing, and data-driven techniques together offer a great opportunity for verticals such as dairy industry to increase productivity by getting actionable insights to improve farming practices, thereby increasing efficiency and yield. In this paper, we present SmartHerd, a fog computing–assisted end-to-end IoT platform for animal behavior analysis and health monitoring in a dairy farming scenario. The platform follows a microservices-oriented design to assist the distributed computing paradigm and addresses the major issue of constrained Internet connectivity in remote farm locations. We present the implementation of the designed software system in a 6-month mature real-world deployment, wherein the data from wearables on cows is sent to a fog-based platform for data classification and analysis, which includes decision-making capabilities and provides actionable insights to farmer towards the welfare of animals. With fog-based computational assistance in the SmartHerd setup, we see an 84% reduction in amount of data transferred to the cloud as compared with the conventional cloud-based approach.

## KEYWORDS

cloud computing, dairy farming, data analytics, data-driven, fog computing, Internet of Things (IoT), microservices, smart farm

## 1 | INTRODUCTION

The demand for dairy products is rapidly rising due to an ever increasing population coupled with an increase in income per capita.<sup>1</sup> The consumption of dairy and dairy products is more in developed countries as compared with developing countries, but this gap is narrowing with rising incomes, population growth, urbanization, and changes in diet.<sup>2</sup> Over 80% of the dairy production in developing countries is accounted to “smallholder farming” or “family farming,” mostly reliant on traditional practices with insignificant technology penetration. Catching up with the increasing production requirements requires better technological platforms that optimize production, even in remote small-scale infrastructures

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2019 The Authors. *Software: Practice and Experience* published by John Wiley & Sons Ltd.

with resource and development constraints. The increase in demand due to factors such as an evolving middle class<sup>3</sup> presents an opportunity of specific interest to pasture-based dairy farming industries like those present in Ireland, New Zealand, and Australia, as it is essential to increase milk productivity and yield to have a substantial presence in the market. This becomes even more important for fast growing and developing economies such as India, where a large percentage of population is vegetarian, and consumers look to dairy and its products to sustain their dietary requirements, especially for protein and calcium-based needs.

Dairy farms have all the constraints of a modern business, ie, a fixed production capacity, a herd to be managed, and farm labor, which is often quite expensive. They face several challenges to maintain their business, including labor shortages, climate changes, land availability, herd management, etc. With such constraints, dairy farmers majorly look to have two controls to increase productivity and profit of their dairy farm business:

- Monitoring the welfare of their livestock for anomalies and diseases, so as to detect health issue early and prevent loss.
- Improve cattle reproduction and milk yield by accurate heat detection (oestrus) so as to increase the size of the farm for further profitability.

It is expected that opting for smart dairy farming principles, which unifies Internet of Things (IoT), data analytics, fog computing, and cloud computing, will assist dairy farmers to overcome these challenges and help move the dairy industry towards sustainable growth in both developing and developed countries.<sup>4</sup> Automating various processes on the farm such as heat detection (oestrus) for accurate reproduction, health assessment (lameness, etc), and monitoring the well-being of animals, is an answer to labor shortage and is realized via data obtained from wearable devices worn by the livestock.<sup>5</sup>

While there have been several approaches towards smart farming by an active incorporation of information and communication technology (ICT) in the agricultural industry, a vast majority of them still remain practically unimplemented by virtue of the fact that a farm environment is much unlike an ideally “connected” IoT use case. The true penetration of ICT in the most granular levels of global farming diaspora depends on the transformative adaptability of the IoT ecosystem, wherein a shift is required from a predominantly urban context towards a more flexible solution that addresses the unique environmental, temporal, and spatial challenges of remote farm locations.

Our work is motivated by the need to enable data-driven dairy farming by extracting timely value out of the data by designing suitable models and analytics, and providing different controls to the farmer and other stakeholders to increase productivity and yield, thus helping farmers adopt the best farming practices for the overall benefit. It is crucial to devise a flexible data-driven solution that enables pervasive data collection and processing in remote farm infrastructures, with a system better adapted to performing critical operations even during outages and bouts of intermittent Internet connectivity.

In this work, we address these issues and propose the emerging fog computing infrastructure as a key enabler for IoT applications and services in situational scenarios of constrained Internet connectivity. We envision a microservices-based approach to assist applications in such a hybrid fog-cloud environment, and propose SmartHerd, which is an IoT platform solution that addresses the connectivity and animal welfare in a smart dairy farming scenario.

The paper has been further structured as follows. Section 2 presents the literature review, background, and motivation. Section 3 outlines limits, opportunities, and challenges in the penetration of ICT in agriculture. Section 4 presents the service orientation in the IoT ecosystem and microservices-based architecture. Section 5 presents the proposed SmartHerd management IoT platform. Section 6 presents the results and analysis. Section 7 outlines the learnings and practical experiences from the real-world deployment, and Section 8 presents the conclusion.

## 2 | LITERATURE REVIEW, BACKGROUND, AND MOTIVATION

There have been proposed systems in industry<sup>6-8</sup> as well as in academia<sup>9-12</sup> for animal health management in dairy farms. A study by Steeneveld and Hogeveen<sup>13</sup> and that by Andonovic et al<sup>14</sup> give an overview of the sensor systems available for health monitoring of animals in dairy farms.

Along with the proposed systems and solutions, there has also been research in integrating technologies to increase productivity and sustainable growth in agriculture, ranging from improving wireless network connectivity in economically underprivileged areas<sup>15-17</sup> to data mining and analytics for agricultural applications<sup>18-20</sup> to providing decision

control in variable environments under constraints.<sup>21-23</sup> A review by Wolfert et al<sup>24</sup> shows that predictive insights in farming operations drive real-time operational decisions, and redesigns business processes for the benefit of various stakeholders in a farming landscape, and that the influence of such systems goes beyond primary production, to the entire supply chain. However, the survey by Rutten et al<sup>25</sup> identifies a serious lack of analytics and intelligence in these systems, thus leading to gaps between the desired requirement of the system and proposed solutions. It articulates the need and requirement of intelligence to be present on the premises, in the on-farm systems.

As a consequence, attention is being drawn towards designing systems with intelligence and data analytics capability being present on premises,<sup>26</sup> and utilizing fog computing comes to shore with those objectives in mind. As per the recent architecture released by OpenFog Consortium,<sup>27</sup> fog should be seen as a horizontal architecture that distributes computation, communication, control, and storage closer to end users along the cloud-to-things continuum, and supports a variety of vertical domains of applications. Fog computing is the approach of providing intelligence and processing capabilities closer to the network edge.<sup>28,29</sup> While it was initially aimed to cater to latency sensitive and time-critical applications and use-case scenarios, we believe that fog computing will act as an active enabler for computing systems, applications, and services that suffer from constrained and unreliable cloud connectivity.

In our case, considering that the farm locations are usually remote with intermittent Internet connectivity, dependence on the cloud for pervasive monitoring and computation is less reliant due to a variety of connectivity issues and other associated factors such as high response latency and high bandwidth requirements.<sup>30</sup> In such situations, fog computing can ensure high availability and less reliance on remote cloud infrastructures. Such a system becomes even of more importance with adverse weather conditions such as during storms and hurricanes,<sup>31,32</sup> when the Internet services get interrupted and affected for various reasons. The efficient use of available computational resources realized by means of fog computing, in turn, promotes the idea of resource optimization.<sup>33</sup>

Several interpretations<sup>34</sup> have been proposed for the implementation of fog nodes and their configuration, either via servers, networking devices, cloudlets, base stations, or vehicles. With the increasing demand of providing smart solutions, the shift towards utilization of fog gateway devices for edge analytics is being supported by an increasing number of industrial<sup>35-37</sup> IoT platform developers and solution providers, including IBM, Intel, and Microsoft.

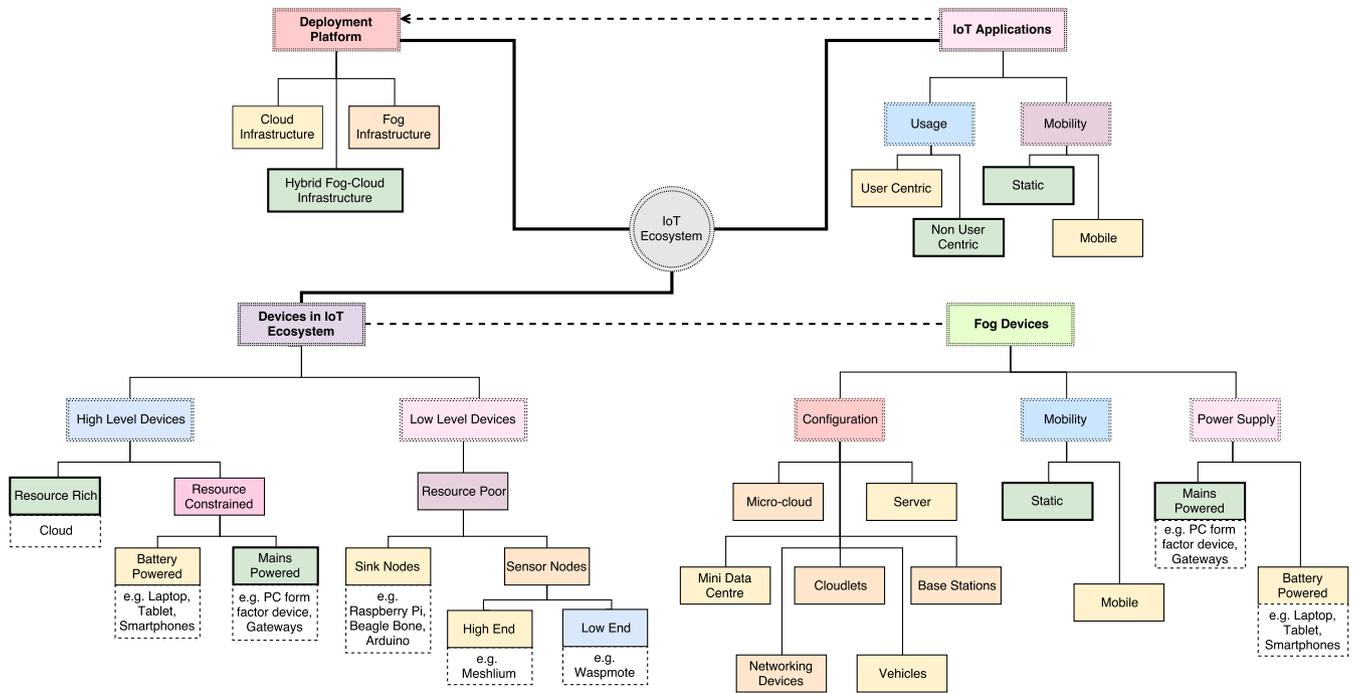
The suitability of fog computing in context of IoT has been studied by Sarkar et al.<sup>38</sup> The approach not only optimizes the use of available in-network compute and storage resources<sup>39</sup> but also reduces the bandwidth and storage requirements to upload data on cloud.<sup>40</sup> Chiang and Zhang<sup>41</sup> survey main research challenges involved in fog and network context of IoT and how fog can help address them. The benefits of combined and continuous fog to cloud architecture have been presented in the work of Ramirez et al,<sup>42</sup> and their evaluation shows a significant reduction in service response time and power consumption over the conventional cloud. A very recent and detailed state-of-the-art survey specifically focusing on fog computing with current research challenges being faced have been presented by Mouradian et al.<sup>43</sup> Hao et al<sup>44</sup> have presented a software architecture for fog environments.

One of the primary limitations possessed by the previously proposed systems is that they take network connectivity for granted, and those who consider that into account do not support real-time analytics. They follow the technique to process and analyze previously collected data and perform offline analysis, which is not suitable for real-time tracking and monitoring of dynamic entities such as dairy cows. The gaps with the existing research are that either it has been developed out of the agricultural context or address the issue of network connectivity, analytics, and control in isolation; this has also been identified as key limitations by Zheleva et al.<sup>45</sup>

Another recent survey by Jukan et al<sup>46</sup> identifies the lack of interoperability provided by such systems and identify the need of developing an integrated system combining edge, fog, and cloud to provide application and services. The authors here also identify that technology solutions with no consideration of interoperability results in vendor lock-in, which not only hinders innovation but also results in higher costs for the farmer/user.

While there has been an initiated movement towards data-driven agriculture in recent times for sustainable and productive growth, no prior work focuses on providing an end-to-end IoT solution integrating edge, fog, and cloud intelligence specifically in case of smart dairy farming IoT settings. The closest work is as done by Vasisht et al<sup>47</sup> primarily towards crop productivity, which has its own set of challenges owing to power, operations, and expenditure that make it unsuitable for a small-scale farm, especially in adverse weather regions.

We position our work as an answer to the issues mentioned above, thus bridging the gap, and providing an innovative way that integrates edge, fog, and cloud computing to provide a solution specifically in case of smart dairy farming in an IoT setup. The novelty from the proposed model comes from the standpoint of view that it has been specifically designed and developed to address a specific vertical of the IoT ecosystem, ie, dairy farming, and within that to address specifically the problems related to animal welfare, and the microservices-oriented design makes it multivendor interoperable.



**FIGURE 1** Visual representation of Internet of Things (IoT) ecosystem illustrating IoT devices, available deployment platforms, and type of IoT applications [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

### 3 | AGRICULTURE AND ICT: LIMITS, OPPORTUNITIES, AND CHALLENGES

Being the key enabler for ICT in smart scenarios, IoT is transforming the way we live and make decisions. Being a multidisciplinary ecosystem, scenarios demanding real-time processing and decision support have been on the increase. Owing to rapid expansion and a vast expanse of possibilities, an IoT ecosystem is hard to define, complex, and difficult to capture.<sup>48</sup> To this end, a multidimensional visual representation of the IoT ecosystem illustrating IoT devices, available deployment platforms and type of IoT applications has been illustrated in Figure 1.

The major limitations, challenges, and opportunities in the context of agriculture and ICT are described in the following.

#### 3.1 | Geographical factors, low Internet connectivity and weather-based outages

What makes a farm scenario unique is the fact that typical farm locations are geographically remote from the urban context and attributed with sparse to low Internet connectivity. The existing IoT systems and applications that address certain specified objectives in the diverse agricultural domain largely depend on Internet connectivity for proper functioning, as the service and application components are traditionally deployed in a cloud-centric manner in remote cloud infrastructures.

Owing to their typical geographical locations, a vast majority of farms already have limited cellular<sup>49</sup> coverage. In that terrain, it is not uncommon to be facing long outages in Internet connectivity due to constraints posed by adverse weather conditions, storms, hurricanes, and other natural disasters. All of these scenarios are apart from general network downtimes, which might be because of another variety of reasons such as network maintenance, broken links, faults, and network/security attacks.

#### 3.2 | Cloud-based solutions

The existing solutions suggested by industry and academia primarily include a lot of closed-form cloud-based solutions that lack flexibility in the management of data and customization owing to on-demand services and operations. They often tend to get expensive, and the trade-off between cost and system utility are not wholly profitable to a small-scale farmer.

### 3.3 | Real-time analytics

One of the key limitations of the existing solutions is the lack of support for real-time processing and analytics for latency-sensitive use cases. For example, while dealing with dairy cattle as in our case, a detailed analysis of patterns from the collected data would enable the development of algorithms, which, once deployed on the fog node closer to the data source, would be able to identify latency-critical scenarios in real time. The existing systems of historical, cloud-based analytics are incapable of serving a time-critical situation, and the bouts of no connectivity further worsen this management.

### 3.4 | End-to-end solution and vendor lock-in

Most of the solutions only focus on one tier of the problem, ie, either data sensing, data analytics, or consumer relations.<sup>50</sup> Architecturally, there is also a major issue of vendor lock in, wherein devices of multiple vendor systems are incompatible with each other, and the lack of features within one solution cannot be complemented with its integration with another.

### 3.5 | WiFi sensors: cost and operational trade-off

Not only is reading data and supporting sensors with WiFi capability challenging in such a remote scenario but also comes at a higher trade-off with the cost. Supporting sensors with an Internet connection hikes their price for both the acquisition and as well as in terms of operational expenses, while Internet connectivity in itself is a challenge in remote regions of operation.

A farm environment thus poses an atypical use case with heterogeneous demands, and the lack of a flexible solution impacts the acceptance of ICT towards what can otherwise be a very fruitful and profitable venture for both the farmer and the solution provider.

### 3.6 | Existing solutions and challenges

There have been several propositions towards enabling connectivity on a farm scenario, as listed in the following.

#### 3.6.1 | Microwave wireless terrestrial link

Being a point-to-point connection, this technology is used by Internet service providers (ISPs) and cellular carriers to connect remote regions towards a backhaul network for last mile connectivity. This link can connect an ISP to a farm, which can then be distributed locally via WiFi access points. However, this has its own sets of challenges and dependencies and is more dependent on Internet providers rather than the individual farmer. Terrain and physical obstacles also impact such a connectivity, as does rain fade and other weather-based factors.

#### 3.6.2 | TV white spaces

To enable connectivity on the farm, solutions such as FarmBeats<sup>47</sup> have proposed the use of unlicensed TV white spaces to set up a link from the farmer's home Internet connection to an IoT base station on the farm, which further provides a WiFi front end. However, this involves its own set of challenges and expenditures that might make such a solution difficult for adoption by a small-scale farmer. For instance, the IoT base station needs power to work, which in itself is a problem. One can use solar electricity, but that is highly reliant on weather conditions and not reliable at this stage, considering the fact that no power on the IoT base station means no WiFi connectivity, and in turn, no data collection and transmission from sensors deployed.

This significantly reduces the reliance and adaptability of ICT on a farm and is not dependable in case of an adverse scenario. Also, currently being unlicensed, it requires permission and authorization from government and federal agencies, which is an additional overhead, thus making it difficult to be used in a number of countries at the moment.

### 3.6.3 | Data mules

A data mule is one such proposition that addresses the issue above in certain use cases where moving objects such as vehicles or people serve as a data mule to collect data from the sensors and to upload it to the cloud as soon as Internet connection becomes available. It was initially proposed for the purpose of delivering emails to remote, rural, or economically underserved regions,<sup>17</sup> which suffer from the Internet connectivity challenge as specified above. The idea is not straightforward applicable and feasible with large-scale sensor deployments, as in such scenarios, the storage capacity of microcomputers or on-chip computers available on sensor devices might get overwhelmed and overutilized with that amount of data. Furthermore, even when the Internet connection is available, the bandwidth available could still be very low, thereby making the upload of large amount of data (in order of megabytes or gigabytes) coming from wide-scale deployments of data infeasible.

## 4 | SERVICE ORIENTATION IN THE IOT ECOSYSTEM AND MICROSERVICES-BASED ARCHITECTURE

The IoT ecosystem possesses some unique requirements for the infrastructure to meet, few of which have been listed in the following.

- **Specified objective:** Typically, the data generated by real-world IoT deployments need to be orchestrated to accomplish a specific objective. IoT services are essentially provided by physical devices present on premises and connected to the Internet.
- **Heterogeneous nature/multivendor interoperability:** The devices in an IoT setting may come from a variety of vendors and thus be widely heterogeneous in terms of characteristics, communication protocol, and networking technologies.
- **Variable requirements:** The latency constraints and data requirements of the varied components of an IoT computing systems might differ extremely and thus demand isolation support for each of the components to fulfill their needs and demand accordingly.

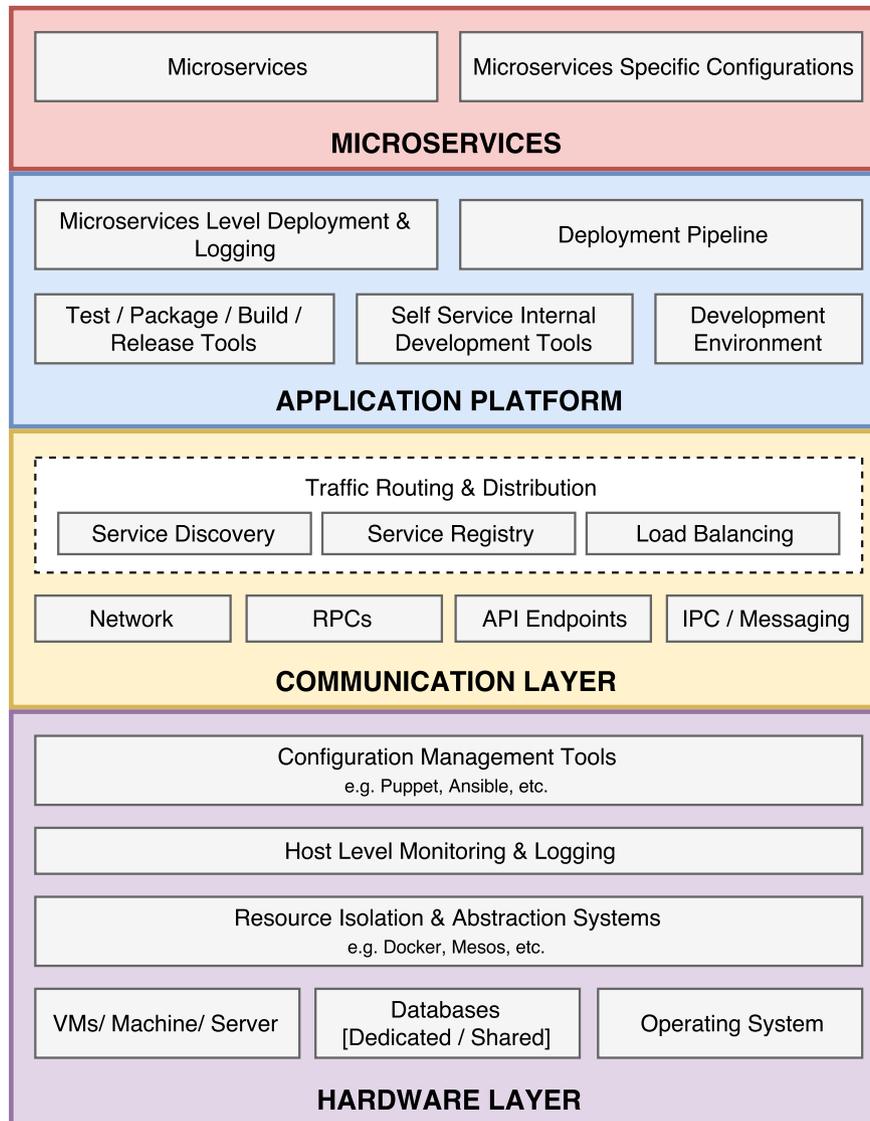
With such diverse and critical requirements, we can clearly infer that a monolithic application design approach would no longer be able to handle the unique requirements imposed by certain IoT use cases and would not be the best fit to handle the data that is produced and consumed by individual “things” and users in the IoT ecosystem. Hence, we believe that a microservices-based approach would be adept at the development and deployment of IoT applications over fog-enabled infrastructures. This involves a proposed change in the conventional application design and development, whereby an IoT application can be decomposed into a collection of microservices, which can be distributed across physical resources available in the cloud and on the network edge.

Ever since the introduction and adaptation of DevOps in practice, the microservices architectural style is the first realization of a service-oriented architecture in this context, and with the plethora of advantages that fall in line with the flexible future applications and service, it is rapidly evolving to be the standard for developing continuously deployed systems.

Some key reasoning outlining similarities in the objectives of microservices and IoT are as follows:

- lightweight communication
- independent deployment facility
- bare-minimum centralized management and control
- isolation support
- building one or multiple applications from a set of different services
- technological independence and support for multivendor interoperability

The microservices architectural model can be depicted to be an abstraction of four<sup>51</sup> layers: hardware, communication, application platform, and microservices. Each of these layers comprises of components, as shown in Figure 2. One of the key takeaways of the microservices architecture is technological independence; each service is independent in terms of deployment and platform and, potentially, the technological stack. They can run their own tasks and processes, and communicate via lightweight protocols and application programming interfaces (APIs). This enables each service to be a business capability that can potentially use its own technological stack but still be a part of a connected comprehensive application structure towards a solution to a larger use case.



**FIGURE 2** The four-layer model of microservice architecture. A comprehensive breakdown of what lies in the four-layered microservices architecture. API, application programming interface; IPC, interprocess communication; RPC, remote procedure call [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

It is also important to apply analytics over the data to get the required information at the right time and in the right place, depending on the use case. Data in IoT deployments typically flow from things to cloud, with fog acting as middleware. It is critical to apply required data analytics at early stages of the pipeline, and the decomposed microservices of an IoT application can provide visualization support for real-time local data processing happening on a fog node via API exposure.

Notably, a microservices-based approach for software development and deployment is currently in wide use by the industry<sup>52</sup> at production level as a part of best DevOps practices.<sup>53</sup> However, that approach has been limited to cloud-based applications only, ie, deploying all the developed microservices for an application within the data center itself, and very limited work<sup>54,55</sup> has been done that focuses on the deployment of applications across the cloud and network edge, especially in the case of IoT applications. The cloud-based applicability of a microservices-style architecture to design a smart city IoT platform has been presented by Krylovskiy et al,<sup>56</sup> where they share their experience of implementing microservices approach in the cloud environment. However, all such cases so far have only been limited to the cloud and not explored in context of the fog to assist distributed IoT scenarios in real time, or as facilitators to enable dynamic characteristics in the network for service provisioning.

## 5 | SMARTHERD MANAGEMENT IOT PLATFORM

In order to address the challenges of remote farm scenarios, we propose the SmartHerd management IoT platform, which is a microservices-based fog computing–assisted IoT platform towards smart dairy farming.

### 5.1 | Goals and objectives

In building SmartHerd, we target the following goals and objectives:

- **Availability:** The downtimes should be minimized, and key tasks such as data collection, processing, and alerts should be resistant to outages.
- **Low dependence on cloud:** Since a primarily cloud-based application requires a pervasive high-bandwidth Internet connectivity, the objective is to reduce the amount of data that is sent to the cloud for computation and to construct the services in a way that outages and constraints in connectivity to the cloud does not adversely impact the deployment scenario.
- **Flexibility:** In case of outages, the devised platform should still be able to collect and process data, as well as support alternative methods of alert.
- **Latency-critical support:** When in a mature stage, the deployment would generate enough periodic data to enable the development of algorithms to detect and identify latency-critical use cases on the farm. The devised platform should be able to potentially support these latency-critical use cases that would arise.

### 5.2 | Smart dairy farm setup: real-world deployment

The trial\* was conducted on a local farm with a full dairy herd of 150 cows in Waterford, Ireland. Out of 150 cows used in the trial, only 147 cows were used in the analysis presented in this article. Given the number of options for the sensors/wearables available for livestock monitoring, we chose a radio-based Long-Range Pedometer (LRP) (433 MHz, ISM band) over WiFi-based sensors, considering it does not rely on Internet for its operation, and serves the purpose of data acquisition in farms with low Internet connectivity and frequent outages. Such sensors involve less operational expenditure, and do not use WiFi-based connectivity to send sensed data to a base station. Therefore, as a part of the deployment, commercially available LRPs (ENGS Systems<sup>®</sup>, Israel) specifically designed for use in dairy cattle were attached to the front leg of cows, as shown in Figure 3.

### 5.3 | Architecture design and system overview

The overall architecture of the test bed is shown in Figure 4. The pedometer consists of an active system with a (backup) data retention capacity of up to 12 hours that measures the activity of cows (such as standing, lying, walking, etc) with a sampling frequency of 8 ms, and the thereby generated data unit is sent to the corresponding receiver and transceiver in every 6 minutes. The range of the antennas attached to the receiver and transceiver is 2 km each, which gives enough coverage to collect data from cows at all times, whether they are grazing in the field, present in their sheds (during adverse weather conditions), or being milked at the milking station.

As shown in Figure 4, the receiver is the master unit, which sends the received data to the communication unit (RS485 to USB) through wired connection, which in turn then sends it to the SmartHerd IoT gateway (a PC form factor device in our case, which acts as controller and fog node; configuration<sup>†</sup> used Intel<sup>®</sup> Core<sup>™</sup> 3rd Generation i7-3540M CPU @ 3.00 GHz, 16.0 GB RAM, and 500 GB local storage) through wired connection via a USB interface.

Most of the dairy farms that are IoT enabled (ie, sensor enabled/smart) have some kind of a farm management system available at hand which the farmer uses to maintain logs or to keep some details. So, our idea was to use the computational resources that are already available in such a setting and leverage them under the fog computing paradigm, so we developed the system (ie, codebase/software framework) as a downloadable piece of software which will run on any PC form factor device. In our case, the laptop available with farmer was with those configurations; hence, we kept the details

\*The ethical approval for the experimentation was taken from Research Ethics Committee of Waterford Institute of Technology, Ireland, prior to the deployment in July 2017.

†The minimum suggested configuration for the given setup is a dual-core processor @ 2.3 GHz, 4.07 GB RAM, and 100 GB local storage.



**FIGURE 3** Pedometer attached as a part of the experiment to the front leg of the cows [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

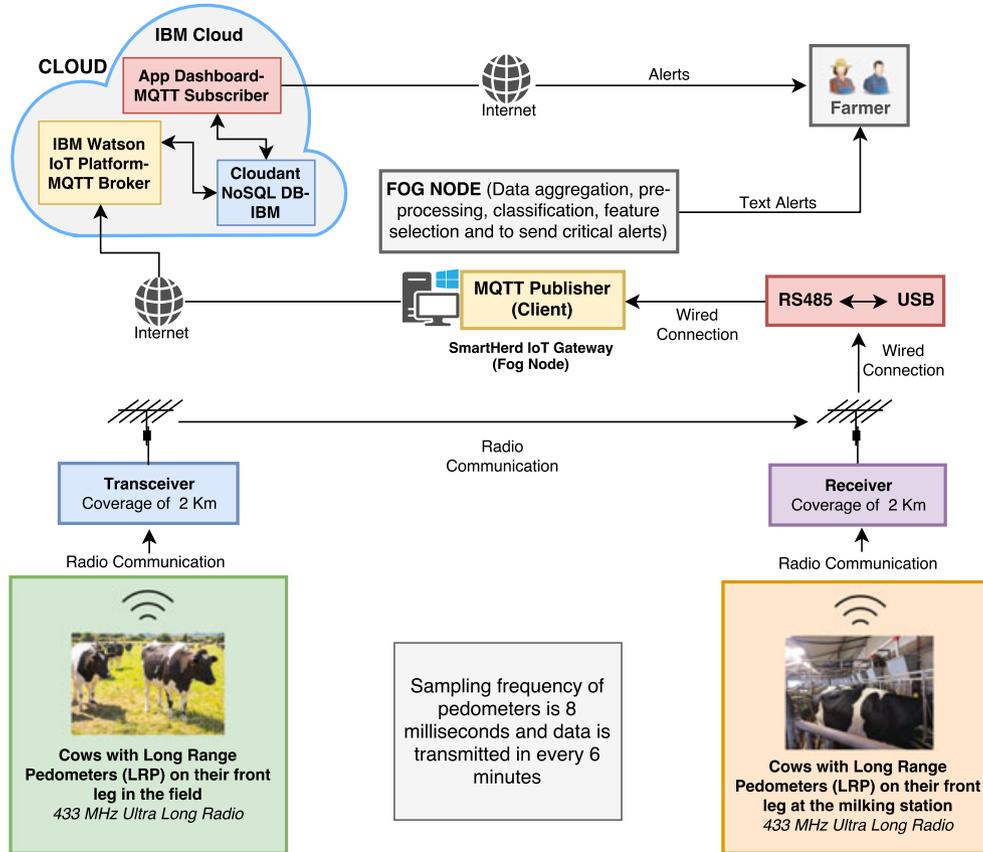
in the manuscript as it is, and suggested the minimum configuration which will work both in case of PC or as per the industry standard<sup>57,58</sup> commercially available gateways.

We have chosen Message Queue Telemetry Transport (MQTT)<sup>59</sup> as the connectivity protocol between fog node (ie, local PC) and cloud (service instances running on IBM Cloud) in our deployment setting. MQTT is an open-source protocol originally invented and developed by IBM.<sup>60</sup> It is a lightweight publish-subscriber model-based protocol designed on top of the transmission control protocol (TCP)/Internet protocol stack.

It is specifically targeted for remote location connectivity with characteristically unreliable network environments such as high delays and low bandwidth,<sup>61</sup> which is one of the issues in remote farm-based deployments such as ours. Hence, we chose MQTT as the connectivity protocol in our deployment.

MQTT provides three quality-of-service (QoS) levels<sup>62</sup> that define the guarantee of delivery for a specific message between sender and receiver:

- At most once (QoS 0): There is no guarantee of delivery here. If the failure happens, no additional attempts are made to rehandle those messages. It is often called “fire and forget” and provides the same guarantee as the underlying TCP.
- At least once (QoS 1): At least once means that messages in a stream are guaranteed to be delivered at least one time to the receiver. If the failure happens, additional attempts will be made to rehandle those messages. This approach may cause unnecessary duplication of data packets in the streams.
- Exactly once (QoS 2): Exactly once means that messages are guaranteed to be handled exactly the same as it would be in the failure-free scenario, even in the event of various failures. It is the safest and slowest QoS level.



**FIGURE 4** SmartHerd management system overview. DB, database; IoT, Internet of Things; MQTT, message queue telemetry transport [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

A developer can specify the QoS level that they want on both publisher and subscriber as per their application requirements.

The MQTT architecture comprises of two components, namely MQTT clients (such as publishers and subscribers) and MQTT broker (for mediating messages between publishers and subscribers). In our setup, these components are as follows:

- **MQTT publisher:** Script running on fog node (ie, SmartHerd IoT gateway).
- **MQTT broker:** IBM Watson IoT platform (as a service on IBM Cloud).
- **MQTT subscriber:** Application designed and hosted on IBM Cloud.

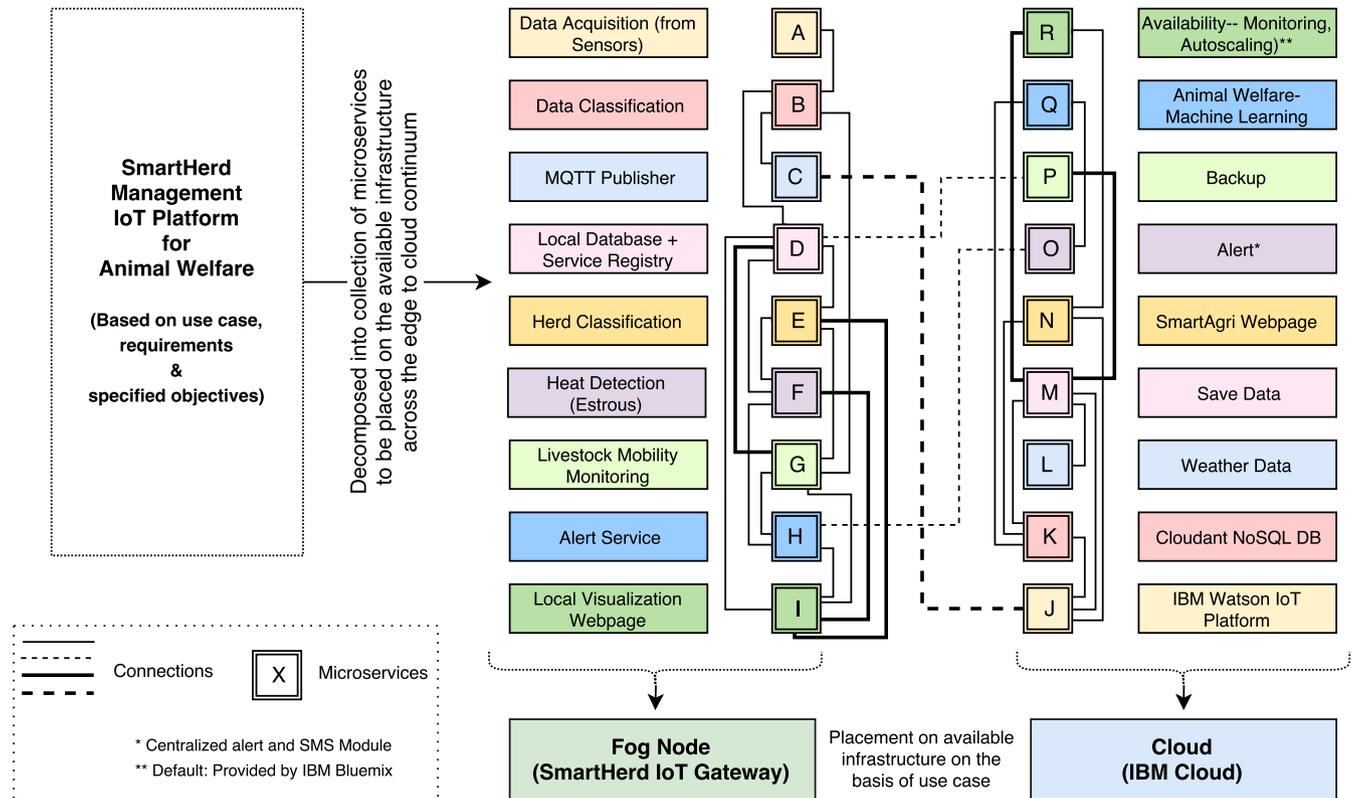
Thus, the data from the fog node after classification as described above is streamed to IBM Watson IoT platform using MQTT; the IBM Watson IoT platform receives all these messages, and the MQTT subscriber listening to the events of this broker picks up all the data and stores it in Cloudant NoSQL JSON Database (Database service on IBM Cloud).

## 5.4 | Microservices-oriented application design

The desired application has been built as a collection of microservices as shown in Figure 5. We envision microservices as one of the key enablers while building an end-to-end IoT solution in such scenarios.

We make the key observation that data requirements and latency constraints of various applications and services in a smart dairy farming scenario can be primarily classified in the categories as shown in Table 1. This along with other reasons mentioned previously motivates the use of microservices-based architecture and fog computing-based approach in smart dairy farm setup.

From an application deployment standpoint, fog computing can be perceived as component of application running at the network edge as well as in cloud, with the components running at fog being latency sensitive and time critical in nature. As fog computing was initially aimed to improve network latency<sup>63</sup> between sensors, applications and users;



**FIGURE 5** Decomposition of SmartHerd Internet-of-Things platform in terms of microservices and placement of these services across edge of the network and on the cloud. DB, database; MQTT, Message Queue Telemetry Transport [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

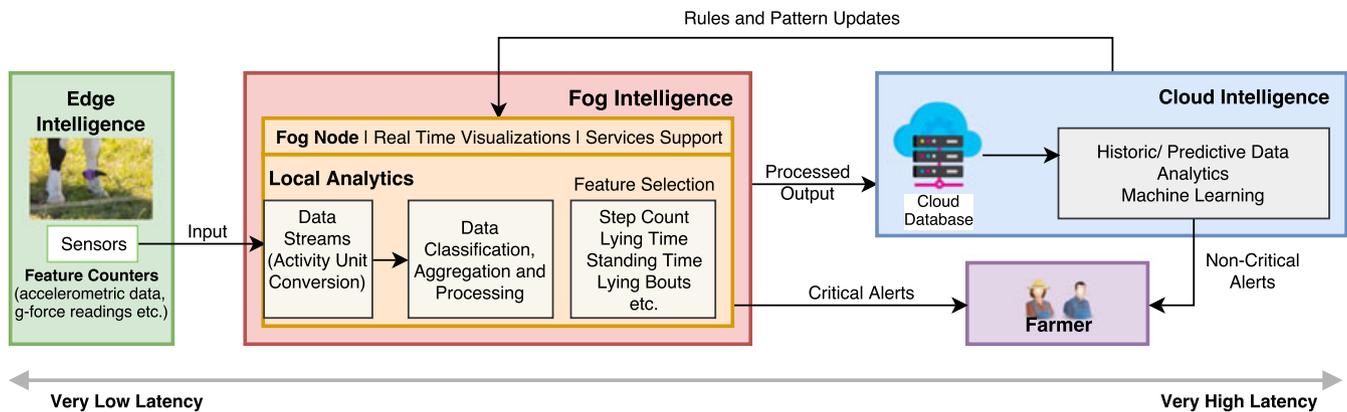
**TABLE 1** Data requirements and latency constraints

Application/Service	Latency Constraints	Data Requirement
Query sensor data	Minutes	Immediate
Livestock location monitoring and mobility analysis	Minutes	Immediate
Heat detection	Hours	Immediate
Lameness and other illness detection	Hours - Days	Nonimmediate
Animal health statistics	Hours - Days	Nonimmediate
Logging and other application performance logging services	Hours - Days	Nonimmediate
Animal behavior and variability Analysis via mathematical modeling	Days	Nonimmediate

however, coupled with a microservices-based application design, we also envision it as an enabler to provide services that efficiently utilize the fog-enabled infrastructures, while also catering to situations where Internet connectivity is constrained.

For ease of orchestration, fog services are abstracted inside containers. The realization of service abstraction becomes more important in scenarios where fog devices are heterogeneous in nature and can range from end-user devices to access points, to routers, and to switches; so to accommodate such heterogeneity, service abstraction is desired and can be realized in terms of containerization. The examples of container technologies include Java Virtual Machines (JVM), Python Virtual Machine (PVM), Linux containers, and Docker<sup>‡</sup>. A wise decision on deciding which container technology to choose acts as an important factor in the efficiency and performance of the overall system. In our deployment setting, we used PVM and JVM as container technologies (or as equivalent possible substitute) on fog node and the default available

<sup>‡</sup><https://www.docker.com/>



**FIGURE 6** Workflow and data flow in the test bed deployment [Colour figure can be viewed at wileyonlinelibrary.com]

container technology in the IBM Cloud (ie, Docker as platform and Kubernetes as container orchestration system for Docker containers) while building SmartHerd IoT system.

We wanted to keep the service management overhead on a fog node as low as possible. The programming platform (languages) used in the development process were Python and Java. Keeping track of developed modular part of the code base as isolated processes (Namespaces and Cgroups) gave us a way to use them as individual and different services with a simple service registry (in form of a simple file) rather than using an additional software/framework to do the same; hence, we used PVM and JVM as containers equivalent on a fog node.

The designed and built microservices of SmartHerd IoT ecosystem are placed across network edge at SmartHerd IoT Gateway and Cloud, ie, IBM Cloud in a manner that latency-sensitive one and the ones with immediate data requirement are placed at the fog node and the remaining ones are placed at the cloud with the desired resource requirement<sup>40</sup> of each of the service being met in an efficient manner.

## 5.5 | Workflow

The detailed work flow of the experimental setup is as shown in Figure 6. The fog node also provides a dashboard for the farmer and serves as a visual medium to see the event information and other related sensor data. After gathering initial insights from the collected data and generating the corresponding alert/response, the aggregated, processed data are sent to the cloud for historical storage and analysis. The cloud is also the site for fusion of the data from other sources, such as weather data. The learning pattern from historical data analysis at cloud is sent back to the fog node for further enhancement of the system and to increase the overall efficiency and responsiveness.

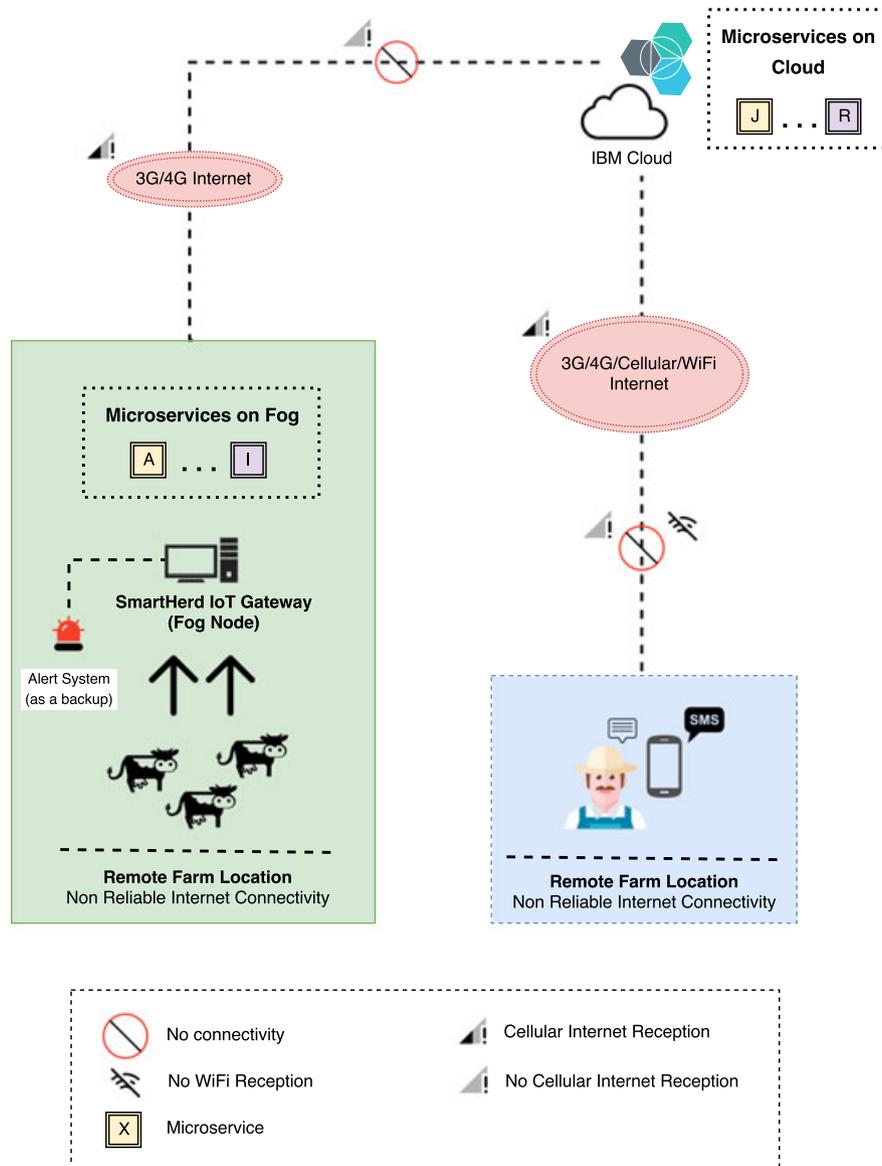
Since fog devices have limited computational capabilities, so in such circumstances, it makes sense to have module-based data analytics components that can be remotely pushed onto fog devices dynamically and on demand. All the long-term information and data are stored in the cloud, and like the fog, it too provides a dashboard where the farmer can input and modify any relevant information related to their livestock and demand further features and services.

## 5.6 | Use case

As shown in Table 1, while the data categorized under nonimmediate requirements is given low priority for fog-level computation and might even be provisioned to the cloud, the main point of focus is the immediate attention and actionable decisions on the data that have immediate requirements of processing for a critical output towards the welfare of the animal.

To determine the attributes of Table 1, we conducted a survey of dairy farmers and dairy-tech service provider companies and consulted the literature. The more insight on this came from survey than from literature. This is in line with the domain-driven design methodology of software development, ie, to work closely with a domain expert (majorly the farmer in our case) and help us understand better how the real-world system works and what are the demands/needs there.

In our use-case scenario, the latency constraint and data requirement of a microservice remains static all the time. However, this might not be the case with other use case scenarios; as a hypothetical example, here, consider a microservice that takes temperature and humidity values from the sensors to determine environmental conditions, for example, in



**FIGURE 7** Use-case scenario with constrained Internet connectivity [Colour figure can be viewed at [wileyonlinelibrary.com](https://onlinelibrary.wiley.com)]

order to regulate air conditioning in an area. Now, a sudden change in temperature might not mean anything, but if the change stays for a long time, it might be an indication of fire in that area. Now in this scenario, the latency constraint and data requirement of the microservice will change. So, these two features of a microservice are use-case dependent.

In the case of detection of an event that would require attention towards the animal, an alert is sent out to the farmer. In summary, SmartHerd, a dairy farm use-case solution towards a smart farm deployment, is as shown in Figure 7.

## 6 | RESULTS AND ANALYSIS

A brief demo video of the developed system is available at this web-link.<sup>64</sup>

### 6.1 | Herd classification and cow profiling

For the system to differentiate between normal and anomalous behavior, there is a need of cow profiling. There needs to be a baseline for the normal activity in the herd, which characterizes normal behavior. Animals grazing within the same pasture can influence the movement, grazing locations, and activities of other animals randomly, with attraction, or

with avoidance<sup>65-67</sup>; therefore, most of the animals will have their activity levels almost equivalent to the herd mean. For such reasons, any deviation from such behavior is classified and analyzed for an anomaly. Such an analysis eliminates the effects of external factors as these will be largely affecting the herd as a whole and only leaves back exclusively the individual behavior of the cow.

As a part of our monitoring and analysis, we use the data of three activities for each cow:

1. **Step count:** The number of steps an animal makes per day. This depends on the age, the level of activity of the cow, and familiarity with the pasture.
2. **Lying time:** The number of hours an animal spends lying down. Lying time has a crucial role to play in determining the emotional and physical health status of a cow, and is cumulative of rumination, sleep, rest, and other factors, which include state of pregnancy, medical conditions, range of illnesses, and potential lameness.
3. **Swaps per hour:** The number of times an animal moves from lying down to standing up, per hour.

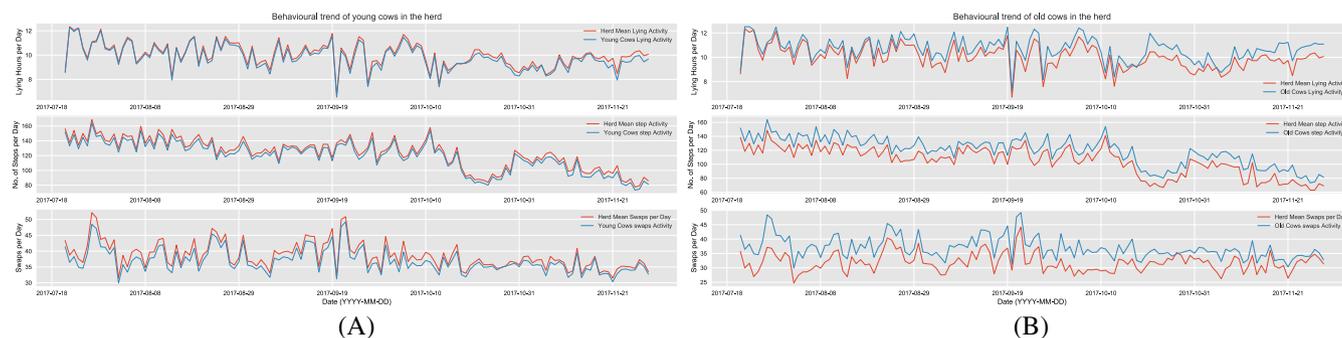
In the following sections, we describe the underlying technique used to form cow profiles to detect anomalies in the herd.

## 6.2 | Age-based classification

As mentioned earlier, 147 dairy cows were used in this analysis from the herd consisting of 150 dairy cows in our experimental setup, the age distribution of which are as depicted in Table 2. The first and most intuitive way of clustering the cows is in terms of age groups: young (age 2 to 5) and old (age 6 to 12). While this matches the intuition that the activity patterns of young cows are closer to the herd mean as opposed to older ones in the herd, there are crucial aspects missed out. For instance, the sample size in young category exceeds the old by 67%, and it is difficult to trace out the subsets of animals that may behave differently in these binary groups. It thus excludes the outliers in each group and generalizes the classification as a whole, which leaves the true behavior of a cow difficult to trace. The behavioral trend of young and old cows for the three activities is as shown in Figure 8A and Figure 8B, respectively. Our initial work has presented

**TABLE 2** Age distribution of cows in the herd

Age (in years)	Number of Cows	Category
2	29	Young cows (age 2 to 5) Number of young cows = 111
3	31	
4	29	
5	22	
6	9	Old cows (age 6 to 12) Number of old cows = 36
7	7	
8	7	
9	5	
10	7	
12	1	
<b>Total</b>	<b>147</b>	<b>147</b>



**FIGURE 8** Young and old cows. A, Behavioral trend of young cows for the three activities in the herd; B, Behavioral trend of old cows for the three activities in the herd [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

age-based clustering of cows combined with data analytics to detect anomalies in their behavior and microservices-based application flow for integration of specific services such as lameness in dairy cattle.<sup>68,69</sup>

### 6.3 | Activity-based clustering

Although behavioral studies indicate that cattle in the same pasture are not considered independent experimental units because of the potential confounding effects of the herd's social interactions,<sup>66</sup> the same authors concluded that under some situations such as larger herds (of around 53-240 cows), movement patterns of subsets of individual cows may have a level of independence that is sufficient for analysis as individual experimental units.

Therefore, there can be no one-size-fits-all solution for determining the behavior of an individual cow, considering the difference in the behavior of subsets within the same herd. Not all cows would have the same levels of activity as others their age or behave similarly.

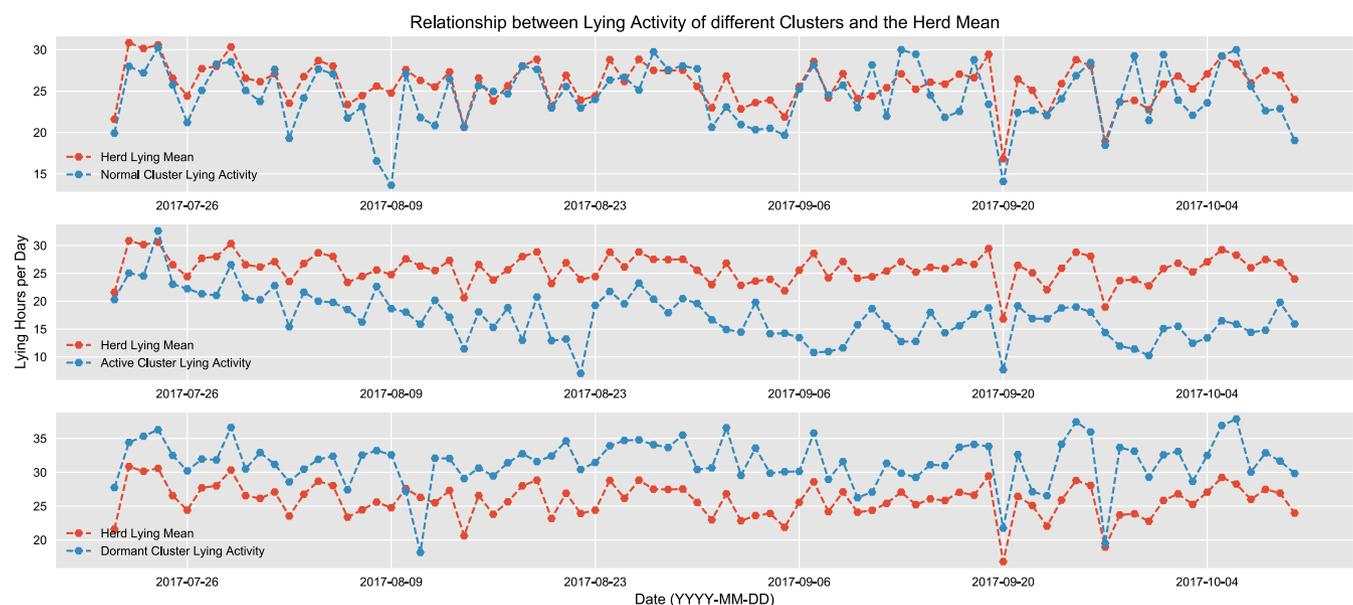
For the purpose of a better analysis of activity, we calculate the absolute deviation (AD) for each of the three monitored activities for each cow against the herd mean, as illustrated in Equation 1.

$$C_{AD} = |H_m - C_a| \quad (1)$$

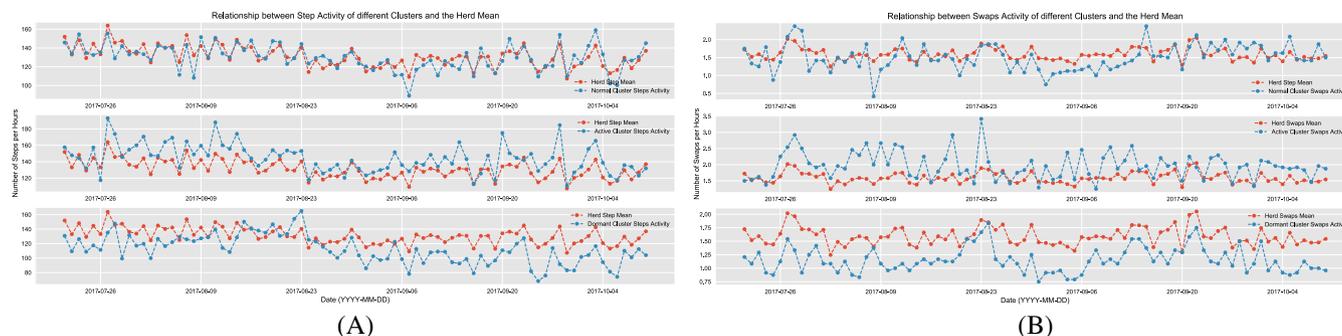
Here,  $H_m$  is the herd mean on a given day,  $C_a$  is the cow's corresponding activity, and  $C_{AD}$  is the AD of cow. Based on a best of three analysis of these ADs for each of the activities, we classify the herd into three clusters based on their behavior: active, normal, and dormant. The cows in the active cluster have an activity level higher than the herd mean, ie, typically higher values of step count and smaller values of lying time. The cows in the normal cluster have their activity levels almost equivalent to the herd mean, and finally, those in the dormant cluster have lower values of step count and spend most of their time lying down. Figure 9 shows the lying activity of the different clusters against the herd mean, and Figure 10A and 10B depict the same for number of steps and swap count, respectively.

This method of classification is unbiased of age; thus, it not only covers the outliers within each age group but also depicts a more in-depth analysis of the activities of each of the cows belonging to each age group, and the overall analysis of the herd activity. The distribution of each cluster against the age is as shown in Table 3. The key observations here include the following:

- The active cluster comprises of 30% young (2-5) and 7% old (6-12) animals in the herd.
- The normal cluster comprises of 45% young (2-5) and 12% old (6-12) animals in the herd.
- The dormant cluster comprises of 0% young (2-5) and 6% old (6-12) animals in the herd.



**FIGURE 9** Lying activity of the clusters against herd mean [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



**FIGURE 10** Step and swap activity of clusters. A, Step activity of the clusters against herd mean; B, Swap activity of the clusters against herd mean [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

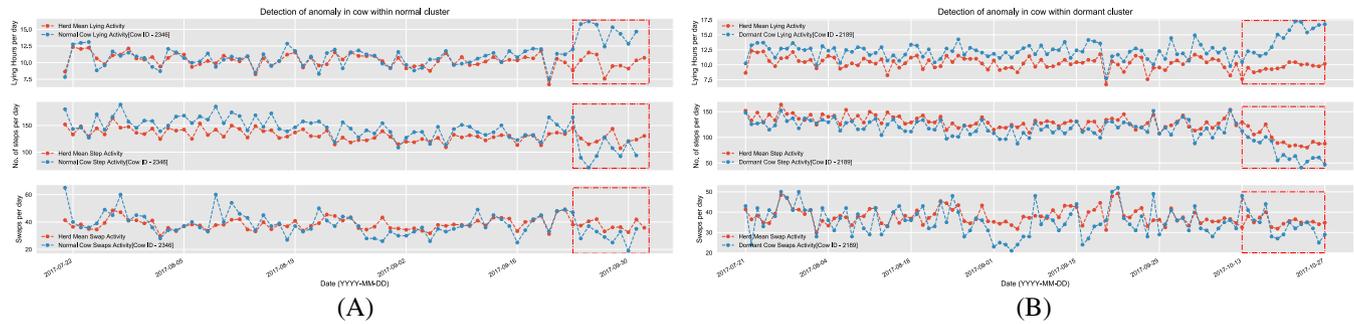
**TABLE 3** Distribution of each cluster against the age

Age (in years)	Active	Normal	Dormant
2	12	17	0
3	9	22	0
4	14	14	1
5	9	13	0
6	7	2	0
7	2	4	1
8	0	4	3
9	1	1	3
10	0	6	1
12	0	0	1
<b>Total</b>	54	83	10

## 6.4 | Animal welfare alert system

Postmonitoring, analyzing, and clustering the cows on the basis of their activity, the SmartHerd platform addresses the critical use cases in a dairy farming scenario where the farmer is alerted about a cow's activity that requires attention and intervention. As shown in Figure 7, the system uses a short messaging system–based service to alert the farmer in this context and, in the case of low cellular/Internet connectivity, displays a physical audio-visual alert on the farm premises. The alerts generated include the following:

- Heat detection (Oestrus):** One of the most profitable and important activities for a dairy farmer is herd reproduction. This is not only important to sustain milk production throughout the years but also to increase the size of the herd. A cow's milk yield gradually decreases after the initial months postcalving, hitting a dry state eventually. Considering such scenarios on a dairy farm, it is critical to accurately identify an animal in heat, which is the best time for insemination of a cow towards pregnancy. This is a 21-day cycle in cows, and the exact heat window only lasts once for about 18 hours in this 3-week cycle, which makes it critical to be identified in time. One of the more accurate methods of detecting an animal in heat is monitoring the step activity, which is an increasingly restless behavior with a sudden spike in step activity, characterizing the beginning of the heat (oestrus) window for the cow, for which the farmer needs to be alerted in time for action to be taken before the insemination window ends.
- Monitoring of an animal's migration between clusters:** Not all cases of a cow being dormant mean that there could be a cause of concern—there might be cows that are dormant and perfectly healthy, as long as they are consistent in their clusters' behavior over time. While the cows are liable to gradually shift their behavior over time between clusters, what is a potential cause of concern is the sudden movement of an animal from a high or normal activity level to being dormant or a sudden drop in its activity even while being dormant. This could indicate a sudden illness, infection, or more commonly, lameness. For example, Figure 11A shows a rapid movement of a cow from being normal to dormant, and Figure 11B shows a sudden drop in the activity of a dormant cow. The farmer was alerted on these cows' behavior, and upon inspection, they were found to be developing lameness.



**FIGURE 11** Detection of anomalies in clusters. A, Normal; B, Dormant [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

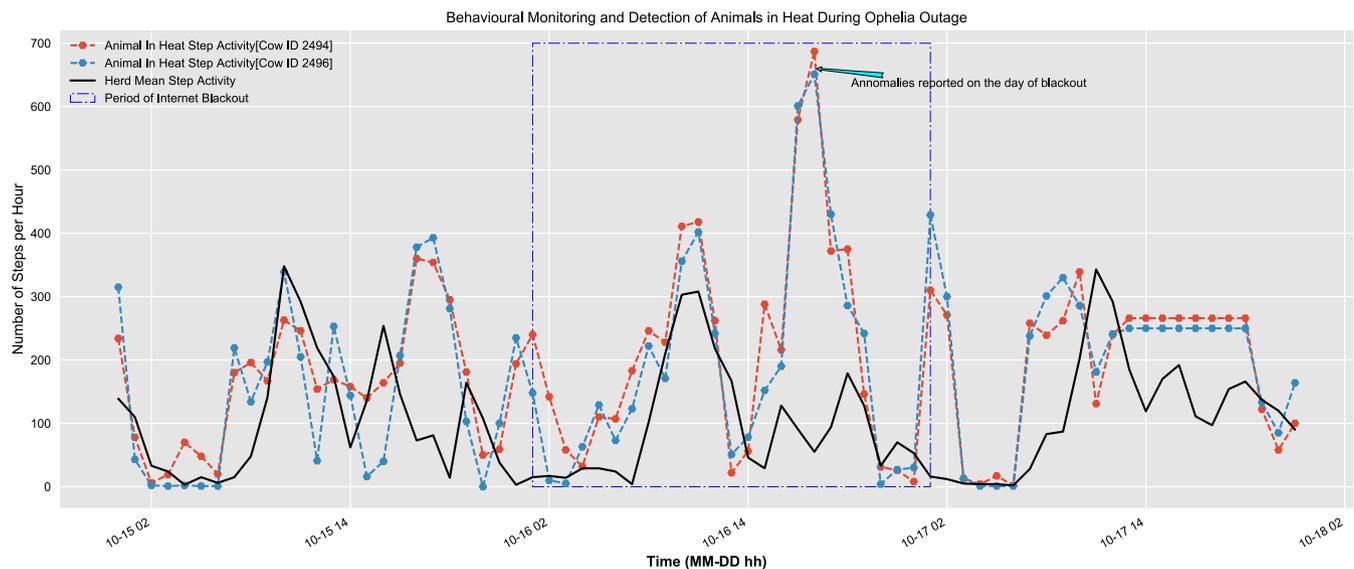
- **Calving alert:** When nearing the time of birth, a pregnant cow tends to find a secluded spot away from the herd and rest, showing an increased lying time. Therefore, lying activity, when tracked with insemination date and pregnancy stage, can quite accurately alert the farmer towards calving activity for a cow.

## 6.5 | System resilience

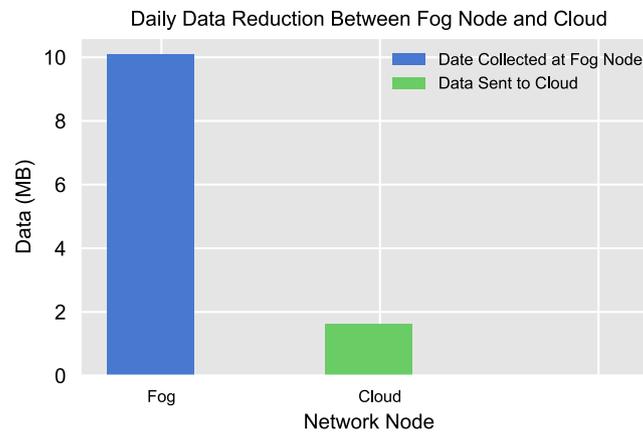
### 6.5.1 | Network outages

Since the SmartHerd IoT platform does not depend on WiFi sensors for data collection, the data collection is not impacted due to Internet constraints, outages, or connectivity issues. Further, the fog node is responsible for the local processing of the data input from sensors and the identification of critical activities such as heat detection. This ensures that critical alerts are not missed out in the state of constrained connectivity to the cloud, and the platform is still able to monitor and analyze the behavior of the herd and track animals' welfare.

A true test of this platform came up at the time Ireland was hit by Hurricane Ophelia<sup>31,32</sup> on October 16, 2017 and the farm location in county Waterford lay in the most severe red alert zone. The area was impacted widespread outages of both power and Internet for two days, and as shown in Figure 12, during the time of the blackout, the platform still remained operational, collected and analyzed the data, and was able to capture the animals in heat at the time and alert the farmer. Thus, even though it was unable to sync up with the cloud due to no connectivity, the SmartHerd IoT gateway was able to function uninterruptedly for a reasonable period of time and provide functionalities even when the connection to cloud was lost, and functioned towards the welfare of the herd.



**FIGURE 12** Working of system during hurricane Ophelia outage [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]



**FIGURE 13** Daily reduction in the amount of data between the fog node and the cloud [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

### 6.5.2 | Failure/overload situation of fog node

In the current setup, we only considered one gateway. In the likely event that the gateway is overloaded or is unavailable, the sensors have a data retention capacity of up to 12 hours, the gateway has a local database that stores the data before preprocessing, the MQTT publisher saves the state of the last successfully transmitted events, and would then continue from where it stopped in case of a network outage. Another possibility to increase resilience of the system is to have coexistence of multiple gateways in the infrastructure, which are synchronized with each other, and data can be redirected between them in case of failure of one gateway; this is something we are considering in our future work.

## 6.6 | Data reduction

From what we have monitored, each sensor/LRP generates around 70 KB of data per day, which, as shown in Figure 13, accounts for 10.1 MB of data collected from the 147 sensors at the SmartHerd gateway each day, with data acquisition and streaming happening every hour. However, given the local processing capability attributed to fog computing, the amount of data that is sent further to the cloud is only 1.62 MB per day, which, as shown in Figure 13, is an 84% reduction in the amount of data that would otherwise have streamed to the cloud throughout the day. This aspect of data reduction becomes even more crucial while scaling up the farm and the herd, as the amount of data collected and streamed would rapidly increase.

## 6.7 | Accuracy of the system developed

We formulated our problem as a binary-class classification problem with irregular (anomalous behavior) being the positive class and regular as the negative class. The data split was made as 80%-20%, ie, 80% of data was used for model training and the remaining 20% was used for testing.

We experimented on a number of classification algorithms ranging from support vector machine, random forest (RF), K-nearest neighbors, and decision trees while noting the accuracy. The best two performing were RF and K-NN with an accuracy of 91% and 1 day before visual signs and 87% with 3 days before visual signs, respectively, as presented in Table 4.

Optimal results for K-NN were obtained at  $k = 4$ , which gave an accuracy of 87% with 3 days before the visual signs could be seen. In all, the normal cluster model had a sensitivity of 89.7% and specificity of 72.5%. The anomalies reported by the system were confirmed as lame cows by the farmer.

**TABLE 4** Accuracy of the developed system

Classification Model	Accuracy (in %)	Number of Days Before the Visual Signs of Anomaly Appears
Random forest	91	1
K-Nearest Neighbors	87	3

## 6.8 | Resource utilization at fog node and discussion on platform performance on using fog node with low-level computational power

The main processing component among the microservices running on the fog node is the MQTT publisher, which is responsible for data preprocessing, aggregation, and streaming processed data to the cloud. We monitored resource (CPU and memory) consumption on the fog node in a time window of every 3 minutes when all the microservices were running on it. Using the psutil (process and system utilities)<sup>70</sup> Python library, we calculated the percentage increase in both CPU and memory before, during, and after streaming (ie, before the MQTT publisher was run, during, and after it was run). We used the percentage increase because this would form a baseline here, ie, irrespective of change in the fog node capabilities, the same increase in resource consumption would be expected.

Figure 14A shows memory usage before, during, and after the MQTT publisher is run. The region highlighted shows the period during which the MQTT publisher was streaming. As it can be seen, there is an increase from about 35.38% to 36.75%. The total usable memory was about 14.50 GB. This would mean the impact of MQTT Component was only 0.12 GB.

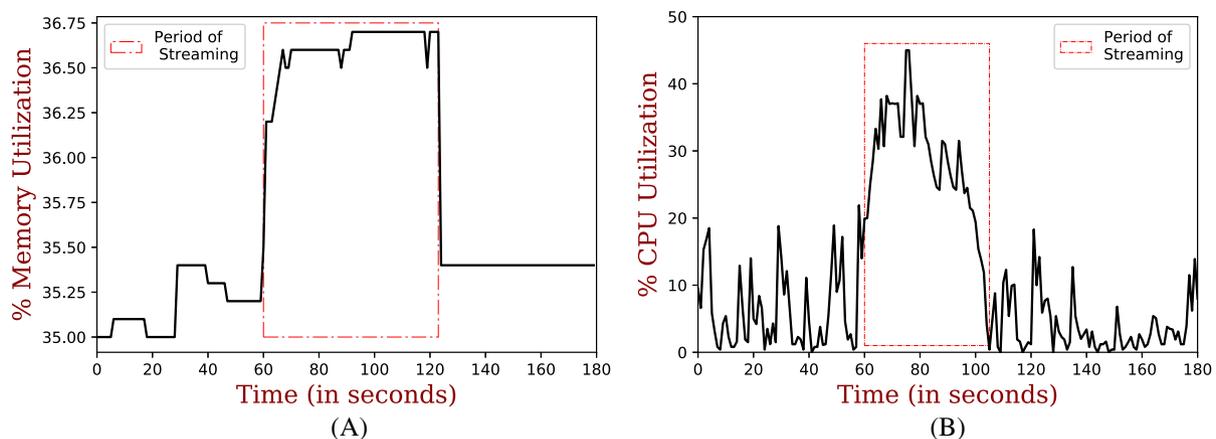
Figure 14B shows the CPU usage before, during, and after the MQTT publisher is run. The region highlighted shows the period during which the MQTT publisher was streaming. As it can be seen, there is an increase from about 20% to 45% in CPU utilization.

Table 5 summarizes the CPU and memory utilization before, during, and after streaming.

In all, the overall resource utilization and effects of MQTT component on the fog node are very minimal for it to run on very resource-constrained compute environments.

Although the real-world experiment was carried on a high-end fog node (computer), but prior to the experiment (ie, before putting the developed code base in the actual deployment), we tested resource (CPU and memory) consumption on another PC setting (Intel® Core™ i5-4300M CPU@ 2.60 GHz, 8.0 GB RAM, and 500 GB Local Storage), and also on a single virtual machine instance running on OpenStack with following configuration: two VCPUs (Intel® Xeon Processors @2.60 GHz), 4 GB RAM, 20 GB disk, initially with artificial data and later with data obtained from ENGS systems (company who provided sensors in the deployment).

A similar behavior as mentioned above was observed on them and the developed system worked well with same settings as in real deployment.



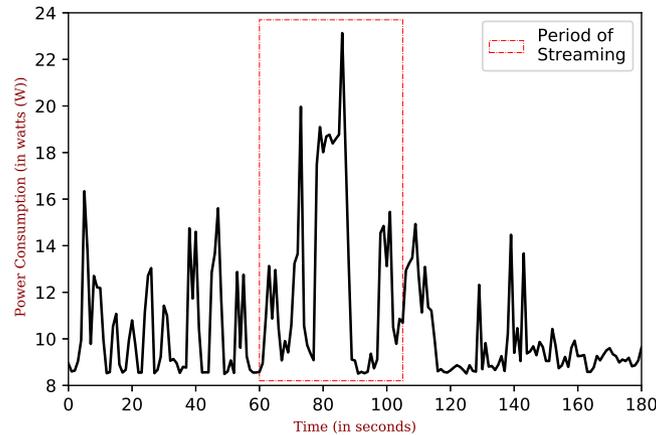
**FIGURE 14** Resource utilization at fog node (ie, SmartHerd IoT Gateway). A, Memory utilization at fog node; B, CPU utilization at fog node [Colour figure can be viewed at wileyonlinelibrary.com]

**TABLE 5** CPU and Memory utilization on fog node before, during, and after streaming for comparison of the developed system to be able to run in resource-constrained environment as well

Resources	Before Streaming With All the Microservices Running	During Streaming With All the Microservices Running	After Streaming With All the Microservices Running
CPU	At max 20% and mostly below that	From 20% to 45% (25% increase)	At max 20% and mostly below that
Memory	At max 35.38% and mostly below that	From 35.38% to 36.75% (1.37% increase)	At max 35.38% and mostly below that

**TABLE 6** Service delivery time of the developed system

Average Query Response Time Total Service Delivery Time SmartHerd System (in ms)	Network Type Available on User's Device	Network Round Trip Time (in ms)	Total Service Delivery Time (in ms) (Sum of Query Response Time and Network Round Trip Time)
	2G	250-300 ms	270-330 ms
20-30 ms	3G	80-100 ms	100-130 ms
	4G	30-50 ms	50-80 ms

**FIGURE 15** Power utilization at fog node [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

Further, in context of efficient in-network resource utilization and increasing system resilience, one of the possible directions of future work is to look into distributed learning-based<sup>71</sup> and distributed data analytics-based<sup>72,73</sup> approaches in such real-world IoT-based deployments.

## 6.9 | Service delivery time

The service delivery time here represents the average query time that we performed on the system with different queries, like a farmer can query the system to see the report about a particular animal or to see the system in general. This time also corresponds to the time when an alert is generated or an anomaly is detected, and the farmer needs to be informed then how much time does the developed system take to do that. It includes the network round trip time along with the processing delays by the system. The network available at user's mobile device can change from 2G, 3G, to 4G. Table 6 summarizes the service delivery time of the developed system.

## 6.10 | Power consumption

We used Intel® Power Gadget tool<sup>74</sup> to measure the power consumption of the fog node (ie, SmartHerd IoT Gateway). It is a software-based power usage monitoring tool enabled for Intel® Core™ processors. While we collected the data for a longer period during the deployment for this analysis, we observed a constant power consumption pattern throughout. Thus, to give a better representation of the power consumption pattern, we present the power consumption at the fog node in a time window of 3 minutes in Figure 15. As visible from the plot, the maximum utilization is 23.2 W and for most of the time, it stays below 18 W. The utilization reaches its maximum value (during the streaming period) when there is a message stream happening between the fog node and the cloud node.

## 6.11 | Deployment cost

The total deployment cost of the system was \$11 711. Table 7 details cost of different components in the deployment.

**TABLE 7** Deployment cost

	Part Description	Quantity	Unit Price (in \$)	Extended Price (in \$)
1	WALL TRANSFORMER 240 Vac IN 24 Vac OUT 500 mA	3	16	48
2	COMMUNICATION CARD - USB	1	228	228
3	RECEIVER LRP - ENGS KIT	1	760	760
4	ANTENNA OMNI 433 MHz N-TYPE 5.5 DBI	1	91	91
5	ANTENNA OMNI LONG RANGE 430-450 MHz 2.3 m	1	148	148
6	Metal Bracket for YAGI/OMNI Antenna	2	24	48
7	CABEL RF RG213 N-TYPEM TO BNCM 8 m	2	68	136
8	TRACK A COW - COWTAG Active LRP V2 (Sensors	150	51	7650
9	2.12.0.3(1.0.6.2) ECO HERD SOFTWARE	1	684	684
10	TRANSCEIVER LRP KIT	1	1280	1280
11	COMMUNICATION CABLE 485	30	2.6	78
12	LIGHTNING PROTECTOR + CABLE	2	63	126
13	SOLAR PANEL KIT	1	224	224
14	Shipment Cost	1	210	210
	<b>Total Cost</b>			<b>\$11 711</b>

## 7 | REAL-WORLD DEPLOYMENT: LEARNINGS AND PRACTICAL EXPERIENCES

The primary challenge is to design an IoT solution to meet the specified objective given the highly variable, harsh, and resource-constrained environment in a smart dairy farming setting. This includes making the system resilient and fault tolerant to cope up with the variable farm environments, including weather-based network outages and connectivity issues because of remote location of the farm. The use of fog computing brings efficiency and sustainability to the overall IoT solution being proposed.

If we are to really envision the future of smart farming, there needs to be a solution that addresses all these challenges that a farm environment can impose. Unlike physical “things” in other IoT use cases, an animal farm environment involves higher stakes due to a direct involvement with cattle and livestock welfare. If we are to increase dependence on technology to further increase the productivity while scaling up, the platform solution needs to be flexible, adaptive, and fault tolerant, keeping the constraints in check.

With respect to our real-world deployment, deciding which sensors to choose to build the sensing infrastructure, the right application protocols, and using the best software development practices, the overall solution demands a dedicated brainstorming for a reasonable period of time to achieve efficiency and sustainability. The trade-off between initial costs for the setup of such a system would be balanced in the long term not just with payback via advantages, but also potential savings, including reduced cost of connectivity to the cloud, low-bandwidth requirements, extended battery life cycles on sensors, etc.

## 8 | CONCLUSION

In this work, we propose the emerging fog computing infrastructure as a key enabler for IoT applications and services in situational scenarios of constrained Internet connectivity. We envision a microservices-based approach to assist applications in such a hybrid fog-cloud environment, and propose a solution that addresses the connectivity and animal welfare in a smart dairy farming scenario.

In the SmartHerd IoT platform, the use of radio-based communication instead of WiFi-based sensors ensures pervasive data collection even in the scenarios of Internet and weather-based outages, while providing the advantages of decreased cost, longer battery life cycles, and wider coverage of the distribution of animals on the farm. Further, the use of SmartHerd IoT gateway leverages fog computing and enables the intelligent processing of data closer to the source, thereby supporting a real-time processing and monitoring of livestock. Insights from our real-world deployment suggest that an activity-based clustering of animals in the herd allows to monitor the behavior of dairy cows and track their welfare, and alert the farmer towards detected anomalies in the behavior of individual cows, even with cloud out of the active loop.

The system in its overall architecture and design is versatile and better adapted to Internet constraints, outages, and connectivity issues on the farm, thus making it a dependable solution for effective livestock monitoring and management in a smart dairy farming scenario.

## ACKNOWLEDGEMENTS

This work has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and is cofunded under the European Regional Development Fund under Grant Number 13/RC/2077. Mohit Taneja is also supported by CISCO Research Gift Fund.

## ORCID

Mohit Taneja  <https://orcid.org/0000-0002-2261-5722>

## REFERENCES

1. World food consumption patterns—trends and drivers. EU agricultural markets briefs. 2015. [https://ec.europa.eu/agriculture/sites/agriculture/files/markets-and-prices/market-briefs/pdf/06\\_en.pdf](https://ec.europa.eu/agriculture/sites/agriculture/files/markets-and-prices/market-briefs/pdf/06_en.pdf). Accessed December 14, 2017.
2. Gateway to dairy production and products. Quebec City, Canada: Food and Agriculture Organization (FAO) of the United Nations. <http://www.fao.org/dairy-production-products/en/#.WdTbDmhSyUl>. Accessed December 14, 2017.
3. Kharas H. *The Emerging Middle Class in Developing Countries*. OECD Development Centre Working Papers. Paris, France: Organisation for Economic Co-operation and Development; 2010.
4. Grogan A. Smart farming. *Eng Technol*. 2012;7(6):38-40. <https://doi.org/10.1049/et.2012.0601>
5. Smith D, Rahman A, Bishop-Hurley GJ, et al. Behavior classification of cows fitted with motion collars: decomposing multi-class classification into a set of binary problems. *Comput Electron Agric*. 2016;131(Supplement C):40-50. <https://doi.org/10.1016/j.compag.2016.10.006>
6. Quantified AG. Cattle health management, tags, and software. <http://quantifiedag.com/>. Accessed December 07, 2017.
7. Dairymaster Ireland. Milking parlours & machines, heat detection, scrapers, feeders, milk cooling. <http://www.dairymaster.ie/>. Accessed December 07, 2017.
8. BouMatic. [https://boumatic.com/us\\_en/](https://boumatic.com/us_en/). Accessed December 07, 2017.
9. Taylor K, Griffith C, Lefort L, et al. Farming the web of things. *IEEE Intell Syst*. 2013;28(6):12-19. <https://doi.org/10.1109/MIS.2013.102>
10. Chen M-C, Chen C-H, Siang C-Y. Design of information system for milking dairy cattle and detection of mastitis. *Math Probl Eng*. 2014;2014:1-9. <https://doi.org/10.1155/2014/759019>
11. Wark T, Swain D, Crossman C, Valencia P, Bishop-Hurley G, Handcock R. Sensor and actuator networks: protecting environmentally sensitive areas. *IEEE Pervasive Comput*. 2009;8(1):30-36. <https://doi.org/10.1109/MPRV.2009.15>
12. Brewster C, Roussaki I, Kalatzis N, Doolin K, Ellis K. IoT in agriculture: designing a Europe-wide large-scale pilot. *IEEE Commun Mag*. 2017;55(9):26-33. <https://doi.org/10.1109/MCOM.2017.1600528>
13. Steeneveld W, Hogeveen H. Characterization of Dutch dairy farms using sensor systems for cow management. *J Dairy Sci*. 2015;98(1):709-717. <http://doi.org/10.3168/jds.2014-8595>
14. Andonovic I, Michie C, Gilroy M, et al. Wireless sensor networks for cattle health monitoring. In: *ICT Innovations 2009*. Berlin, Germany: Springer-Verlag Berlin Heidelberg; 2010:21-31.
15. Schmitt P, Iland D, Zheleva M, Belding E. HybridCell: cellular connectivity on the fringes with demand-driven local cells. Paper presented at: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications; 2016; San Francisco, CA.
16. Heimerl K, Ali K, Blumenstock J, Gawalt B, Brewer E. Expanding rural cellular networks with virtual coverage. In: *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*; 2013; Lombard, IL.
17. Pentland AS, Fletcher R, Hasson A. DakNet: rethinking connectivity in developing nations. *Computer*. 2004;37(1):78-83. <https://doi.org/10.1109/MC.2004.1260729>
18. Vázquez Diosdado JA, Barker ZE, Hodges HR, et al. Classification of behaviour in housed dairy cows using an accelerometer-based activity monitoring system. *Animal Biotelemetry*. 2015;3(1):15. <https://doi.org/10.1186/s40317-015-0045-8>

19. Gandhi N, Armstrong LJ. A review of the application of data mining techniques for decision making in agriculture. Paper presented at: 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I); 2016; Noida, India.
20. Williams ML, Parthaláin NM, Brewer P, James WPJ, Rose MT. A novel behavioral model of the pasture-based dairy cow from GPS data using data mining and machine learning techniques. *J Dairy Sci.* 2016;99(3):2063-2075. <https://doi.org/10.3168/jds.2015-10254>
21. Zois D-S, Mitra U. Controlled sensing: a myopic fisher information sensor selection algorithm. Paper presented at: 2014 IEEE Global Communications Conference; 2014; Austin, TX.
22. Zois D-S, Levorato M, Mitra U. Heterogeneous time-resource allocation in wireless body area networks for green, maximum likelihood activity detection. Paper presented at: 2012 IEEE International Conference on Communications (ICC); 2012; Ottawa, Canada.
23. Zois D-S, Levorato M, Mitra U. Active classification for POMDPs: a Kalman-like state estimator. *IEEE Trans Signal Process.* 2014;62(23):6209-6224. <https://doi.org/10.1109/TSP.2014.2362098>
24. Wolfert S, Ge L, Verdouw C, Bogaardt MJ. Big data in smart farming—a review. *Agricultural Systems.* 2017;153:69-80. <http://doi.org/10.1016/j.agsy.2017.01.023>
25. Rutten CJ, Velthuis AGJ, Steeneveld W, Hogeveen H. Invited review: sensors to support health management on dairy farms. *J Dairy Sci.* 2013;96(4):1928-1952. <http://doi.org/10.3168/jds.2012-6107>
26. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: vision and challenges. *IEEE Internet Things J.* 2016;3(5):637-646.
27. OpenFog Consortium. OpenFog reference architecture for fog computing. [https://www.openfogconsortium.org/wp-content/uploads/OpenFog\\_Reference\\_Architecture\\_2\\_09\\_17-FINAL.pdf](https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL.pdf). Accessed November 12, 2017.
28. Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the Internet of Things. In: Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC); 2012; Helsinki, Finland.
29. Bonomi F, Milito R, Natarajan P, Zhu J. 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 International Publishing; 2014:169-186.
30. Stojmenovic I, Wen S. The fog computing paradigm: scenarios and security issues. Paper presented at: 2014 Federated Conference on Computer Science and Information Systems; 2014; Warsaw, Poland.
31. NASA Sees Hurricane Ophelia Lashing Ireland. Washington, DC: NASA. <https://www.nasa.gov/feature/goddard/2017/ophelia-atlantic-ocean>. Accessed December 03, 2017.
32. Irish Mirror Online. Hurricane Ophelia hits Ireland - live tracker shows what the weather will be like where you live. <http://www.irishmirror.ie/news/irish-news/hurricane-ophelia-hits-ireland-live-11349140>. Accessed December 03, 2017.
33. Taneja M, Davy A. Resource aware placement of data analytics platform in fog computing. *Procedia Comput Sci.* 2016;97:153-156. Part of special issue: 2nd International Conference on Cloud Forward: From Distributed to Complete Computing. <http://doi.org/10.1016/j.procs.2016.08.295>
34. Mahmud R, Kotagiri R, Buyya R. Fog computing: a taxonomy, survey and future directions. CoRR. 2016. <https://arxiv.org/abs/1611.05539>
35. Perera C, Qin Y, Estrella JC, Reiff-Marganiec S, Vasilakos AV. Fog computing for sustainable smart cities: a survey. *ACM Comput Surv.* 2017;50(3):32:1-32:43.
36. Foghorn systems: edge intelligence software for IIoT. <https://www.foghorn.io/>. Accessed November 03, 2017.
37. IBM Watson - IBM and Cisco cloud analytics. <https://www.ibm.com/internet-of-things/partners/ibm-cisco/>. Accessed November 03, 2017.
38. Sarkar S, Chatterjee S, Misra S. Assessment of the suitability of fog computing in the context of Internet of Things. *IEEE Trans Cloud Comput.* 2018;6(1):46-59. <https://doi.org/10.1109/TCC.2015.2485206>
39. Taneja M, Davy A. Poster abstract: resource aware placement of data stream analytics operators on fog infrastructure for Internet of Things applications. Paper presented at: 2016 IEEE/ACM Symposium on Edge Computing (SEC); 2016; Washington, DC.
40. Taneja M, Davy A. Resource aware placement of IoT application modules in fog-cloud computing paradigm. Paper presented at: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM); 2017; Lisbon, Portugal.
41. Chiang M, Zhang T. Fog and IoT: an overview of research opportunities. *IEEE Internet Things J.* 2016;3(6):854-864. <https://doi.org/10.1109/JIOT.2016.2584538>
42. Ramirez W, Masip-Bruin X, Marin-Tordera E, et al. Evaluating the benefits of combined and continuous fog-to-cloud architectures. *Computer Communications.* 2017;113(Supplement C):43-52. <https://doi.org/10.1016/j.comcom.2017.09.011>
43. Mouradian C, Naboulsi D, Yanguis S, Glitho RH, Morrow MJ, Polakos PA. A comprehensive survey on fog computing: state-of-the-art and research challenges. *IEEE Commun Surv Tutor.* 2017;20(1):416-464.
44. Hao Z, Novak E, Yi S, Li Q. Challenges and software architecture for fog computing. *IEEE Internet Comput.* 2017;21(2):44-53. <https://doi.org/10.1109/MIC.2017.26>
45. Zheleva M, Bogdanov P, Zois D-S, Xiong W, Chandra R, Kimball M. Smallholder agriculture in the information age: limits and opportunities. In: Proceedings of the 2017 Workshop on Computing Within Limits (LIMITS); 2017; Santa Barbara, CA.
46. Jukan A, Masip-Bruin X, Amla N. Smart computing and sensing technologies for animal welfare: a systematic review. *ACM Comput Surv.* 2017;50(1):10:1-10:27. <https://doi.org/10.1145/3041960>
47. Vasisht D, Kapetanovic Z, Won J, et al. FarmBeats: an IoT platform for data-driven agriculture. In: Proceedings of the 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI); 2017; Boston, MA.
48. IEEE. IEEE Standards Association - IoT ecosystem study. <http://standards.ieee.org/innovate/iot/study.html>. Accessed November 19, 2017.
49. McLaughlin K. Gaps in 4G network hinder high-tech agriculture: FCC prepares to release \$500 million to improve coverage. <http://www.bendbulletin.com/home/4535283-151/gaps-in-4g-network-hinder-high-tech-agriculture>. Accessed December 02, 2017.

50. AgGateway. <http://www.aggateway.org/>. Accessed December 03, 2017.
51. Fowler SJ. *Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization*. Sebastopol, CA: O'Reilly Media; 2017.
52. Familiar B. *Microservices, IoT, and Azure: Leveraging DevOps and Microservice Architecture to Deliver SaaS Solutions*. New York, NY: Apress; 2015.
53. Balalaie A, Heydarnoori A, Jamshidi P. Microservices architecture enables devOps: migration to a cloud-native architecture. *IEEE Software*. 2016;33(3):42-52. <https://doi.org/10.1109/MS.2016.64>
54. de Sousa Ramalho F. SmartEdge: fog computing cloud extensions to support latency-sensitive IoT applications. 2016. <https://repositorio.ufrn.br/jspui/handle/123456789/22557>. Accessed September 13, 2017.
55. Yannuzzi M, van Lingem F, Jain A, et al. A new era for cities with fog computing. *IEEE Internet Comput*. 2017;21(2):54-67. <https://doi.org/10.1109/MIC.2017.25>
56. Krylovskiy A, Jahn M, Patti E. Designing a smart city Internet of Things platform with microservice architecture. Paper presented at: 2015 3rd International Conference on Future Internet of Things and Cloud; 2015; Rome, Italy.
57. Dell. Dell edge gateway 5100. <https://goo.gl/vWcdSW>. Accessed March 12, 2019.
58. Intel. Intel<sup>®</sup> IoT Gateway. <https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/gateway-solutions-iot-brief.pdf>. Accessed March 12, 2019.
59. MQTT. <http://mqtt.org/>. Accessed August 03, 2017.
60. Getting to know MQTT. <https://www.ibm.com/developerworks/library/iot-mqtt-why-good-for-iot/index.html>. Accessed August 03, 2017.
61. Lee S, Kim H, Hong D-K, Ju H. Correlation analysis of MQTT loss and delay according to QoS level. Paper presented at: The International Conference on Information Networking 2013 (ICOIN); 2013; Bangkok, Thailand.
62. MQTT essentials part 6: quality of service 0, 1 & 2. <https://www.hivemq.com/blog/mqtt-essentials-part-6-mqtt-quality-of-service-levels/>. Accessed March 12, 2019.
63. Hong K, Lillethun D, Ramachandran U, Ottenwalder B, Koldehofe B. Mobile fog: a programming model for large-scale applications on the Internet of Things. In: Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing (MCC); 2013; Hong Kong, China.
64. Taneja M, Jalodia N, Byabazaire J, Davy A, Olariu C. Smartherd- connected\_cows\_demo.mp4-Google Drive [video]. <https://drive.google.com/file/d/1QlrKSp8SkAZRRAFQDQHdPXu1TVYaVv3/view>. Accessed October 04, 2018.
65. Cattle | animal behaviour. <http://www.animalbehaviour.net/cattle/>. Accessed December 03, 2017.
66. Stephenson MB, Bailey DW. Do movement patterns of GPS-tracked cattle on extensive rangelands suggest independence among individuals? *Agriculture*. 2017;7(7). <https://doi.org/10.3390/agriculture7070058>
67. Minta SC. Tests of spatial and temporal interaction among animals. *Ecological Applications*. 1992;2(2):178-188.
68. Taneja M, Byabazaire J, Davy A, Olariu C. Fog assisted application support for animal behaviour analysis and health monitoring in dairy farming. Paper presented at: 2018 IEEE 4th World Forum on Internet of Things (WF-IoT); 2018; Singapore.
69. Byabazaire J, Olariu C, Taneja M, Davy A. Lameness detection as a service: application of machine learning to an internet of cattle. Paper presented at: 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC); 2019; Las Vegas, NV.
70. psutil- PyPI. <https://pypi.org/project/psutil/>. Accessed March 13, 2019.
71. Konečný J, McMahan HB, Ramage D, Richtárik P. Federated optimization: distributed machine learning for on-device intelligence. *CoRR*. 2016. <https://arxiv.org/abs/1610.02527>
72. Taneja M, Jalodia N, Davy A. Distributed decomposed data analytics in fog enabled IoT deployments. *IEEE Access*. 2019;7:40969-40981.
73. Chang T-C, Zheng L, Gorlatova M, Gitau C, Huang C-Y, Chiang M. Decomposing data analytics in fog networks. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems (SenSys); 2017; Delft, The Netherlands.
74. Intel. Intel<sup>®</sup> Power Gadget | Intel<sup>®</sup> Software. <https://software.intel.com/en-us/articles/intel-power-gadget-20>. Accessed March 14, 2019.

**How to cite this article:** Taneja M, Jalodia N, Byabazaire J, Davy A, Olariu C. SmartHerd management: A microservices-based fog computing-assisted IoT platform towards data-driven smart dairy farming. *Softw: Pract Exper*. 2019;49:1055-1078. <https://doi.org/10.1002/spe.2704>