Wireless Sensor Based Data Analytics for Precision Farming

Kriti Bhargava (No. W20066590)

Department of Computing and Mathematics

Waterford Institute of Technology

Thesis submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy

Supervisors: Dr. Stepan Ivanov and Prof. William Donnelly

Submitted to Waterford Institute of Technology, April 2019

I would like to dedicate this thesis to my family who have been my inspiration and strength throughout the course of this study.

Declaration

I hereby declare that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy, is entirely my own work and has not been taken from the work of others save to the extent that such work has been cited and acknowledged within the text of my work.

Kriti Bhargava

Submitted to Waterford Institute of Technology, April 2019

Acknowledgements

This thesis is the result of the guidance, love and prayers of a lot of people who never failed to believe in me. I take this opportunity to express my deepest gratitude and appreciation for all their support throughout this journey.

I begin by thanking my supervisors, Dr. Stepan Ivanov and Prof. William Donnelly, for their invaluable guidance, critiques, encouragement and patience throughout these years. Stepan, it has been such an honor to be your first PhD student. It is by your example that I have learnt to persevere and work hard to achieve my goals. I will be forever grateful for all your help and motivation that made my PhD experience productive and stimulating. Willie, your passion and ethics for research are truly inspiring. I am extremely thankful for the excellent example that you have provided as a successful researcher and professor.

I am thankful to my research unit manager at TSSG, Mr. Eric Robson, for his enthusiastic support and encouragement, and the entire data mining group, especially, the Precision Dairy team - Dr. Chamil Kulatunga and Mr. Dixon Vimalajeewa, for their immeasurable help and advice over the years. I extend my gratitude to the entire team and staff at TSSG, and the wider community at WIT, especially Dr. Brendan Jennings and Dr. Sasitharan Balasubramaniam, for their constant support throughout these years and providing me with various opportunities to grow as a researcher. I also thank the Precision Dairy team at TEAGASC, Cork, Ireland, especially Dr. Laurence Shalloo, Dr. Niall O'Leary, and Mr. Diarmuid McSweeney for their help during the course of field experiments. The funding of Precision Dairy provided by SFI is also gratefully acknowledged.

I thank my parents, Mr. Mukul Bhargava and Mrs. Preeti Bhargava, my brother Mr. Rishabh Bhargava, and my extended family for being a constant source of moral and emotional support. Your encouragement, love and faith gave me the strength to pursue my dreams. Last but not the least, I thank all my friends for their constant motivation. I, especially thank my friends from Ireland, Dr. Radhika Loomba, Mrs. Mandy Lalrindiki, Dr. Michael T. Barros, Mr. Sidhant Hasija, Mr. Genaro Longoria and Ms. Kanika Sharma, for all their help, endless conversations and wonderful memories over the years.

I couldn't have done it without you all.

Abstract

With advances in the Internet of Things, the use of Wireless Sensor Networks (WSN) has been widely proposed for monitoring and automation of farm processes under the umbrella of Precision Farming. In conventional WSN systems, data gathered by sensors is transmitted to remote cloud servers for analysis. These systems, however, incur delay in getting insights into the processes due to the high volume of data generated on the farms coupled with the poor Internet connectivity. This negatively affects the delay-sensitive applications that require immediate response. The Fog Computing paradigm suggests a shift in intelligence from the cloud towards the network edges to cater to the requirements of delay-sensitive applications. It proposes the use of compute, memory and networking resources available at edge devices such as gateways, routers and sensors to reduce dependency on cloud and, thereby, improve the responsiveness of the system. In this work, we focus our attention on the development of on-board intelligence for sensor devices in the context of Precision Farming. Firstly, we identify gaps in the current WSN-based Precision Farming technologies and examine the suitability of Edge Mining, an instance of Fog Computing, for real-time event detection in farm processes. In addition, we propose an extension of the Edge Mining approach to allow for context-aware operation of sensor devices in farms. A WSN prototype consisting of a plug-n-play universal sensor device and gateway node has been designed to validate the performance of these algorithms. Next, we develop two cooperative frameworks - Collaborative Edge Mining and Iterative Edge Mining, to represent the analytic problems as a set of cooperative Edge Mining-based tasks for parallel and sequential analysis respectively within WSN. The cooperation between tasks allows for scaling of analysis within and across devices to improve computational capability of the network. Finally, we discuss resource management through cooperative computing within WSN. Cooperation between devices is considered to improve accuracy and timeliness of in-network analytics while optimizing the use of energy resources of sensor devices for improved network longevity.

Contents

Chapter 1

Introduction

1.1 Background and Motivation

In this section, we provide the background and motivation for the research presented in this thesis. We describe modern-day Wireless Sensor Networks (WSN) in section 1.1.1, followed by the envisioned future WSN in section 1.1.2. We discuss the Fog Computing paradigm that, in our opinion, lays foundation for the design of future WSN in section 1.1.3, and present the key design challenges that must be addressed for realization of future WSN in section 1.1.4.

1.1.1 Wireless Sensor Networks today

A typical WSN consists of small, inexpensive, battery-operated sensor devices that are deployed over a large area for continuous monitoring of the environment, and gateway nodes to collate the sensor data and upload it onto the cloud for analysis and sharing with the end-users. With developments in Information and Communication Technologies (ICT) along with design of micro-services, WSN presents itself as a powerful tool for real-time monitoring/surveillance, and has been identified as the key enabler of the Internet of Things (IoT) paradigm. As a result, significant attention has been given to improve the design and capabilities of WSN over the past decade. For instance, tremendous efforts have been made to increase energy efficiency and computation capability of sensor devices. Conventionally limited to sense and send, certain tasks assigned to these devices, today, are of relatively high computational intensity (e.g. data fusion [1], localization [2]). A few sensor devices, today, are also capable of harvesting energy from natural sources such as wind and solar energy [3] for continuous use. Furthermore, several improvements have been achieved in development of communication technologies for WSN. Various low-power, short-range (e.g. Zigbee [4], bluetooth-low-energy [5]) and long-range (e.g. LoRa [6]) communication techniques have been designed and used in the recent past. A comparative study of the emerging wireless technologies for WSN is presented in [7]. Design of these technologies ensures the suitability of using WSN systems in both urban and rural areas.

Standardization of communication technologies for WSN along with the simultaneous efforts to allow inter-operability of sensor devices, have diversified the types of sensors being deployed in a given system. A WSN, today, may include devices not only of different types (e.g. temperature, CO_2) but also of different makes and models (e.g. TelosB, Micaz, Iris). This diversification makes WSN systems thorough and multi-purpose for use in different application domains such as agriculture, health-care, transportation, environmental sensing and industrial monitoring. In health-care, for instance, the use of WSN has been proposed for numerous applications such as remote patient monitoring and tracking, remote diagnostics and examination, and medical asset tracking as discussed in [8]. WSN will allow design of early warning systems and reduce costs otherwise incurred in periodic hospital checks. In transportation, WSN has been used to build smart roads, rails and runways, augmented maps and Intelligent Transportation Systems (ITS) to improve safety of people and reduce delays while travelling. An instance of a WSN system for ITS has been presented in [9]. In agriculture, use of WSN has been suggested for monitoring and automation of tasks relating to environment management, crop growth, yield mapping, and health and mobility management of livestock under the umbrella of Precision Farming [10]. A detailed survey on characterization and classification of WSN applications is presented in [11].

An important aspect of WSN is timely and efficient management of data for extraction of information. The use of Cloud Computing has been suggested for analyzing the ‘big data’ collected by sensors. Accordingly, several communication protocols for efficient transmission of data from sensor network to cloud have been proposed, to date. Data collected by sensors can be categorized as either delay-critical, delay-tolerant or delay-insensitive depending on the application scenario. Subsequently, the networking approach used within WSN can be either periodic or event-driven, deterministic or opportunistic, topology or location-based, direct-delivery or multi-hop to meet the latency, accuracy and energy requirements [12]. For instance, the authors in [13] use opportunistic networking to transfer delay-tolerant data in smart dairy farms. The approach uses milking cows to gather data from farms that is transferred to a cloud gateway, hosted in the parlour, using direct-delivery as cows go for milking. Alternatively, numerous routing protocols have been designed for data transmission in a multi-hop WSN. In [14], the authors present a context-aware, gradient gravity routing protocol that uses interactions between WSN and end-user devices for targeted delivery of data. The authors in [15] discuss the design of a fuzzy logic-based multi-dimensional link quality indicator that is integrated in the Collection Tree Protocol (CTP) for reliable data communication in multi-hop WSN. The use of LoRaWAN for low-power data transfer over wide area networks has also been discussed in [6].

For analysis of data, Cloud Computing facilitates on-demand and pay-as-you-go access to computational power, memory, and other resources on remote servers via the Internet. Several cloud-based data mining platforms have been designed, to date. Apache Hadoop software library, for instance, is a framework that allows for distributed batch processing of large data sets across clusters of computers [16]. Alternatively, stream processing platforms including Storm [17] and Spark [18] by Apache have also been developed to improve timeliness of analysis. MapReduce is a widely used distributed programming model that is used by these platforms for easy parallelization of the processing tasks [19]. A large number of distributed and parallel data mining algorithms have been implemented using the MapReduce model. These algorithms are roughly divided into 4 categories - Association Rule Mining (e.g. Apriori, Frequent Pattern Growth), Classification (e.g. Decision Trees, Neural Networks), Clustering (e.g. K-means) and Stream data mining (e.g. MILE) as presented in [20]. While Cloud Computing enables informed decision making, big data mining brings in series of new challenges and complexities related to data storage and energy consumption. Additionally, big data mining algorithms must address issues related to scalability, heterogeneity and security of data. Ongoing work is, therefore, focused on improving the existing approaches to deliver secure and timely information to the end-users for improved decision making.

1.1.2 Wireless Sensor Networks tomorrow

Despite the recent advances in WSN technology, continuous efforts are being made to further improve the design and efficiency of sensor devices. This ranges from the design of sensor devices for large IoT objects such as buildings and vehicles to small personal devices such as mobile phones, health monitoring wearables and medical pills. Several attempts at reducing the dimensions of sensor devices have been made to make them more suited for embedded systems, as well as allow their deployment in relatively inaccessible areas, for instance inside human body, for continuous monitoring. With parallel advances in nanotechnology, design of nano-sensors is a natural extension of WSN systems. Development of man-made or natural nano-sensors using techniques such as top-down lithography, bottom-up assembly, and molecular self-assembly is being investigated. Moreover, design of networking protocols suitable for communication at nano-scale has been proposed to pave way for nano-networks within WSN. Such nano-networks are expected to monitor processes that ensue at molecular scale for design of early forecasting systems. However, due to limitations of size and resources, these networks must be integrated with WSN, in future, for communication and analysis of data. Furthermore, with increase in deployment of WSN systems in rural areas, improvements in the design of energy harvesting WSN must be realized. Energy harvesting WSN are expected to enable indefinite use of sensor devices while reducing the energy footprint of operation to ensure 'green computing'. This is also useful for the design of nano-sensors where inclusion of an external battery-unit may not be feasible.

The continuous use of future WSN is expected to generate unprecedented volume and variety of data causing information overload on the cloud. While Cloud Computing offers a cost-effective solution for analysis of sensor data, such a data explosion would result into communication and processing delays on cloud servers, in turn, leading to latency in getting useful insights in the data. These delays will be further enhanced in remotely connected application domains such as smart agriculture [21] that suffer from poor communication bandwidth and may not have a continuous, ubiquitous network connectivity to the cloud. To mitigate these challenges, we envisage the design of future smart sensor devices that would incorporate certain intelligence to perform real-time, on-board data processing before transmitting data to cloud. Besides deciding where and how to route data packets, we presume that the sensor devices will be capable of deciding what information to send to cloud on-the-go. As such, the design of future WSN is expected to be user-centric and capable of performing in-network analytics to not only reduce redundancy in sensor data but generate real-time alerts based on the given specifications. Through localized processing of data, future WSN design would allow reduced communication to the cloud and facilitate optimization of both sensor and cloud resources. In doing so, it would also improve the security and privacy of the sensor data.

Furthermore, owing to the dynamic nature of application/user requirements, we envisage the sensor-based intelligence to exhibit flexibility to adapt to these changes. That is, WSN-based services must be re-configurable on-the-go to meet changes in requirements so as to ensure computational accuracy while optimizing the use of device resources. Given the large number of devices in a typical WSN, we also envision cooperation between nearby smart sensor devices to improve computational capability of the network. Collaborative in-network analysis between different sensor devices would allow integration of various types of information for provision of more complex services, which are beyond the capabilities of an individual device, within the network. Moreover, cooperation through computation offloading among devices would allow load balancing across the network for improved energy balance and, in turn, network longevity. Collective participation of sensor devices in analytic tasks could also lead to a potential improvement in the accuracy of predictions made by individual devices, thereby, improving the performance of WSN. Finally,we expect the design of future WSN to facilitate contextualization of the sensor data/services to improve the decision making process. Location and user-awareness owing to on-board data analysis would help develop spatial correlation between data-sets to add relevance to the data. Moreover, location-awareness could help improve communication of data to the cloud by means of opportunistic and location-based networking, to improve energy utilization within WSN.

1.1.3 Fog Computing

The Cloud Computing approach facilitates on-demand use of a cluster of servers, hosted remotely on the Internet, for storage, analysis and management of data. It allows easy scaling up and down of resources in case of variable task loads to avoid under/over-provisioning of resources. By facilitating such flexibility in resource utilization, it saves the capital expenditure, in particular the energy cost of maintaining the ICT infrastructure, incurred by the small and medium enterprises. This computing architecture functioned well until WSN systems generated some large data-sets in remotely connected applications such as in rural agriculture where back-haul connectivity is limited between the remote rural farms/factories and the cloud. The intermittent Internet connectivity in such applications causes unnecessary delay in communication and computation of data, which is undesirable especially in delay-critical scenarios. This issue is further enhanced as advances in IoT and mobile technology have caused majority of our interactions with the world to be increasingly dependent on real-time, context-aware services. For IoT to ensure reliable delivery of services to end-users and improve Quality of Experience, we, therefore, suggest the design of future WSN to incorporate smart sensors that are self-sustaining and operate autonomously to perform real-time, on-board data analysis. To do so, we expect the design of future WSN to borrow principles from Fog Computing.

Fog Computing [22] is a new computing paradigm that proposes partial migration of intelligence away from the cloud and towards the network edges. It follows the concept of data gravity and suggests the use of compute, storage and networking resources available closer to the data sources for hierarchical and scalable analysis of data along the sensor-cloud continuum, prior to sending it to the cloud. On one hand, Fog Computing aims at optimizing the network resource utilization through managing, disseminating and responding to queries at the edge. On the other hand, it allows for faster insights into the data despite intermittent connectivity or high mobility of devices. Due to resource-constraints of the edge, however, the fog works in junction with the cloud for big data analysis. While edge analytics is suggested for real-time alert generation, results of edge-based analysis are sent to cloud for future analysis and deep learning. The reduced packet transmissions to cloud allows for improved energy profile of devices as well as reduced resource (memory and energy) requirements in the cloud. The suitability of Fog Computing over Cloud Computing for IoT applications has been discussed in [23] and [24]. The studies highlight the benefits of using the Fog Computing approach, especially, in scenarios where majority of the applications are delay-critical. A reference architecture has been recently proposed for Fog Computing in [25]. The architecture is based on a set of core principles (referred to as pillars) including security, scalability, autonomy, agility and programmability that must be adhered to while designing future IoT/WSN systems.

Apart from the above considerations, one of the key challenges in Fog Computing is to determine what and how much computation to offload at the edge in order to balance the resource trade-off between communication and computation tasks on the edge devices. This is primarily determined based on the application requirements (latency vs. accuracy) as well as the type of fog agent (edge device) used for analysis. Over the past few years, numerous interpretations of fog agents within IoT have been discussed. While some approaches propose the use of computational resources at network devices such as switches, with increase in number of mobile-phones, a number of studies have suggested the use of free computation slots on user mobile phones [26]. A cluster of mobile-phones in vicinity is considered as an ad-hoc cloudlet and used for computation offloading [27]. This concept has evolved in the recent years and has come to be known as Mobile Edge Computing (MEC) where mobile operators leverage resources of the edge devices rather than the centralized servers in cloud computing for data processing. Recent studies further extend the concept of fog to extreme edge of the network, i.e. sensor devices, to cater to WSN applications that lack such relatively powerful edge devices. This is primarily supported by the tremendous improvements in computation capabilities of sensor devices [28] as well as the design of pervasive low-power wireless technologies like ULP-PAN and LP-WAN. Certain in-network processing within WSN (referred here as sensor analytics) has already been proposed. In [29], for instance, the authors discuss use of in-network data fusion to improve the accuracy of a WSN-based tracking system. While data fusion techniques allow localized data reduction, they lack predictive capabilities and limit the analytics performed within WSN. In this work, we review the existing sensor analytic techniques and aim to extend the current work for the design of contextualized, user-centric services within WSN in the context of Precision Farming.

1.1.4 Challenges

While the opportunities for WSN are vast, some fundamental gaps still exist in the understanding and preparedness of WSN at scale. We envision that features of future WSN would include autonomy, context-awareness, flexibility and self-sufficiency, resulting in their rapid adoption in varied application domains. These features pose certain design challenges, some of which are considered in this work. The challenges below are presented in the context of Precision Farming (discussed in detail in section 2.1). However, the relevance of these challenges is not limited exclusively by the example, and remains significant for realizing future WSN in other application domains.

- **C1. Robustness in the absence of external connectivity and diminishing resources**

  WSN systems are usually deployed for monitoring remote areas, an easy access and continuous network connectivity to which is often unavailable. To ensure uninterrupted, efficient delivery of services, these systems must be capable of autonomous operation for real-time analysis of data without relying on third party components such as cloud servers. That is, on-board functionality of sensor devices must extend beyond data collection and communication to include significant analytics for timely detection of events. Depending on the application scenario and availability of cloud resources, the functionality should offer various levels of autonomy to WSN operation. In extreme conditions (with no Internet connectivity) sensor-based analytics should facilitate real-time services within WSN. For instance, a WSN system in rural farms should operate autonomously to perform localization of animals as they move around the farm for on-farm navigation and real-time detection of behaviour anomalies. However, a lack of suite of analytics suitable for sensors has been identified. While certain data mining techniques have been designed for edge devices such as mobile-phones, these techniques are usually too complex for deployment on sensors. Design of low-power, light-weight analytics for sensor devices is, therefore, of utmost importance.

- **C2. Robustness when device capabilities are limited**
  Despite the tremendous improvements in design and capabilities of sensor devices, these devices are characterized by limited compute-power, memory and energy resources. In addition, their design is constrained with the type and number of sensors featured, and communication range. The analytics deployed on sensor devices must adhere to these resource constraints for smooth operation. As such, the analytics adds to the WSN functionality besides its original data collection and communication tasks.

Hence, the WSN should perform analytics such that it does not significantly impact the monitoring tasks itself. A trade-off between analytics functionality implemented by the WSN and resource expenditure must be considered. Furthermore, current sensor analytic approaches consider operation on sensor devices in isolation with each other, thereby, limiting functionality delegated to WSN. Future WSN must facilitate outsourcing and collaboration of analytic tasks to sensor devices in close vicinity of each other to enable scaling of analysis within the network. Such cooperation would also ensure load balancing among devices to prevent formation of bottlenecks. Cooperative analytics between different environmental sensors (e.g. temperature and humidity) in a farm, for instance, can be used to deduce the joint effect of the farm variables on animal health to ensure their well-being.

- **C3. The need to adapt**
  Owing to difficulty in replacing sensor devices in remote areas, lifetime maximization is one of the key challenges in the design of a WSN system. Whereas on-board analysis has the potential to improve energy profile of devices (through reduced packet transmissions to cloud), data analytics with very stringent application deadlines (that require higher clock speeds) may exhaust the device resources and cause failure. As such, the configuration of sensor analytics is determined by the application requirements that usually vary over time. The functionality in sensor devices today, however, is hard-coded and does not allow re-configuration based on changing requirements and scenarios. To optimize the use of on-board resources, analytics should be flexible and must accommodate changes on-the-go to maintain the trade-off between accuracy of results and energy efficiency (i.e. longevity) of the devices. For instance, while continuous (possibly rapid) computations may be required for localization of animals as they move around a farm during the day, the frequency of localization should be reduced during the evening as animals rest within the sheds. Moreover, while certain applications require precise analysis (e.g. sub-meter location accuracy), others may have more flexible requirements, thereby, requiring reduced computations.

- **C4. Energy Efficiency**
  The sensor devices, today, are primarily battery-powered. Even with energy harvesting in future, these devices will remain energy constrained owing to limited battery capacity. It is, therefore, of utmost importance to ensure efficient use of these scare resources. While computation forms one aspect of energy consumption on these devices, communication of data to cloud is the most expensive task performed by

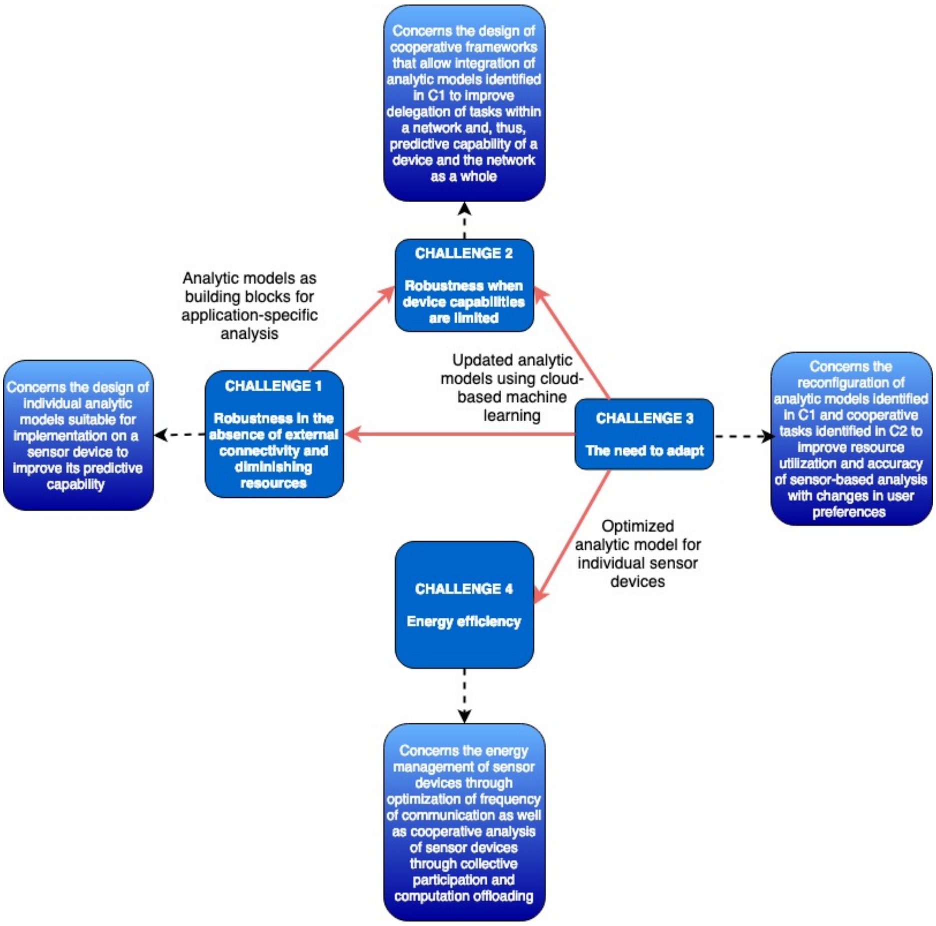


Fig. 1.1 Conceptual diagram to depict the relationship between WSN design challenges

the devices. The energy cost depends on frequency (on-off of radio) and number of packet transmissions, and the distance between devices and the cloud. Endeavours to minimize the transmissions over long distances to cloud must be made to optimize energy efficiency of the devices while ensuring timely access to information. Care must be taken to avoid the formation of bottlenecks so as to improve the network lifetime. Furthermore, cooperative computing via computation offloading must be considered to make efficient use of computational resources within the network and avoid over-utilization of individual devices. Such cooperation would ensure energy balance across devices, thereby, improving sustainability of the network operation. Energy-aware computation offloading, for instance, could improve computation capabilities of a heterogeneous energy harvesting WSN by minimizing total energy cost required by analytic tasks.

The research challenges discussed above complement each other as shown in fig. 1.1 and must be addressed sequentially to allow for the design of future WSN. The first challenge, C1, is concerned with the design of robust sensor devices that are capable of performing on-board analytics in addition to sense and send tasks for improved responsiveness of WSN applications, especially, in the absence of external resources and Internet connectivity. The pool of analytics must adhere to the resource constraints of sensor devices to ensure smooth and uninterrupted operation. Next, C2 is concerned with the design of frameworks that allow cooperation between the analytic tasks (identified in C1) distributed within and across sensor devices. These frameworks should allow improved scalability of in-network analysis as well as ensure load balancing and fairness among devices. As such, while C1 considers the design of analytic models suitable for sensor-based implementation, C2 considers the outsourcing and collaboration between these analytic models for improved network-wide analytics. The third challenge, C3, addresses the need to adapt the analytic models identified in C1 as well as the interaction between the models based on frameworks designed in C2 with changes in user requirements to improve the accuracy of predictions. The adaptation is also aimed to enhance the energy efficiency of sensor operation and complements the last challenge C4. C4, in particular, considers energy management through collective and cooperative analysis across devices to jointly optimize computation and communication energy consumption of the network as a whole.

1.2 Research Scope of the thesis

In this section, we discuss the scope of research presented in this thesis. Section 1.2.1 presents the limitations of current WSN systems that are the focus of this work, followed by the research objectives in section 1.2.2.

1.2.1 Limitations

The compute, memory and battery constraints of sensor devices pose significant limitations to their adoption. While increasing capacity of each sensor device or installing additional infrastructure (e.g. increase in number of sensors or gateways) would improve the capabilities of a WSN system, it is often cumbersome and leads to an increase in the operational costs. In this work, we, therefore, aim at achieving performance improvements in WSN by incorporating intelligence within sensor devices to allow real-time predictions as well as optimize the use of already existing resources. We also consider application and context-aware operation of sensor devices to manage the trade-off between prediction accuracy and resource-utilization. The limitations as considered by this work are discussed as under.

- **Sensor analytics:** The design of on-board intelligence for sensor devices is limited to data analytic techniques. While the use of machine learning (ML) algorithms such as neural networks over WSN topology would also facilitate real-time analysis, the learning involved with these algorithms is computationally-intensive and may adversely affect the performance of WSN. Moreover, the learning is particularly difficult in case of dynamic network topology with large number of mobile nodes, such as WSN consisting of animal wearable devices as considered in this work. Therefore, we consider a WSN architecture wherein light-weight analytic tasks are performed on sensor devices while the compute-intensive learning is performed on the cloud. Interaction between sensor-cloud is facilitated to allow exchange of sensor data and update of analytic models. Use of sensor analytics ensures efficient and timely response to events and thus, correlates with the challenge ***C1 (Autonomy)***.

- **Isolated analysis:** As discussed previously, WSN comprises of devices that are resource-constrained (challenge ***C2 (Resource-constrained devices)***). In addition, current WSN systems consider sensor operation in isolation from each other. This limits the extent of analytics that can be performed by WSN. To allow scalability of analysis, we study cooperation and integration of different sensor analytic tasks, as opposed to the use of other more powerful edge devices (e.g. gateways or mobile-phones) which are often unavailable in remote applications such as smart farming, or use of data from third party services (e.g. weather stations). While cooperation between sequential tasks is considered for vertical scaling of analysis within a device, cooperation between parallel tasks is realized for horizontal scaling of analysis across the network. Such cooperation within WSN not only allows efficient use of existing resources by offering load balancing but also improves context-awareness of device operation with respect to each other.

- **Performance and resource trade-off:** Since analytic tasks add to the existing functionality of sensor devices, a resource trade-off between computation and monitoring tasks must be resolved. Whereas devices today do not allow ease of programmability, we strive to offer flexibility in sensor-based analytics to optimize frequency of computation while meeting application requirements and ensuring continuous monitoring. The flexibility is incorporated using ML-based learning on the cloud, and delay-tolerant integration of sensors with cloud. Note, however, we do not consider the cost-metric associated with the use of cloud resources for performance measurement. Moreover, we consider cooperative analytics as well as context-aware communication within WSN to optimize the energy consumption of devices. The optimal number of sensor devices for such analysis or handover of tasks between devices has not been considered. This limitation correlates with the challenges ***C3 (Flexibility)*** and ***C4 (Energy Efficiency)***.

1.2.2 Objectives

To address the challenges identified in section 1.1.4, our work focuses on the design of an autonomous WSN system capable of integrating, processing and managing data from different sensors to deliver real-time services. To do so, firstly, we explore and design a suite of analytic techniques suitable for sensor devices. In particular, we focus on the development of on-board intelligence for wearable sensor technology in the context of Precision Farming. Secondly, we design cooperative frameworks that allow interaction between different analytic tasks for scaling of analysis within WSN. And finally, we address the issue of flexibility in device intelligence for adapting sensor operation to changing user-requirements and ensuring energy management within WSN. The objectives of the study are represented by the following Research Questions (RQ).

- ***RQ1. Sensor analytics:*** *What pool of data analytics techniques are most suitable for on-board execution on miniature IoT devices? How are these techniques relevant in the context of Precision Farming?*
- ***RQ2. Cooperative WSN:*** *What algorithmic frameworks can be designed to represent real-life services as a set of cooperative data analytic tasks deployed within WSN?*
- ***RQ3. Flexibility and Energy Management in Cooperative WSN:*** *How can the services be re-configured on-the-go to meet the changes in application requirements while ensuring resource efficiency of the sensor devices?*

1.3 The approach

In this section, we describe the research approach that has been developed to answer the research questions. The three questions are addressed sequentially, and each research question is based on findings of previous questions. The methodology used to validate the findings of the research questions is discussed in section 1.3.1.

- ***RQ1. Sensor-based analytics:*** To determine the design considerations that must be

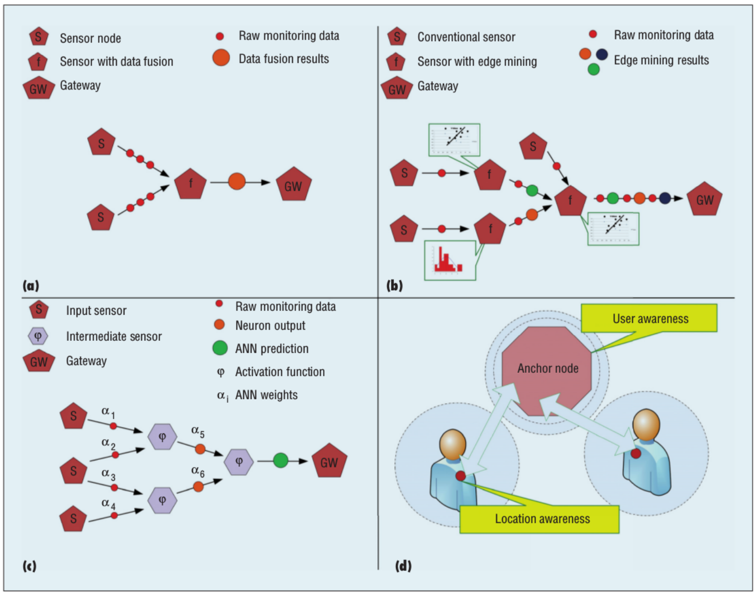
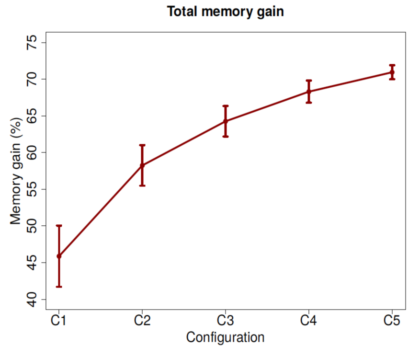


Fig. 1.2 Sensor analytics a) Data fusion b) Edge Mining c) ANN d) Location awareness [30]

taken into account while developing the suite of analytics for sensor devices, we first identify the limitations of the current WSN solutions and the challenges to be addressed for the realization of future WSN, in the context of Precision Farming [30]. As highlighted by existing studies, a major drawback of the existing WSN technologies is lack of decision-support systems that allow real-time decision making for automation of processes. While the existing systems enable fine-grained monitoring of a farm, they fail to provide timely and easy access to insights that can help farmers make informed decisions. To overcome this limitation and improve responsiveness of the existing systems, we consider the suitability of sensor-based analytics. We review certain analytic techniques such as Data Fusion [1], Edge Mining [31] and Artificial Neural Networks [32] (fig. 1.2) that have been previously proposed for sensor-based execution, and reflect on the benefits and challenges associated with their use in Precision Farming applications. These techniques incorporate some level of intelligence and can facilitate on-board alert generation for delay-critical events. Moreover, they reduce redundancy in data and improve the quality of data exchange within the network to, in turn, enhance energy efficiency of sensor operation. We present the pool of analytics that would best fit the capability and constraints of sensor devices, while meeting the application requirements. The select techniques are light-weight for ease of implementation on sensor devices, and are extensible to act as building blocks for design of in-network services in future WSN. Chapter 2 provides a detailed discussion on the existing techniques.

procedure :
Calculate new state
Using dEWMA filtering
$v_{x,t} \leftarrow \alpha_x * z_{x,t} + (1-\alpha_x) * (v_{x,t'} + r_{x,t'} * (t-t'))$
$r_{x,t} \leftarrow \beta_x * (v_{x,t} - v_{x,t'})/(t-t') + (1-\beta_x) * r_{x,t'}$
Estimate new state
Using linear extrapolation
$v'_{x,t} \leftarrow \begin{bmatrix} 1 & (t-t') \\ 0 & 1 \end{bmatrix} v_{x,t'}$
Eventful?
yes, if $(|v'_{x,t} - v_{x,t}| > \varepsilon_x)$
then, store $(v_{x,t}, r_{x,t}, t)$

(a)



(b)

Fig. 1.3 (a) L-SIP algorithm (b) Memory gain [34]

Of the existing techniques, the Edge Mining approach presents a particularly good fit for the design of smart sensor devices. It proposes the use of data mining techniques on-board sensor devices such that instead of reporting raw data, each sensor reports only the occurrence of significant, application-specific events. The sensor systematizes historical readings using simplistic models based on linear, decision-tree or histogram representation of data, which are updated continuously with regards to new readings. Changes in models are assessed against user-specified thresholds and reported as important events if they cannot be predicted using past estimates with reasonable accuracy. The Edge Mining approach is based on the Spanish Inquisition Protocol (SIP) [33] and represents a new stage in sensor intelligence. To validate the suitability of Edge Mining for Precision Farming applications, we consider the use of Linear SIP (L-SIP), an instance of Edge Mining, for data compression and event detection on-board sensor devices [34]. The approach is developed as a proof-of-concept and is shown to increase longevity of device operation while improving responsiveness of WSN-based systems. Furthermore, we present an extension of the ClassAct approach, another instance of Edge Mining, for activity monitoring and localization of cows in dairy farms [35]. The approach allows for location-aware sensing as cows move around the farm. These techniques are discussed as under.

We propose the use of L-SIP on-board sensor device for intelligent data collection (sense and send) in smart dairy farms [34]. Owing to poor Internet connectivity in rural farms, we suggest delay-tolerant communication for reliable data transfer to cloud. We discuss the design of our animal wearable sensor device that monitors animal health and mobility, as well as acts as a mobile agent to collect data from static in-field sensors. All data is stored locally on the wearable device until it is in the vicinity of a

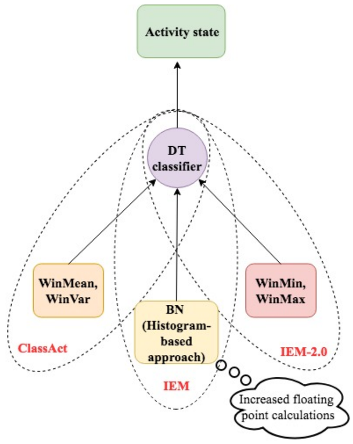
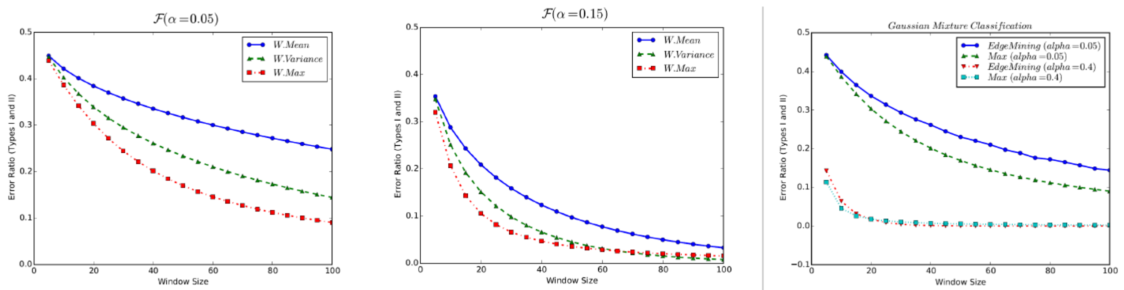


Fig. 1.4 Difference between ClassAct, IEM and IEM2.0 [35]

gateway. While this approach provides a solution for data communication in remotely connected areas, given the wide-variety of data that must be gathered on a farm, the memory constraints of sensor devices pose a major challenge in its realization. We examine the suitability of using L-SIP for lossless data compression on-board sensor devices as shown in fig. 1.3a. L-SIP represents sensor data as a linear model wherein the application state is encoded as a vector of smoothed point-in-time value and rate of change. While a number of techniques have been proposed for estimating the state, we use the double Exponentially Weighted Moving Average (dEWMA) to calculate the new state, per sensing cycle. An event is detected if the new state value cannot be predicted using linear extrapolation of the the previous state with a desired accuracy specified by a threshold ε. As such, instead of storing raw data, we suggest conversion of sensor data into linear models that are stored only at the occurrence of such events. In doing so, L-SIP reduces memory requirements of sensor devices and improves the operational time of the network. Moreover, it reduces redundancy in data and also allows real-time detection of events to improve responsiveness of the system. The approach is shown to be data agnostic to suit the variety of farm data such as temperature, humidity and acceleration, and achieve a memory gain of up to 70% for user-specified ε values as shown in fig. 1.3b. Furthermore, use of L-SIP has been shown to facilitate signal reconstruction with reasonable accuracy, if required.

ClassAct [36] is a decision-tree classifier that represents the sensor data as a smoothed probability distribution over a given set of states. The index of the most probable state is chosen and sent to the cloud if it differs from the previous estimate. The approach is used for activity recognition of people in [36], and has been shown to perform reasonably well for classification of low-level activity states such as walking, standing



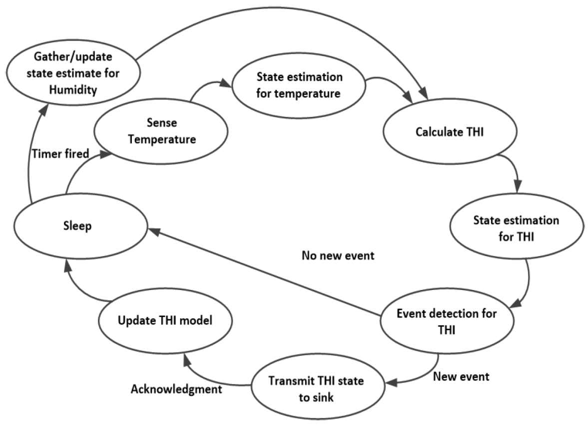
(a) (b) (c)

Fig. 1.5 Gaussian Mixture Effect on single-feature classification for low ($\alpha = 0.05$ (a)) and low-to-medium mixture effects ($\alpha = 0.15$ (b)); Joint windowed mean & variance classification for low and medium mixture effects ($\alpha \in 0.05, 0.4$ (c)) [35]

and sitting. ClassAct, however, basis its prediction on values of low-order moments such as windowed mean and variance at fixed points in time. This limits its use in applications where the given states have significantly overlapping values (and therefore similar mean and variance), despite coming from different distributions. We study the effect of mixed Gaussian signals, for instance overlapping signals represented by a normal distribution and a Gaussian mixture, on accuracy of ClassAct-based classification, and present an extension of the ClassAct approach, IEM2.0, to cater to the above limitation [35]. Our approach considers the distribution of signal over time and basis prediction on windowed minimum and maximum values that capture the temporal pattern of the signal. IEM2.0 is an extension of ClassAct and IEM approaches (discussed as part of RQ2) and significantly improves the prediction accuracy while reducing the number of computations. The difference in the three approaches is illustrated in fig. 1.4. We analyze the predictive capabilities of our approach as compared to ClassAct for different Gaussian mixture proportions. The approach is shown to be particularly useful when the mixture is highly imbalanced as shown in fig. 1.5. Furthermore, we demonstrate its suitability with respect to activity recognition and localization of animals in dairy farms. We discuss the use of our classification technique for high-level activity classification (i.e. milking, grazing), and further propose Activity Sequence-based Map Matching (ASMM) for localization of animals in paddock, parlour and in-transit between parlor and paddock using the given farm topology. The proposed approach allows for contextualization of sensor data collected by the devices for improved decision making. An extensive evaluation of the approach along with memory analysis for the algorithm has been carried out using real-world

procedure :
$t \leftarrow$ current time
At static humidity node
$hum_t \leftarrow$ obtain vector of sensor readings
estimate new state for humidity - dEWMA filtering
$v_{H,t} \leftarrow \alpha_H * hum_t + (1-\alpha_H) * (v_{H,t-1} + r_{H,t-1} * \delta t)$
$r_{H,t} \leftarrow \beta_H * (v_{H,t} - v_{H,t-1})/\delta t + (1-\beta_H) * r_{H,t-1}$
At static temperature node
$tmp_t \leftarrow$ obtain vector of sensor readings
estimate new state for temperature - dEWMA filtering
$v_{T,t} \leftarrow \alpha_T * tmp_t + (1-\alpha_T) * (v_{T,t-1} + r_{T,t-1} * \delta t)$
$r_{T,t} \leftarrow \beta_T * (v_{T,t} - v_{T,t-1})/\delta t + (1-\beta_T) * r_{T,t-1}$
if request for new THI state estimate received or
temperature or humidity event occurred, then
Obtain humidity state $(v_{H,t}, r_{H,t})$ and calculate
$thi_t \leftarrow 1.8 * v_{T,t} - (1 - v_{H,t})(v_{T,t} - 14.3) + 32$
estimate new state for THI - dEWMA filtering
$v_{THI,t} \leftarrow \alpha_{THI} * thi_t + (1-\alpha_{THI}) * (v_{THI,t-1} + r_{THI,t-1} * \delta t)$
$r_{THI,t} \leftarrow \beta_{THI} * (v_{THI,t} - v_{THI,t-1})/\delta t + (1-\beta_{THI}) * r_{THI,t-1}$
predict sink value using linear extrapolation
$THI_{sink,t} \leftarrow \begin{bmatrix} 1 & t - t_{sink} \\ 0 & 1 \end{bmatrix} THI_{sink,t_{sink}}$
if eventful $(|v_{sink,t} - v_{THI,t}| > \varepsilon_{THI})$
or $t - t_{sink} \geq t_{heartbeat}$ then
a. Transmit$((v_{THI,t}, r_{THI,t}), n, t)$
b. $n \leftarrow n + 1$ (increment sequence number)
c. when acknowledgement received
i. $THI_{sink,t} \leftarrow s_{THI,t}$
ii. $t_{sink} \leftarrow t$
iii. $t_{heartbeat}$ reinitialized

(a)



(b)

Fig. 1.6 (a) CEM algorithm (b) CEM state diagram for THI detection [37]

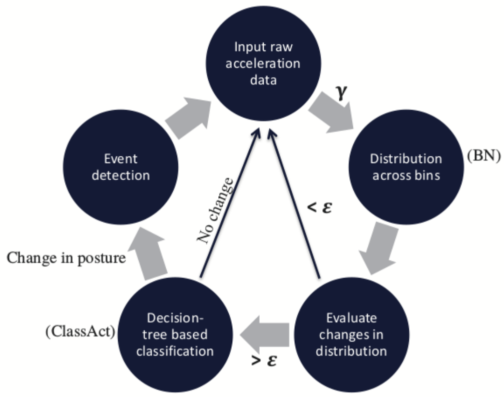
animal mobility data collected in dairy farms to assess the suitability of our methods for sensor-based execution.

- ***RQ2. Cooperative WSN:*** Although the Edge Mining-based analytic techniques (discussed above) improve autonomy of sensor operation, they consider analysis on each device in isolation from each other. This not only constraints the amount of computation that can be offloaded to WSN, it also limits the predictive capabilities of system wherein integration of multilpe sub-tasks/data-sets is required for detection of application-specific events. To address these limitations, we propose two cooperative frameworks - Collaborative Edge Mining (CEM) [37] and Iterative Edge Mining (IEM) [38], that allow distributed computing within WSN for provision of real-time services. Both frameworks have been implemented by considering distribution of Edge Mining algorithms that act as building blocks for required services on sensor devices, and facilitating interaction between them for integration of results.

  CEM has been proposed to facilitate distribution and parallelization of analysis within WSN. A subset of application logic or an Edge Mining task is delegated to each sensor device based on the analytic model, application requirements, and availability of resources such as type of sensors. The intermediate results of individual analysis are then integrated within the network to detect more complex, application-specific

procedure :
Update bin count $\forall i \in B$
    $x_{i,t} \leftarrow \gamma \cdot x_{i,t-1} + b(i, z_t)$
Update bin distribution (simplify) $\forall i \in B$
    $y_{i,t} \leftarrow x_{i,t} \ / \ \Sigma_{i \in B} x_{i,t}$
$y' \leftarrow$ previous significant distribution at time t'
If $\exists i \in B : |y_{i,t} - y'_i| > \varepsilon$ or $t - t' \geq t_{heartbeat}$
    Update $y' \leftarrow y_t$ and $t' \leftarrow t$
    Estimate new state
    $s_t \leftarrow f(DT, y_t)$
    Eventful?
        Yes, if state differs from last update
            Store in Flash state s_t and time t

(a)



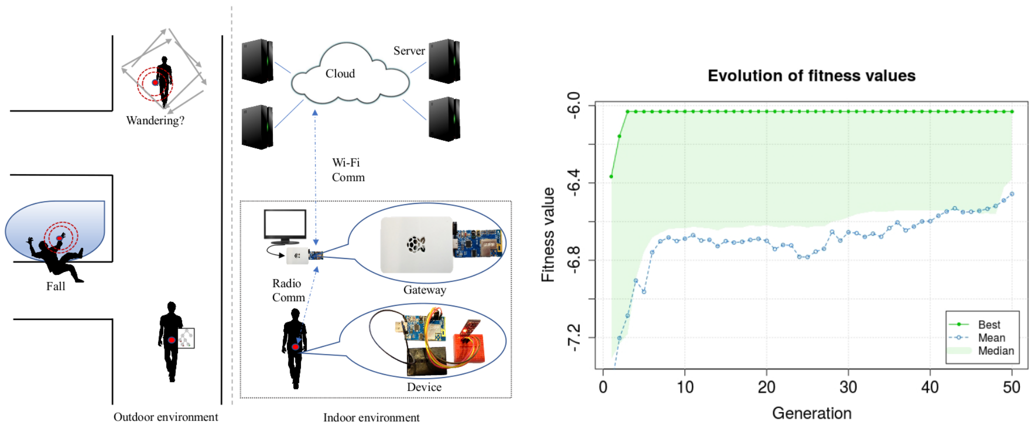
(b)

Fig. 1.7 (a) IEM algorithm (b) IEM state diagram for activity recognition [38]

events. While Edge Mining-based tasks on individual devices are performed per sensing cycle, their results are integrated only if an intermediate event is detected by any device to check for significant updates in the application-state. The intelligent (event-based) interaction between different tasks prevents over-utilization of sensor resources. Moreover, the horizontal scaling of analysis within WSN improves the load and energy balancing across devices. Such cooperation between devices allows for improved computation capability of network, and lays foundation for design of integrated systems in future WSN. Combined analysis of information from different devices also reduces packet transmissions to cloud as compared to the conventional Edge Mining approach, thereby, improving energy efficiency of the sensor operation. We illustrate the use of CEM framework for detection of Heat Stress in animals as shown in fig 1.6. The severity of Heat Stress is estimated using Temperature-Humidity Index (THI), which can be calculated using static temperature and humidity sensors on a farm. We discuss the design of an animal wearable collar device that acts as a master node to initiate the THI calculation on static sensors, as well as a mobile sink node to collect THI updates from sensors. Each sensor device that participates in analysis runs the L-SIP algorithm to estimate changes in temperature or humidity states. Upon detection of event by any device, the state information is exchanged to update the THI value. As opposed to Edge Mining approach wherein both temperature and humidity states would be sent to sink for calculation of THI, the CEM approach requires transmission of only the THI updates. It, thus, improves the network compute capability as well as improves energy efficiency of WSN operation.

Whereas CEM aims at improving the computational capacity of a WSN system-as-a-whole, IEM is proposed to improve compute capability of each sensor device. It is implemented by superimposing correlated and sequential Edge Mining tasks on a single device, and facilitating intelligent interaction between them to output the application state. The tasks are designed and integrated such that the analysis performed by each subsequent task is based on the output of the previous task. The application-specific state is determined by the output of the final task. Moreover, frequency of execution of the subsequent tasks is governed by occurrence of intermediate events resulting from analysis of the prior tasks. By doing so, IEM regulates the number of computations performed by each device to prevent over-utilization of its resources. IEM, thus, allows vertical scaling of analysis within a device. The joint analysis performed by superimposed tasks, in turn, leads to significant reduction in packets transmitted to the cloud. We illustrate the use of IEM for real-time activity monitoring of animals within a farm as shown in fig. 1.7. IEM is implemented based on interaction between two Edge Mining algorithms - Bare Necessities (BN) and ClassAct. Firstly, BN takes raw acceleration readings and converts them into an intermediate state represented as histogram distribution across bins, where each bin defines an intermediary, application-relevant state. The distribution suggests the relative time spent in each of the states, and is updated per sensing cycle. The distributions are also smoothed over the past estimates on account of no sudden changes in activity state. An intermediate event is detected if change in distribution of any bin exceeds a given threshold (after smoothing). The ClassAct algorithm is invoked only at occurrence of such events. It takes the updated distribution generated by BN as input to the classifier, and identifies the activity state. The frequency of classification is, thus, governed by the threshold parameter to ensure optimal use of CPU resources while meeting the accuracy requirements. The reduced frequency of computations, as compared to conventional ClassAct approach that runs per sensing cycle, is particularly beneficial in case of large decision-trees. The performance of IEM is evaluated in terms of accuracy of classification as well as frequency of classification across different mobility patterns and scenarios. The approach has also been shown suitable for localization of elderly in behavioural-tracking Ambient Assisted Living (AAL) applications [39].

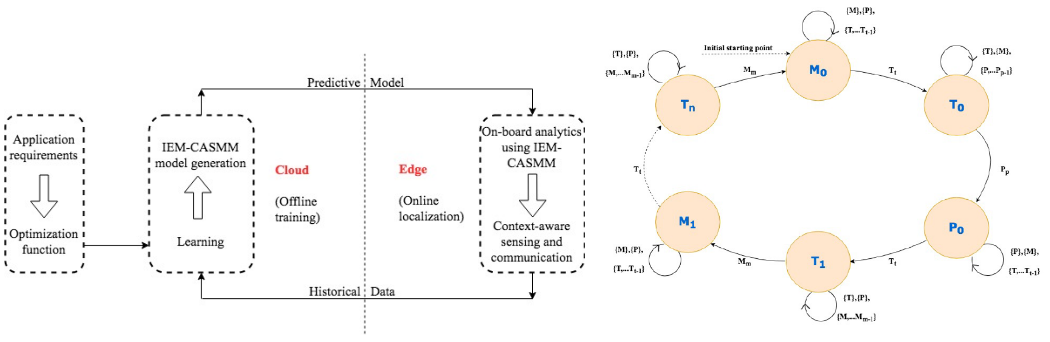
- ***RQ3. Flexibility and Energy Management in Cooperative WSN:*** Depending on the application, WSN-based services can be deployed using either one or combination of the cooperative frameworks discussed above. Owing to varying application requirements as well as dynamic network topology, design of future WSN must facilitate



(a) (b)

Fig. 1.8 (a) WSN architecture for user localization (b) Evolution of GA fitness value [40]

re-configuration of the services to adapt to these changes. For instance, in case of IEM-based localization, a certain behavioural tracking AAL application may have varying accuracy requirements based on the scenario. While it may need precise localization in some instances, it may suffice to obtain coarse location at other times. The frequency of computations, governed by the input parameters of the IEM model, must then be changed accordingly to optimize the use of sensor resources while ensuring computational accuracy. To allow such flexibility in analysis, we consider integration of the fog-enabled WSN with the cloud. We propose the design of a dual analytical framework wherein the sensor-based analytics is performed for real-time predictions, and cloud-based learning is performed using historical data to update the sensor analytic models based on changing application requirements [40]. Delay-tolerant communication is proposed to facilitate exchange of data and the updated models between sensor and cloud. We illustrate the use of this framework for achieving flexibility in IEM-based localization to balance trade-off between prediction accuracy and energy consumption of device with changes in user-specified requirements in the context of location-based AAL applications as shown in fig. 1.8a. We implement cloud-based learning using Genetic Algorithms (GA) to find the optimal set of parameter values for sensor-based IEM model. The use of GA is considered owing to its ability to generate high quality solutions for a large search space in polynomial time as shown in fig. 1.8b. We define a fitness function that minimizes error in classification and localization along with number of on-board classifications. The performance of GA is primarily governed by the weights assigned to each term in the fitness function. The IEM model can, thus, be easily updated by varying the weights of terms as desired. We evaluate the



(a) (b)

Fig. 1.9 (a) System architecture (b) Behavioural state transitions using CASMM [35]

performance of GA-based model for different activity sequences and mobility patterns obtained from the Kasteren data-set [41].

Energy management plays a crucial role in the design of future WSN. The energy consumed by the computation tasks is proportional to the number of CPU cycles required as well as the frequency of computations. As discussed above, while real-time computations are performed on-board sensor devices, input parameters for sensor analytic models can be updated using cloud-based learning as shown in fig. 1.9a. The optimal parameter values are determined based on an objective function defined by application requirements. For instance, such an objective function can be used to relax either accuracy requirements for reduced computations or the task completion deadlines for reduced clock frequency to, in turn, reduce energy consumption through on-board computations. Whereas the on-board analysis forms one aspect of energy consumption on devices, data communication from sensor to cloud is the most energy-intensive task performed by these devices. To minimize this cost, we propose a context-aware, event-driven communication approach for data transfer to cloud. We exploit the location information of devices obtained through IEM2.0-based activity classification and collaborative ASMM (CASMM) approach, and propose transmission of data only at the occurrence of change in location [35]. Since real-time predictions are made independently on-board sensor devices, delay-tolerant communication of these results to cloud allows improved energy efficiency of devices by reducing the unnecessary redundant and periodic packet transmissions. As such, the energy cost associated with communication is then directly proportional to the accuracy of localization. While a high frequency of classification can be used to improve performance, it may burden the device resources. Instead, we exploit the spatial-temporal coherence of neighboring devices and suggest cooperative activity state detection, prior to ASMM, for improved accuracy of localization (referred to as CASMM as shown in fig. 1.9b). We envisage a set of participating devices in close vicinity of each other as a coalition that exhibits a common activity state based on the location (e.g. milking in parlor), and facilitate exchange of updates in activity state between devices to improve accuracy of individual predictions. We adopt a majority-voting scheme to make the activity state consistent within the coalition. However, the cooperation between devices itself incurs an additional overhead. We model the communication cost associated with both cooperation between devices as well as packet transmissions to cloud, and aim at reducing the net cost for the network. The effect of different coalition size on the accuracy of localization and, thereby, communication cost is studied for resource optimization within WSN.

Furthermore, we suggest cooperative computing via computation offloading in heterogeneous energy harvesting WSN for optimizing the use of in-network computational resources [42]. In computation offloading, a sensor device (initiating node (IN)) offloads partial computation to a neighbouring device, known as the cooperating node (CN), such that the given task completion deadline is met while optimizing the energy resources of the network. Accordingly, we model the computation and communication (device-to-device) costs associated with cooperative computing as well as micro-solar energy harvesting capacity of sensor devices, and discuss an energy-aware optimal task partitioning algorithm for computation offloading in energy harvesting WSN. The cooperation allows for improved computational capability of WSN while delivering timely responses for latency-sensitive applications and reducing the net energy cost of the network. The energy-aware nature of our algorithm takes into account status of both stored and harvested energy on device and has been considered for different energy-harvesting scenarios such as presence of IN and CN in shadow-shadow, shadow-light and light-light respectively. The approach improves energy balance of a WSN which is an important factor for its long-term autonomous operation as well as reduces energy waste due to overflows. We discuss the use of Lagrange Multipliers to solve the equal constrained optimization problems. We also present an energy-aware, utility-based technique for selection of the CN. The approach aims at improving fairness within the network to avoid over-utilization of certain devices. The proposed techniques are evaluated using the Simgrid simulator. The results show reduced energy consumption of the network along with an improved average operational time of sensor devices.

(a) (b) (c)

Fig. 1.10 (a) Collar device (b) Animal wearable collars (c) Cloud gateway

1.3.1 Validation

To validate the approach developed for each research question, we perform analysis of real-world sensor data collected using a WSN prototype. The WSN comprises of two kinds of devices - wearable sensor devices and cloud gateway. The wearable device is a plug-n-play universal sensor node consisting of various sensors for application-specific data collection. On-board implementation of sensor analytic techniques is proposed on these devices for real-time event generation. The gateway node is used to gather data collected by the wearable devices for uploading on the cloud. R-based analysis is performed on the cloud to train the analytic models and test their performance using the sensor data.

- Collar device: Collar device (fig. 1.10a) forms the most integral part of our WSN prototype. The primary component of the collar device is the IEEE 802.15.4 compliant, low-power CM5000 mote [43]. It consists of a MSP430F1611 micro-processor, a CC2420 802.15.4, 2.4GHz wireless module for radio communication, and an on-board SHT11 sensor to collect temperature and humidity readings. The mote features a program memory of 48KB, a 10KB RAM for storing state of program variables, as well as an additional flash memory of 1MB for storing sensor data. The non-volatile nature of the flash prevents loss of data despite device failures. The CM5000 mote also consists of a 6 and 10-pin connector to allow for additional sensors. To facilitate mobility tracking, we connect a 10 degrees of freedom (DOF) MPU9255 Inertial Measurement Unit (IMU) [44] to the mote. The IMU consists of an MPU6050 and HMC5883L for measuring 3-axis acceleration, orientation (gyroscope) and 3-axis magnetic field (magnetometer). It features a user-programmable full scale range to ensure accurate tracking for both slow and fast motion. We also connect a Ublox NEO-6M Global Positioning System (GPS) receiver [45] to our collar device. While the GPS enables node localization, it adversely affects the battery-life of devices. It's use is, therefore, limited in this work for estimating memory requirements while using GPS readings on sensor nodes. The entire device is powered using 2xAA batteries (3V) and boxed in a pelican casing as shown in fig. 1.10b. Based on the application, a TinyOS program [46] sits on the collar devices for data collection and analysis tasks. The design of program varies for each study and is discussed in the respective publications.

- Cloud gateway: The gateway node consists of a CM5000 mote connected to a Raspberry Pi (v. B2) [47] module as shown in fig. 1.10c. As mentioned above, the role of gateway is to collect data from collar devices and upload it onto the cloud for future analysis. Accordingly, a TinyOS application runs on the CM5000 mote of the gateway for data collection from the collar devices via mote-to-mote communication (using Zigbee protocol). To establish connection with the gateway, a device temporarily joins the 802.15.4 Personal Area Network (PAN) of the gateway by sending an association request. At any given time, the gateway can connect to a predefined number of collar devices. If the node is currently connected to the predefined maximum, it does not confirm association and a random back-off mechanism is activated on the collar device to retry association. Otherwise, an acknowledgement is sent from the CM5000 mote on the gateway to the device confirming its association. Once the device is connected to the gateway, it sends its data packets over the radio until the flash is empty. These packets are further transferred by the gateway mote to the Raspberry Pi. A JAVA application is built for Raspberry Pi, using TinyOS tools, to collect and store the incoming data. The data, thus, generated is periodically pushed to a private Gitlab (cloud) repository using a WiFi dongle. Upon completion of data transmission, the device sends a disassociation request to the gateway requesting to leave the PAN, and subsequently resumes its operation.

The use of above hardware is considered owing to the ability to program the sensor devices. The implementation of algorithms (included in respective publications) is generic, making them suitable for use with any other off-the-shelf WSN devices. We conduct experiments outside our laboratory for collecting data to validate the approach developed in RQ2, and parts of RQ1 (data compression using L-SIP) and RQ3 (flexibility in analysis). Partial findings of RQ1 (localization) and RQ3 (cooperation via collective participation for energy management) are validated using data collected from a pilot study conducted in Dairygold-sponsored farm located in Kilworth, Co. Cork, Ireland. The experiment aimed at collecting mobility data of 5 cows selected randomly from a herd of 46, over a period

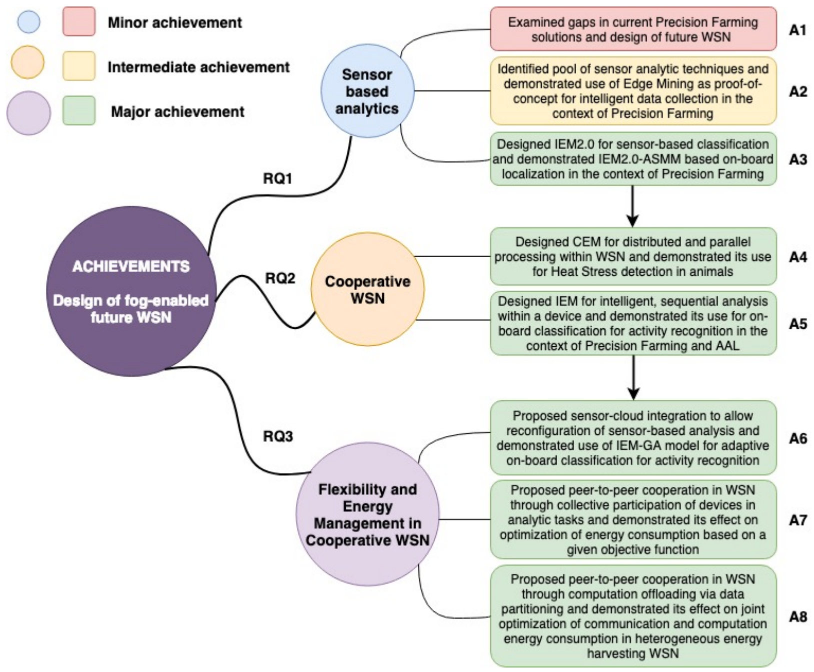


Fig. 1.11 Key achievements of the research work

of 5 days in June 2017, which was later analyzed to evaluate the performance of sensor-based localization. The details of the experiment are discussed in appendix H. Besides the prototype-based analysis, partial findings of RQ3 (cooperation via computation offloading for energy management) are also validated via a series of SimGrid simulations [48]. SimGrid is an open-source simulator that allows to study the behavior of large-scale distributed networks such as P2P systems.

1.4 The Contribution

We believe this thesis makes a significant contribution towards the design of future, fog-enabled WSN systems. In this section, we summarize the key achievements of this work as depicted in fig. 1.11. The achievements are mapped to each research question, and are color-coded to highlight their significance. The extent of contribution from each research question to this thesis has also been illustrated. Achievements 1 and 2 (A1 and A2) relate to the findings from the background research carried out for this work to identify gaps in existing WSN-based solutions for Precision Farming as well as limitations of the current sensor analytic techniques (RQ1). As part of A2, a WSN prototype was also developed to validate the suitability of Edge Mining for real-time event detection in Precision Farming applications. A3 corresponds to the design of IEM2.0 that allows improved on-board classification and adds to the pool of sensor-based analytics. Achievements 4 and 5 (A4 and A5) result from the design of novel cooperative frameworks - CEM and IEM, for distributed analysis within WSN and their validation using our prototype for various applications. These achievements emanate from research carried out as part of RQ2. A6 corresponds to the design of dual-analytical framework that allows sensor-cloud integration to enable flexibility in sensor analytic models. We consider the design issues in implementation of such a system in rural applications as well as the selection of algorithms for both cloud-based learning and sensor analytics. Lastly, achievements 7 and 8 (A7 and A8) result from the design of peer-to-peer cooperative analytic approaches that aim at optimizing energy consumption within WSN and their use in different WSN topology. The work addresses a novel area of work in WSN and presents possible solutions to improve scalability of analysis within WSN while ensuring load balancing and fairness among devices. A6, A7 and A8 emanate from research carried out as part of RQ3. The achievements have been discussed in greater detail below.

– ***RQ1. Sensor-based analytics***

- **A1.** We have examined the need for WSN to improve Precision Farming practices, along with the gaps in existing solutions in [30]. A serious lack of autonomy has been identified as a major drawback in the design of WSN-based systems. This arises owing to a significant lack of intelligence in sensor operation coupled with high dependency on intermittently-available cloud resources for storage and analysis of data. In future, these problems are expected to be enhanced due to the adoption of Internet of Nano-Things in farming practices as discussed in [49]. To enable autonomous operation of WSN-based systems, we, therefore, suggest realization of smart sensor devices for future WSN.
- **A2.** We have identified the pool of data analytic techniques suitable for sensor-based execution, along with the benefits and challenges associated with their use in the context of Precision Farming in [30]. Real-time analytics on-board sensor devices is expected to provide fast accessibility to data insight, improve resource efficiency of the network as well as ensure security of data. Of the proposed techniques, the Edge Mining approach, in our opinion, is the most suitable for sensor-based analytics. We have validated the use of L-SIP, an instance of Edge Mining, for intelligent data collection in dairy farms using our WSN prototype (discussed in 1.3.1) in [34] as a proof-of-concept. The approach is shown to be data agnostic and allows real-time event detection while significantly reducing the redundancy in data. The reduced memory requirement on sensor devices is,

in turn, expected to improve the operational time of devices when adopting a delay-tolerant communication approach.

- **A3.** We proposed an extension of the ClassAct approach, an instance of Edge Mining, for real-time activity recognition and localization on-board sensor devices in [35]. The approach relies on self-measurements from IMU and uses a decision tree classifier for activity state recognition. The sequence of activities generated is used as input for ASMM to identify the device location. Unlike ClassAct, the approach basis classification on distribution of data. In doing so, it is shown to be better suited for separating two different signals, represented by Gaussian mixtures, that have significantly overlapping values and an unequal likelihood of occurrence. The approach is extensively evaluated for localization of animals within a farm using real-world acceleration data of cows collected during a pilot study in Dairygold-sponsored farm in Kilworth, Ireland. The analysis shows that our approach can achieve a localization accuracy of up to 99%. Moreover, an array-based implementation of the approach is discussed and a resource assessment is carried out to verify its suitability for device-based implementation. The analysis confirms a very low memory footprint of the approach.

– ***RQ2. Cooperative WSN***

- **A4.** We developed the CEM framework for distributed and parallel processing within WSN as presented in [37]. The framework delegates separate tasks (we considered Edge Mining algorithms) to devices that participate together in on-site analysis, and integrates their intermediate results to detect application-specific events. CEM allows scaling of analysis across WSN based on sensor capabilities, and lays foundation for the design of integrated sensor systems on-farm. The performance of CEM has been evaluated for estimation of Temperature Humidity Index, an important metric for Heat Stress detection in cows, across different values of input parameters and compared to the original Edge Mining approach. The analysis shows that CEM facilitates on-site detection of more complex events while reducing packet transmissions within the network, as compared to Edge Mining, for improved energy profile of the devices.
- **A5.** We developed the IEM framework for intelligent, sequential analysis of tasks within a device in [38]. IEM allows interaction of correlated tasks on the same device such that the execution of the subsequent tasks is dependent on the output of the previous task. The application state is determined by the outcome of the

final task. As such, IEM improves compute ability of a device while ensuring efficient resource utilization. As a proof-of-concept, IEM is implemented as the superimposition of two Edge Mining algorithms - BN and ClassAct, for activity monitoring and behaviour analysis of animals. The approach is evaluated for accuracy of predictions and frequency of computations across different input parameters using real-life data from our prototype. The analysis confirms high prediction accuracy of IEM with very few computations on-board. Furthermore, the suitability of IEM has been demonstrated for localization of the elderly in AAL applications in [39]. Given the topology information and the activity sequence generated by IEM, the user location is estimated by calculating the distance covered over time.

– ***RQ3. Flexibility and Energy Management in Cooperative WSN***

- **A6.** We have proposed flexibility in sensor-based analytics with changes in user-preferences through sensor-cloud integration, and demonstrated its use for re-configuration of IEM-based localization in AAL applications in [40]. The use of cloud-based learning using GA approach is shown to optimize the sensor-based IEM model, using historical data, in order to balance the trade-off between classification accuracy and frequency of computations in IEM-based localization. A delay-tolerant communication framework is used to send the updated model back to the device. An evaluation of the dual analytical framework (IEM-GA) is carried out using acceleration data collected by our wearable device across different activity sequences obtained from the Kasteren data-set. The analysis shows that GA-based learning allows easy adaptation of analysis (using different fitness functions) to changes in user requirements, achieves a fast learning curve, as well as optimizes sensor performance.

- **A7.** We proposed a coalition-based cooperative data analytic approach within WSN in [35]. Depending on availability of devices, we conceive a group of devices in vicinity of each other as a coalition. Collective participation of devices (within a coalition) in analytic tasks is suggested to improve accuracy of individual sensor-based predictions (using a majority voting scheme). We demonstrate the use of cooperative analysis for improving localization accuracy of devices using approach developed in **A3**. We exploit this information to minimize number of transmissions to cloud following a context-aware, event-driven communication framework wherein data packets are transmitted to the cloud only at the

Table 1.1 Research achievements

Res. Question	Challenge	Achievement	Presented in publication	Appx.
RQ1	C1	A1	IEEE Intelligent Systems [30]	A
			ACM NanoCom [49]	B
		A2	IEEE Intelligent Systems [30]	A
			IEEE SenseApp [34]	C
		A3	ACM Trans. Sen. Netw. [35]	H
RQ2	C2	A4	IEEE WD [37]	D
		A5	IEEE ICNC [38]	E
			IEEE ICE/ITMC [39]	F
RQ3	C3	A6	IEEE PIMRC [40]	G
		A7	ACM Trans. Sen. Netw. [35]	H
		A8	Sustainable Computing [42]	I

occurrence of a change in location. We define an optimization function that considers the communication cost between device-to-device and device-to-cloud, and study the effect of different coalition sizes on the accuracy of predictions and short and long-range transmissions in order to minimize the total energy cost. The analysis shows that our cooperation approach along with the delay-tolerant communication framework can reduce the net energy consumption of a network by 90%.

- **A8.** We developed theoretical models for cooperative computing via computation offloading in micro-solar powered heterogeneous energy harvesting WSN as presented in [42]. We discuss optimal data partitioning so as to minimize the total energy consumption (communication and computation) while meeting the application deadline requirements based on the energy harvesting status of sensor nodes under different scenarios. The evaluation of our models shows a reduction in both energy losses, and waste due to energy conversion and overflows respectively, as compared to a data partitioning algorithms that offload computation tasks without taking the energy harvesting status of nodes into consideration. In addition, our approach reduces net energy consumption of the network while improving energy balance across sensor devices for sustainable operation. We also discuss an energy-aware cooperating node selection strategy based on a utility function for improved fairness within the network.

As highlighted in fig. 1.11, few of the achievements discussed above are minor and result primarily from the background research performed during the study. The achievements resulting from RQ2 (A4 and A5) marked the first milestone for our research through design of CEM and IEM frameworks. These frameworks are novel and lay foundation for distributed, peer-to-peer cooperative analysis in WSN, which is particularly important in applications where input data is distributed across devices. This contribution allows to overcome one of the significant limitation of current analytic approaches that consider operation on each sensor device in isolation with each other. Another milestone was achieved through design of IEM-GA based analytical framework (A6) that allows reconfiguration of sensor intelligence, and addresses the challenge of flexibility in current WSN solutions. Finally, the last key milestone for our research was the design of peer-to-peer cooperative analysis frameworks through collective participation in analytics (A3 and A7) and computation offloading (A8) for energy management in WSN. Together, all these achievements allow for the design of future WSN. The achievements have been mapped to the 9 research publications that have resulted from this work and are shown in table 1.1. As part of the continuous study of the research questions, a single publication partially answers more than one research question and multiple publications combined try to address a single question.

1.4.1 Publications

Results of the research carried out as part of this PhD have been documented in a number of research articles. A complete list of the articles is presented in chapter 4.

Articles related to the first research question include 3 peer-reviewed articles and parts of 1 peer-reviewed article. Of the four articles, one article has been published in *IEEE Intelligent Systems* (appx. A), two articles have been presented in the proceedings of *2nd ACM International Conference on Nanoscale Computing and Communication (ACM NanoCom '15)* (appx. B) and *Eleventh IEEE International Workshop on Practical Issues in Building Sensor Network Applications (IEEE SenseApp '16)* (appx. C), and one article has been published in *ACM Transactions on Sensor Networks* (appx. H).

Articles related to the second research question include 3 peer-reviewed articles that have been presented in the proceedings of three conferences - *Eighth Wireless Days Conference* (appx. D), *International Conference on Computing, Networking and Communications (ICNC 2017)* (appx. E) and *23rd ICE/IEEE International Technology Management Conference (ICE/IEEE ITMC 2017)* (appx. F).

Articles related to the third research question include 2 peer-reviewed articles and parts of 1 peer-reviewed article. Of the three articles, one article was presented in the proceedings of the *28th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC 2017)* (appx. G), and other two articles were published in *Sustainable Computing: Informatics Systems* (appx. I) and *ACM Transactions on Sensor Networks* (appx. H).

Chapter 2

State-of-the-art

In this chapter, we present state-of-the-art solutions that address the problems considered in this research. Firstly, we discuss the existing WSN solutions for Precision Farming applications in section 2.1. We highlight the challenges associated with current technologies that have been addressed in our work. Next, we discuss the currently proposed sensor analytic approaches in section 2.2, followed by an overview of the state-of-the-art localization techniques used in WSN in section 2.3. Finally, we review the recent work carried out with respect to resource management via in-network analysis in section 2.4.

2.1 Precision Farming

The mounting population coupled with diminishing arable land and unpredictable weather conditions raises concerns of food security in near future, thus, making it imperative to efficiently utilize the available natural resources. The use of ICT in agriculture has been proposed to allow precise monitoring and automation of farm processes under the umbrella of Precision Farming. This is expected to improve control over the farm processes and, in turn, increase the productivity and sustainability of farming. Originally, Remote Sensing along with Geographic Information Systems (GIS) and GPS was used for monitoring the farms [50]. However, these systems are expensive and offer limited spatial-temporal resolution. Today, WSN-based systems have been widely proposed for accurate real-time monitoring of farming practices. Sensor devices facilitate collection of a wide variety of farm data such as soil composition and dynamics, crop growth, climate changes and animal health and mobility. Timely analysis of the sensor data allows prediction of the onset of diseases or adverse weather conditions in early warning systems to help farmers make informed decisions [51]. Furthermore, analysis of data enables precise application of chemicals/fertilizers to specific areas in a farm. This can significantly reduce the input costs by minimizing wastage while allowing improved productivity. A review of WSN applications in Precision Farming has been presented in [52].

Certain agricultural sensor systems exist already. K. Taylor et al., for instance, describe a WSN system deployed in Kirby farm near Armidale, New South Wales [53]. The system incorporates various sensors to monitor soil moisture, temperature, humidity and pressure, rainfall, and hail. Monitoring data from sensors is pushed to a centralized entity, where it is enriched and analyzed to be sent to farmers. A survey conducted in Netherlands [54] shows that almost two fifths of the farms surveyed have adopted some sensor-based farm monitoring. Another study [55] discusses the use of unmanned robotic systems for farming applications. These systems aim at automation of certain farm monitoring and mapping tasks (such as yield mapping) to reduce manual labor. Several WSN-based systems have also been designed for monitoring animal health and mobility, with the aim of early detection of diseases to promote animal welfare. A review of various sensor systems for animal health management in dairy farming has been presented in [56]. These systems are primarily designed to monitor animal fertility, metabolism, and Mastitis. A few systems have also been developed for mobility monitoring of animals. Mobility patterns give an understanding of animal behavior and can be used to detect health issues such as Lameness [57]. Additionally, mobility tracking facilitates implementation of the Virtual Fence (VF) technology that uses acoustic and electric stimuli to control the movement of animals within a farm. Current VF solutions make use of either electromagnetic coupling between animal wearable sensor devices and an insulated wire unrolled in the farm [58] or GPS receivers fitted to the wearable devices to estimate the position of animals with respect to the VF [59].

Despite the numerous advantages, very few WSN-based systems have been put into use for farming practices. This is primarily due to the limited capability of sensor devices coupled with the lack of infrastructure in a typical farm. Conventionally, the tasks assigned to these devices are limited to data collection and transmission while the analysis takes place on the cloud. A study conducted by Rutten et. al. [56] describes such a system for animal health management and highlights the lack of analytics and intelligence in sensor devices. This introduces latency in analysis and poses a major constraint in WSN implementation in large-scale, rural farm environments that suffer from intermittent or no Internet connectivity. While additional infrastructure may resolve certain issues, it would increase the deployment and maintenance costs of the system causing reluctance among farmers to embrace use of WSN systems. Consequently, there is a need to improve operation of WSN to allow on-site analysis and prediction, especially, for latency-sensitive phenomenons to develop cost-effective and autonomous farming solutions. Such systems will not only allow easy accessibility to data insights but also promote information security through local data reduction. The need for intelligent WSN becomes inevitable with the design of nano-sensors for future farming [60]. While nano-sensors would allow early detection of processes that ensue at molecular scale, they will cause explosion of data generated by the farms, thereby, necessitating in-network analytics for ensuring real-time responsiveness of the system. Furthermore, while different WSN systems have been designed to cater to various aspects of a farm - crops, soil, yield and animals, these systems work independent of each other. This causes difficulty and delay in correlating data from different systems to improve the decision making process. Cooperation between these systems is, thus, desirable for the design of effective decision-support systems that aim at integrated farm management. Such a system, for instance, will allow timely analysis of the effects of feed on animal behaviour and, in turn, milk quality in a dairy farm to help improve the overall productivity of the farm. We consider these gaps in existing solutions for the design of future WSN through the research presented in this thesis.

2.2 Sensor analytics

Although recently proposed, few sensor analytic techniques have already been implemented. This section presents the key techniques that are currently in use.

1. Data fusion: Often, data generated within WSN has a high degree of redundancy owing to overlap in the area monitored by individual sensors. Data fusion techniques aim at combining the data originating from adjoining sensor nodes to reduce this redundancy as well as produce a more accurate result. Such reductions in data result into lesser packet exchange within the network, thereby, improving the energy efficiency of the sensor devices. The reduced traffic also leads to fewer collisions within the network, improving the reliability of data transfer within WSN. For a given signal, fusion algorithms can also be used to remove any noise and outliers. In [1], for instance, data fusion models are used to merge noisy measurements from multiple sensors to explore the fundamental limits of sensing coverage. Through extensive analysis, data fusion is shown to significantly improve the coverage of WSN by exploiting the collaboration among sensor devices. In [61], the authors propose a multi-sensor fusion technique for abnormal behaviour detection in elderly people with cognitive problems. The technique forms a major component of a multi-modal scheme to increase life autonomy of elderly people. Furthermore, data fusion algorithms have been shown

to be particularly useful in multimedia applications that generate bulky data. In [62], for instance, an information fusion-based mechanism is proposed for reducing the volume of data being transferred from a Wireless Multimedia Sensor Network to the cloud. Based on the granularity of data required by the end-users, two levels of information fusion have been proposed - low-level (data-level or feature-level fusion) and high-level (decision-level fusion). At first, larger granularity data is uploaded on the cloud while raw values are stored temporarily within the network. If the end user requirements are met satisfactorily, the raw values are deleted. Otherwise, sensors repeatedly extract fine-grained information from raw data and upload it on cloud till the end user is satisfied. In doing so, the system aims at optimizing the energy efficiency of sensor operation. A similar approach has been suggested in [63] for the design of energy-efficient Visual Sensor Networks. Merging of overlapping image data on-board visual sensors is expected to improve the longevity of such networks while improving the quality of information. Despite these advantages, however, data fusion algorithms are not widely used. This is because of the signal-specific nature of these algorithms, making it cumbersome to design specific techniques for the multitude of signals generated everyday.

2. Edge mining: Edge Mining proposes the implementation of light-weight data mining tasks on sensor devices [31]. Edge Mining has been realized using the Spanish Inquisition Protocol (SIP) as presented in [33]. Instead of sending raw data to the gateway, SIP converts the raw data into application relevant states that are transmitted to the cloud gateway only if the new state value cannot be predicted using the past estimates and an approximation model within a desired accuracy. Such changes are considered significant and marked as the occurrence of an event. Similar to data fusion, Edge Mining improves energy efficiency of the network by reducing the number of packet transmissions to the gateway. Moreover, it caters to the drawback of data fusion algorithms in that it is not signal-specific. So far, three Edge Mining algorithms have been proposed (as described below) based on different instantiations of the general-SIP approach.

    - Linear SIP (L-SIP): L-SIP represents the application state as a point-in-time value and rate of change. That is, it generates a linear model of the measured data. A number of techniques such as Kalman Filter, Exponentially Weighted Moving Average (EWMA) and Normalized Least Mean Squares (NLMS) can be used for state estimation. Only the state values that cannot be predicted from the previous state, within user-specified error bounds, are considered significant. Such values along with the rate-of-change are sent to the cloud to update the data model. L-SIP is preferred in applications that may require reconstruction of the signal.

- ClassAct: ClassAct is a decision tree-based classifier that represents the application state as a probability distribution over a given set of states [36]. Upon simplification, the index of the most probable state is chosen, and identified as an event if it differs from the last stored value. In [36], ClassAct is evaluated for acceleration-based classification of low-level activity states such as walking, standing and sitting. In comparison to L-SIP, ClassAct significantly reduces the number of packet transmissions by discarding most of the raw data. However, the signal transformation is destructive since the original signal cannot be reproduced with a good accuracy.
- Bare Necessities (BN): BN further reduces the raw data by storing only the summary of data over time. It estimates the application state as a distribution across non-overlapping bins and provides the relative time spent in each bin [64]. For any measurement, the change in state is considered eventful if the distribution of any bin changes by more than the given threshold. In [64], BN has been implemented for residential building monitoring to provide the proportion of time for which a room was in cold, comfortable, warm or overheated state based on temperature readings. The evaluation shows reduction in transmission frequency of temperature readings of the order of 99% for an allowed error rate of 10%. Owing to the drastic reduction in number of transmissions, an additional heartbeat mechanism is used to update the state if the time since the last packet transmission exceeds a given threshold. The periodic transmissions prevent large approximations in signal and also enable detection of failed nodes.

The Edge Mining-based event detection on-board sensor devices has the potential to improve the real-time responsiveness of WSN while optimizing the energy consumption of the network. However, a major drawback of Edge Mining is that it assumes WSN as a network of individual smart-sensing devices that perform mining tasks in isolation, with the primary aim of reducing network traffic. This limits its use where collaboration between sensor nodes is necessary for the design of application-specific services.

3. Artificial Neural Networks (ANN): ANN are distributed networks that work similar to the human nervous system. Every autonomous device imitates a neuron and performs

analysis based on multiple inputs and pre-defined functions. If the output is eventful, it is propagated to the next set of neurons/devices and further processed until relayed to the output node. ANN are widely deployed in applications of pattern recognition and classification. Owing to the distributed and parallel nature of WSN, researchers are investigating the possibility of mapping ANN onto WSN topologies. Towards this end, numerous parallels have been drawn between ANN and WSN [32]. The sensor nodes can be envisaged to form the input layer of the ANN which sense data and periodically transmit it to their neighbours (hidden nodes) for processing (using a sigmoid function). The nodes in the hidden layer of the topology then work in parallel to determine significant events to be relayed to the output node. Few neural network algorithms have been proposed for implementation in a WSN. In [65], authors discuss the implementation of a 'Smart Table' for identifying the presence of objects on a furniture and classifying them based on their size. In [66], Fuzzy Adaptive Resonance Theory (ART) and ARTMAP algorithms have been proposed for programming sensor devices to form a wireless electronic nose network that can decipher the presence of mal-odour gases and calculate their concentration. However, implementation of ANN algorithms in WSN is very limited, to date. This is because modifying WSN topology to map ANN algorithms is a complex task. Moreover, adjusting edge weights between nodes in order to facilitate the learning process within the network is compute-intensive. While some of the above applications (e.g. [65]) propose implementation an entire ANN within a single device for ease of implementaion, it may adversely affect the resource-efficiency of the sensor devices.

2.3 Localization in WSN

With increase in the number of WSN deployments, contextualization of sensor data is key for the design of future WSN. Accordingly, localization of sensor devices has gained significant importance to improve context-awareness of the system, as well as provide Location Based Services (LBS). In this regard, we propose real-time localization using sensor-based analytics as a proof-of-concept. The remainder of this section presents the state-of-the-art for localization in WSN.

Over the past years, numerous LBS such as navigation [59] and target tracking [67] have been proposed. In [59], for instance, design of LBS has been discussed for virtual fence applications that enable automated navigation of animals through a farm, under the umbrella of Precision Farming. In addition, LBS can be used to limit the use of pesticides to specific areas in farm to ensure efficient pest and waste management [52]. Similarly, user-mobility is studied to provide location-based health-care and entertainment services. In [68], outdoor and indoor localization of the elderly has been discussed to provide LBS for AAL applications. Continuous monitoring of user location facilitates accurate real-time responsiveness to behavioural anomalies in the elderly people, and thereby, enables independent living. Furthermore, localization of sensor devices has been used in computing applications such as routing that require details of the network traffic patterns for load balancing.

To facilitate these applications, several localization techniques have been proposed, to date. Traditionally, the use of GPS has been suggested for outdoor localization due to high accuracy as well as ease of integration of GPS receivers with sensor devices. The performance of these systems, however, often degrades in crowded or indoor environments due to the absence of line of sight to GPS satellites. The use of local positioning techniques in conjunction with GPS has accordingly been proposed to improve accuracy of localization. In [69], for instance, the authors propose a cooperative positioning technique that uses a local coordinate system to estimate relative position of mobile entities and, thereby, improve accuracy of GPS-based localization. Despite the performance improvements, GPS receivers are energy hungry units and negatively affect the lifetime of sensor devices. The use of WSN itself has also been proposed for localization purposes using anchor-based and anchor-less techniques. In the former, anchor nodes whose positions are pre-determined and fixed are marked as reference points, and distances between the unknown (mobile) nodes and reference points is calculated using range-based (e.g. received signal strength (RSS), angle of arrival (AoA)) or range-free measures (e.g. hop count) for localization [70]. A light intensity-based positioning system, for instance, has been discussed in [2]. The approach performs predictions using RSS measures in indoor environments. The latter is usually used in indoor applications where pre-determination of the co-ordinates of anchor nodes is not possible. An anchor-less localization approach using factor graphs and sum product algorithms, for instance, is discussed in [71]. An experimental evaluation of WSN-based indoor localization is presented in [72]. While these approaches are shown to perform well, their performance degrades in case of highly mobile targets wherein a large number of sensors have to remain active to track targets in all potential directions or in case targets move into holes in the deployment area. To solve these problems, authors in [67] propose the use of mobile sensors to follow targets directly for tracking applications.

Alternatively, Pedestrian Dead Reckoning (PDR) systems that make use of built-in inertial sensors in user wearable/smartphone have been proposed for localization. These systems rely on self-tracking, and estimate the user location based on past estimates and displacement over short intervals of time. An instance of a PDR system has been discussed in [73]. The system uses 8 IMU worn on the body, and a force sensor worn under the feet to capture joint movements for user localization. Another study in [74] presents a blind localization algorithm that combines data from built-in inertial and acoustic sensors in user smartphone to gauge the location of the smartphone. While these systems operate well in both outdoor and indoor environments, standalone PDR systems often accumulate error due to drift with walking distance over time. Assisted-PDR systems have, accordingly, been proposed to overcome these limitations. Certain examples make use of iBeacons [75] or ASMM [76] to correct the drift, and improve accuracy of localization. Of these, the ASMM approach presents a cost-effective solution as it requires minimum interaction with external third-party components.

2.4 Resource management via in-network analysis

As discussed earlier, the Fog Computing paradigm proposes computation offloading on nearby edge devices, instead of cloud-based servers. The offloading decision is taken so as to maximize the computation gain subject to a given objective function that aims at optimizing either the response time (i.e. minimize latency to meet application deadlines) or the network energy consumption (i.e. maximize network lifetime) or both. In [77], for instance, the authors design an optimization framework, namely MobiQoR, that minimizes service response time and application energy consumption through joint optimization of Quality of Result (QoR) and offloading strategy in a mobile edge computing environment. The key idea behind the framework is motivated by the observation that a growing number of edge applications allow a lower quality result. Thus, relaxing QoR in these applications can alleviate the workload of edge devices and, in turn, enable a significant reduction in response time and energy consumption. In another study [78], the authors propose mobile opportunistic computing wherein mobile devices leverage the compute resources in nearby heterogeneous devices including mobile devices, cloudlets, and cloud, in order to reduce the execution time and energy consumption. A high-level architecture of the system that considers routing, scheduling, discovery, securing and computation offloading has been presented. Alternatively, the use of computational resources at peer devices has been proposed. In [79], for instance, the authors present the design of a system, namely Serendipity, that allows computation offloading from one mobile device only to other other mobile devices intermittently connected with it, as opposed to remote cloud resources. The system incorporates a task allocation algorithm that decides how to apportion the computational task into sub-tasks and the allocation of such tasks to nearby devices in order to minimize the job completion time. In addition, a utility function that considers the energy consumption of all nodes participating in computation tasks as well as residual energy available on these nodes is proposed to allow energy-aware computing.

Given the constrained nature of sensor devices, recent studies have explored similar computation offloading and cooperative analysis for resource optimization within WSN. In [28], the authors present certain concepts that are relevant for the theory of distributed computation within WSN. The study reviews two-party and multi-hop communication complexity theories to address the computation of functions with distributed inputs between two or more nodes respectively, with an objective to minimize the computation time. Furthermore, distributed computation in networks subject to noise as well as randomized gossip-based approaches to compute aggregate functions has been discussed. In another study [80], the authors present a node cooperation-based scheme to ensure real-time processing of complicated tasks within large scale WSN. The proposed approach performs task grouping (using profiling) and task allocation (using node cooperation) to minimize processing delay as well as communication overheads within WSN. An energy-efficient cooperative computing model for battery-operated WSN has been proposed in [81]. The study models the application profile, computation and communication energy of a sensor, and presents an optimal task partition to minimize the total energy consumption (communication and computation) required for processing the application at local and remote sensor nodes in cooperative computing, given a target completion deadline. Moreover, an energy-aware cooperative node selection strategy is discussed. The use of these models with node selection strategy is expected to achieve a desirable trade-off between fairness and energy consumption at each node within the network. Furthermore, joint optimization of sensing, computation and communication tasks has been discussed for dynamic energy harvesting WSN in [82]. Based on realistic energy and network models, the authors formulate a stochastic optimization problem to make sensing rate control, communication and computation decisions, and propose a light-weight algorithm, namely Recycling Wasted Energy, to solve it. While these approaches study computation offloading for resource management in WSN, the use of distributed computing within WSN is still in it's infancy, and hasn't yet been realized in any practical study to the best of our knowledge. Moreover, the cooperative computation for executing functionality with distributed input data within the network has not yet been discussed. We attempt to address these gaps through the research presented in this thesis.

Chapter 3

Research Summary

In this work, we have addressed some of the key challenges in the design of future WSN as depicted in fig. 3.1. These networks are characterized as ubiquitous, intelligent, flexible, user-centric, cooperative, robust and energy efficient. In this regard, we have proposed the design of smart sensor devices that are capable of operating autonomously to predict application-relevant events in real-time despite lack of external resources and Internet connectivity [30]. We have considered sensor-based analytics using Edge Mining, a light-weight data mining approach for real-time event detection, and validated its performance for intelligent data collection in the context of Precision Farming [34]. We have also proposed IEM2.0, a decision-tree based classification technique, to allow for on-board localization to ensure context-awareness in future WSN applications [35]. Given the resource-constrained nature of sensor devices, we have further proposed the design of cooperative WSN that would allow distributed and parallel processing within WSN for improved predictive capabilities. We have designed two cooperative frameworks - Collaborative Edge Mining [37] for horizontal scaling of analysis across devices and Iterative Edge Mining [38] [39] for vertical scaling of analysis within a device, and illustrated their use for Precision Farming and Ambient Assisted Living applications. To ensure flexibility in analysis with changes in user requirements, we have suggested sensor-cloud integration wherein cloud-based machine learning is used to reconfigure sensor analytic models [40]. Finally, we have proposed energy management in future WSN through cooperative analytics. We have considered coalition-based cooperative analysis among devices for improved accuracy of event detection, which is, in turn, used for event-driven communication to minimize energy consumption on sensor devices [35]. Energy-aware cooperative analysis via computation offloading has also been considered for improved fairness in heterogeneous energy harvesting WSN of the future [42]. We believe that our work lays a strong foundation for the design of future WSN that are robust,

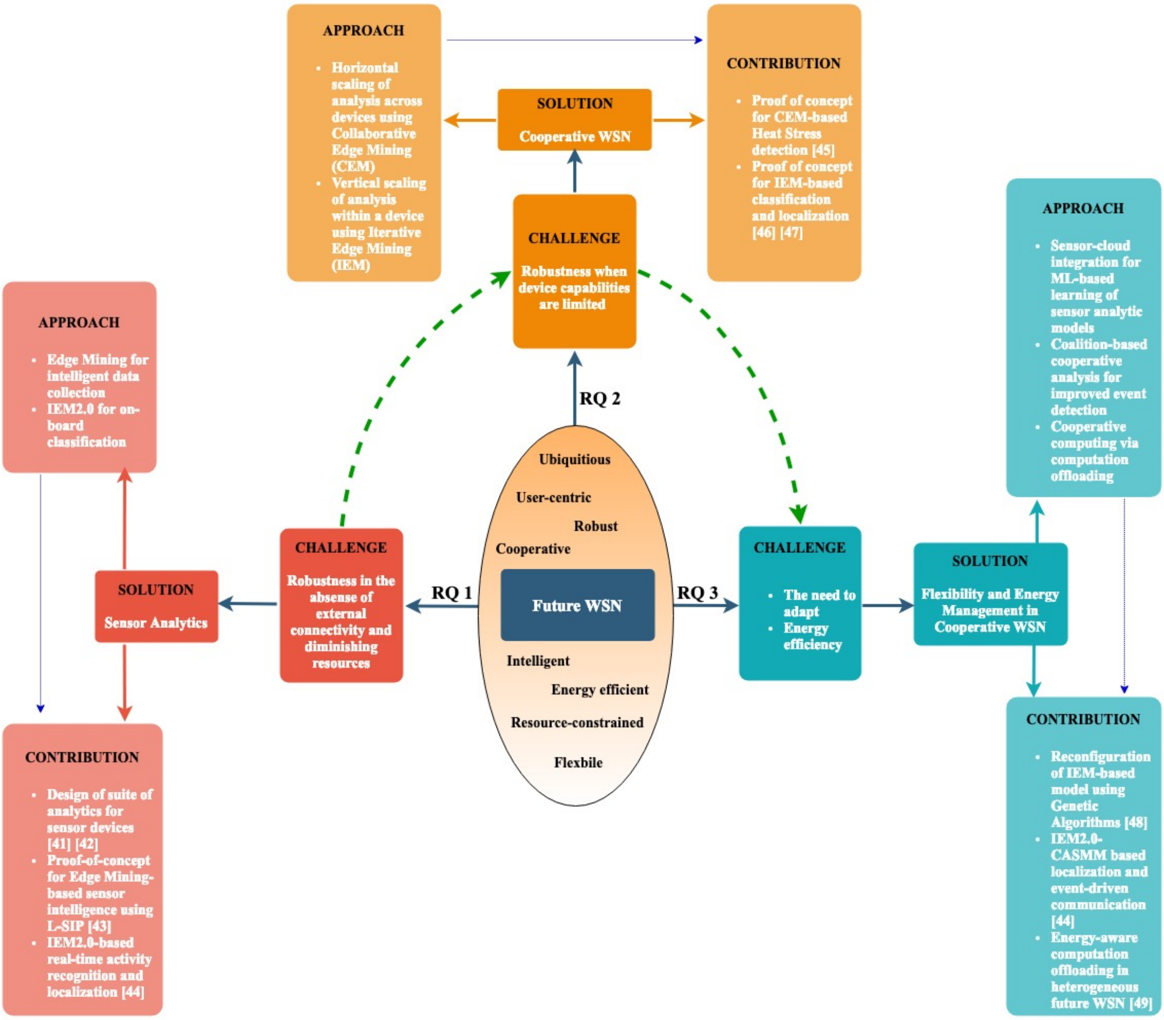


Fig. 3.1 Conceptual framework

intelligent, cooperative, flexible and energy-efficient, and suitable for deployment both in urban and rural IoT applications. The conclusions from each of the three research questions are discussed in further detail below, followed by certain future research trends in section 3.2.

3.1 Conclusions

The conclusions are presented as answers to the research questions that are also the objectives of this work as discussed in section 1.2.2.

- ***RQ1. Sensor analytics:*** *What pool of data analytics techniques are most suitable for on-board execution on miniature IoT devices? How are these techniques relevant in the context of Precision Farming?*

Conclusions: To support autonomous operation of WSN, the sensor devices should incorporate intelligence that is light-weight, extensible and allows real-time detection of events. This is particularly important in remote applications such as for Precision Farming that have limited access to other powerful edge devices such as gateways, and experience intermittent Internet connectivity [30]. The Edge Mining algorithms, namely L-SIP, BN and ClassAct, present a good solution for on-board data analytics. The algorithms perform light-weight data mining tasks on sensor devices and allow real-time detection of application state changes. This is, especially, beneficial for delay-critical applications related to animal health monitoring and mobility tracking. The use of L-SIP on-board sensor devices, for instance, has been shown to allow intelligent data collection within farms [34]. The approach generates linear models of data that are stored only at the occurrence of events. The approach, thus, reduces redundancy in data while improving real-time responsiveness of the system. Moreover, the approach is data agnostic, thus, making it suitable for Precision Farming applications that cater to large variety of data. Furthermore, since contextualization is key for the design of future WSN, as a proof-of-concept, real-time localization of animals has been proposed on-board sensor devices [35]. The approach implements decision-tree based activity recognition, and further performs ASMM for localization. It allows context-aware sensing as animals move around the farm, without relying on third-party components such as GPS and cloud. The approach is shown to be resource-efficient for implementation on sensor devices as well as extensible to other application domains.

- ***RQ2. Cooperative WSN:*** *What algorithmic frameworks can be designed to represent real-life services as a set of cooperative data analytic tasks deployed within WSN?*

  Conclusions: The algorithmic frameworks should allow scalability of analysis within WSN to improve the computational capability of the network, while ensuring load balancing and effective resource utilization for each device. Cognizant of this view, we designed two cooperative frameworks - CEM and IEM. CEM considers the availability of resources on devices, and allows horizontal scaling of computation by delegating suitable tasks to neighboring sensor devices [37]. The integration of results on each device generates application-specific information for delivering real-time services. In the context of Precision Farming, as a proof-of-concept, CEM has been used for energy-efficient (due to reduced packet transmissions) and timely detection of Heat Stress in cows. Alternatively, IEM can be used to scale-up the analysis performed by sensor devices through intelligent interaction between different tasks allocated to each device [38]. The framework allows sequential execution of co-related tasks on the same device such that the frequency of performing subsequent task(s) is governed by the output of previous process. The application-specific information is obtained through serial processing of all tasks. The coordinated interaction between tasks ensures that application requirements are met while not overwhelming the device. In Precision Farming, IEM can be used to facilitate acceleration-based behaviour analysis and mobility monitoring of animals. Furthermore, the framework has been shown suitable to extend to other application domains, for instance, localization of the elderly in behavioural-tracking AAL applications [39].

- ***RQ3. Flexibility and Energy Management in Cooperative WSN:*** *How can the services be re-configured on-the-go to meet the changes in application requirements while ensuring resource efficiency of the sensor devices?*

  Conclusions: We have shown that cloud-based learning using ML algorithms such as Genetic Algorithms (GA) can facilitate optimization of the sensor-analytic model based on user-preferences [40]. Flexibility in device intelligence and, in turn, network-wide services can be achieved by integrating WSN with cloud systems. A bi-directional link between sensor-cloud would allow transfer of data to cloud as well as transfer of the updated model to the devices. Based on application requirements, such flexibility can be used to improve accuracy as well as energy efficiency of sensor-based predictions. We demonstrate flexibility in IEM-based localization on-board sensor devices using GA-based learning on cloud. The use of GA allows fast learning and ease of adaptation to change in requirements by simply adjusting the weight of terms in the fitness function. Furthermore, cooperation between devices has a tremendous potential to improve network-wide resource management. Cooperative analysis through collective participation of devices in analytic tasks has been shown to improve accuracy of individual sensor-predictions [35]. Given an objective function, the improved accuracy of analysis can, in turn, be used for resource optimization. We illustrated its use for context-aware sensing and communication to reduce unnecessary packet transmissions to cloud and, in turn, minimize communication cost incurred by sensor devices. Alternatively, for tasks with large datasets, cooperative analysis using data partitioning and computation offloading has been discussed for joint optimization of computation and communication costs [42]. Evaluation of the theoretical models for heterogeneous energy harvesting WSN shows that the approach can significantly reduce the net energy consumption of the network while improving the energy balance across devices to improve network longevity.

3.2 Future work

While our research lays foundation for the design of future WSN, further work is required to improve the pool of analytics that can be deployed on sensor devices and improve scalability of our algorithms for large-scale networks. The following topics may serve as possible directions for continuation of this research.

- ***Using ML to model sensor functionality:*** Today, ML-based cloud learning is used to model crop growth and disease management. Moreover, it has been shown useful in optimizing performance of sensor-based analytics. In future, with design of more powerful and energy-harvesting sensor devices, it may be suitable to implement ML techniques within WSN. These models will facilitate real-time learning and enable adaptation/control of sensor-based services such as sensing frequency, computation complexity as well as communication framework to improve resource efficiency of devices based on different objective functions. Moreover, localized computations will improve the privacy of data. This, however, requires further work in identifying suitable ML techniques as well as the learning approach for mobile WSN.

- ***Validation and verification:*** Currently, research is primarily focused on the design of low-cost, energy-efficient sensor devices. In the context of Precision Farming, for instance, multiple vendors are working towards design of sensors for animal health monitoring such as SmartBow [83] and RumiWatch [84]. We have developed a suite of analytic algorithms that have a low-memory and energy footprint to be incorporated within such IoT devices. In future, there is a need to develop tools that would allow seamless integration of these algorithms with the different devices. Standardization of the suite of algorithms that operate on sensor devices would allow farmers to benchmark performance and improve decision-making within the farms. Furthermore, it would allow validation and verification of sensor-based analytics across different applications.

- ***Self-organizing WSN:*** So far, we have discussed re-configuration of sensor analytics on a device with change in application requirements. In addition to this, WSN themselves must be re-configurable and resilient to any disruption caused by energy depletion or physical damage to a certain device in order to ensure uninterrupted services. As such, future WSN must ensure adaptability by re-organizing themselves autonomously to allow lossless handover/migration of services between devices in case of a node failure. Such decisions should be made in real-time while ensuring minimum communication overhead between devices.

Chapter 4

List of Research Articles

In this chapter, we present a complete list of research articles that resulted from the PhD work. All articles are presented in reverse chronological order.

Published articles

P1. K. Bhargava, S. Ivanov, D. McSweeney, W. Donnelly, "Leveraging Fog Analytics for Context-Aware Sensing in Cooperative Wireless Sensor Networks," vol. 15, no. 2, pp. 23:1-23:35, Mar. 2019.

P2. C. Kulatunga, K. Bhargava, D. Vimalajeewa and S. Ivanov, "Cooperative in-network computation in energy harvesting device clouds," Sustainable Computing: Informatics and Systems, vol. 16, pp 106-116, Dec. 2017.

P3. K. Bhargava and S. Ivanov, "A fog computing approach for localization in WSN," IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Montreal, QC, pp. 1-7, Oct. 2017.

P4. K. Bhargava, G. McManus and S. Ivanov, "Fog-centric localization for ambient assisted living," International Conference on Engineering, Technology and Innovation (ICE/ITMC), Funchal, pp. 1424-1430, Jun. 2017.

P5. K. Bhargava, S. Ivanov, C. Kulatunga and W. Donnelly, "Fog-enabled WSN system for animal behavior analysis in precision dairy," International Conference on Computing, Networking and Communications (ICNC), Santa Clara, CA, pp. 504-510, Jan. 2017.

P6. K. Bhargava, S. Ivanov, W. Donnelly and C. Kulatunga, “Using Edge Analytics to Improve Data Collection in Precision Dairy Farming,” IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops), Dubai, pp. 137-144, Nov. 2016.

P7. K. Bhargava and S. Ivanov, “Collaborative Edge Mining for predicting heat stress in dairy cattle,” Wireless Days (WD), Toulouse, pp. 1-6 Mar. 2016.

P8. K. Bhargava, S. Ivanov and W. Donnelly, “Internet of Nano Things for Dairy Farming,” 2nd Annual International Conference on Nanoscale Computing and Communication (NANOCOM’ 15). ACM, New York, NY, USA, Sep. 2015.

P9. S. Ivanov, K. Bhargava and W. Donnelly, “Precision Farming: Sensor Analytics,” IEEE Intelligent Systems, vol. 30, no. 4, pp. 76-80, Jul-Aug. 2015.

Bibliography

[1] R. Tan, G. Xing, B. Liu, J. Wang, and X. Jia. Exploiting data fusion to improve the coverage of wireless sensor networks. *IEEE/ACM Transactions on Networking*, 20(2):450–462, Apr 2012.

[2] Bo Xie, Kongyang Chen, Guang Tan, Mingming Lu, Yunhuai Liu, Jie Wu, and Tian He. Lips: A light intensity–based positioning system for indoor environments. *ACM Trans. Sen. Netw.*, 12(4):28:1–28:27, September 2016.

[3] S. Sudevalayam and P. Kulkarni. Energy harvesting sensor nodes: Survey and implications. *IEEE Communications Surveys Tutorials*, 13(3):443–461, Jul 2011.

[4] J. Lee, C. Chuang, and C. Shen. Applications of short-range wireless technologies to industrial automation: A zigbee approach. In *2009 Fifth Advanced International Conference on Telecommunications*, pages 15–20, May 2009.

[5] J. Hughes, J. Yan, and K. Soga. Development of wireless sensor network using bluetooth low energy (ble) for construction noise monitoring. *Int. J. Smart Sens. Intell. Syst.*, 8(2):1379–1405, Jun 2015.

[6] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia, and N. Strachan. Evaluation of lora and lorawan for wireless sensor networks. In *2016 IEEE SENSORS*, pages 1–3, Oct 2016.

[7] M. Elkhodr, S. A. Shahrestani, and H. Cheung. Emerging wireless technologies in the internet of things: a comparative study. *CoRR*, abs/1611.00861, 2016.

[8] J. Ko, C. Lu, M. B. Srivastava, J. A. Stankovic, A. Terzis, and M. Welsh. Wireless sensor networks for healthcare. *Proceedings of the IEEE*, 98(11):1947–1960, Nov 2010.

[9] M. Franceschinis, L. Gioanola, M. Messere, R. Tomasi, M. A. Spirito, and P. Civera. Wireless sensor networks for intelligent transportation systems. In *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, pages 1–5, Apr 2009.

[10] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad, and M. Ismail. Energy-efficient wireless sensor networks for precision agriculture: A review. *Sensors (Switzerland)*, 17(8), 8 2017.

[11] L. M. Borges, F. J. Velez, and A. S. Lebres. Survey on the characterization and classification of wireless sensor network applications. *IEEE Communications Surveys Tutorials*, 16(4):1860–1890, 2014.

[12] J. Hao, B. Zhang, and H. T. Mouftah. Routing protocols for duty cycled wireless sensor networks: A survey. *IEEE Communications Magazine*, 50(12):116–123, Dec 2012.

[13] C. Kulatunga, L. Shalloo, W. Donnelly, E. Robson, and S. Ivanov. Opportunistic wireless networking for smart dairy farming. *IT Professional*, 19(2):16–23, Mar 2017.

[14] S. Ivanov, S. Balasubramaniam, D. Botvich, and O. B. Akan. Gravity gradient routing for information delivery in fog wireless sensor networks. *Ad Hoc Networks*, 46:61 – 74, 2016.

[15] Z. Guo, Q. Wang, M. Li, and J. He. Fuzzy logic based multidimensional link quality estimation for multi-hop wireless sensor networks. *IEEE Sensors Journal*, 13(10):3605–3615, Oct 2013.

[16] Apache. Apache hadoop. http://hadoop.apache.org/, 2011. [Online; accessed Aug. 2016].

[17] Apache. Apache storm. http://hortonworks.com/hadoop/storm/, 2011. [Online; accessed Sep. 2018].

[18] Apache. The apache spark project. http://spark.apache.org/, 2012. [Online; accessed Aug. 2016].

[19] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[20] T. Hu, H. Chen, L. Huang, and X. Zhu. A survey of mass data mining based on cloud-computing. In *Anti-counterfeiting, Security, and Identification*, pages 1–4, Aug 2012.

[21] A. Grogan. Smart farming. *Engineering Technology*, 7(6):38–40, Jul 2012.

[22] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. 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 '12, pages 13–16, 2012.

[23] S. Sarkar, S. Chatterjee, and S. Misra. Assessment of the suitability of fog computing in the context of internet of things. *IEEE Transactions on Cloud Computing*, 6(1):46–59, Jan 2015.

[24] S. M. A. Oteafy and H. S. Hassanein. IoT in the fog: A roadmap for data-centric IoT development. *IEEE Communications Magazine*, 56(3):157–163, Mar 2018.

[25] OpenFog Consortium Architecture Working Group. Openfog reference architecture for fog computing. openfogconsortium.org, 2017. [accessed Aug. 2018].

[26] S. Dey, A. Mukherjee, H. S. Paul, and A. Pal. Challenges of using edge devices in iot computation grids. In *2013 International Conference on Parallel and Distributed Systems*, pages 564–569, Dec 2013.

[27] M. Chen, Y. Hao, Y. Li, C. Lai, and D. Wu. On the computation offloading at ad hoc cloudlet: architecture and service modes. *IEEE Communications Magazine*, 53(6):18–24, June 2015.

[28] A. Giridhar and P. R. Kumar. Toward a theory of in-network computation in wireless sensor networks. *IEEE Communications Magazine*, 44(4):98–107, Apr 2006.

[29] S. Tennina, M. Valletta, F. Santucci, M. D. Renzo, F. Graziosi, and R. Minutolo. Entity localization and tracking: A sensor fusion-based mechanism in wsns. In *2011 IEEE International Conference on High Performance Computing and Communications*, pages 983–988, Sep 2011.

[30] S. Ivanov, K. Bhargava, and W. Donnelly. Precision farming: Sensor analytics. *IEEE Intelligent Systems*, 30(4):76–80, July 2015.

[31] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic. Edge mining the internet of things. *IEEE Sensors Journal*, 13(10):3816–3825, Oct 2013.

[32] G. Serpen, J. Li, L. Liu, and Z. Gao. Wsn-ann: Parallel and distributed neurocomputing with wireless sensor networks. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Aug 2013.

[33] D. Goldsmith and J. Brusey. The spanish inquisition protocol - model based transmission reduction for wireless sensor networks. In *2010 IEEE Sensors*, pages 2043–2048, Nov 2010.

[34] K. Bhargava, S. Ivanov, W. Donnelly, and C. Kulatunga. Using edge analytics to improve data collection in precision dairy farming. In *2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops)*, pages 137–144, Nov 2016.

[35] Kriti Bhargava, Stepan Ivanov, Diarmuid McSweeney, and William Donnelly. Leveraging fog analytics for context-aware sensing in cooperative wireless sensor networks. *ACM Trans. Sen. Netw.*, 15(2):23:1–23:35, 2019.

[36] R. Rednic, E. Gaura, and J. Brusey. Classact: Accelerometer-based real time activity classifier. In *Proceedings of 2nd WiSIG Showcase*, pages 21–25, Jul 2009.

[37] K. Bhargava and S. Ivanov. Collaborative edge mining for predicting heat stress in dairy cattle. In *2016 Wireless Days (WD)*, pages 1–6, March 2016.

[38] K. Bhargava, S. Ivanov, C. Kulatunga, and W. Donnelly. Fog-enabled wsn system for animal behavior analysis in precision dairy. In *2017 International Conference on Computing, Networking and Communications (ICNC)*, pages 504–510, Jan 2017.

[39] K. Bhargava, G. McManus, and S. Ivanov. Fog-centric localization for ambient assisted living. In *2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, pages 1424–1430, Jun 2017.

[40] K. Bhargava and S. Ivanov. A fog computing approach for localization in wsn. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7, Oct 2017.

[41] Tim van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, UbiComp '08, 2008.

[42] C. Kulatunga, K. Bhargava, D. Vimalajeewa, and S. Ivanov. Cooperative in-network computation in energy harvesting device clouds. *Sustainable Computing: Informatics and Systems*, 16:106 – 116, 2017.

[43] Advanticsys. Mtm-cm5000-msp. advanticsys.com, 2017. [accessed Aug. 2017].

[44] Invensense. Mpu-9255 product specification (rev. 1.0). stanford.edu, 2014. [accessed Sep. 2017].

[45] U-blox. Neo-6 (data sheet). www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_%28GPS.G6-HW-09005%29.pdf, 2010. [accessed Aug. 2018].

[46] Wiki. Tinyos documentation wiki. tinyos.stanford.edu/tinyos-wiki, 2015. [accessed Nov. 2016].

[47] Raspberry Pi Foundation. Raspberry pi 2 model b. raspberrypi.org, 2016. [accessed May. 2016].

[48] Inria. Simgrid: Versatile simulation of distributed systems. http://simgrid.gforge.inria.fr/simgrid/latest/doc/, 2017. [accessed Aug. 2018].

[49] K. Bhargava, S. Ivanov, and W. Donnelly. Internet of nano things for dairy farming. In *Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication*, NANOCOM' 15, pages 24:1–24:2. ACM, 2015.

[50] S. K. Seelan, S. Laguette, G. M. Casady, and G. A. Seielstad. Remote sensing applications for precision agriculture: A learning community approach. *Remote Sensing of Environment*, 88(1):157 – 169, 2003.

[51] O. Elijah, T. A. Rahman, I. Orikumhi, C. Y. Leow, and M. N. Hindia. An overview of internet of things (iot) and data analytics in agriculture: Benefits and challenges. *IEEE Internet of Things Journal*, pages 1–1, 2018.

[52] A. Rehman, A. Z. Abbasi, N. Islam, and Z. A. Shaikh. A review of wireless sensors and networks' applications in agriculture. *Computer Standards & Interfaces*, 36(2):263 – 270, 2014.

[53] K. Taylor, C. Griffith, L. Lefort, R. Gaire, M. Compton, T. Wark, D. Lamb, G. Falzon, and M. Trotter. Farming the web of things. *IEEE Intelligent Systems*, 28(6):12–19, Nov 2013.

[54] W. Steeneveld and H. Hogeveen. Characterization of dutch dairy farms using sensor systems for cow management. *Journal of Dairy Science*, 98(1):709 – 717, 2015.

[55] F. A. Auat Cheein and R. Carelli. Agricultural robotics: Unmanned robotic service units in agricultural tasks. *IEEE Industrial Electronics Magazine*, 7(3):48–58, Sep 2013.

[56] C.J. Rutten, A.G.J. Velthuis, W. Steeneveld, and H. Hogeveen. Invited review: Sensors to support health management on dairy farms. *Journal of Dairy Science*, 96(4):1928–1952, 2013.

[57] M. Alsaaod, C. Römer, J. Kleinmanns, Kathrin Hendriksen, S. Rose-Meierhöfer, L. Plümer, and W. Büscher. Electronic detection of lameness in dairy cows through measuring pedometric activity and lying behavior. *Applied Animal Behaviour Science*, 142(3):134 – 141, 2012.

[58] M. O. Monod, P. Faure, L. Moiroux, and P. Rameau. A virtual fence for animals management in rangelands. In *MELECON 2008 - The 14th IEEE Mediterranean Electrotechnical Conference*, pages 337–342, May 2008.

[59] T. Wark, D. Swain, C. Crossman, P. Valencia, G. Bishop-Hurley, and R. Handcock. Sensor and actuator networks: Protecting environmentally sensitive areas. *IEEE Pervasive Computing*, 8(1):30–36, Jan 2009.

[60] H. Chen and R. Yada. Nanotechnologies in agriculture: New tools for sustainable development. *Trends in Food Science & Technology*, 22(11):585–594, 2011.

[61] G. Hernández-Peñaloza, A. Belmonte-Hernández, M. Quintana, and F. Álvarez. A multi-sensor fusion scheme to increase life autonomy of elderly people with cognitive problems. *IEEE Access*, 6:12775–12789, 2018.

[62] Z. J. Zhang, C. F. Lai, and H. C. Chao. A green data transmission mechanism for wireless multimedia sensor networks using information fusion. *IEEE Wireless Communications*, 21(4):14–19, Aug 2014.

[63] A. Marcus and O. Marques. An eye on visual sensor networks. *IEEE Potentials*, 31(2):38–43, Mar 2012.

[64] E. I. Gaura, J. Brusey, and R. Wilkins. Bare necessities - knowledge-driven wsn design. In *2011 IEEE SENSORS Proceedings*, pages 66–70, Oct 2011.

[65] S. G. Soares, A. F. da Rocha, T. M. G. de A. Barbosa, and R. A. de Matos Araújo. Embedding a neural network into wsn furniture. In *2010 10th International Conference on Hybrid Intelligent Systems*, pages 219–222, Aug 2010.

[66] Y. W. Kim, S. J. Lee, G. H. Kim, and G. J. Jeon. Wireless electronic nose network for real-time gas monitoring system. In *2009 IEEE International Workshop on Robotic and Sensors Environments*, pages 169–172, Nov 2009.

[67] Tian Wang, Zhen Peng, Junbin Liang, Sheng Wen, Md Zakirul Alam Bhuiyan, Yiqiao Cai, and Jiannong Cao. Following targets for mobile tracking in wireless sensor networks. *ACM Trans. Sen. Netw.*, 12(4):31:1–31:24, September 2016.

[68] L. Mainetti, L. Patrono, A. Secco, and I. Sergi. An iot-aware aal system for elderly people. In *2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech)*, pages 1–6, Jul 2016.

[69] H. Hlavacs and K. A. Hummel. Cooperative positioning when using local position information: Theoretical framework and error analysis. *IEEE Transactions on Mobile Computing*, 12(10):2091–2104, Oct 2013.

[70] U. Nazir, N. Shahid, M. A. Arshad, and S. H. Raza. Classification of localization algorithms for wireless sensor network: A survey. In *2012 International Conference on Open Source Systems and Technologies*, pages 1–5, Dec 2012.

[71] U. Ferner, H. Wymeersch, and M. Z. Win. Cooperative anchor-less localization for large dynamic networks. In *2008 IEEE International Conference on Ultra-Wideband*, volume 2, pages 181–185, Sep 2008.

[72] A. Maddumabandara, H. Leung, and M. Liu. Experimental evaluation of indoor localization using wireless sensor networks. *IEEE Sensors Journal*, 15(9):5228–5237, Sep 2015.

[73] Q. Yuan and I. . Chen. 3-d localization of human based on an inertial capture system. *IEEE Transactions on Robotics*, 29(3):806–812, Jun 2013.

[74] D. Ayllón, H. Sánchez-Hevia, R. Gil-Pita, and M. Rosa-Zurera. Indoor blind localization of smartphones by means of sensor data fusion. In *2015 IEEE Sensors Applications Symposium (SAS)*, pages 1–6, Apr 2015.

[75] Z. Chen, Q. Zhu, and Y. C. Soh. Smartphone inertial sensor-based indoor localization and tracking with ibeacon corrections. *IEEE Transactions on Industrial Informatics*, 12(4):1540–1549, Aug 2016.

[76] B. Zhou, Q. Li, Q. Mao, W. Tu, and X. Zhang. Activity sequence-based indoor pedestrian localization using smartphones. *IEEE Transactions on Human-Machine Systems*, 45(5):562–574, Oct 2015.

[77] Y. Li, Y. Chen, T. Lan, and G. Venkataramani. Mobiqor: Pushing the envelope of mobile edge computing via quality-of-result optimization. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 1261–1270, Jun 2017.

[78] A. Mtibaa, K. A. Harras, K. Habak, M. Ammar, and E. W. Zegura. Towards mobile opportunistic computing. In *2015 IEEE 8th International Conference on Cloud Computing*, pages 1111–1114, Jun 2015.

[79] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura. Serendipity: Enabling remote computing among intermittently connected mobile devices. In *Proceedings of the Thirteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '12, pages 145–154, 2012.

[80] J. Jiang, G. Han, and C. Zhu. A complicated task solution scheme based on node cooperation for wireless sensor networks. In *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pages 264–269, Dec 2016.

[81] Z. Sheng, C. Mahapatra, V. C. M. Leung, M. Chen, and P. K. Sahu. Energy efficient cooperative computing in mobile wireless sensor networks. *IEEE Transactions on Cloud Computing*, 6(1):114–126, Jan 2018.

[82] S. Yang, Y. Tahir, P. Chen, A. Marshall, and J. McCann. Distributed optimization in energy harvesting sensor networks with dynamic in-network data processing. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, Apr 2016.

[83] SmartBow. User manual: E093 smartbow eartag. smartbow.com, 2014. [accessed Oct. 2017].

[84] J. Werner, L. Leso, C. Umstatter, J. Niederhauser, E. Kennedy, A. Geoghegan, L. Shalloo, M. Schick, and B. O'Brien. Evaluation of the rumiwatchsystem for measuring grazing behaviour of cows. *Journal of Neuroscience Methods*, 300:138 – 146, 2018.

Appendix A

Precision Farming: Sensor Analytics

Journal Title:	IEEE Intelligent Systems
Article Type	Magazine
Complete Author List	Stepan Ivanov, Kriti Bhargava and William Donnelly
Status	Published: vol. 30, no. 4, pp. 76-80, Jul.-Aug. 2015
Contribution	My contribution to this research article was identification of applications in Precision Farming, particularly automation, surveying existing data analytics techniques for WSN, and recognizing challenges in deployment of sensor analytics. I also took part in writing some parts of the paper and proof-reading.

Editor: **Daniel E. O'Leary,** University of Southern California, oleary@usc.edu

Precision Farming: Sensor Analytics

Stepan Ivanov, Kriti Bhargava, and William Donnelly, *Waterford Institute of Technology*

In agriculture, the use of wireless sensor networks (WSNs) is strongly advocated under the umbrella of precision farming. Today, a substantial amount of research focuses on developing efficient WSN systems that will provide fine-grained monitoring and automation of the farming processes. In the future, multiple WSN systems deployed on every farm will form an integrated environment (see Figure 1) covering various aspects of farm management. Intelligent insight gained using the environments will help improve future farming.

Analytics in Precision Farming

Some agricultural sensor systems exist already. In this article, we describe how these systems' designs and intelligence levels vary depending on the application area.

Monitoring

Kerry Taylor and colleagues describe a WSN system deployed on Kirby Farm near Armidale, New South Wales.[1] The network incorporates several different sensors for soil moisture, air temperature, humidity and pressure, rainfall, and hail. Monitoring data, provided by the sensors, is pushed to a centralized entity, where the data is enriched and analyzed. Results of the analysis are communicated back to farm personnel via their personal mobile devices. In this way, the system presents an exemplar decision-support solution for pasture management.

At the same time, C.J. Rutten and colleagues have conducted a comprehensive survey showing lack of analytics and intelligence in sensor systems for animal-health management on a dairy farm.[2] The survey presents a four-tier classification of the existing sensor solutions. The first tier is intended for systems that measure specific animal-health aspects. Readings obtained by such systems (for example, milk composition) are presented as is to farm personnel, who draw their own conclusions. Second-tier systems interpret the readings. For instance, the electrical conductivity of milk can be used to identify mastitis (inflammation of the udder tissue and mammary gland, which typically occurs due to bacterial infection), and 3D acceleration can be used to detect locomotion problems. Third-tier systems integrate information coming from various sources, including other sensor systems and nonsensor data. Such a system would help farm personnel comprehend potentially large amounts of data, which could become overwhelming. Rutten and colleagues have identified a serious lack of such systems.[2] The fourth tier is for decision support systems. Even though some of the sensor systems have been identified as tier four, their functionality is based on alerting second-tier events, such as mastitis or estrus (a period of fertility and sexual receptivity in female mammals).

Automation

Context and user awareness are common trends in modern farm-process automation systems. For example, Fernando Auat Cheein and Ricardo Carelli surveyed unmanned robotic systems proposed for farming applications.[3] Such a system typically attempts to replace manual labor in tasks, such as monitoring the farm environment, mapping monitoring results (such as yield mapping), and taking particular actions (for example, blossom thinning). The system might include several different sensors (for example, for pollen or CO_2 level) and operate within the deployment area of a static WSN.

Virtual fence systems represent another type of context awareness.[4] Traditionally, physical barriers and fences are used to control animal movement within a farm's geographical limits. A virtual fence system aims to accomplish such control without using physical barriers. The system attempts to identify an animal's location (for example, via GPS) and if needed (for example, if the animal is outside a particular area), direct the animal's

1541-1672/15/$31.00 © 2015 IEEE
Published by the IEEE Computer Society

movement using a specific stimulus (such as electric or audio). Such systems must be aware of both their surroundings and the context. Navigation within the farm's limits requires specific landscape knowledge, which might need to incorporate weather information or the presence of specific farm objects and people.

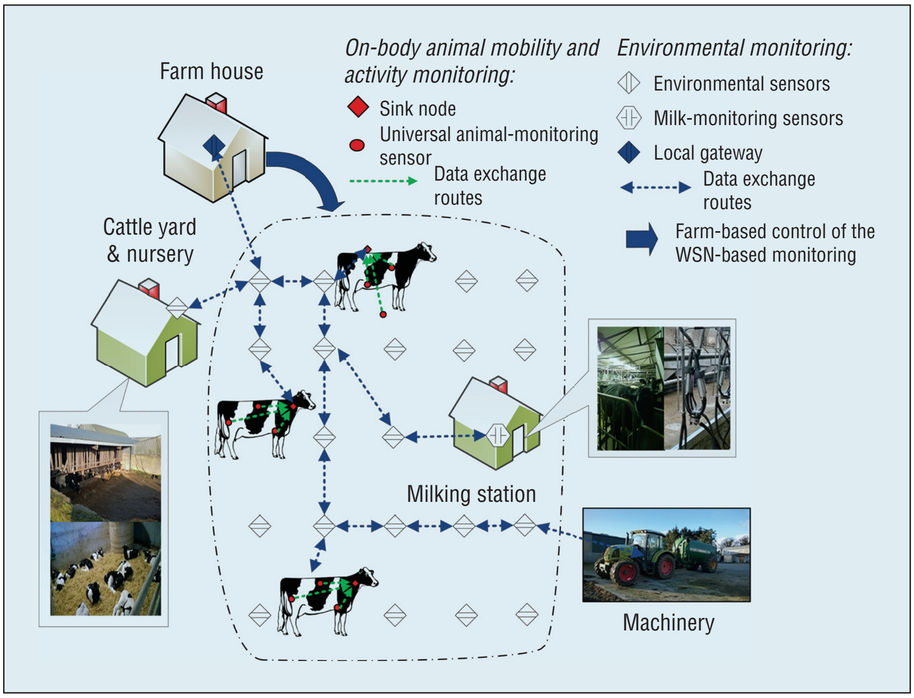


Figure 1. Integrated farm management environment based on wireless sensor networks (WSNs). The environment incorporates several sensor systems, including those for on-body animal health monitoring, environmental surveillance, and automated farm machinery. The local gateway, part of the environment, is used for remote control of the environment and access to the sensor-based monitoring data.

Wireless Sensor Networks Today

In the future, the demand for intelligent agricultural systems will become apparent. Accurate, timely analyses of vast amounts of data provided by integrated WSN environments will become paramount in increasing efficiency and sustainability of agriculture. Conventionally, analyzing monitoring data is considered to be beyond WSN capabilities and carried out remotely. Meanwhile, in recent years, design and manufacturing of WSNs have improved dramatically. Modern wireless sensors possess computational capacity sufficient for certain data analytics functionality. In this section, we overview modern WSN functionality that already incorporates some level of intelligence.

Data Fusion

Often, data generated within a WSN has a degree of redundancy owing to overlapping in monitoring of individual sensors. In this case, merging data from different sensors reduces information exchange within the WSN and therefore increases its efficiency, energy consumption, and reliability (see, for example, work by Rui Tan and colleagues[5]). Reaching the best possible traffic reduction is the main objective of data fusion (see Figure 2a). Data fusion can also involve such network-wide analytics tasks as noise suppression and outlier identification. Some later examples of sensor data fusion consider rather complex applications, such as multimedia streaming. Fusion of such data requires intelligence from the sensor nodes. Yet this intelligence is very specific and seldom can be applied outside a particular application range.

Edge Mining

Elena Gaura and colleagues have proposed a novel approach called *edge mining* (see Figure 2b) to reduce network traffic in a WSN.[6] Each sensor reports (for example, to a cloud-based service via a gateway) occurrence of events within the environment rather than raw sensor readings. The sensor systematizes its historical readings using a simplistic internal representation model that is based on linear, decision-tree, or histogram representation of the readings. The model is constantly updated with regards to the new readings, and substantial changes of the model are attributed to events within the environment. This represents a new stage in sensor intelligence. The representation model is indeed the view of the environment that the sensor builds autonomously and uses to make decisions (detect events). The generic nature of the models used by the sensors makes edge mining suitable for a range of sensor readings. However, the approach is limited to analysis conducted by the sensors in isolation from each other, which is a substantial limitation.

Bioinspired Analytics

Bioinspired solutions are common for both distributed data analytics and WSNs. Thus, the recent appearance of seminal joint WSNs and data analytics solutions on the fertile ground of bioinspired computing is not surprising. Gursel Serpen and colleagues mapped operation of an artificial neural network (ANN; see Figure 2c) to an ad hoc WSN.[7] Each sensor, apart from its own original monitoring task, hosts functionality of a neuron, whereas sensor-node communication is used for information exchange between the neurons. Although the

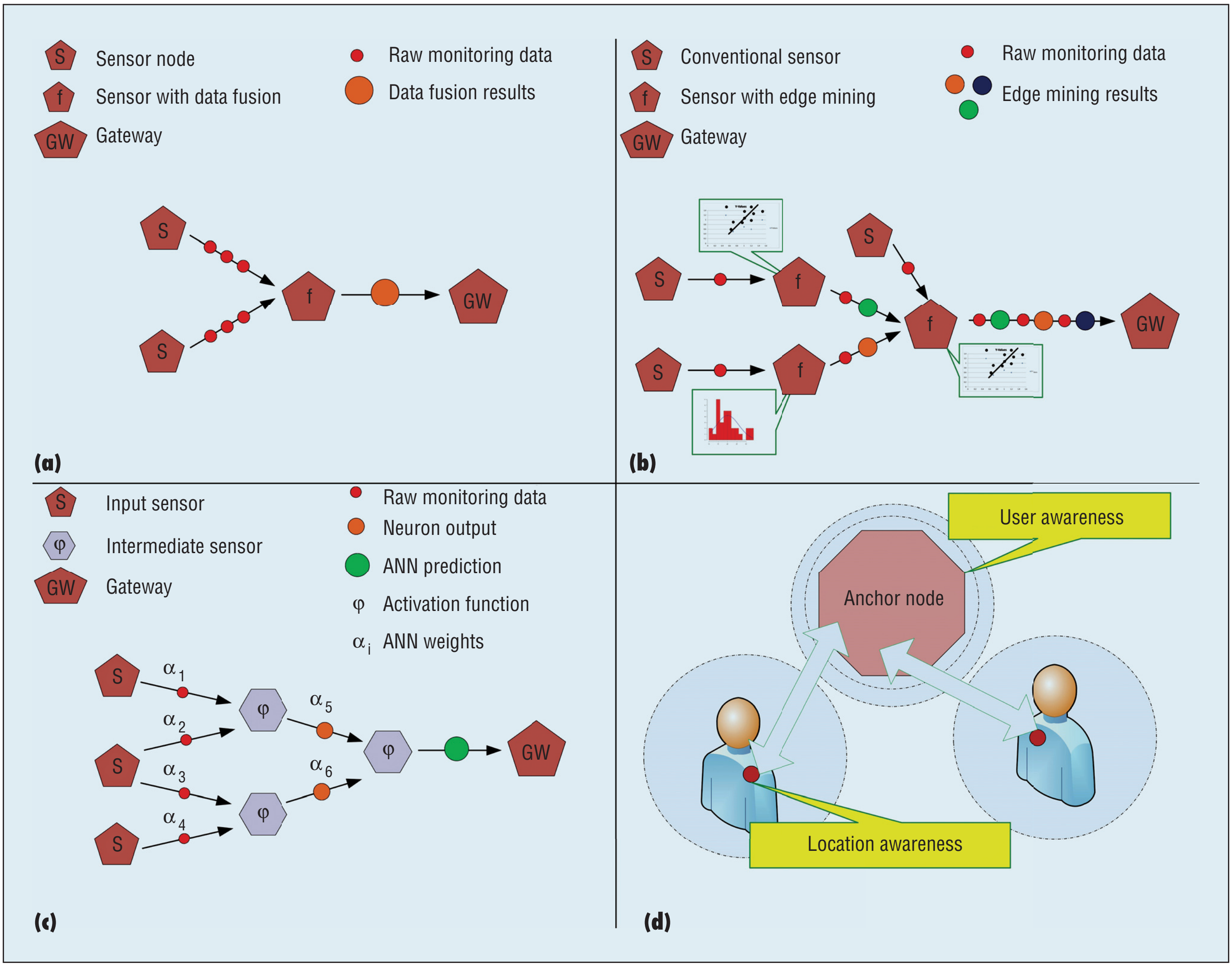


Figure 2. WSN intelligence. (a) Data fusion. (b) Edge mining. (c) Bioinspired analytics. (d) Location and user awareness. The figure depicts schematics of typical implementations of the four types of WSN intelligence.

proposed solution considers only one application, the overall approach can be extended for other analytics tasks. This comes together with the high adaptation of an ANN, where changing the ANN's topology modifies its analytics. However, identification of a suitable ANN topology and network learning are typically computationally complex tasks.

Location Awareness

Researchers have developed several methods to identify a sensor's location, representing a separate strain of sensor intelligence (see Figure 2d). Thus, particular nodes could be aware of their static location or equipped with specialized modules that can determine node positioning directly (such as GPS or GLONAS, the Global Navigation Satellite System). Nodes with known locations could serve as so-called anchor points to identify the rest of the network's location (for example, using received signal strength or delivery delay). Node acceleration and speed (which may be available from the on-board accelerometer of the node itself or its immediate neighbors) can be used for continuous location correction and update. Enriching WSN data with the location information can increase its location awareness and therefore improve its services.[8]

User Awareness

The broadcast nature of wireless transmission requires devices operating within the same frequency range to cooperate to ensure their coexistence. For instance, concurrent transmissions are typically scheduled at the media access control layer of the WSN architecture to reduce mutual interference and avoid collisions. Such cooperation lets nodes learn about each other and their current and potential future use (Figure 2d). For example, the presence of a particular type of wireless device (such as a smart watch worn by the primary user) could indicate the user's presence. Enriching operation of the WSN with such knowledge increases awareness of its services, making them user aware. Yet, secondary WSN data is typically used exclusively to improve and support its ability to transport sensor reading.

Table 1. WSN intelligence today: technique comparison.

Characteristics	Data fusion	Edge mining	Bioinspired analytics	Location and user awareness
Provides network-wide analysis	Yes	No	Yes	No
Is application agnostic	No	Yes	Yes	Yes
Implies computationally complex learning	—	No	Yes	—

Challenges and Benefits

In agriculture, the need for data analytics and the ongoing expansion of the WSNs (see Table 1) create an exciting opportunity for a new technological development on the brink of the two seemingly distant branches. Thus, the WSNs themselves can execute some of the data analytics' functionality. This development promises many benefits, including the following:

- *Accessibility of data insight*. Results from analytics incorporated by a farm-based WSN (that is, farm insight) will become immediately available within the farming environment, whereas additional delays due to external communication (such as an analytics engine hosted on a cloud) will be avoided. This is particularly important for farms situated in rural areas with poor Internet connectivity.
- *Sustainable computing*. Originally used to increase the life of battery-powered sensors, energy harvesting has become an imminent part of modern WSN design. Operation of a WSN is typically powered (even partially) by various renewable sources, such as solar and wind energy. Therefore, analytical tasks completed within the network will also be powered by renewable energy, leading to a greener, more eco-friendly way of computing.
- *Information security*. Performing some of the data analytics directly within the farm also has a number of benefits from an information security viewpoint. Thus, only a summary (such as an average of particular measurements across the farm) of the farm-monitoring data will need to be shared with a third party (cloud-based analytics). This will subsequently complicate obtaining specific sensitive information from the data shared.

Meanwhile, building a viable agricultural WSN system with elements of sensor analytics poses several challenges that must be addressed to not only increase robust operation of the system but also guarantee its value to the end user—the farmer:

- *Data analytics functionality*. Particular attention must be dedicated to identifying the pool of data analytics methods that will be delegated to WSNs. The methods must not only be highly parallel, but their implementation must account for the potential instability of WSNs (such as low energy budget of specific nodes or impaired communication due to interference). To ensure market uptake, the methods implemented by the WSN must be of potentially high practical value to the farmer. The methods must incorporate a degree of flexibility to reflect potential modifications in farm-management strategy.
- *Resource tradeoff*. Analytics will present WSN functionality additional to its original farm-monitoring tasks. Hence, the WSN should perform an analytics task such that it does not significantly impact the farm monitoring itself. A tradeoff between analytics functionality implemented by the WSN and resource expenditure must be considered.
- *Integrated cyber-physical systems*. Systems deployed on a particular farm will form an integrated environment. The environment will need to be catered to various types of information, stretching from delay tolerant (for example, milk fat/protein content) to real time (heartrate of newborn cubs kept in a nursery). Communication of the environment needs to incorporate autonomy, robustness to harsh environmental conditions, high efficiency (such as low energy consumption and high transmission quality), and flexibility (to reflect changes in analytics functionality).

Addressing these challenges and introducing analytics into farm-based WSN systems has the potential to increase these systems' practical value and significantly extend their usage.

Multiple WSN systems deployed on a farm will form an integrated environment providing farm personnel with valuable farming insight rather than raw sensor readings. This will significantly simplify management of the farm and increase efficiency and sustainability of its operation. ■

Acknowledgments

This work has received support from the Science Foundation Ireland (SFI) and the Agriculture and Food Development Authority, Ireland (TEAGASC) as part of the SFI

TEAGASC Future Agri-Food Partnership, in a project titled "Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems."

References

1. K. Taylor et al., "Farming the Web of Things," *IEEE Intelligent Systems*, vol. 28, no. 6, 2013, pp. 12–19.
2. C.J. Rutten et al., "Invited Review: Sensors to Support Health Management on Dairy Farms," *J. Dairy Science*, vol. 96, no. 4, 2013, pp. 1928–1952.
3. F.A. Auat Cheein and R. Carelli, "Agricultural Robotics: Unmanned Robotic Service Units in Agricultural Tasks," *IEEE Industrial Electronics*, vol. 7, no. 3, 2013, pp. 48–58.
4. C. Umstatter, "The Evolution of Virtual Fences: A Review," *Computers and Electronics in Agriculture*, vol. 75, no. 1, 2011, pp. 10–22.
5. R. Tan et al., "Exploiting Data Fusion to Improve the Coverage of Wireless Sensor Networks," *IEEE/ACM Trans. Networking*, vol. 20, no. 2, 2012, pp. 450–462.
6. E.I. Gaura et al., "Edge Mining the Internet of Things," *IEEE Sensors*, vol. 13, no. 10, 2013, pp. 3816–3825.
7. G. Serpen et al., "WSN-ANN: Parallel and Distributed Neurocomputing with Wireless Sensor Networks," *Proc. Int'l Joint Conf. Neural Networks* (IJCNN), 2013, pp. 1–8.
8. N. Thang et al., "Location-Aware Visual Radios," *IEEE Wireless Comm.*, vol. 21, no. 4, 2014, pp. 28–36.

Stepan Ivanov is a postdoctoral researcher in the Telecommunication Software and Systems Group at the Waterford Institute of Technology. Contact him at sivanov@wit.ie.

Kriti Bhargava is a PhD student in the Telecommunication Software and Systems Group at the Waterford Institute of Technology. Contact her at kbhargava@tssg.org.

William Donnelly is the president of the Waterford Institute of Technology. Contact him at wdonnelly@wit.ie.

Appendix B

Internet of Nano Things for Dairy Farming

Conference Title:	2nd ACM International Conference on Nanoscale Computing and Communication (NanoCom 2015)
Article Type	Work-in-progress
Complete Author List	Kriti Bhargava, Stepan Ivanov and William Donnelly
Status	Published: Sep. 2015

Internet of Nano Things for Dairy Farming

Kriti Bhargava
Telecommunications Software & Systems Group
Waterford Institute of Technology
kbhargava@tssg.org

Stepan Ivanov
Telecommunications Software & Systems Group
Waterford Institute of Technology
sivanov@tssg.org

William Donnelly
Telecommunications Software & Systems Group
Waterford Institute of Technology
wdonnelly@wit.ie

ABSTRACT

This material is based on work in progress.
Over the last decade, precision agriculture has grown in importance in order to meet the increasing food demand and ensure sustainability of farming. Today, advances in the Internet of Things (IoT) paradigm have promoted the use of Wireless Sensor Networks (WSN) for precision farming. However, recent technological developments suggest that use of Nanotechnology has immense potential to further improve the farming productivity. In this paper, we present some use-cases for the application of Internet of Nano Things (IoNT) in dairy farming. Although the use of IoNT involves several challenges, we envisage a multitude of benefits associated with its implementation.

1. INTRODUCTION

Precision agriculture is a recent trend that applies Information and Communication Technologies (ICT) in farming practices with a view to improve crop yield and ensure sustainable growth. Due to recent advances in the Internet of Things (IoT) paradigm along with mounting population, market pressures and growing environmental concerns, the use of Wireless Sensor Networks (WSN) has been proposed for precision farming. A recent survey conducted in the Netherlands by Steeneveld and Hogeveen [7] reveals that almost two fifths of the farms surveyed have adopted some sensor-based farm monitoring. Rutten et al. have reviewed various sensor systems recently proposed for animal health management in dairy farming [6]. These sensor systems can be used for monitoring animal fertility, mobility, metabolism and Mastitis detection. In a dairy farming scenario, however, fine-grained monitoring may be required since most of the crop and animal conditions ensue at molecular level. This necessitates data collection at the nano-scale to achieve significant improvements in the farm productivity.

Concurrent advances in Nanotechnology and IoT have stimulated the evolution of a new networking paradigm, Internet of Nano Things (IoNT). Developing suitable IoNT applications for dairy farming may be beneficial to monitor activities at the nano-meter range. However, real-world implementation of any IoNT solution requires addressing a number of significant challenges as discussed in [1] and [2]. In this paper, we consider the use of the IoNT paradigm in the scope of dairy farming.

The remainder of the paper is organized in the following order. In Section 2, we briefly discuss some of the use-cases for an IoNT solution in dairy farming. The challenges and benefits associated with the implementation are discussed in Section 3. We present the conclusions and future work in Section 4.

NANOCOM' 15, September 21 - 22, 2015, Boston, MA, USA
 ISBN 978-1-4503-3674-1/15/09 ...$15.00
DOI: http://dx.doi.org/10.1145/2800795.2800830

2. DAIRY FARMING SCENARIO

The use of IoNT has the potential to usher the development of several precision farming applications. Accordingly, design of nano-devices (e.g nano-sensors and nano-actuators) has been proposed in order to monitor environment variables, soil fertility, crop growth and animal health at the nano-scale. We consider the implementation of Wireless Nano Sensor Networks (WNSN) in dairy farming and discuss some of the key use-cases below.

1. ***Grass monitoring***: P. Creighton et al. conducted a survey to understand the various grassland management practices that have been employed for dairy farming [4]. Their study advocates the use of ICT, although at macro-scale, to enhance the farm productivity. The suitability of using Nanotechnology solutions for agriculture has been discussed [3]. IoNT may have the ability to realize some of these solutions. Nano-devices may be used to sense, compute and communicate precise real-time farm data. For instance, nano-devices may be used to closely monitor the dynamics between plant cell organelles and pathogens to enable early prediction and prevention of diseases. Nano-sensors may be used to monitor the soil fertility while nano-scale actuators may be designed to allow controlled delivery of fertilizers to the soil, thereby, preventing soil depletion. Dynamic irrigation can be practised by observing the soil moisture levels in real-time, to avoid water-logging and wastage (see page 586 in [3]).

2. ***Animal health and feed management***: Since running laboratory tests for disease diagnostics can be expensive and cumbersome, alternate techniques for easy disease detection are desirable. For instance, design of a breath sampling device which can be fitted to the nostrils of cattle has been proposed for disease detection [8]. However, owing to the ability to monitor molecular processes, use of nano-sensors for disease prediction is preferable. Nano-scale drug delivery systems shall enable controlled release of medication to specific sites in order to alleviate the spread of diseases. Nano-sensors to monitor hormone levels in the cattle, for improved fertility, may also be designed. Additionally, nano-scale carriers may be used to improve the nutrient profile of the feedstock. Feeding efficiency may be further enhanced by adding nano-scale digestive-aids.

3. Monitoring field conditions: Climate change adversely affects the food security and may cause pest and disease invasion [5]. Weather monitoring and forecasting using nano-devices may be beneficial to alleviate these harmful effects. Moreover, nano-scale carriers may be used for drug delivery to ensure timely prevention of pest attacks and diseases. Nano-devices may also be used to monitor air and water quality in dairy farms. Contaminants can be detected at parts per billion and disinfectants may be discharged using nano-actuators for purification. Real-time monitoring using nano-sensors may also assist in controlling weeds.

4. Reducing resistance to antibiotics: Excessive use of fertilizers and drugs can result into tissues developing resistance against them. The slow and controlled release of chemicals and medication, facilitated by the use of nano-carriers, may improve the ability of tissues to absorb them for efficient use. Reduced use of chemicals also decreases the input costs and minimizes wastage.

3. BENEFITS AND CHALLENGES

The use of WNSN in dairy farming promises a multitude of benefits, some of which are discussed below.

1. Agricultural sustainability: Nano-devices enable fine-grained control over the farm processes. Real-time monitoring of soil condition can help replenish soil nutrients, ensuring good quality grass. Early diagnosis and treatment of plant viruses and animal diseases is beneficial for their welfare. These applications can, in turn, improve the quality of the derived foods such as milk and eggs.

2. Self-powered nano-devices: Energy harvesting using background flows or bioconversion of animal wastes appears to be the most feasible solution for powering the nano-devices. This promotes the idea of green computing.

3. Network reliability: Due to molecular dimensions, a large number of nano-devices may be deployed to form a WNSN in dairy farms. This improves the reliability of data transmissions in-spite of the possible node failures due to molecular absorption or interactions with farm environment. Meanwhile, we envisage several challenges in the implementation of WNSN in dairy farms, as discussed below.

1. Design of nano-devices: Nano-sensors,nano-carriers, nano-actuators and nano-chemicals are still at the research and developmental stage. A number of design questions such as composition of nano-devices, type of sensors for dairy farming, computational capability of processors, data security and farm friendliness of devices are yet to be resolved.

2. Communication: Nano-devices will be used for monitoring activities at a nano-meter range. However, in order to monitor farm conditions at micro or macro scale, collaboration amongst a large number of nano-devices is necessary. Molecular and electromagnetic networking protocols for intra WNSN communication are, therefore, required. These protocols must address the issues concerning the battery-constrained low power and lossy nano-networks.

3. Interfacing WNSN and the Internet: WNSN in the farms must be connected to the Internet in order to exchange data with the cloud. Designing hardware and middle-ware for connecting WNSN to the existing WSN deployments in the farm may be beneficial in this regard. Moreover, development and use of communication paradigms to transfer data from WNSN to the Internet via WSN may help improve the transmission reliability of the data and control information between WNSN and the Internet.

4. CONCLUSIONS AND FUTURE WORK

Precision farming is the key for ensuring agriculture sustainability. Growth in IoT paradigm has promoted the idea of using WSN for efficient farm monitoring. However, Nanotechnology solutions have the potential to enhance the productivity of future agriculture. In this paper, we have discussed the application of IoNT in dairy farming. Currently, we are working towards designing communication techniques to optimize data transfer from various on-farm WSN systems to the cloud. Although several design and communication challenges are yet to be addressed, implementation of WNSN may help realize the dream of precision dairy farming.

5. ACKNOWLEDGMENTS

This work has received support from the Science Foundation Ireland (SFI) and the Agriculture and Food Development Authority, Ireland (TEAGASC) as part of the SFI TEAGASC Future Agri-Food Partnership, in a project titled "Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems".

6. REFERENCES

[1] I. F. Akyildiz and J. M. Jornet. The internet of nano-things. *IEEE Wireless Communications*, 17(6):58–63, December 2010.

[2] S. Balasubramaniam and S. Kangasharju. Realizing the internet of nano things: Challenges, solutions and applications. *Computer*, 46(2):62–68, February 2013.

[3] H. Chen and R. Yada. Nanotechnologies in agriculture:new tools for sustainable development. *Trends in Food Science and Technology*, 22(11):585–594, November 2011.

[4] P. Creighton, E. Kennedy, L. Shalloo, T. M. Boland, and M. O. Donovan. A survey analysis of grassland dairy farming in ireland, investigating grassland management, technology adoption and sward renewal. *Grass and Forage Science*, 66(2):251–264, June 2011.

[5] J. J. Lennon. Potential impacts of climate change on agriculture and food safety within the island of ireland. *Trends in Food Science and Technology*, 2014.

[6] C. J. Rutten, A. G. Velthuis, W. Steeneveld, and H. Hogeveen. Invited review: Sensors to support health management on dairy farms. *Journal of Dairy Science*, 96(4):1928–1952, April 2013.

[7] W. Steeneveld and H. Hogeveen. Characterization of dutch dairy farms using sensor systems for cow management. *Journal of Dairy Science*, 98(1):709–717, January 2015.

[8] C. Turner, H. Knobloch, J. Richards, P. Richards, T. Mottram, D. Marlin, and M. Chambers. Development of a device for sampling cattle breath. *Biosystems Engineering*, 112(2):75–81, June 2012.

Appendix C

Using Edge Analytics to Improve Data Collection in Precision Dairy Farming

Conference Title:	Eleventh IEEE International Workshop on Practical Issues in Building Sensor Network Applications (IEEE SenseApp 2016)
Article Type	Regular
Complete Author List	Kriti Bhargava, Stepan Ivanov, William Donnelly and Chamil Kulatunga
Status	Published: Nov. 2016

2016 IEEE 41st Conference on Local Computer Networks Workshops

Using Edge Analytics to Improve Data Collection in Precision Dairy Farming

Kriti Bhargava, Stepan Ivanov, William Donnelly, Chamil Kulatunga
Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland
Email: (kbhargava, sivanov)@tssg.org, wdonnelly@wit.ie, ckulatunga@tssg.org

***Abstract*—Despite the numerous advantages of using Wireless Sensor Networks (WSN) in precision farming, the lack of infrastructure in the remote farm locations as well as the constraints of WSN devices have limited its role, to date. In this paper, we present the design and implementation of our WSN based prototype system for intelligent data collection in the context of precision dairy farming. Due to the poor Internet connectivity in a typical farm environment, we adopt the delay-tolerant networking paradigm. However, the data collection capability of our system is restricted by the memory constraints of the constituent WSN devices. To address this issue, we propose the use of Edge Mining, a novel fog computing technique, to compress farming data within the WSN. Opposed to the conventional data compression techniques, Edge Mining not only optimizes memory usage of the sensor device, but also builds a foundation for future real-time responsiveness of the prototype system. In particular, we use L-SIP, one of the Edge Mining techniques that provides real-time event-driven feedbacks while allowing accurate reconstruction of the original sensor data, for our data compression tasks. We evaluate the performance of L-SIP in terms of Root Mean Square Error (RMSE) and memory gain using R analysis.**

I. Introduction

Over the last decade, the use of Wireless Sensor Networks (WSN) in precision farming has been widely advocated in order to improve the agricultural productivity and sustainability. WSN facilitate collection of farm data, using battery-powered sensors, which is, in turn, used for better monitoring and understanding of the farm processes such as weather changes, soil composition and dynamics, crop growth, and animal health and mobility patterns. A review of WSN applications in precision farming has been presented in [1]. In spite of the numerous advantages, however, very few WSN based systems have been put into practice, to date. This is primarily due to the constrained nature of the WSN devices along with the lack of infrastructure in a typically remote farm environment.

In this paper, we address some of the practical issues related to the deployment of WSN in the context of precision dairy farming. We present our WSN based prototype system for data collection in a dairy farm. Due to the intermittent or no Internet connectivity over the large area of farms, the data collected using the in-field sensors cannot be transmitted to the cloud storage in a timely manner. We, therefore, adopt the delay-tolerant networking paradigm for our system to facilitate reliable data transfer to the cloud. We discuss the design of our sensor node, referred to as the collar device, that is used to implement the delay-tolerant communication and is so-named as it will be worn around the neck by dairy cows. The collar device is tailored to ensure animal welfare and comprises of a variety of sensors to monitor cow health, activity and location. The device also acts as a mobile node that collects data from the different in-field sensors (e.g. grass monitoring) as the cow moves across the farm. All data is stored locally on the collar device itself until the cow is in the vicinity of the cloud gateway, presumably housed in a milking station, and offloads data onto it.

Given the wide variety of data that must be gathered periodically from the farm, a major challenge in implementing the delay-tolerant framework is the storage constraint of the collar device. Although sensor motes, today, feature a non-volatile flash memory, it is limited in capacity and is usually insufficient to store the large amounts of data that is gathered during the day. This, in turn, limits the data collection capability and the operational time of our prototype system. For instance, we collected temperature, humidity, acceleration, gyroscope, magnetometer and GPS (latitude, longitude and timestamp) data at a frequency of 1Hz and stored it on our collar device. The device could only gather data for a maximum of 4.5 hours before overwriting the least recent values in the flash. To address this limitation, we propose data compression on collar devices. We evaluate the feasibility of using Edge Mining, a novel fog computing approach, as opposed to the traditional compression techniques for reducing the memory requirements. Edge Mining algorithms are lightweight in nature and reduce the amount of data, rather than the size of each data entry, by storing only those values that cannot be predicted accurately using the past readings. Additionally, localised reduction of data builds the foundation for future real-time responsiveness of our system. This is key to the timely detection of critical events in precision farming. For instance, mobility pattern of cows must be monitored and analysed in real-time for virtual fence and feed management applications in order to facilitate corrective measures, if necessary, and redirect the cows in the desired way [2]. Moreover, real-time monitoring and evaluation of cow health is important for the early detection of diseases to alleviate the spread of any infection and ensure animal welfare.

In [3], authors implement Edge Mining using three instantiations of the Spanish Inquisition Protocol (SIP): Linear SIP (L-SIP), ClassAct and Bare Necessities (BN). SIP transforms

DOI 10.1109/LCNW.2016.9

raw data into an application-relevant state that is considered significant only if the data value cannot be predicted using the past estimates and an approximation model with the desired accuracy [4]. Accordingly, we propose the implementation of Edge Mining, using SIP, on the collar devices by storing only those states where the approximation error in data value exceeds a given threshold ε. We use the L-SIP algorithm since it reduces data on the device while preserving sufficient information to reconstruct the signal at the gateway, if needed. The performance of L-SIP for data compression is primarily governed by the user-specified ε values for each signal. A higher value of ε allows larger approximations in the estimated values, leading to higher values of memory gain as well as the Root Mean Square Error (RMSE). We evaluate the performance of L-SIP for the data collected using our collar device based on the above two metrics. We study the changes in quality of compression across different values of ε and variations in the signal using R analysis. L-SIP provides a significant memory gain of $\sim$70% for a given set of ε values in our scenario. Implementation of L-SIP, thus, not only improves data collection for delay-tolerant networking but also provides real-time event-driven feedbacks.

The remainder of the paper is organized as follows. In section II, we review some of the techniques for data compression and sensor analytics. We discuss the implementation of our testbed in section III. We evaluate the performance of L-SIP for data compression in section IV followed by the conclusions in section V.

II. Related Work

In this section, we present some of the existing approaches for data analysis in WSN. Since we are primarily concerned with optimizing memory usage for sensor devices, we review the proposed data compression algorithms for WSN along with other sensor analytics and Edge Mining techniques that can be used for localised data reduction.

A. Data Compression

Data compression techniques aim at storing data using the minimum number of bits possible, without any significant loss in information. An extensive survey on the compression techniques for WSN has been presented in [5]. While distributed compression techniques such as Data Transform Coding (DTC), Data Source Coding (DSC), and Compressive Sensing (CS) are used in dense sensor networks, local compression approaches such as Two-Modal Transmission (TMT) scheme based on predictive coding, and Lightweight Temporal Compression (LTC) scheme have been proposed for sparse sensor networks. Another novel approach based on distributed and adaptive signal processing has been proposed in [6]. The approach exploits the existing correlations in sensor data by adopting the principles of DSC and reaches a maximum energy saving of 65%. While selecting the suitable compression algorithm for a given application, the different techniques are compared on the basis of their code size, net energy saving, and compression performance i.e. the compression ratio vs the information gain. Additionally, the accuracy of data required and the nature of the WSN are considered. However, since compression techniques only reduce the number of bits per data value, they do not provide any insights into the data in near real-time, thereby, introducing latency in event detection.

B. Sensor Analytics

Although several techniques have been implemented for cloud-based data mining, the existing approaches cannot be directly used for edge analytics owing to the computational constraints of the sensor devices. Certain light-weight algorithms have, therefore, been proposed to perform localized data analysis in WSN applications. Data Fusion is one of the most basic approach that performs data reduction in WSN by merging the redundant data that emerges from the neighbouring sensor nodes [7]. The study shows that Data Fusion can be used to improve the sensing coverage and, in turn, the monitoring of the field. However, Data Fusion algorithms are signal specific and do not cater well to systems with heterogeneous streams of data. Data reduction can also be achieved through the implementation of Artificial Neural Networks (ANN) on top of the existing hardware-software platform of WSN [8]. These techniques improve the network intelligence by performing classification, clustering and prediction tasks on the sensor devices. However, the network learning involved is compute-intensive and may significantly reduce the battery lifetime of motes.

C. Edge Mining

Edge Mining is a novel fog computing approach that aims at improving the energy efficiency of a device by reducing the number of packet transmissions to a remote sink node. For doing so, it performs localized data analysis through implementation of light-weight data mining algorithms on the sensor devices. Accordingly, in a delay-tolerant framework, Edge Mining can be used to optimize the storage requirements by reducing the number of readings that are stored on a device as opposed to the number of bits per value as in case of compression techniques. Furthermore, the localized data mining facilitates real-time detection of events, thereby, improving responsiveness of the system. Edge Mining has been implemented using three different instantiations of general SIP as shown in [3]. SIP encodes raw data into state estimates that are considered significant/eventful only if the new data value cannot be predicted using the past estimates and an approximation model with a desired accuracy [4]. That is, a state estimate must be stored only if the error in prediction exceeds a user specified threshold ε. The three Edge Mining techniques differ on the basis of encoding schemes used for state estimation and are described in the context of our delay-tolerant scenario as under.

1) Linear SIP (L-SIP): In L-SIP, the state vector is represented as point-in-time value and rate of change. A number of techniques such as Kalman Filter, Expoenentially Weighted Moving Average (EWMA) and Normalised Least Mean Squares (NLMS) can be used for state estimation. A change of state is considered eventful only if the difference in the calculated point-in-time value and the estimated value exceeds the threshold ε. L-SIP is data agnostic and provides a significant reduction in the memory usage while storing enough data to allow reconstruction of the signal, if required.

2) ClassAct: ClassAct is a decision tree based classification technique. Given the application knowledge, the new state estimates are represented as a probability distribution over a set of activities that form the tree. The distribution is simplified to index of the most likely activity and the state estimate is stored only if the calculated index differs from the predicted value. While the decision tree is built through network learning at the sink node, classification of data can be performed using only a few comparisons on the sensor devices. Although this technique provides greater reduction of raw data compared to L-SIP, it is a destructive approach since the original signal cannot be reproduced in future.

3) Bare Necessities (BN): BN further reduces the memory usage by storing only the summary of data over time. The state vector is represented as a distribution over non-overlapping bins. A new state is calculated by assigning the raw value to a bin and updating the distribution for each bin. If the distribution of any bin changes by more than a threshold, it is considered eventful and the updated state is stored at the sensor node. Unlike L-SIP, BN discards most of the raw data which, in turn, affects the quality of future cloud-based analysis.

III. Testbed Implementation

In this section, we present our WSN based prototype system that is used for delay-tolerant data collection for precision dairy farming. In a dairy farm environment, we envisage a WSN comprising of three kinds of sensor nodes: in-field sensor nodes, collar devices and gateway node as illustrated in figure 1. The in-field sensors are static nodes that are used to monitor farm conditions such as weather changes and grass growth. The collar device is worn by dairy cows and comprises of a number of sensors to monitor cow health and mobility. Additionally, it acts as a mobile data carrier that collects data from the in-field sensors as the cow moves across the farm, stores it locally on the device, and brings it back to the milking station that houses the cloud gateway. Data from the collar device is transmitted to the gateway via mote-to-mote communication and is further uploaded on the cloud using Raspberry Pi connected to the gateway mote. The raw data, thus, collected is used by farmers to gain further insights into the farm conditions and take remedial actions, if necessary. Moreover, this data can be used to identify correlations between different farm processes and, in turn, improve the overall productivity.

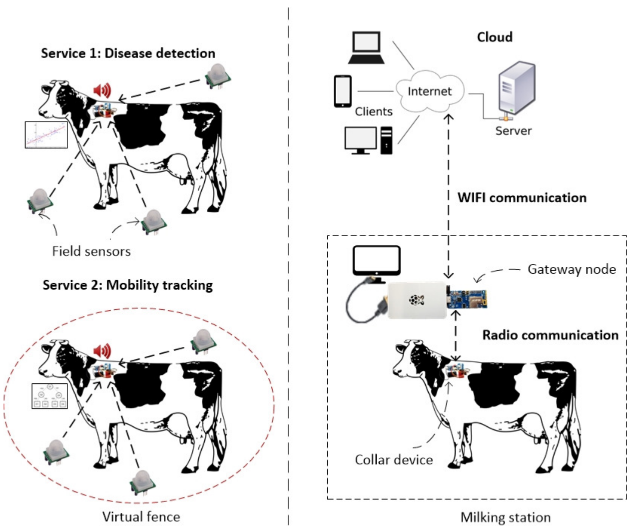


Fig. 1. Delay-tolerant networking framework for precision dairy farming

In this work, we implement a WSN testbed consisting of the collar device and gateway node and consider the memory collection capability of our system via data collection using device sensors. We present the design of our collar device and gateway node and address the challenges posed by the memory constraints of the device. We also review the L-SIP algorithm used for data compression.

A. Collar device

Collar device as shown in figure 2 forms the most integral part of our prototype system. The primary component of the collar device is the IEEE 802.15.4 compliant, low-power CM5000 mote that is based on the design of TelosB motes [9]. It consists of the MSP430F1611 processor and a CC2420 802.15.4, 2.4GHz wireless module for radio communication. The mote also comprises of an on-board SHT11 sensor to collect temperature and humidity readings, and supports three serial interfaces, namely UART, I^2C and SPI, to connect with external sensors. In order to facilitate mobility tracking for cows, we connect a 10 degrees of freedom (DOF) Inertial Measurement Unit (IMU) to the mote via the I^2C interface [10]. The IMU consists of three ICs, MPU6050, HMC5883L and BMP180, for measuring 3-axis acceleration and 3-axis orientation (gyroscope), 3-axis magnetic field, and barometric pressure respectively. The IMU features a user-programmable full scale range to ensure accurate tracking for both slow and

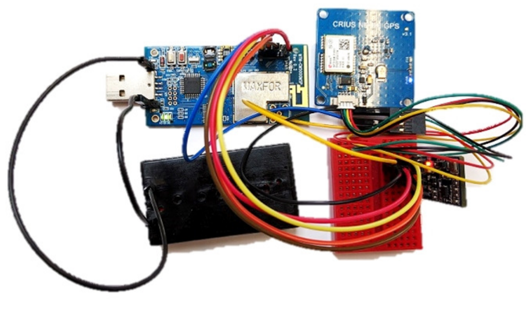


Fig. 2. Collar device comprising of CM5000 mote connected externally to a 10 DOF IMU and ublox NEO6 GPS receiver

fast motion [11], [12]. Data, thus, obtained can be used for feed management and detection of mobility-related diseases such as lameness [13]. Further, we connect a ublox NEO-6M Global Positioning System (GPS) receiver to our collar device via the UART interface [14]. The GPS unit enables context awareness for node localization in applications like virtual fences [2]. In our scenario, we primarily use GPS data to identify whether a cow is in the milking station or a dairy farm. Accordingly, we extract the values for latitude, longitude and time of position fix from the Geographic Position - Latitude/Longitude (GPGLL) factor of the National Marine Electronics Association (NMEA) stream.

Since we are in the development phase, the three components have been temporarily connected using breadboard, and jumpers and pin headers. The VCC of the external sensors is connected to VCC of the CM5000 mote which is itself powered using 2xAA batteries (3V). Although CM5000 is both TinyOS and ContikiOS compatible, we use TinyOS programming owing to its small footprint of 400 bytes [15] in the program memory. The programs are installed on the device using a USB interface and stored in a program memory of size 48KB. A 10KB RAM is available for storing the variable states along with an additional flash memory of 1MB that is used to store data. The non-volatile nature of the flash prevents loss of data owing to device failures. To examine the data collection capability of our device, and, in turn, the prototype system, we have designed a TinyOS application that runs on the collar devices for collection of temperature, humidity, acceleration, orientation, compass and GPS data at a given frequency from the device sensors. The gathered data is periodically pushed to the flash memory in fixed size heaps, using log appends, and stored locally for a specified period of time after which the device tries association with the gateway to offload its data. To establish connection with the gateway, the device temporarily joins the 802.15.4 Personal Area Network (PAN) of the gateway. Once the device is connected to the gateway, it sends its data packets over the radio until the flash is empty. Since we have implemented the IEEE 802.15.4 MAC and PHY layers, care must be taken that the payload size of each packet does not exceed the maximum transmission unit of 127 bytes. Once all packets have been transmitted from the device, it sends a disassociation request to the gateway requesting to leave the PAN.

B. Cloud gateway

The gateway node comprises of a CM5000 mote connected to Raspberry Pi (model B2) [16] via the USB interface as shown in figure 3. A TinyOS application runs on the CM5000 mote for data collection from the collar devices. At any given time, the gateway can connect to a predefined number of collar devices that is decided on the basis of the expected amount of data that must be transmitted by each device. If the node is currently connected to the predefined maximum number of collar devices, it does not confirm association and a random back-off mechanism is activated on the collar device to retry association. Otherwise, an acknowledgement is sent from the

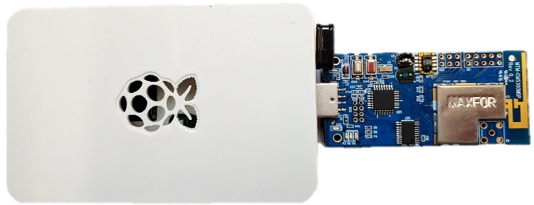


Fig. 3. Cloud gateway consisting of a CM5000 mote connected to a Raspberry Pi (model 2B)

CM5000 mote on the gateway to the device confirming its association. The node then starts listening to its radio for any incoming packets on the specified channel. These packets are transferred to the Raspberry Pi using the underlying UART interface. A JAVA application is built for Raspberry Pi, using TinyOS tools, to collect and store the incoming data. The data files, thus, generated are periodically pushed to Gitlab (cloud) using a WiFi module as shown in figure 1.

C. Data compression using L-SIP

Whereas the delay-tolerant approach provides a solution for transferring data from the sensor nodes in a remote farm environment to the cloud, the memory resources of the device pose a major constraint in its realization. Although we implement log storage using the device flash, the available memory is insufficient considering the vast amount of data collected during the day. For instance, at a sampling frequency of 1Hz, we could store the above mentioned values over a period of 4.5 hours only before the log storage was overwritten by the new values. Since we use GPS data to obtain only a broad idea of a cow's location, we reduce the sampling frequency of the GPS data to once per 15 minutes. This not only improves the data collection capability of our system but also increases the lifetime of our device since the energy cost for the ublox unit is quite significant compared to the other ICs. At a sampling rate of 1 second for the remaining sensors, this increased the operational time by two-fold. While reducing the sensing frequency for the other sensors is a plausible solution for reducing the data volume, it may cause loss of information.

Therefore, we propose localized compression of raw data on the collar devices in order to further optimize storage and improve the operational time of our system. The technique used should be data agnostic to accommodate the variety of farm data and must preserve the meaning of the signals after decompression. As mentioned before, we use Edge Mining rather than the conventional techniques for data compression since it not only reduces the data volume on the device but also builds the foundation for event-driven feedbacks for our prototype. While issues related to soil dynamics and weather changes may be treated at a later instance without any significant consequences, most processes related to grass

Algorithm 1 Linear SIP for improved data collection

procedure :

Calculate new state

  Using dEWMA filtering

    $v_{x,t} \leftarrow \alpha_x * z_{x,t} + (1-\alpha_x) * (v_{x,t'} + r_{x,t'} * (t-t'))$

    $r_{x,t} \leftarrow \beta_x * (v_{x,t} - v_{x,t'})/(t-t') + (1-\beta_x) * r_{x,t'}$

Estimate new state

  Using linear extrapolation

    $v'_{x,t} \leftarrow \begin{bmatrix} 1 & (t-t') \\ 0 & 1 \end{bmatrix} v_{x,t'}$

Eventful?

  yes, if $(|v'_{x,t} - v_{x,t}| > \varepsilon_x)$

    then, store $(v_{x,t}, r_{x,t}, t)$

TABLE I

CONFIGURATIONS USED FOR EVALUATION

ε	C1	C2	C3	C4	C5
ε_T	14*β_T	28*β_T	42*β_T	56*β_T	70*β_T
ε_H	6*β_H	12*β_H	18*β_H	24*β_H	30*β_H
ε_{Acc_x}	2*β_{Acc_x}	4*β_{Acc_x}	6*β_{Acc_x}	8*β_{Acc_x}	10*β_{Acc_x}
ε_{Acc_y}	1*β_{Acc_y}	2*β_{Acc_y}	3*β_{Acc_y}	4*β_{Acc_y}	5*β_{Acc_y}
ε_{Acc_z}	2*β_{Acc_z}	4*β_{Acc_z}	6*β_{Acc_z}	8*β_{Acc_z}	10*β_{Acc_z}
ε_{Gyro_x}	3*β_{Gyro_x}	4*β_{Gyro_x}	5*β_{Gyro_x}	6*β_{Gyro_x}	7*β_{Gyro_x}
ε_{Gyro_y}	0.2*β_{Gyro_y}	0.4*β_{Gyro_y}	0.6*β_{Gyro_y}	0.8*β_{Gyro_y}	1*β_{Gyro_y}
ε_{Gyro_z}	0.1*β_{Gyro_z}	0.2*β_{Gyro_z}	0.3*β_{Gyro_z}	0.4*β_{Gyro_z}	0.5*β_{Gyro_z}

management such as pest and disease attacks, and animal health and mobility issues demand real-time responsiveness. A real-time fog service based on Collaborative Edge Mining, an extension of the Edge Mining approach, in WSN for detection of Heat Stress in dairy cattle is presented by the authors in [17]. We adopt the L-SIP algorithm over ClassAct and Bare Necessities for data compression as it allows accurate reconstruction of the signal in future.

L-SIP is the linear instantiation of the SIP and encodes raw data as a state vector containing smoothed point-in-time value ($v_{x,t}$) and rate of change ($r_{x,t}$) at time t where x is the variable for which the state estimation is performed. We use the double EWMA (dEWMA) technique for state estimation due to its fast calculation and ease of implementation. dEWMA exponentially reduces the dependency of the current state on the past estimates by calculating the data value and rate of change as weighted averages of the current raw data value (z_t) and past estimates as shown in algorithm 1. Here, t' is the time associated with the previously stored state estimate, and α_x and β_x are the data and trend smoothing factors respectively and range between 0 and 1. Once the new state estimate is calculated by the device, the expected value at time t is calculated through the linear extrapolation of the previous state. If the difference in the calculated and predicted value is less than the given threshold ε_x for the variable, the new state is discarded by the device. Otherwise, the change is considered eventful and the new state vector is stored in the memory along with the corresponding timer value to allow future predictions. Resource efficiency is, thus, improved by reducing the number of state estimates stored. Moreover, the rate of change value improves the accuracy of the decompressed signal and prevents the propagation of error in case of packet loss past the subsequent packet during the reconstruction phase.

IV. EVALUATION

To evaluate the performance of L-SIP for data compression in our scenario, we gathered temperature, humidity and IMU data at a sampling rate of 1 second, and GPS data once per 15 minutes and stored it against the timer values for 5 hour intervals. While the application is proposed for farming practices, the data for this study was collected by us (human measurements) both inside and outside our laboratory. The data collection was repeated 8 times for different levels of activity (sit and walk) across 5 days. Since there was not much variation in the magnetometer and GPS readings, we base our analysis on 8 signal streams: temperature (°C), humidity (%RH), x,y and z-axis acceleration (g), and normalized x,y, and z-axis orientation/gyroscope (Least Significant Bit (LSB)). The α value for all variables is set to 0.94 following the best-fit approach. The β value for all datasets is calculated as the expectation value of the variable and represents the average of difference between any two consecutive readings. Since quality of compression varies with ε values, we evaluate L-SIP for different ε based on the following two metrics:

1) Root Mean Square Error (RMSE): Accuracy of the reconstructed signal with respect to the original signal is an important factor in evaluating the quality of compression. We calculate the RMSE for each variable by calculating the difference between the estimated and calculated data value at each instance. RMSE depends on the ε value for each variable. A higher threshold permits larger approximations in the estimated values, leading to higher values of RMSE. An upper bound on ε values must, therefore, be set to ensure that RMSE is within acceptable bounds. Conversely, we fix upper bounds on the RMSE values as shown below and calculate the corresponding upper bounds for thresholds.
   a) Temperature: 0.5°C
   b) Humidity: 0.5%RH
   c) Acceleration: 0.1g that corresponds to a positional inaccuracy of ∼1m
   d) Normalized Gyroscope: 0.05LSB that corresponds to an inaccuracy of approximately ∼5°/s
2) Memory gain (%): Edge Mining compresses data by storing only those state vectors where the data value changes significantly compared to the previous state estimate. Accordingly, we create a data frame to store only those instances where the difference between calculated smoothed data value and the estimated value exceeds ε. Each entry of the data frame requires 6 Bytes to store the current data value along with the corresponding timer and rate of change. Memory used in Byte units per

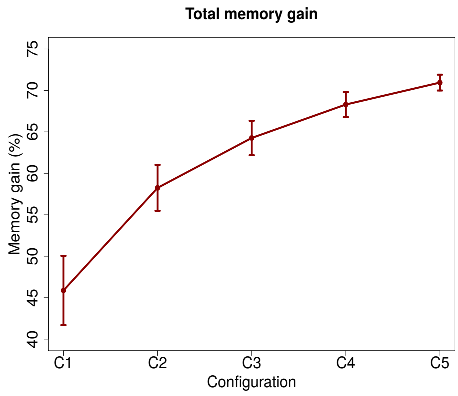


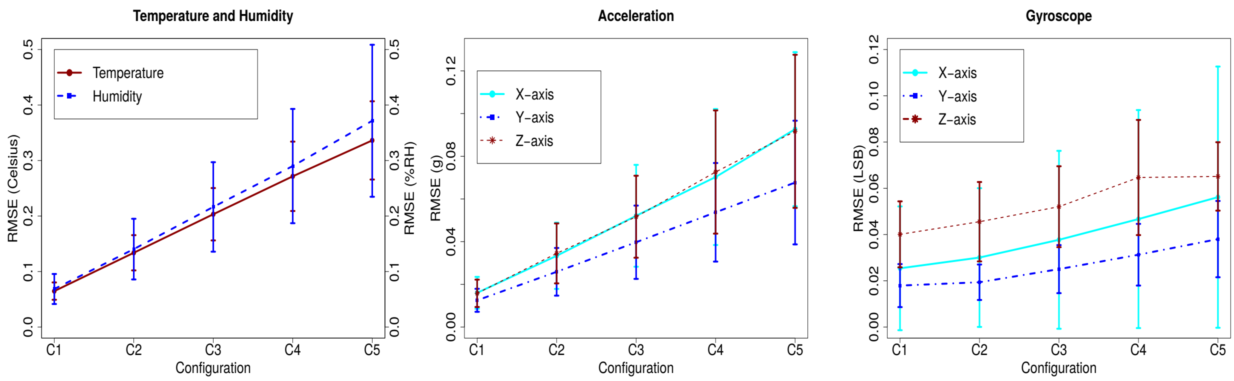
Fig. 4. Total memory gain averaged over 8 iterations for different configurations

variable is shown in eq. 1, where N' is the number of instances for which the error in approximation exceeds ε. Total memory gain (%) across all 8 signals and timer values can then be calculated as shown below, where N is the total number of readings collected.

$$\begin{aligned} MemUsed &= N' \cdot 6 \\ MemTotal &= \sum_{n=1}^{8} MemUsed_n \\ MemGain &= \frac{(N \cdot 9 \cdot 2 - MemTotal) \cdot 100}{N \cdot 9 \cdot 2} \end{aligned} \tag{1}$$

We calculate the RMSE and the memory gain for 5 different configurations as shown in Table I. We assign ε as a multiple of β value from the first dataset such that the largest ε for each variable, as shown in C5, corresponds to the upper bounds in the RMSE. The remaining ε are calculated as evenly spaced values between 0 and the upper bounds (C5) in order to study the compression quality at both comparatively small and large thresholds. In figure 4, we illustrate the memory gain averaged over the 8 iterations across the different configurations along with the respective confidence intervals at a confidence level of 95%. A large ε permits larger approximations from the original signal, resulting in fewer entries in the data frame and, in turn, an increase in memory gain. For the given thresholds, we achieve close to 47% reduction in the memory requirements for smallest set of ε values. At higher thresholds, the memory gain is as high as 70% and would considerably improve the operational time of our system. Although we increase the ε values in a fixed proportion, memory gain from C1 to C5 does not increase by a fixed percentage. The different growth rate of memory gain between C1 and C5 is attributed to the small changes in actual ε values for some variables. For instance, the increase in ε_{Acc_i} and ε_{Gyro_i}, where i can be x,y and z, values between any two consecutive configurations is marginal for most datasets and does not cause significant reduction in N' and, in turn, the memory gain. The average value of RMSE over 8 iterations corresponding to the above memory gains is shown in figure 5 for all data streams. We calculate the confidence intervals at a level of 95%. As discussed above, RMSE rises with an increase in the threshold value. However, L-SIP ensures that RMSE stays within acceptable bounds by storing the calculated data value each time the approximation error crosses the threshold. Moreover, the rate of change value prevents the indefinite propagation of reconstruction error due to packet loss, thereby, ensuring small values of RMSE during the decompression phase. Similar to memory gain, the RMSE for each signal increases at different rates across different configurations owing to the marginal changes in absolute values of ε.

Further, we analyse the changes in the quality of compression with changes in the distribution of values, i.e. the change in variation in raw signal. We study the compression for the first dataset with respect to 3 variables: temperature, x-axis acceleration and normalized x-axis gyroscope using ε values from C3. Figures 6a, 6b and 6c show the reconstructed signal



(a) (b) (c)

Fig. 5. RMSE for all signals averaged over 8 iterations for different configurations

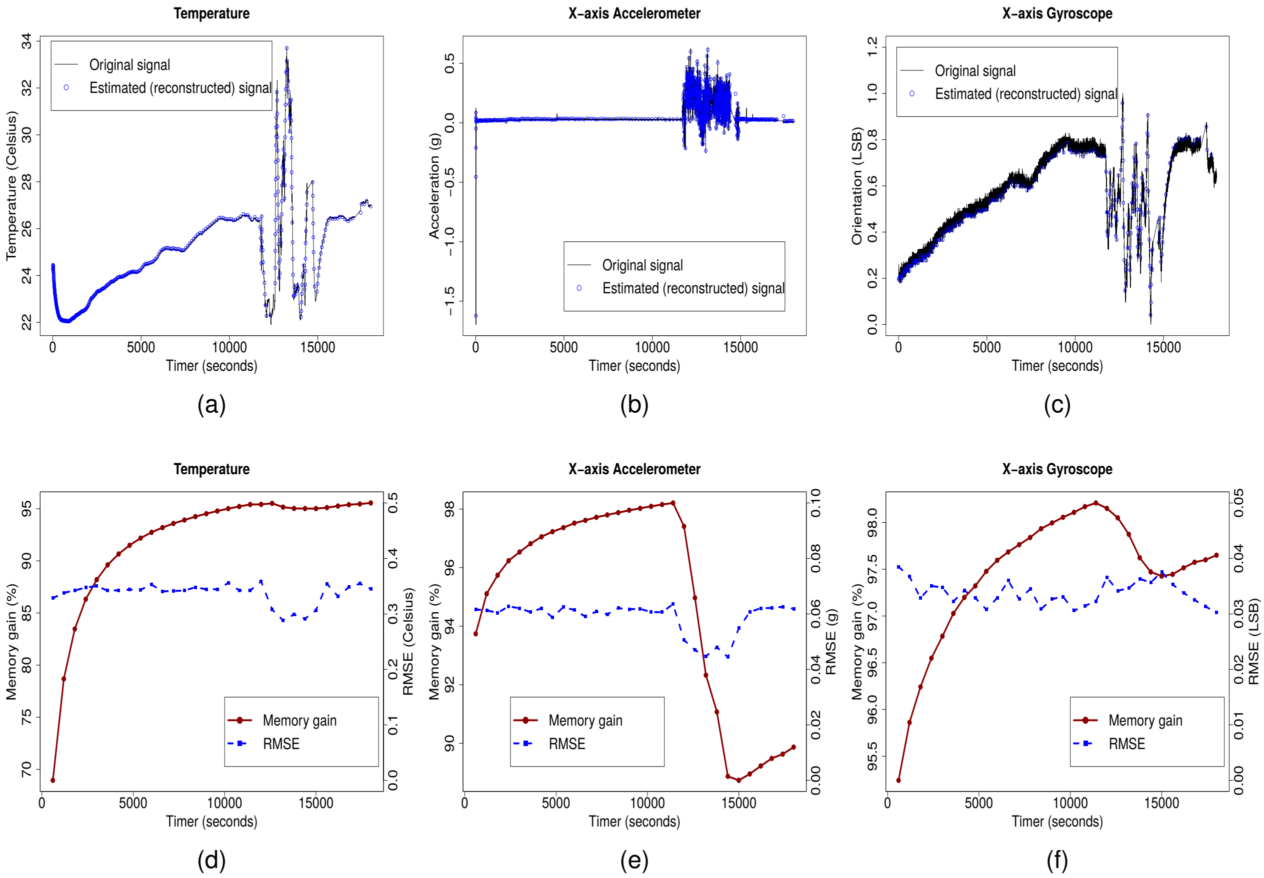


Fig. 6. Estimated signals superimposed onto the original signals along with the cumulative memory gain and RMSE for ε values from C3

superimposed on the original signal over the 5 hour period. As is evident, L-SIP reconstructs an accurate signal while giving an overall memory gain around 60-65% (figure 4). Next, we calculate the cumulative gain in memory corresponding to each variable. We assume that our system stores the timer and data values for only one variable at a time and calculate the memory used from start to time t in 10-minute windows. The individual memory gains calculated for temperature, acceleration and gyroscope data are as high as 95%, 98% and 99% respectively as shown in figures 6d, 6e and 6f. While the value increases for small variations with the time elapsed, a drop in memory gain is observed corresponding to the larger fluctuations owing to the more frequent entries in data frames. The drop in value is more visible for accelerometer compared to gyroscope, and is very slight in case of temperature. This is because the value of ε_T is much larger compared to ε_{Acc_x} and ε_{Gyro_x} and, therefore, accommodates larger approximations in the signal with minor changes in the value of N'. We also calculate the average values of RMSE over 10-minute windows for the 5 hour period in order to understand the changes in error with different variations in the signal. While RMSE is stable for small changes in the data value, a drop in RMSE coincides with the drop in memory gain. This is because the more frequent entries in the data frame accurately capture the nature of variation, thereby, avoiding large approximation errors. The average RMSE in the three signals remains below the allowed maximum at all times.

As shown above, L-SIP gives a significant increase in memory gain with relatively small values of RMSE for different configurations of ε and different variations in the signal. The key challenge is to balance the trade-off between the memory gain and RMSE while meeting the application requirements and attaining reasonable information gain. Although, we use the same multiples of β for all iterations, the ϵ values for the same configurations differ between datasets due to change in β values in each iteration. As a result, the maximum RMSE obtained for variables is much less than the allowed maximum in some cases. Since ε values are user-programmable, the compression results can be improved by changing the ε between different iterations, depending on the user requirements, through cloud-based network learning. Further reduction in storage requirements can be achieved through compression of the key samples that are stored on the device after mining.

V. Conclusions

In this paper, we have addressed some practical issues concerning the implementation of WSN technology in the context of precision dairy farming. We present the design of our prototype system and collar device that is used for data collection in dairy farms. Due to the remote location of a typical farm, we implement the delay-tolerant framework for data communication where data is stored on the collar device itself until the cow is in vicinity of the cloud gateway. However, the data collection capability of our application is limited due to the memory constraints of the constituent devices. This, in turn, reduces the operational time of our WSN system. To address this issue, we propose the implementation of light-weight Edge Mining algorithms on our collar device to perform localized data compression. Edge Mining algorithms convert the raw data into state vectors and reduce memory usage by storing only those instances that cannot be predicted from the past estimates using a given approximation model. Compared to the traditional compression techniques, Edge Mining not only optimizes the storage requirements but also provides a foundation for future real-time responsiveness of the system. This is of utmost importance for detecting critical issues such as those related to animal health and mobility. We use the L-SIP algorithm over other Edge Mining techniques since L-SIP preserves sufficient information on the sensor device to allow reconstruction of original signal at the gateway. The performance of L-SIP for data compression is evaluated with respect to 8 signals namely temperature, humidity, x,y,z-axis acceleration, and x,y,z-axis gyroscope on the basis of RMSE and memory gain, using R analysis. With an upper bound on ε values corresponding to RMSE of 0.5°C, 0.5%RH, 0.1g and 5°/s for temperature, humidity, acceleration and orientation respectively, L-SIP provides an overall memory gain of $\sim$70%. This, in turn, would lead to a significant improvement in the operational time of our prototype system. Since the quality of compression varies with the user-programmable ε value, the information gain using L-SIP can be further improved, depending on the application requirements, using feedbacks from cloud-based network learning. The compression performance also changes with change in variation of the signal. Even though larger fluctuations in signal result in an increase in the number of readings that must be stored on the device, the cumulative memory gain calculated for individual variables was above 95% for most part of the experiment with RMSE below the allowed maximum at all times.

Acknowledgment

This work has received support from the Science Foundation Ireland (SFI) and the Agriculture and Food Development Authority, Ireland (TEAGASC) as part of the SFI TEAGASC Future Agri-Food Partnership, in a project (13/IA/1977) titled Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems.

References

[1] A. Rehmana, A.Z. Abbasib, N. Islamb and Z.A. Shaikhb, *A review of wireless sensors and networks' applications in agriculture*, Journal on Computer Standards & Interfaces, Elsevier, vol. 36, pp. 263-270, issue 2, Feb. 2014.

[2] T. Wark, D. Swain, C. Crossman, P. Valencia, G. Bishop-Hurley and R. Handcock, *Sensor and Actuator Networks: Protecting Environmentally Sensitive Areas*, IEEE Pervasive Computing, vol. 8, no. 1, pp. 30-36, Jan-Mar 2009.

[3] E.I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith and R. Rednic, *Edge Mining the Internet of Things*, IEEE Sensors Journal, vol. 13, no .10, pp. 3816-3825, Oct. 2013.

[4] D. Goldsmith and J. Brusey, *The spanish inquisition protocol: Model based transmission reduction for wireless sensor networks*, Proceedings of IEEE Sensors 2010, pp. 2043-2048, Nov. 2010.

[5] M. A. Razzaque, C. Bleakley and S. Dobson, *Compression in wireless sensor networks: A survey and comparative evaluation*, ACM Transactions on Sensor Networks (TOSN), v.10 n.1, p.1-44, November 2013.

[6] J. Chou, D. Petrovic and K. Ramachandran, *A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks*, Twenty-Second Annual Joint Conference of the IEEE Computer and Communications INFOCOM 2003, IEEE Societies, San Francisco, CA, vol. 2, pp. 1054-1062, 2003.

[7] T. Rui, X. Guoliang, L. Benyuan, W. Jianping and J. Xiaohua, *Exploiting Data Fusion to Improve the Coverage of Wireless Sensor Networks*, IEEE/ACM Transactions on Networking, vol. 20, no. 2, pp. 450-462, Apr. 2012.

[8] G. Serpen, J. Li, L. Liu and Z. Gao, *WSN-ANN: Parallel and Distributed Neurocomputing with Wireless Sensor Networks*, The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, Aug. 2013.

[9] Advanticsys, *MTM-CM5000-MSP*, Available at http://www.advanticsys.com/shop/mtmcm5000msp-p-14.html, Last accessed in May 2016.

[10] Waveshare Electronics, *10 DOF IMU Sensor User Manual*, v1.2, pp. 1-6, Mar. 2015.

[11] Invensense, *MPU-9255 Product Specification*, InvenSense Inc., revision 1.0, Sep. 2014.

[12] Invensense, *MPU-9255 Register Map and Descriptions*, InvenSense Inc., revision 1.0, Oct. 2014.

[13] A. Poursaberi, C. Bahr, A. Pluk, D. Berckmans, I. Veerme, E. Kokin and V. Pokalainen, *Online lameness detection in dairy cattle using Body Movement Pattern (BMP)*, 11th International Conference on Intelligent Systems Design and Applications (ISDA), Cordoba, pp. 732-736, 2011.

[14] U-blox, *NEO-6 - Data Sheet*, www.u-blox.com, GPS.G6-HW-09005-E, Last accessed in Mar. 2016.

[15] TinyOS Documentation Wiki, Available at http://tinyos.stanford.edu/tinyos-wiki/index.php/, Last accessed in Nov. 2015.

[16] Raspberry Pi Foundation, *RASPBERRY PI 2 MODEL B*, www.raspberrypi.org, Last accessed in May 2016.

[17] K. Bhargava and S. Ivanov, "Collaborative Edge Mining for predicting heat stress in dairy cattle," Wireless Days 2016, Toulouse, France, pp. 1-6, Mar. 2016.

Appendix D

Collaborative Edge Mining for Predicting Heat Stress in Dairy Cattle

Conference Title:	Eighth Wireless Days Conference 2016
Article Type	Regular Paper
Complete Author List	Kriti Bhargava and Stepan Ivanov
Status	Published: Mar. 2016

Collaborative Edge Mining for Predicting Heat Stress in Dairy Cattle

Kriti Bhargava, Stepan Ivanov
Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland
Email: (kbhargava, sivanov)@tssg.org

***Abstract*—Edge Mining (EM), a novel Fog Computing technique, has been proposed to perform data analysis on sensor devices at the edge of Internet of Things (IoT). The approach, however, is limited to analysis conducted by each sensor node in isolation. In this paper, we propose Collaborative Edge Mining (CEM), an extension of the EM technique, wherein multiple sensor devices participate together in on-site data analysis and prediction. Our model detects contextually relevant events by integrating and analysing data arising from different sources and, thereby, lays the foundation of a sensor-based implementation of Apache Storm like framework. We have evaluated our approach with respect to the Linear Spanish Inquisition Protocol for a precision farming application. We illustrate CEM for the estimation of Temperature Humidity Index, an important metric to predict Heat Stress in dairy cattle, and compare its performance to EM. CEM performs well in most cases, especially, latency-sensitive scenarios.**

***Index Terms*—wireless sensor networks, edge mining, apache storm, precision farming, heat stress.**

I. Introduction

In the recent years, there has been widespread adoption of the Internet of Things (IoT) paradigm owing to its ability to provide localization and context awareness. Wireless Sensor Networks (WSN) technology has been identified as one of the key enablers in realizing the potential of IoT. With advances in the IoT technology, there has been a constant improvement in the design of these devices. Sensor nodes, today, have become quite powerful due to tremendous growth in their computational and storage capabilities. Originally limited to data collection and packet forwarding, the tasks assigned to these devices have become more diverse and computationally complex. Moreover, sensor nodes have not only become energy efficient but are also capable of harvesting energy.

Use of WSN has been proposed for precision farming with a view to improve the agriculture efficiency and sustainability. One of the significant sectors in agriculture is the pasture-based dairy industry. A Dairy Farm Management System (DFMS) requires a wide variety of sensors to collect environmental data such as temperature and humidity, soil management data such as chemical composition and moisture content, and livestock related data for managing animal health, milk yield, feed quality and mobility tracking. Although some of the DFM applications are low pace, on-site prediction of latency-sensitive phenomenons, for instance, animal health related issues is of significant importance. This would not only minimize animal health hazards but also the economic losses, otherwise incurred. However, a recent survey conducted to review the sensor systems available for dairy health management suggests a significant lack of analytics in the available systems [1]. Currently deployed sensor devices are only used to sense and send data to the sink which is then subjected to cloud-based analysis. This approach is largely dependent on the Internet connectivity which is often poor in dairy farms and could cause delay in disease detection.

A new networking paradigm called Fog Computing has been introduced in [2] to cater to the needs of latency-sensitive applications by providing compute, storage and networking resources between the network edges and Cloud Computing data centres. The benefits of Fog Computing for IoT applications have been discussed in [3]. Furthermore, E.I. Gaura et al. propose Edge Mining (EM), a novel Fog Computing technique, where data analysis takes place at the network edges itself i.e. at the wireless, battery-constrained sensor devices [4]. The idea behind EM is to localize data analysis and improve energy efficiency of the network by reducing the number of transmissions to the sink. The paper presents a generalized form of the Spanish Inquisition Protocol (SIP) algorithm, general SIP (G-SIP), and examines the benefits of EM with respect to three instantiations of G-SIP: Linear SIP (L-SIP), ClassAct and Bare Necessities (BN). The algorithms are lightweight and can be easily implemented in WSN. However, the approach is limited to analysis conducted by each sensor node in isolation and may not be suitable for real-time applications where interaction between various nodes is necessary for event detection.

In this paper, we propose Collaborative Edge Mining (CEM), a novel data analysis approach for IoT. CEM facilitates parallel and distributed processing for on-site prediction of latency-sensitive phenomenons that require cooperation between sensors. The technique bases on EM and adapts the main principles of operation from the Apache Storm framework [5]. A Storm cluster consists of Spouts, data sources, and Bolts, data processors, that follow a Storm topology (application logic) to facilitate parallel and distributed analysis for real-time applications. Likewise, CEM allows complex data analysis within WSN by distributing a subset of application logic or an EM task to each sensor and integrating the results for further analysis and event detection. Distribution of tasks between sensor devices ensures efficient use of the available in-network compute resources. Moreover, the lightweight na-

978-1-5090-2494-0

ture of the EM tasks prevents overburdening of the sensor nodes. We illustrate our CEM model with respect to L-SIP for a precision farming application. Heat Stress is a major condition in dairy cattle that causes adverse effects on the health and productivity of the animal. The severity of Heat Stress is estimated using the Temperature Humidity Index (THI) which can be calculated using CEM on temperature and humidity sensors on farm. We compare the performance of CEM and EM for L-SIP using R programming. CEM provides localisation of analysis and performs well in most scenarios.

The remainder of the paper is organized in the following order. In Section 2, we discuss the related work. Our proposed solution is presented in Section 3. We evaluate and compare our approach to EM with respect to L-SIP in Section 4 followed by the conclusions in Section 5.

II. Related Work

Prior to discussing the CEM technique, we overview some of the existent mechanisms for sensor analytics. We also discuss the significance of Heat Stress in dairy cows.

A. Sensor Analytics

In the past few years, there has been a dramatic improvement in the complexity of tasks that can be performed by the sensor devices. Data analytics techniques such as Data Fusion, Edge Mining and Artificial Neural Networks, as described below, have already been implemented in WSN. The benefits and challenges associated with the implementation of sensor analytics for precision farming have been discussed in [6].

1) Data Fusion: Data Fusion is one of the most basic analytics technique that has been implemented in WSN. It minimizes data redundancy by merging the overlapping data generated by different sensors and, in turn, reduces the network traffic [7]. Consequently, it improves the energy efficiency of the network and the quality of data exchange between the devices. Additionally, Data Fusion can be used to perform analytical tasks such as outlier detection and target tracking. However, Data Fusion algorithms are signal specific and cannot be easily extended to new applications.

2) Edge Mining: EM techniques improve network intelligence by converting raw sensor data into contextually relevant information (state) within the WSN. These state estimates are stored locally on the sensor devices and are forwarded to the sink only at the occurrence of unexpected events. In [4], the benefits of EM are examined based on three algorithms viz. L-SIP, ClassAct and Bare Necessities. Although EM improves the network efficiency, it assumes WSN as a network of individual smart-sensing devices that perform mining tasks in isolation. This limits its use where collaboration between sensor nodes is necessary for detecting events.

3) Artificial Neural Networks (ANN): A certain resemblance in the operation of an ANN and WSN has been recently identified in [8]. The paper proposes use of the already existing WSN technology as the hardware-software platform for implementation of ANN algorithms to perform prediction, classification and clustering tasks. As an example, a Hopfield Neural Network has been mapped to a WSN to solve the problem of minimum weakly connected dominating set. The approach is generic and can potentially be extended for other analytical tasks. However, identification of the correct ANN topology and network learning are computationally intensive tasks and may increase the complexity of the implementation.

B. THI

One of the potential applications of CEM is the estimation of THI for the timely prediction of Heat Stress in dairy cows. Dairy cows experience Heat Stress due to increase in temperature beyond a thermo-neutral range of 12-21°C and a simultaneous change in humidity levels. Heat Stress causes changes in their physiological status i.e. respiration rate, heart rate, metabolism and fertility rate. Moreover, severe Heat Stress conditions can cause huge economic losses of up to $698 per dairy cow per year [9].

THI combines the effects of ambient temperature and relative humidity to evaluate the severity of Heat Stress (eq. 1)

$$THI = 1.8 * Ta - (1 - RH) * (Ta - 14.3) + 32 \tag{1}$$

where Ta is the measured ambient temperature (°C) and RH is the relative humidity as a fraction of the unit [10]. Recent studies have shown that the incidence of Heat Stress in dairy cows occurs at a THI value of 68 which corresponds to 22°C at 45 % RH, 23°C at 35% RH and 24°C at 20% RH [11].

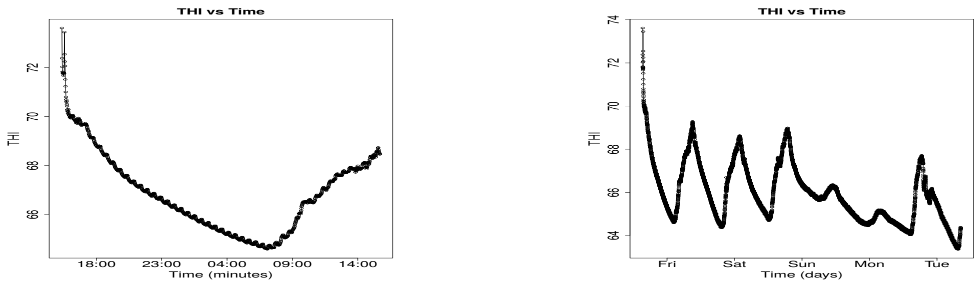
The severity of Heat Stress increases proportionally with the THI value and can ultimately lead to the death of an animal. This necessitates continuous monitoring of THI variable to maintain it within the acceptable bounds. Based on the THI value, farmers can use appropriate cooling techniques such as dietary management, water sprinklers on farm, and provision of ventilation, cooling and shade for the cows. A timely cure can significantly help in minimizing the adverse effects on animal health and economic losses incurred due to Heat Stress.

III. Proposed solution

As stated above, in this paper, we consider the implementation of CEM using L-SIP in the context of THI state estimation for predicting Heat Stress in dairy cows. Figures 1a and 1b illustrate the variation in THI, with changes in temperature and humidity, over time. THI values have been calculated using temperature and humidity data gathered continuously in one minute intervals. Time periods with THI greater than 68 signify Heat Stress conditions. In this section, we first describe EM with respect to L-SIP, and subsequently, discuss the implementation of CEM using L-SIP for THI prediction.

A. EM using L-SIP

EM has been proposed to improve energy efficiency of a network through local data processing that, in turn, reduces the network traffic. One of the techniques to realize EM in WSN is by using the SIP [12]. SIP reduces the energy cost of a network by sending only the unexpected information to the sink. Instead of sending raw data, SIP encodes the data into states using an approximate model of the phenomenon.



(a) Variation in THI values over 1 day

(b) Variation in THI values over 5 days

Fig. 1: Variation in THI over time

This model is shared between the sensor nodes and the sink to allow future state predictions. While a sensor node calculates the state vector periodically, it also keeps a track of the predicted value at sink using its local copy of the sink state and the approximation model. Only if the difference in the calculated and predicted value exceeds a permissible threshold ε, it transmits the new state vector to the sink. Thereafter, the sensor node updates its local copy of the sink state.

L-SIP is the linear model of SIP that encodes the state estimate at time t (s_t) using the point-in-time value v_t and rate of change r_t. The linear model contains sufficient information to allow future predictions and also reconstruction of the signal, if necessary. Furthermore, in case of packet loss, r_t prevents the propagation of reconstruction error beyond the subsequent packet. Although different techniques can be used for state estimation, the use of double Exponentially Weighted Moving Average (dEWMA) is preferred due to ease of implementation and good performance in WSN. Using dEWMA, v_t and r_t are calculated as the weighted average of the current observation and past estimates as shown in eq. 2 and 3.

$$v_t = \alpha * x_t + (1 - \alpha) * (v_{t-1} + r_{t-1} * \delta t) \tag{2}$$

$$r_t = \beta * (v_t - v_{t-1})/\delta t + (1 - \beta) * r_{t-1} \tag{3}$$

where x_t is the most recent data value and δt is the time difference between the current and previous observation. The coefficients α and β are the data and trend smoothing factors respectively such that $0 < \alpha, \beta < 1$.

Once the state estimate is calculated, the sensor node predicts the current value at sink through linear extrapolation of its local copy of the previous sink state. If the error in prediction exceeds a given threshold ε, a new packet containing (v_t, r_t) is sent to the sink. Upon receiving the acknowledgement, it updates its local copy of the sink state.

B. CEM using L-SIP for THI state estimation

EM techniques are limited to analysis conducted by each sensor independently. Hence, event detection for phenomenons that require data from multiple sensors can only be carried out later at the sink, thus, affecting the timeliness of predictions. We, therefore, propose CEM, an extension of the EM approach, where information from different sensors is gathered and further analysed on farm to enable application-specific event detection in latency-sensitive scenarios. The basis of our CEM model is the Apache Storm framework for parallel and distributed processing. The core abstraction in Storm is Stream, an unbounded sequence of tuples, that is sourced into the network by Spouts and processed by Bolts, that run the worker processes, according to the given Storm topology i.e. application logic. Similarly, CEM facilitates parallelization and distribution of data analysis tasks among sensor nodes in IoT. While the master node distributes the application logic and approximation model to the network, each sensor node first runs a basic EM algorithm to convert raw data into intermediate states. Using the application logic and the given model, a sensor node then integrates these states to derive a contextually relevant state. The first state estimate is transmitted to the sink and a copy of it is maintained at the sensor node for future predictions. At any given time t, if there is a significant update in the intermediate states, the sensor node recalculates the final state estimate. Moreover, it predicts the current sink value using the previous sink state and the given model. An event is detected only if the difference between the calculated and predicted value exceeds a threshold. Accordingly, a packet containing the updated state is sent to the sink. In the remainder of this section, we describe the CEM approach for our scenario.

Our WSN comprises of static temperature and humidity sensors, and mobile animal wearable devices (cow collars). Since THI is a cow-related phenomenon, we adopt a new computing framework called ‘cowputing’, where a cow is the center of all data analysis, for the evaluation of THI. The role of the cow collar is 2 fold: it acts as the master node and initiates THI calculation in WSN by sending the application logic (eq. 1) and approximate model for state estimation to the network, and a mobile sink node to collect THI updates from the sensor nodes. In this paper, we consider the implementation of CEM using L-SIP and dEWMA filtering. Each static temperature and humidity sensor synchronously runs L-SIP to convert its raw readings into state estimates per sensing cycle. If a static node, for instance temperature sensor, receives a request for THI value from the mobile cow collar, it gathers the current humidity state from a neighbouring sensor and computes THI using the temperature and humidity state estimates in eq. 1. Next, dEWMA filtering is used to smooth data values and calculate the rate of change. The first state estimate for THI is forwarded to the sink and a local copy of the sink state is maintained at the sensor node. While temperature and humidity states are computed periodically, THI state is recalculated only if an event is detected at either of the static sensors. Using the local copy of the previous sink state, temperature sensor predicts the current THI value at the sink. If the difference between the calculated and predicted value is significant, an event is detected and a new packet containing the updated THI state is transmitted to the sink. Upon receiving acknowledgement, the temperature sensor updates the local copy of sink state. Fig 2 illustrates this process.

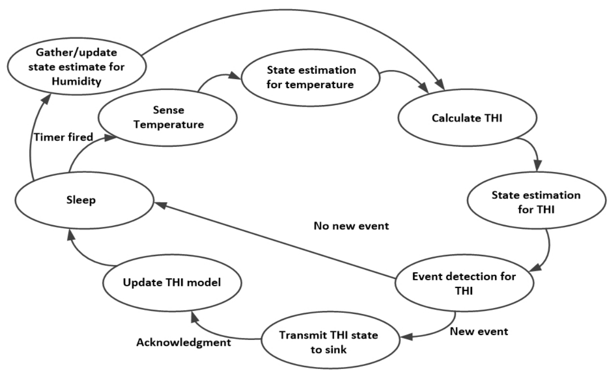


Fig. 2: Collaborative edge mining model for THI state estimation at a temperature sensor

Algorithm 1 summarises the above approach. The basic idea behind using L-SIP is that linear integration of two or more linear functions generates another linear function. Our approach is generic and can be used for diverse IoT applications where input variables follow a linear model. The state estimate for temperature ($s_{T,t}$) and humidity ($s_{H,t}$) is calculated per sensing cycle using the raw data values tmp_t, hum_t and coefficients α_T, β_T and α_H, β_H respectively. The two state estimates consist of the smoothed data value and rate of change pair $(v_{T,t}, r_{T,t})$ and $(v_{H,t}, r_{H,t})$ respectively where δt denotes the time difference between the current and previous state. If a temperature sensor receives a request from the mobile cow collar or an event is detected at the humidity sensor, it obtains $s_{H,t}$ from the nearest static sensor and calculates the THI value (thi_t) using eq.1. dEWMA filtering is used over thi_t to estimate the smoothed THI state $s_{THI,t}$ using the coefficients α_{THI} and β_{THI}. If this is the first THI estimate, it is transmitted to the sink and a copy is maintained at the temperature sensor. Otherwise, the local copy of previous sink state is linearly extrapolated to predict the current sink state vector $THI_{sink,t}$ ($v_{sink,t}, r_{sink,t}$). If the calculated and predicted value differs by more than the threshold ε_{THI}, an output tuple containing $(s_{THI,t}, n, t)$ is transmitted to the sink. The sequence number n is used to account for any loss in packets. Alternatively, if time since the last transmission exceeds a threshold $t_{heartbeat}$, a new packet

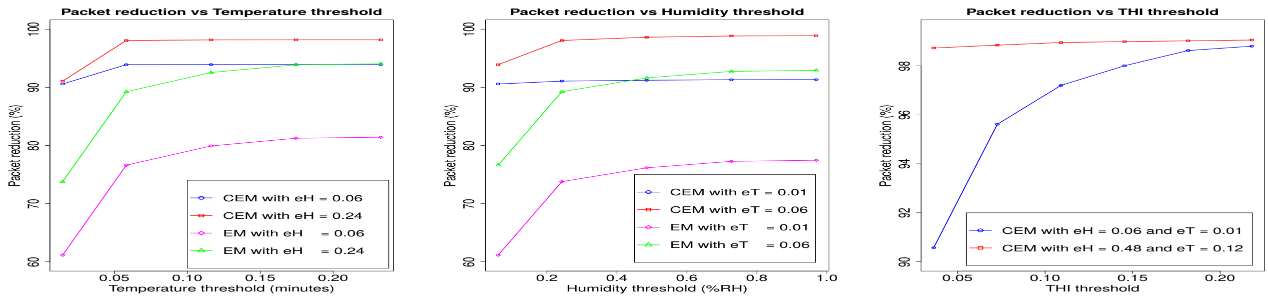
Algorithm 1 CEM using L-SIP for Heat Stress prediction

procedure :
$t \leftarrow$ current time
At static humidity node
  $hum_t \leftarrow$ obtain vector of sensor readings
  estimate new state for humidity - dEWMA filtering
  $v_{H,t} \leftarrow \alpha_H * hum_t + (1-\alpha_H) * (v_{H,t-1} + r_{H,t-1} * \delta t)$
  $r_{H,t} \leftarrow \beta_H * (v_{H,t} - v_{H,t-1})/\delta t + (1-\beta_H) * r_{H,t-1}$
At static temperature node
  $tmp_t \leftarrow$ obtain vector of sensor readings
  estimate new state for temperature - dEWMA filtering
  $v_{T,t} \leftarrow \alpha_T * tmp_t + (1-\alpha_T) * (v_{T,t-1} + r_{T,t-1} * \delta t)$
  $r_{T,t} \leftarrow \beta_T * (v_{T,t} - v_{T,t-1})/\delta t + (1-\beta_T) * r_{T,t-1}$
  if request for new THI state estimate received or
  temperature or humidity event occurred, then
    Obtain humidity state $(v_{H,t}, r_{H,t})$ and calculate
    $thi_t \leftarrow 1.8 * v_{T,t} - (1 - v_{H,t})(v_{T,t} - 14.3) + 32$
    estimate new state for THI - dEWMA filtering
    $v_{THI,t} \leftarrow \alpha_{THI} * thi_t + (1-\alpha_{THI}) * (v_{THI,t-1} + r_{THI,t-1} * \delta t)$
    $r_{THI,t} \leftarrow \beta_{THI} * (v_{THI,t} - v_{THI,t-1})/\delta t + (1 - \beta_{THI}) * r_{THI,t-1}$
    predict sink value using linear extrapolation
    $$THI_{sink,t} \leftarrow \begin{bmatrix} 1 & t - t_{sink} \\ 0 & 1 \end{bmatrix} THI_{sink,t_{sink}}$$
    if eventful $(|v_{sink,t} - v_{THI,t}| > \varepsilon_{THI})$
    or $t - t_{sink} \geq t_{heartbeat}$ then
      a. Transmit$((v_{THI,t}, r_{THI,t}), n, t)$
      b. $n \leftarrow n + 1$ (increment sequence number)
      c. when acknowledgement received
        i. $THI_{sink,t} \leftarrow s_{THI,t}$
        ii. $t_{sink} \leftarrow t$
        iii. $t_{heartbeat}$ reinitialized

containing the updated state is sent to the sink. $t_{heartbeat}$ checks, using periodic packet transmissions, if the system is running correctly and avoids large approximations in THI values when higher values of thresholds are used. On receiving the acknowledgement, the local variables are updated at the static sensor. A similar approach shall be used in case the initial request is received by a humidity sensor. This process continues unless the cow collar disconnects from the static sensor. In case a failure occurs at the static sensor, it must hand over its current state to the neighbouring sensors. Each cow collar must maintain a connection with at least one static sensor at all times. CEM, thus, allows continuous monitoring of THI as experienced by cows moving across different farms.

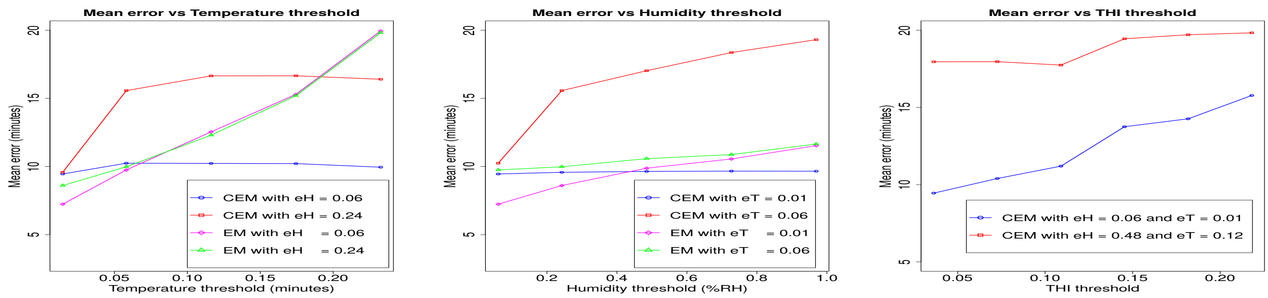
IV. Evaluation

In this section, we present our evaluation of EM and CEM for the estimation of THI values. In accordance with the EM technique, temperature and humidity readings are transmitted to the sink only at the occurrence of events at the respective sensors. THI values are then calculated at the sink node. Contrary to this approach, THI values are



(a) Packet reduction at fixed eH & $\varepsilon_{THI} = 0.04$ (b) Packet reduction at fixed eT & $\varepsilon_{THI} = 0.04$ (c) Packet reduction at fixed eT & eH

Fig. 3: Packet reduction for EM and CEM for different thresholds and $t_{heartbeat} = 60$



(a) Mean error at fixed eH & $\varepsilon_{THI} = 0.04$ (b) Mean error at fixed eT & $\varepsilon_{THI} = 0.04$ (c) Mean error at fixed eT & eH

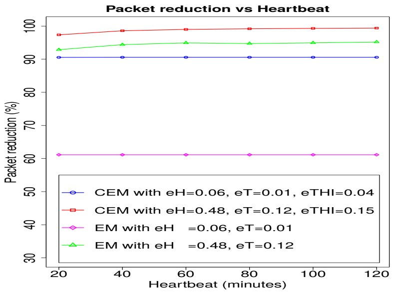
Fig. 4: Mean error for EM and CEM for different thresholds and fixed $t_{heartbeat} = 60$

locally computed at the static sensors within the network using CEM. Packets are sent only when a THI event is detected or $t - t_{sink} \geq t_{heartbeat}$. Both EM and CEM do not add any overhead to WSN functionality nor do they increase packet transmissions within WSN. They only affect the quality of data at sink since drop in the number of packet transmissions might reduce the accuracy of readings sent. We, therefore, evaluate the performance of EM and CEM based on the following two metrics using R programming:

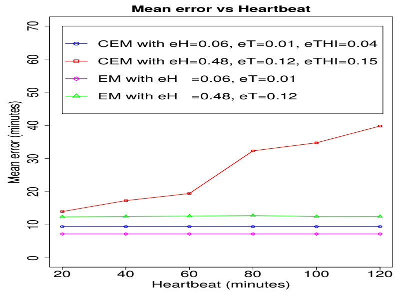
1) Packet reduction (%): An obvious advantage of the two approaches is the reduction in the number of messages that are forwarded to the sink node. The extent by which transmissions can be minimized is dependent on the input parameters, namely, temperature threshold (eT), humidity threshold (eH), ε_{THI} and $t_{heartbeat}$. We evaluate the average value of packet reduction over a five day period for different values of input parameters.
2) Mean error (minutes): Since the severity of Heat Stress increases directly with increase in THI levels, it is important that we maintain the THI value within permissible bounds. However, the two approaches of EM and CEM approximate the THI values until an event is detected. This leads to premature or delayed detection, depending on the nature of the curve, of the changes in THI levels during signal reconstruction. We have calculated the average time difference incurred using EM and CEM while estimating THI levels from 64 to 73.5 with a step of 0.01 over a five day period and termed it as the mean error. Its value varies with the input parameters.

While we have assigned β values for temperature, humidity and THI by calculating the $\mathbb{E}\left[|tmp_t - tmp_{t-1}|\right]$, $\mathbb{E}\left[|hum_t - hum_{t-1}|\right]$ and $\mathbb{E}\left[|thi_t - thi_{t-1}|\right]$ respectively, the preferred value of α's for evaluation is 0.94, based on the best fit approach. We approximate THI values based on EM and CEM for different values of eT, eH, ε_{THI} (set as multiples of β values) and $t_{heartbeat}$ and analyse the performance based on the above two metrics. Small values of these parameters imply high frequency of packet transmissions and, in turn, small mean error which is crucial in latency-sensitive applications. We limit threshold values such that the error rate for all three variables is less than 3%. Fig. 3 illustrates the packet reduction for EM and CEM for different values of thresholds and constant $t_{heartbeat}$. Packet reduction increases with the increase in threshold values for both EM and CEM till it reaches a plateau. This occurs as packet forwarding at large thresholds is primarily governed by $t_{heartbeat}$. We have fixed $t_{heartbeat}$ equal to one hour since it is considered as a reasonable frequency to update variable states in our scenario.

Figure 4 depicts the mean error caused by EM and CEM for different values of thresholds and fixed $t_{heartbeat}$. Mean error, in case of EM, increases continuously with the rising thresholds. For CEM, however, mean error increases with increase in threshold values and approaches a plateau at larger eT, eH and ε_{THI} values. This is explained by the dominance of $t_{heartbeat}$



(a) Packet reduction at fixed eT, eH and ε_{THI}



(b) Mean error at fixed eT, eH and ε_{THI}

Fig. 5: Performance comparison for EM and CEM for different heartbeat values

over threshold values in determining packet transmissions. In some cases, mean error in CEM is greater than EM since packet transmissions in CEM not only depend on eT and eH but also ε_{THI}. Further, we study the performance of EM and CEM across different values of $t_{heartbeat}$ and at fixed thresholds eT, eH and ε_{THI} (fig. 5). While message transfer at small values of input parameters is primarily controlled by the thresholds, $t_{heartbeat}$ plays an important role in improving the performance at higher threshold values. Increase in heartbeat implies lower frequency of packet transfer, thereby, increasing both the packet reduction and mean error.

As is evident, CEM performs well in most scenarios, especially, in latency-sensitive applications where small values of input parameters are used. In comparison to EM, CEM significantly reduces the number of packets sent. This, in turn, improves the energy efficiency of the network since switching the radio on/off and packet transmission is more resource intensive than simple computing. Additionally, CEM minimizes storage requirements at the sink by improving network intelligence through on-site data analysis and interpretation. One of the challenges of CEM, however, is resolving the trade-off between packet reduction and mean error by fixing the threshold and heartbeat values.

V. Conclusion

In this paper, we have discussed the shortcomings of the currently used sensor analytics techniques and proposed an extension of the EM model. Based on the Apache Storm framework, we have described a novel data analysis technique, CEM, for latency-sensitive applications in IoT. It exercises parallel and distributed computing and facilitates the collective participation of different sensors for event detection. We have presented our CEM model and its L-SIP-based implementation in the scope of a precision farming application. We have evaluated our approach for the estimation of THI values based on temperature and humidity states using R programming. While CEM is preferred over EM for latency-sensitive applications where small values of eT, eH and ε_{THI} are required, mean error in case of CEM is slightly higher than EM for larger ε values until $t_{heartbeat}$ dominates packet forwarding. A key decision in implementing CEM in sensor networks, therefore, is assigning values to the input parameters such that optimal performance is achieved while also realizing the application requirements and timeliness. The approach is generic and can be applied to a wide range of IoT applications.

Acknowledgment

This work has received support from the Science Foundation Ireland (SFI) and the Agriculture and Food Development Authority, Ireland (TEAGASC) as part of the SFI TEAGASC Future Agri-Food Partnership, in a project (13/IA/1977) titled "Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems".

References

[1] C.J. Rutten, A.G. Veilthuis, W. Steeneveld, H. Hogeveen, "Invited review: sensors to support health management on dairy farms," Journal of Dairy Science, Elsevier, vol. 96, no. 4, pp. 1928-1952, Apr. 2013.

[2] F. Bonomi , R. Milito , J. Zhu, S. Addepalli, "Fog computing and its role in the internet of things," Proceedings of the first edition of the MCC workshop on Mobile cloud computing, Aug. 2012.

[3] S. Sarkar, S. Chatterjee, S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," IEEE Transactions on Cloud Computing, vol.PP, no.99, pp.1-1, Oct. 2015.

[4] E.I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, R. Rednic, "Edge Mining the Internet of Things," IEEE Sensors Journal, vol. 13, no .10, pp. 3816-3825, Oct. 2013.

[5] Apache Storm, Documentation available at http://storm.apache.org/documentation.html, Last accessed Nov. 2015.

[6] S. Ivanov, K. Bhargava, W. Donnelly, "Precision Farming: Sensor Analytics," IEEE Intelligent systems, vol.30, no.4, pp.76-80, July-Aug. 2015.

[7] T. Rui, X. Guoliang, L. Benyuan, W. Jianping, J. Xiaohua, "Exploiting Data Fusion to Improve the Coverage of Wireless Sensor Networks," IEEE/ACM Transactions on Networking, vol. 20, no. 2, pp. 450-462, Apr. 2012.

[8] G. Serpen, J. Li, L. Liu, Z. Gao, "WSN-ANN: Parallel and Distributed Neurocomputing with Wireless Sensor Networks," The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, Aug. 2013.

[9] A.D. Vries, "Economics of Heat Stress: Implications for Management," Dairy, Jun. 2014.

[10] V. Gantner, P. Mijić, K. Kuterovac, D. Solić, Ranko Gantner, "Temperature - humidity index values and their significance on the daily production of dairy cattle," Daily production of dairy cattle, Mljekarstvo vol. 61, pp. 56-63, 2011.

[11] Lallemand Animal Nutrition, "New insights on heat stress evaluation and impact in dairy cows and its management through microbial solutions," Levucell, 2012.

[12] D. Goldsmith, J. Brusey, "The spanish inquisition protocol: Model based transmission reduction for wireless sensor networks," Proceedings of IEEE Sensors 2010, pp. 2043-2048, Nov. 2010.

Appendix E

Fog-enabled WSN System for Animal Behaviour Analysis in Precision Dairy

Conference Title:	International Conference on Computing, Networking and Communications (ICNC 2017)
Article Type	Regular Paper
Complete Author List	Kriti Bhargava, Stepan Ivanov, Chamil Kulatunga and William Donnelly
Status	Published: Jan. 2017

Fog-enabled WSN System for Animal Behavior Analysis in Precision Dairy

Kriti Bhargava, Stepan Ivanov, Chamil Kulatunga, William Donnelly
Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland
Email: (kbhargava, sivanov, ckulatunga)@tssg.org, wdonnelly@wit.ie

***Abstract*—Monitoring and analysis of animal behavior are two of the prominent applications of Wireless Sensor Networks (WSN) in modern Dairy Farming. Behavioral information collected by sensor devices worn by the animals is expected to provide early detection of stress and onset of specific diseases. Animal mobility coupled with farm-based contextual information is expected to automate and increase efficiency of the pasture. Though some WSN solutions have been proposed for these applications, their realizations commonly depend on high availability of third-party components (e.g. cloud-environment for behavior analysis). This reduces suitability of these solutions for pasture-based dairy farms, where large scale and remote locations significantly restrict accessibility to external components (e.g. poor or no internet connectivity). Meanwhile, continuous design improvement of WSN devices has significantly increased their computational capacity. To take advantage of this, a novel Edge Mining (EM) concept has been proposed under the umbrella of Fog Computing, where to increase availability, data analysis is partially hosted by WSN. In this article, we propose an Edge Mining implementation of our WSN system for analyzing animal mobility and behavior. We develop a novel EM method that could be used for a range of animal activity and behavior analysis. Performance of the method is evaluated regarding the accuracy and suitability for WSN-based execution.**

***Index Terms*—wireless sensor networks, behavior analysis, virtual fence, fog computing, edge mining.**

I. Introduction

Real-time monitoring and analysis of animal behavior is of utmost importance for early detection of diseases in order to promote animal welfare. Wireless Sensor Networks (WSN) have been widely proposed to monitor animal health and mobility, owing to their ability to collect real-time data, under the umbrella of Precision Dairy Farming. Mobility patterns give an understanding of the cattle behavior and can be used to detect health stress. Irregularities in the behavior, for instance, may be indicative of the onset of Lameness, a major heath issue for dairy cattle that adversely affects animal well-being as well as milk production [1]. Additionally, mobility tracking is used to improve feeding efficiency and prevent overgrazing within the farms. Virtual Fence (VF) technology has been implemented to restrict the movement of cows within a given boundary. VF replace the physical barriers and redirect the cows based on acoustic and electric stimuli. The initial implementations of VF use electromagnetic coupling between sensor devices carried by collared cows and an insulated wire unrolled in the farm to detect when a cow reaches the boundary [2]. This approach relies on the installation of additional infrastructure which is often cumbersome in remote farm locations. Alternative techniques use GPS receivers fitted to the collar devices to determine the location of the animal and estimate its distance with respect to the VF [3]. The role of sensor devices in all of the above techniques, however, is limited to data collection and transmission while the analysis takes place on the cloud. This introduces latency in analysis and poses a major constraint in WSN implementation in large-scale, rural farm environments that suffer from intermittent or no Internet connectivity. Furthermore, GPS modules are energy intensive and their use for VF applications adversely affects the network longevity.

Meanwhile, with the recent advances in technology, there has been a tremendous improvement in the design of sensor devices that constitute the WSN. Traditionally limited to sense and send, these devices are now capable of performing more complex tasks. Accordingly, a new computing paradigm, called Fog Computing [4] has been proposed to bring down certain computations away from the cloud and closer to the network. Fog Computing aims at optimizing the in-network resource utilization while providing faster insights into the data which is crucial, especially, for latency-sensitive applications. The suitability of Fog Computing over Cloud Computing for Internet of Things (IoT) applications has been analyzed in [5]. Edge Mining [6] is a novel Fog Computing approach wherein light-weight data mining algorithms are offloaded on the sensor devices in order to facilitate near real-time data analysis. The reduction in data is based on the assumption that prior knowledge of the application requirements is available. Edge Mining facilitates event detection within the network through identification of significant changes in the application state from the previous values. Moreover, it improves the quality of data exchange within the network by sending only the significant state estimates to the sink. The reduction in number of packet transmissions, in turn, improves the overall energy efficiency of the network.

In this paper, we present our WSN system for behavior analysis in dairy cows. We discuss the design of our sensor nodes that are worn by the dairy cows (collar devices) and are used to collect acceleration data. Real-time analysis of this data for mobility monitoring is performed on the collar devices itself as a cow moves along the farm. Thus, animal health assessment and VF boundary enforcement is performed autonomously. However, given the limited computational capa-

978-1-5090-4588-4/17/$31.00 ©2017 IEEE

bility of the collar devices, we perform the compute-intensive network learning on the cloud. We adopt a delay-tolerant communication framework wherein the results of mobility monitoring are stored locally on the collar devices until the cows are in vicinity of a cloud gateway. The monitoring data is then uploaded to the cloud where it is carefully analysed. Results of the analysis are sent back to the collar devices in a delay-tolerant manner via the gateway nodes. Each collar will use these results to fine-tune its own animal health and mobility analysis. For the collar-based analysis, we propose Iterative Edge Mining (IEM), a data agnostic technique for classification in WSN. IEM is realized by super imposing two Edge Mining approaches: Bare Necessities (BN) and ClassAct, and performs decision tree based classification on the distribution of signal over time. We consider the implementation of IEM on the collar devices to identify the activity state of a cow based on the acceleration data. The histograms for each state over time can be used to understand the mobility pattern of dairy cattle which is, in turn, indicative of animal stress and other issues. Moreover, given the farm-based topology information, real-time analysis of the mobility gives an estimate of the relative position of cattle with respect to the VF, without the need of any external infrastructure. We evaluate the performance of IEM in terms of classification accuracy and study the changes in performance across different values of the input parameters.

II. Related Work

Although a number of WSN systems have been proposed for animal health management in dairy farms, a survey in [7] shows the need for these systems to become intelligent. Accordingly, the quality of collar-based analysis is one of the key factors impacting the efficiency of the overall system. In this section, we present an overview of the existing approaches for sensor analytics that could be used by collar devices.

Owing to the increase in computational power of sensor devices, a number of data analysis techniques have been implemented within WSN. Data fusion, for instance, is one of the most basic approaches that has been proposed to reduce data redundancy within WSN by merging the overlapping data that emerges from the neighboring sensor nodes [8]. This, in turn, reduces the number of packet transmissions within WSN and improves the overall energy efficiency of the network. Fusion algorithms, however, are signal specific and cannot be easily extended to heterogeneous data streams. The implementation of Artificial Neural Networks (ANN) on top of the existing WSN topology has also been proposed to perform localized classification, clustering and prediction tasks in [9]. Although ANN algorithms are data agnostic, the network learning involved is computationally intensive and may adversely affect the lifetime of the network.

In the recent years, Edge Mining techniques have been proposed to perform light-weight data mining tasks at the network edges with an aim to reduce packet transmissions in the network. Edge Mining has been implemented using the Spanish Inquisition Protocol (SIP) [10] that suggests transmission of only the unexpected information from the sensor nodes to the sink. SIP first encodes the data into an application relevant state. It then uses a local copy of the previous sink state along with a shared approximation model to predict the expected state at the sink node. Only if the difference between the predicted and calculated value exceeds the given threshold ε, the new state is considered eventful and is sent to the sink. Edge Mining has been implemented using three instantiations of the SIP - Linear SIP (L-SIP), ClassAct and BN. L-SIP uses a linear model to encode the data as a point-in-time value and rate of change. Whereas, L-SIP allows the reconstruction of signal within a given error bound, ClassAct and BN, on the other hand, are relatively destructive approaches that are used for activity classification and data summarization respectively. Edge Mining algorithms have been shown to significantly reduce the number of packet transmissions and, thereby, the energy consumption within the network. In the remainder of this section, we describe the ClassAct and BN approaches that form the basis of IEM.

A. *ClassAct*

ClassAct is a decision tree based activity classifier that encodes the state estimate as a probability distribution over a given set of activities. The distribution is simplified to the index of the most likely activity using Exponentially Weighted Voting and a packet containing the new state is generated only if there is a change from the previous estimate. While building the decision tree is computationally intensive and is carried out on a remote server, the evaluation of the decision tree can be performed in a few instructions at the sensor node. In [11], ClassAct has been implemented for posture recognition in an enclosed bomb disposal unit using accelerometer data from a Body Sensor Network. A decision tree constructed using the C4.5 algorithm has been used to classify the states into nine given postures based on the data collected from a set of eleven sensors at a frequency of 10Hz. ClassAct provides a significant packet reduction of the order of 98-99% and an accuracy of 88%. The performance is shown to further improve upto 97% through classification based on time domain features such as windowed mean and windowed variance. A major drawback of ClassAct, however, is that the classification is based on a small set of probabilistic moments represented by the feature values. Therefore, it may not be able to distinguish between signals that may follow different distributions but exhibit the same feature values. For instance, the algorithm may not differentiate between two signals that follow normal and uniform distribution respectively but exhibit the same value of mean and variance, thereby, increasing the misclassification rate.

B. *Bare Necessities (BN)*

BN significantly reduces packet transmissions by storing only the summary of data over time at the sink node. It estimates the application state as a distribution across non-overlapping bins and provides the relative time spent in each bin. For each reading z, BN updates the count x and

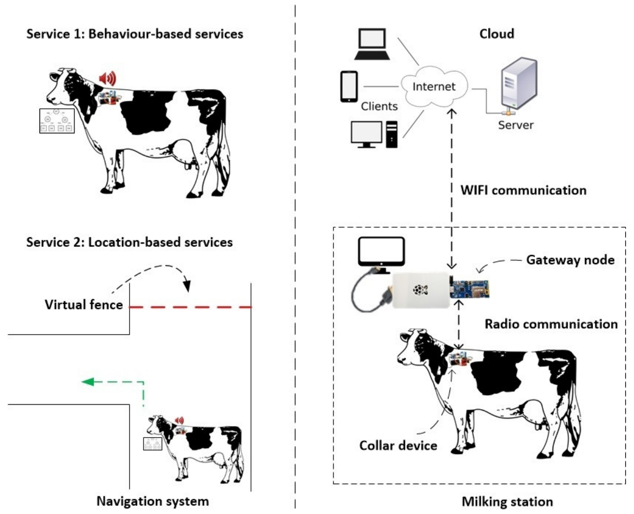


Fig. 1: WSN architecture for precision dairy farming

distribution y for every bin i belonging to a set of bins B as shown in eq. 1.

$$\left.\begin{array}{l} x_i \leftarrow \gamma \cdot x_i + b(i,z) \\ y_i \leftarrow x_i / \Sigma_{i \in B} x_i \end{array}\right\} \tag{1}$$

The function b yields 1 if the value z belongs to the band i and 0 otherwise. Here, γ is a smoothing parameter, called the decay factor, that is applied to the previous estimate and is calculated as shown in eq. 2 where $t_{1/2}$ is the decay half-life and T is the sensing period such that $0 < \gamma < 1$.

$$t_{1/2} = \frac{T \cdot ln2}{1-\gamma} \tag{2}$$

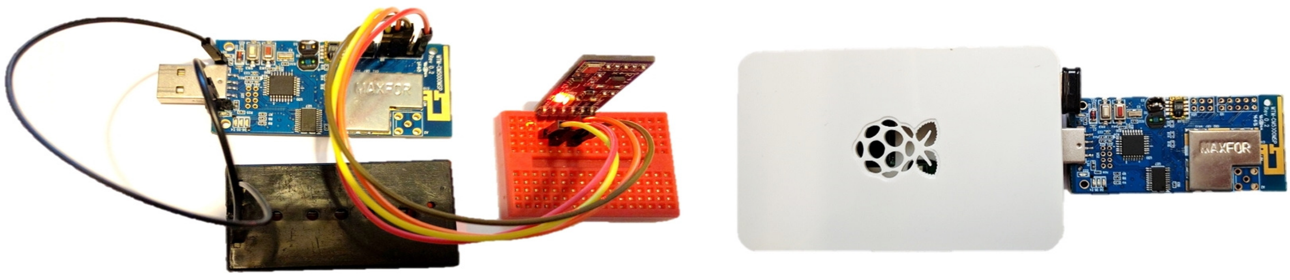
The change in state is considered eventful if the distribution of any bin changes by more than the given threshold ε. In [12], the performance of BN has been evaluated in the context of a residential building monitoring. BN is used to provide the proportion of time for which a room was in cold, comfortable, warm or overheated state based on the temperature readings and is shown to reduce the transmission frequency of temperature data from once every 5 minutes to once in 38 days for an allowed error of 10%. Furthermore, the average transmission of temperature, humidity and CO_2 is reduced to once per 13 days for the same threshold, thereby, improving the network lifetime. Given that BN reduces packet transmissions to the order of 99%, an additional heartbeat mechanism is used to update the state if the time since last packet transmission exceeds a threshold $t_{heartbeat}$. The periodic transmissions also enable detection of failed nodes.

III. Proposed solution

In this section, we present an overview of our WSN prototype system for behavior analysis in dairy cows. Our proposed architecture comprises of two kinds of sensor nodes: mobile sensor devices (cow collars) and a cloud gateway as shown in fig. 1. Each cow is equipped with a collar device that is used to monitor its acceleration as it moves along the farm. The IEM algorithm runs locally on each device, per sensing cycle, to analyze the acceleration values and identify the activity state of the cow. Due to lack of Internet connectivity in a farm environment, the results of this analysis are stored locally on the device until the cow is in the vicinity of the gateway node, hosted inside the milking station. Once a cow reaches the milking station, data from the collar devices is transmitted to the gateway node over the radio and is further uploaded on the cloud through WiFi connectivity of the Raspberry Pi of the gateway node. Although the execution of IEM takes place on the device without the use of any external infrastructure, the network learning is performed on the cloud. Cloud-based analysis uses the past estimates and provides depper insights into the data to allow revision of the IEM configuration and, thereby, improve the quality of collar-based behavior analysis. Configuration revisions are delivered to each animal-collar individually via the gateway node.

The relative time spent in each state as obtained from the IEM algorithm gives us the mobility pattern of the cow. Specific irregularities in behavior indicate the onset of animal health-related issues. As IEM is continuously executed by the collar device, results of the IEM analysis of cow-mobility (since previous milking) are available before a cow reaches the milking station. This facilitates immediate identification of sick cows from the healthy ones. The identification does not require a stable Internet connectivity between the gateway node and the cloud which is particularly important for remote farms. Real-time analysis of the mobility can be further used to calculate the distance traveled by each cow, over time. Given the topology information of the farm, this value gives an estimate of the location of the cow inside the farm. Each time a change in activity is recorded, a behavior algorithm provides real-time navigation instructions to the cow based on its relative position with respect to the VF. As illustrated in fig. 1, when a cow approaches/stops at a T-point, its location is estimated relative to the VF and appropriate instructions are given by the collar to turn left.

Fig. 2 illustrates the design of our collar devices and the gateway node. The basic unit of the collar device is CM5000 mote [13] that consists of a MSP430 family processor and a CC2420 802.15.4, 2.4GHz wireless module for radio communication. A 10 degrees of freedom Inertial Measurement Unit (IMU) is externally connected to the mote via the I^2C interface. The IMU comprises of a MPU6050 IC [14], for measuring the acceleration and orientation (gyroscope) on X,Y and Z-axis, along with a magnetometer and pressure sensor. We design a TinyOS application that runs on the collar device to collect acceleration readings at a frequency of 1Hz. The data is analyzed and stored locally in the device flash. The gateway node comprises of a CM5000 mote connected to a Raspberry Pi (model B2) [15] as shown in fig. 2b. Once the cow reaches the milking station, it requests to join the Personal Area Network (PAN) of the gateway in order to initiate the data transfer. If the association request is confirmed, the device sends data packets to the CM5000 mote of the gateway



(a) (b)

Fig. 2: (a) Cow collar device (b) Cloud gateway node

node via mote-to-mote communication. A TinyOS application running on the gateway mote transfers these packets to the Raspberry Pi, using the underlying UART interface, from where they are forwarded to the cloud for further analysis. Upon completion of the data transfer, the device disassociates from the gateway node. In the remainder of this section, we describe the IEM algorithm that runs on the collar devices for mobility tracking in dairy cows.

A. *Iterative Edge Mining*

As stated previously, IEM is based on the interaction between two Edge Mining algorithms: BN and ClassAct. BN takes the raw acceleration reading and converts it into an intermediate state that represents the distribution of signal across different bins. The sequence of distributions generated by BN over time acts as the input for the ClassAct algorithm and is fed into a decision tree classifier that identifies the application state from the given library for each distribution. An event is generated if the state value corresponding to any distribution differs from the previous estimate.

The distribution of signal is calculated using the bin counts and is characterized by the decay factor γ. The parameter γ smooths the distribution of signal over time on the assumption that a signal does not change its value abruptly and stays in the same state for a certain period of time. Since BN detects events based on changes in the smoothed signal distribution, higher values of γ will reduce the number of classification events generated by BN. While this reduces the number of computations required, it leads to misclassification of activity around the actual state transitions, referred to as cross-state misclassification, causing a delay in detecting change in application state. Moreover, if an event is misclassified without an actual change in the application state, the effect of this event lasts longer for higher values of γ, thereby, increasing the number of misclassifications. For a fixed γ, the number of classification events also vary with the values of ε and $t_{heartbeat}$. The value of ε decides the percentage of error allowed in the distribution of each bin and is determined by the application requirements. Higher values of ε allow larger approximations from the original signal, leading to fewer classification events. Here, $t_{heartbeat}$ value ensures periodicity of events generated by BN and, in turn, the classification

Algorithm 1 Iterative Edge Mining for classification

```
procedure :
Update bin count ∀i ∈ B
    x_{i,t} ← γ · x_{i,t−1} + b(i, z_t)
Update bin distribution (simplify) ∀i ∈ B
    y_{i,t} ← x_{i,t} / Σ_{i∈B} x_{i,t}
y' ← previous significant distribution at time t'
If ∃i ∈ B : |y_{i,t} − y'_i| > ε or t − t' ≥ t_{heartbeat}
    Update y' ← y_t and t' ← t
    Estimate new state
    s_t ← f(DT, y_t)
    Eventful?
        Yes, if state differs from last update
            Store in Flash state s_t and time t
```

checks, by fixing the maximal duration for which any two consecutive BN events can be apart. That is, it injects an artificial BN event if no change in distribution is detected for the duration of $t_{heartbeat}$. Once an intermediate event is detected by BN, ClassAct is performed on the current distribution of signal to identify the application state and detect events. The overall performance of classification also depends on the user-specified number and sizes of each bin. The values of input parameters must, therefore, be chosen carefully to balance the trade-off between the classification accuracy and number of computations in order to optimally utilize network resources as well as meet the application requirements.

IEM realization: Since network learning is computationally intensive and may reduce the network longevity, we use cloud-based analysis to determine an optimal set of non-overlapping and jointly exhaustive bins B and calculate the values of input parameters namely γ, ε, $t_{heartbeat}$. Once the learning is complete, classification is performed using IEM in a few simple steps on the sensor nodes. Algorithm 1 summarizes our approach. For every reading z at time t, we update the bin count x by the decay factor γ and increment the currently active bin using the predicate function $b(i, z_t)$. The count is updated for each bin i belonging to the set B where i represents a subset of values that a given signal can take. The bin distribution y for all bins in B is then calculated

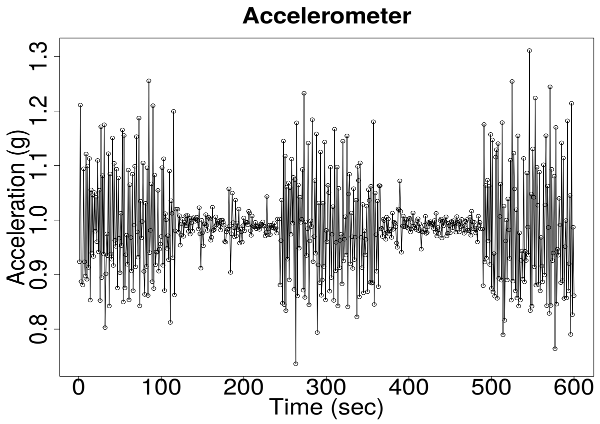


Fig. 3: Acceleration data over time

by normalizing the bin counts. The distribution suggests the proportion of time spent in each bin and reflects the variation in values. Next, we calculate the change in distribution with respect to the previous significant distribution y' at time t'. The distribution y is considered significant if one of its components differs from y' by greater than the allowed error threshold ε or the time since y' was last updated exceeds a threshold $t_{heartbeat}$. If y is significant, y' is updated and the application relevant state is estimated using decision tree classification based on y. The calculated state s_t is compared to the previously stored application state and an event is detected if the two states differ from each other. If so, the new state s_t is stored in the flash along with the current timestamp t. Whereas the ClassAct approach in Edge Mining traverses the decision tree to identify the application state per sensing cycle, IEM performs classification only if the change in distribution of signal exceeds the given threshold ε. This significantly reduces the number of computations, especially, where the decision tree is of a large size.

IV. Evaluation

To validate the proposed method, human acceleration data is used in this article. We implement IEM to classify stand and walk activities using data collected from a single MPU6050 accelerometer. MPU6050 measures 0g on the X and Y, and +1g along the Z-axis when placed horizontally on a flat surface. We collect acceleration data along the three axes at a sampling frequency of 1Hz by alternating between walk and stand activities in approximately 2 min intervals. We repeat the experiment 4 times for a stretch of 10 min each. The total acceleration a_{total} is calculated as shown in eq. 3, where a_X, a_Y and a_Z represent acceleration along X,Y and Z axis respectively.

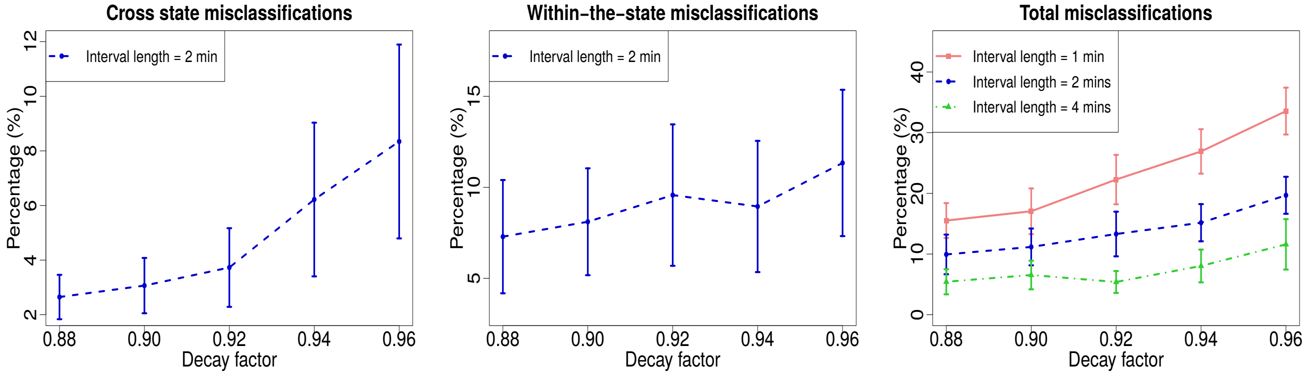
$$a_{total} = \sqrt{a_X^2 + a_Y^2 + a_Z^2} \tag{3}$$

The resulting data for one dataset is as shown in fig. 3. Small variations from 1g are representation of stand periods while slightly larger variations correspond to the walking intervals. Since stabilization period does not reflect the performance of the system, we ignore the data values for the first 2-minute interval and evaluate the performance of IEM for stretches of 8-minutes of activity on the basis of classification accuracy. Of the four data sets, we leave one out for training the classifier using C5.0 decision tree algorithm in R and evaluate the performance of IEM for the remaining three sets. To improve the reliability of results, we perform a 4-fold cross-validation by rotating the training set between the four files. We consider the distribution of signal across three equal-sized bins representing the range of accelerometer data and fix the value of $t_{heartbeat}$ to 15 seconds.

A. *Effect of γ on performance*

Since IEM should achieve high classification accuracy regardless of the number of events generated by BN, we first consider the effect of decay factor γ on the performance of IEM by assigning ε to zero. This selection for the ε value corresponds to the highest sensitivity of BN to changes between distributions. This, in turn, allows us to evaluate classification quality for all possible distributions for a particular γ. For each training set, we generate five models for five different values of γ and evaluate their performance against the corresponding test data using R analysis. Using cross-validation, we get results from 12 test files for each value of γ. For each file, we calculate the average delay in detecting the state transitions by calculating the average number of cross-state misclassifications. We calculate the mean value for the same across all 12 files and estimate the percentage error with respect to the total number of acceleration readings over 8 minutes for each γ. A higher value of γ increases the smoothing in distribution, thereby, increasing the effect of the previous estimates on the current distribution. This makes the detection of state transitions more difficult leading to higher percentage errors in cross-state misclassifications as depicted in fig. 4a. While the error for smaller values of γ is as low as 3%, it increases to approximately 8% for a γ value of 0.96.

Furthermore, as seen in fig. 3, there is an overlap between walk and stand acceleration values which causes an inaccuracy in classifying the activity states, at certain points, without an actual change in application state. We refer to these inaccuracies as within-the-state misclassification. We calculate the average number of these misclassifications within and across all files for each value of γ. Within-the-state misclassifications as a percentage of total number of readings per file is as shown in fig. 4b. A higher value of γ smooths the distribution of signal and, therefore, ignores the small fluctuations in the acceleration values. However, if a state is misclassified, its effect lasts longer on the future distributions leading to an increase in the number of misclassifications. Due to this combined effect, the overall number of within-the-state misclassifications remains in the same range across different values of γ. We then calculate the total number of misclassifications across the entire length of activity monitoring. As expected, an increase in the percentage value is observed with the increase in γ values (fig. 4c). While the classification accuracy is as high as 90% for smaller γ, it decreases upto 83% for large γ.



(a) (b) (c)

Fig. 4: (a) cross state, (b) within-the-state and (c) total misclassifications for $\varepsilon = 0$

To understand the performance of IEM across different mobility patterns, we use ranking statistics to transform our data sets to generate files for two other movements: jerky motion with 1 min activity intervals and smooth motion with 4 min activity intervals for the same duration of 8 min. Owing to a large number of state transitions, the average number of cross-state misclassifications increases for the jerky motion while a drop in value is observed in case of smooth motion. Moreover, in case of jerky motion, the previous activity state has a significant impact on the current distribution owing to the small window size of 1 min for each activity, further increasing the stretch of cross-state misclassifications. This effect is less for the smooth changes due to the larger activity intervals. Accordingly, the total percentage of misclassifications is higher than the original data for the jerky motion as shown in fig. 4c. On the contrary, the classification accuracy increases upto 95% for the smooth motion. To ensure misclassification within the allowed error rate, frequent classifications are, therefore, required in case of jerky motion i.e. a small ε or $t_{heratbeat}$ value, while the number of calculations can be significantly reduced for smooth motion.

B. Effect of γ and ε on performance

Next, we consider the combined effect of γ and ε on the performance of IEM for the original data files. For the same set of training models as above, we generate test files corresponding to the five values of γ for three different ε values. While ε has no impact on the distribution of signal, it determines the interaction between BN and ClassAct by varying the number of classification events generated. The number of classification events as a percentage of total readings across different values of γ is shown in fig. 5. For a fixed value of γ, a reduction in number of checks is observed with increase in ε value as the model allows greater approximations in the signal. The value further decreases with increase in the value of γ as smoothing of the distribution generates fewer events. The reduced frequency of checks, however, adds to the delay in detecting state changes and leads to an increase in the number of cross-state misclassifications. As shown in fig. 6a, the percentage error increases from 3% to 5% compared to fig. 4a for an increase in ε value from 0 to 0.02 (2% error). This value further increases with the value of ε due to lower probability of classification checks, thereby, increasing the stretch of misclassified values. Heartbeat value ($t_{heartbeat}$) plays an important role here to ensure periodicity of checks and avoid large number of misclassifications. On the contrary, larger ε ignores the small fluctuations within the state resulting in fewer number of checks and, in turn, within-the-state misclassifications as shown in fig. 6b. As a result, while the total percentage of misclassifications increases compared to the values in fig. 4c, the change is comparable across different values of ε for a fixed γ (fig. 6c). This implies that a similar classification accuracy can be achieved for large ε while permitting fewer calculations. However, care must be taken in deciding the ε value since large ε results in a larger delay in detecting state changes.

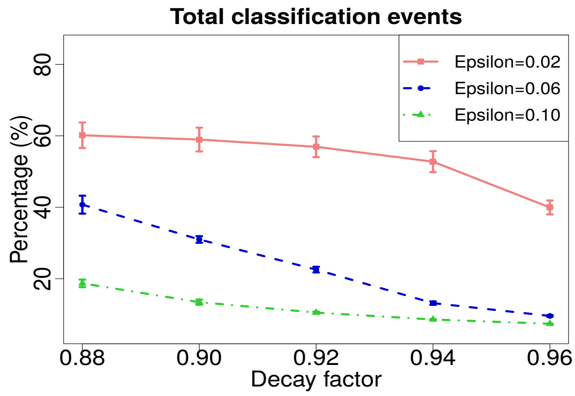
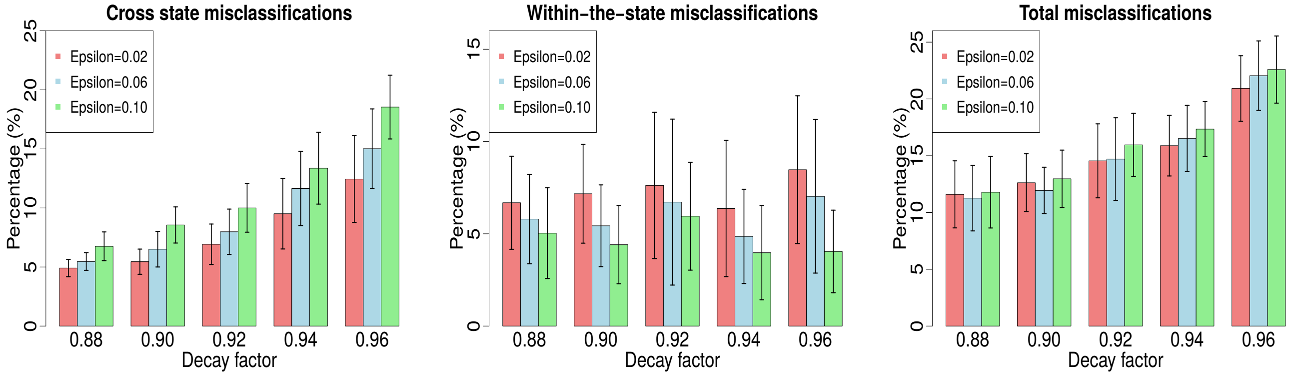


Fig. 5: Classification events for 2 min activity intervals at $t_{heartbeat}$=15 sec



(a) (b) (c)

Fig. 6: (a) cross-state, (b) within-the-state (c) total misclassifications for 2 min activity intervals at $t_{heartbeat}$=15 sec

V. Conclusion

In this paper, we present the design and proposed implementation of our WSN system for behavior analysis in dairy cows. We have built a fog service for real-time monitoring and analysis of mobility patterns of a cow that can be used to determine animal well being as well as provide location-based services such as Virtual Fence applications. Unlike the existing techniques that rely on third party components such as a cloud environment, we build an autonomous system wherein the analysis is performed locally on the collar devices worn by dairy cows as they move along the farm. We propose IEM, a data agnostic, decision tree based classification technique that identifies application specific states based on the distribution of signal over time. IEM is implemented on the collar devices to identify the activity state of a cow from a fixed library and understand the mobility pattern using the histogram of activity states over time. We evaluate the performance of IEM, based on human acceleration data, in terms of classification accuracy for stand and walk activities for different mobility patterns. Although the performance of IEM varies with the input parameters, an accuracy of the order of 95% is achieved for small values of decay factor γ and error threshold ε. Furthermore, a significant drop is seen in the number of classification events and, in turn, the number of computations required with increase in the parameter values. In future, we may consider the use of collaborative sensing to understand the herd behavior. The underlying approach is generic and can be applied to a wide variety of WSN applications.

Acknowledgment

This work has received support from the Science Foundation Ireland (SFI) and the Agriculture and Food Development Authority, Ireland (TEAGASC) as part of the SFI TEAGASC Future Agri-Food Partnership, in a project (13/IA/1977) titled "Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems".

References

[1] M. Alsaaod, C. Rmer, J. Kleinmanns, K. Hendriksen, S.R. Meierhfer, L. Plmer, W. Bscher, "Electronic detection of lameness in dairy cows through measuring pedometric activity and lying behavior," Applied Animal Behaviour Science, ELSEVIER, vol. 142, no. 3-4, pp. 134-141, Oct. 2012.

[2] M.O. Monod, P. Faure, L. Moiroux, P. Rameau, "A virtual fence for animals management in rangelands," The 14th IEEE Mediterranean Electrotechnical Conference, Ajaccio, pp. 337-342, May 2008.

[3] T. Wark, D. Swain, C. Crossman, P. Valencia, G. Bishop-Hurley, R. Handcock, "Sensor and Actuator Networks: Protecting Environmentally Sensitive Areas," IEEE Pervasive Computing, vol. 8, no. 1, pp. 30-36, Jan.-Mar. 2009.

[4] F. Bonomi , R. Milito , J. Zhu, S. Addepalli, "Fog computing and its role in the internet of things," Proceedings of the first edition of the MCC workshop on Mobile cloud computing, Aug. 2012.

[5] S. Sarkar, S. Chatterjee, S. Misra, "Assessment of the Suitability of Fog Computing in the Context of Internet of Things," IEEE Transactions on Cloud Computing, vol.PP, no.99, pp.1-1, Oct. 2015.

[6] E.I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, R. Rednic, "Edge Mining the Internet of Things," IEEE Sensors Journal, vol. 13, no .10, pp. 3816-3825, Oct. 2013.

[7] C.J. Rutten, A.G.J. Velthuis, W. Steeneveld, H. Hogeveen, "Invited Review: Sensors to Support Health Management on Dairy Farms," J. Dairy Science, vol. 96, no. 4, pp. 19281952, Feb. 2013.

[8] T. Rui, X. Guoliang, L. Benyuan, W. Jianping, J. Xiaohua, "Exploiting Data Fusion to Improve the Coverage of Wireless Sensor Networks," IEEE/ACM Transactions on Networking, vol. 20, no. 2, pp. 450-462, Apr. 2012.

[9] G. Serpen, J. Li, L. Liu, Z. Gao, "WSN-ANN: Parallel and Distributed Neurocomputing with Wireless Sensor Networks," The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, Aug. 2013.

[10] D. Goldsmith, J. Brusey, "The spanish inquisition protocol: Model based transmission reduction for wireless sensor networks," Proceedings of IEEE Sensors 2010, pp. 2043-2048, Nov. 2010.

[11] R. Rednic, E. Gaura, J. Brusey, "Classact: Accelerometer-based real time activity classifier," Proceedings of 2nd WiSIG Showcase, Teddington, UK: Sensors KTN, pp. 2125, Jul. 2009.

[12] E. I. Gaura, J. Brusey, R. Wilkins, "Bare necessities-knowledge driven WSN design," Proceedings of 10th IEEE Sensors, pp. 6670, Oct. 2011.

[13] Advanticsys, MTM-CM5000-MSP, Available at http://www.advanticsys.com/shop/mtmcm5000msp-p-14.html, Last accessed in May 2016.

[14] Invensense, MPU-6000 and MPU-6050 Product Specification Revision 3.4, InvenSense Inc., Aug. 2013.

[15] Raspberry Pi Foundation, RASPBERRY PI 2 MODEL B, www.raspberrypi.org, Last accessed in May 2016.

Appendix F

Fog-centric Localization for Ambient Assisted Living

Conference Title:	23rd ICE/IEEE International Technology Management Conference
Article Type	Regular Paper
Complete Author List	Kriti Bhargava, Gary McManus and Stepan Ivanov
Status	Published: Jun. 2017

Fog-centric Localization for Ambient Assisted Living

Kriti Bhargava, Gary McManus, Stepan Ivanov
Telecommunications Software &Systems Group
Waterford Institute of Technology
Waterford, Ireland
(kbhargava, gmcmanus, sivanov) @tssg.org

***Abstract*—Ambient Assisted Living (AAL) is a novel discipline that aims at improving the quality of life for all generations, especially the elderly, with the help of information and communication technologies. Behavioral tracking AAL systems necessitate the monitoring and understanding of daily activities and preferences of the user for design of customized, context-aware services and detection of behavior anomalies. Localization of the user is, therefore, key to facilitate real-time activity monitoring in AAL applications. Although several localization techniques have been proposed to date, majority of them incur a high operational cost owing to dependency on dense sensor deployments for ambient intelligence or use of expensive hardware such as GPS receivers. In this paper, we propose a low-cost Wireless Sensor Networks (WSN) system, comprising of a single wearable device and a cloud gateway, for outdoor localization in the context of AAL. With the inception of the Fog Computing paradigm, we consider the implementation of a light-weight data mining technique, Iterative Edge Mining (IEM), on the wearable device for on-board activity recognition. IEM is based on the classification of signal distributions to enable real-time mobility tracking as the user moves around an environment. Given the topology information and the activity sequence generated by the algorithm, we estimate the user location by associating the distance covered over time with the orientation values. Alerts are signaled locally upon detection of behavior anomalies and transmitted to the gateway node using a delay-tolerant communication framework. As such, IEM runs autonomously on the sensor node without interaction with external objects, thereby, improving the responsiveness as well as the operational cost of our system. We evaluate the performance of IEM in terms of localization accuracy in an outdoor environment.**

Keywords—ambient assisted living; localization; fog computing; edge mining; wireless sensor network

I. INTRODUCTION

With advancing age, the elderly often experience physical disabilities and require support with mobility and the activities of daily living. Moreover, they may develop some form of Dementia, a chronic syndrome that causes deterioration in the cognitive function beyond what might be otherwise expected with ageing. This, in turn, leads to challenging behavioral and psychological changes such as repetition, aggression, agitation and psychosis. Alzheimer's is the most prevailing form of Dementia that affects the short-term memory, orientation and intellectual capacity of an individual [1]. It may result in loss of identity, thereby, increasing distress for the patient as well as the caregivers. Wandering is a common symptom for Alzheimer patients that poses serious threat to their safety and may lead to traumatic experiences. Personalized monitoring and care of the elderly is, therefore, important to assist them with daily activities and ensure their well-being. Ambient Assisted Living (AAL) is a recent trend that combines Information and Communication Technologies (ICT) with the social environment with a view to improve the quality of life for all generations, primarily the ageing population with cognitive disabilities [2]. An important aspect of AAL is localization of the user to enable activity monitoring for safe and independent living and minimize the risk of wandering [3]. AAL solutions have the potential to not only allow patients to restore their usual routine but also to reduce the burden on caregivers. Although a few activity tracking systems have been proposed for AAL, their implementation is constrained due to the high operational cost incurred by use of expensive hardware such as GPS modules or dense sensor deployments and cloud infrastructure required for ambient sensing, communication and data analysis.

Meanwhile, owing to the growth in ICT, there has been a tremendous improvement in the design and computational capabilities of small devices that constitute edge of the network in the Internet of Things (IoT). A new networking paradigm, Fog Computing, proposes a partial migration of intelligence away from the cloud towards the network edges [4]. That is, Fog Computing aims at facilitating localized data processing and event detection at the end-user terminals. The concept has gained importance owing to its ability to efficiently utilize the in-network resources while minimizing dependency on the cloud infrastructure. It not only reduces the operational cost but also improves the responsiveness of the system for alert generation. Over the past few years, numerous interpretations of fog nodes within IoT have been discussed. While some approaches propose the use of computational resources at edge devices such as network switches [5], others suggest the use of free computation slots on user mobile phones [6]. Recent studies have further brought down the concept of Fog Computing to wireless, battery-operated sensor devices that sit at the edge of Wireless Sensor Networks (WSN). Edge Mining is a novel approach that suggests the implementation of light-weight data mining tasks on the sensor devices [7]. While resource-intensive network learning is performed on the cloud, minor computations carried out at the sensor nodes enable real-time event detection. Furthermore, Edge Mining algorithms improve the energy efficiency of WSN by reducing packet transmissions to the cloud gateway via localized data reduction and, in turn, increase operational time of the system.

In this paper, we implement fog-enabled mobility tracking within WSN for user localization in the context of AAL. Our WSN system consists of a low-cost wearable activity tracker and a cloud-gateway, and assumes prior knowledge of the application environment and user-specific information. The wearable device consists of an Inertial Motion Unit (IMU) that gathers accelerometer and gyroscope data as the user moves around the application environment. Real-time analysis of the data is carried out on the device itself using a Fog Computing approach called Iterative Edge Mining (IEM), proposed by the authors in [8]. IEM is based on the Edge Mining algorithms and performs activity state recognition using a decision-tree classifier on signal distributions. Given the topology information of the environment, the mobility pattern produced by IEM is used along with the gyroscope data to determine user location. Alerts are generated locally at the occurrence of unexpected events and transmitted to a cloud gateway using delay-tolerant communication. While our device performs localization autonomously, the resource-intensive network learning for IEM is performed on the cloud to modify the parameters for on-board analytics, if necessary, and tune the results according to changes in application environment and requirements. Results of the learning are sent back to sensor device, in a delay-tolerant manner, to adjust the analytic model. The performance of our system has been evaluated for outdoor localization using analysis in R.

The remainder of the paper is organized as follows. In section II, we discuss the related work. The application scenario proposed solution is presented in section III. The evaluation of our system is discussed in section IV followed by the conclusions in section V.

II. Related Work

In this section, we present an overview of some of the localization techniques and AAL solutions, proposed to date. We also discuss the state-of-the-art in sensor analytics with emphasis on the Edge Mining approach that forms the basis of IEM.

A. *Localization Techniques*

So far, numerous approaches have been proposed for localization in both outdoor and indoor applications. The use of Global Positioning Systems (GPS) based systems is well-known for outdoor positioning owing to the high availability of GPS modules in current IoT devices and the positioning accuracy. However, GPS units are expensive as well as energy exhaustive, thereby, affecting the lifetime of a system. Additionally, their performance deteriorates significantly in crowded and indoor areas due to the absence of line of sight to GPS satellites. Consequently, cooperative techniques have been proposed that use hybrid positioning systems to improve the performance of GPS systems [9]. Alternatively, radio frequency based solutions have been proposed for localization in indoor environments such as smart buildings as discussed in [10]. The role of WSN for node localization has also been explored. The techniques proposed are either anchor-based where fixed nodes with known GPS coordinates are used to estimate the coordinates of mobile nodes using different ranging techniques [11] or anchorless that aim at determining only the relative distance between two nodes. Majority of the solutions, however, rely on dense sensor networks for accurate sensing and communication making the network installation a tedious task. In recent years, Pedestrian Dead Reckoning (PDR) systems, comprising of wearable inertial sensors for self-tracking, have been designed to calculate user position based on the past estimates and displacement over short intervals of time. Personalized monitoring with PDR systems allows better understanding of the user behavior and mobility patterns for customization of services. A 3D localization technique using multiple wearable sensors has been presented in [12]. The system monitors the spatial location of users based on the orientation of body segments and lower limb movements. Although, the experimental results show an accuracy of up to 99%, the suitability of the approach is arguable due to use of multiple sensors that may cause discomfort. Moreover, standalone PDR systems often accumulate error over time due to sensor drift. Their use is, therefore, combined with contextual information or low-cost beacons that facilitate recalibration as shown in [13].

B. *Ambient Assisted Living*

With improvements in the average life-expectancy of people worldwide, there has been a simultaneous increase in the number of people suffering from cognitive disabilities, such as Dementia, that appear with age. Dementia is a progressive disorder that deteriorates the memory, comprehension and behavior of an individual. The most common cause of Dementia is the Alzheimer's disease that occurs owing to the death of nerve cells and loss of brain tissue. It has a severe impact on the short-term memory, orientation and mobility of the patient, increasing the risks associated with wandering [14]. This urges the development of smart solutions to monitor the health and activities of the patients, and provide timely care. AAL proposes the use of ICT to assist people, especially the elderly, with daily activities and mobility to allow independent living and ensure their well-being. An activity recognition and assessment technique using the smart home technology has been discussed in [15]. The system proposes dense sensor deployment inside the apartment to monitor user interaction with objects of interest. Machine learning is performed on the sequence of sensor events to classify the daily activities such as cooking, cleaning, eating and telephone use. Furthermore, the authors propose a method to develop generalized models corresponding to each activity that abstract over different application scenarios and residents [16]. More recently, activity trackers have replaced the use of static sensors to personalize care and improve behavior analysis for the individuals. In [17], wearable devices consisting of environmental and inertial sensors have been designed to continuously monitor health status and mobility of the elderly. The system combines GPS and BLE technologies to assist in outdoor and indoor mobility respectively. An outdoor navigation system that facilitates independent visits to the exhibition for the cognitively impaired has been discussed in [18]. The approach aims at social inclusion of the individuals under the umbrella of AAL. Although, the solutions perform reasonably well, their implementation is challenging due to the high operational costs. To ensure validity and usability of AAL solutions, five evaluation metrics including accuracy, availability, installation complexity and user acceptance have been outlined in [19].

C. *Sensor Analytics*

With advances in the IoT, there has been an immense improvement in the design and computational capability of sensor devices that constitute WSN. Traditionally constrained to sense and send, the tasks assigned to these devices nowadays incorporate an analytic component. WSN-based localization, for instance, is a form of sensor analytics that has been implemented to improve the context of sensor data. Other approaches such as Data Fusion [20] and Edge Mining [7] utilize the on-board sensor resources for reducing data redundancy within the network with an aim to improve the quality of data exchange. Reduced packet transmissions to the cloud gateway, in turn, improves the energy profile of the network. Furthermore, mapping of Artificial Neural Networks (ANN) on top of the existing WSN hardware has been proposed to facilitate classification and prediction tasks within the network [21]. We base our localization approach on the Edge Mining algorithms that inherent a certain degree of intelligence and allow real-time event detection on the sensor devices as discussed below.

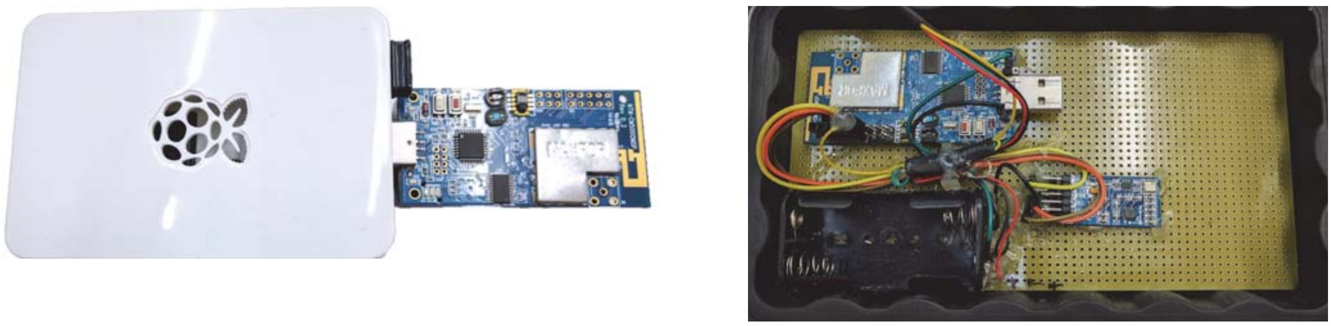
1) *Edge Mining*

The aim of Edge Mining is to improve the energy efficiency of WSN by reducing data communication to the cloud gateway or sink node. Accordingly, it suggests the implementation of light-weight data mining tasks on the sensor devices for localized data reduction. Edge Mining has been realized using the Spanish Inquisition Protocol (SIP), described in [22]. SIP proposes the use of a shared approximation model between the sensor devices and sink node to locally predict the expected application state at sink based on the past estimates. A packet containing the new state value is transmitted only if the new state differs from the estimated value by more than a threshold. Three instantiations of general-SIP have been used for the design of Edge Mining algorithms, namely Linear-SIP (L-SIP), ClassAct and Bare Necessities (BN), as presented in [7]. The algorithms differ based on the representation of application states. L-SIP encodes the state as a point-in-time value and rate of change. The state value is calculated at the sensor node per sensing cycle and compared to the estimated value at sink node. An event is generated if the difference between the two exceeds a user-specified threshold. ClassAct is a decision tree-based activity classifier that models the state value as a smoothed probability distribution over a given set of activities [23]. The state is simplified to the index of the most probable activity and transmitted to the sink node if it varies from the previous estimate. The state recognition, however, relies on a fixed set of probabilistic moments and may not distinguish signals with different distributions but same feature values. The BN approach is primarily designed for applications that only require the summary of data over time [24]. It represents the state as a distribution across non-overlapping bins, where each bin corresponds to a range of value the variable can take, and generates events based on changes in the bin distributions. The ClassAct and BN algorithms discard majority of the raw data and significantly reduce packet transmissions to sink. The two approaches are, therefore, preferred over L-SIP for applications that do not require the reconstruction of the original signal.

III. Proposed Solution

In this work, we consider the challenge of mobility monitoring and outdoor positioning for the elderly suffering from Alzheimer's to detect behavioral anomalies and alleviate the risk of wandering. Although numerous solutions have addressed the issue of outdoor localization in the past, the technologies proposed present several implementation challenges. For instance, use of expensive GPS modules for each user is impractical due to significant operational costs. Alternatively, installation of distributed systems for pervasive computing is cumbersome and labor intensive. Considering the evaluation metrics discussed in [19], we propose a low-cost WSN-based solution for mobility tracking and user localization. Our system comprises of only two nodes - a wearable device and cloud gateway, and relies on self-measurements rather than the range-based techniques, thereby, ensuring ease of deployment. The wearable device is designed to gather IMU data and performs on-board data processing using IEM for real-time activity recognition as the user moves around the environment. Given the topology information, the user location is estimated using the mobility model generated by the algorithm after short intervals of time. The above analysis is performed autonomously on the sensor device without interaction with external objects. Alerts are signaled at the occurrence of unexpected events such as detection of mobility patterns corresponding to wandering behavior. Furthermore, a delay-tolerant communication framework is used to transmit results of the analysis to the cloud gateway. Cloud-based analysis facilitates the implementation of complex learning techniques to modify input parameters, performance metrics and user information for on-board analysis, if necessary, to tune the performance according to the application requirements. For instance, while some applications may only require coarser information such as user presence in specific zones, others may require a more precise location as in case of fall detection to facilitate immediate care. The updated model is, in turn, transmitted to the user device in a delay-tolerant manner, thereby, eliminating the need for continuous Internet connectivity.

Fig. 1 illustrates the design of our prototype devices - wearable activity tracker and cloud gateway node. The main component of the wearable is CM5000 [25] mote that consists of a MSP430 processor and CC2420 radio module. A 10 degrees of freedom IMU consisting of the MPU6050 IC [26], is connected externally to the CM5000 board to measure acceleration and orientation of the user. The components are soldered together on a PCB and placed inside a pelican casing. The device is powered up using 2 AA batteries. A light-weight TinyOS [27] program runs on the device for periodic data collection and analysis using IEM. While our system runs autonomously, it assumes prior knowledge of user-specific



(a) (b)

Fig. 1. (a) Cloud gateway node (b) Wearable activity tracker

information, such as the average pace and normal activity levels, and topology of the application environment using initial supervised learning. The distance travelled by the person is accordingly calculated using the time series data generated by IEM and the average speed of the person. Given the topology and gyro data, user can then be localized within the environment using the displacement measure. Alerts are signaled upon identification of significant deviations from the normal behavior. The results of analysis are stored locally in the flash memory of the device as the user moves around and transmitted to the cloud gateway, hosted indoors, once the user is in its vicinity. The gateway node (fig. 1(b)) consists of a CM5000 mote connected to a Raspberry Pi 2B [28] module. The data is, in turn, uploaded on the cloud for further learning using a Wi-Fi module connected to the Pi. Once the learning is complete, the updated parameters are transmitted back to the wearable device in a similar manner. Our system design, thus, ensures the autonomy of user mobility while providing timely interventions when required.

A. Iterative Edge Mining

A key component of our device based analytics is activity state recognition that is performed by the IEM [8] algorithm. IEM is a fog-centric, sensor analytics approach that is implemented by superimposing two Edge Mining algorithms - BN and ClassAct, on a single node. IEM reads the raw acceleration values from the IMU and converts it into a smooth signal distribution using BN, per sensing cycle. An intermediate event is detected if the change in any bin distribution is significant. The sequence of BN events is fed as input to the ClassAct algorithm that determines the user activity state. The interaction between the two algorithms is controlled by the value of three input parameters - decay factor (γ), error threshold (ε) and heartbeat ($t_{heartbeat}$) that are determined using cloud-based analysis. The γ parameter ranges between 0-1 and is introduced to smooth the signal distributions on the assumption that the application state does not change abruptly. A higher value of γ, increases the weight of past estimates and reduces the fluctuations in the distribution. The resultant smoothing leads to fewer BN events and classification checks. This improves the energy efficiency and, in turn, the lifetime of the device. The reduced frequency of classification, however, also results in an increase in misclassifications and latency in detecting activity changes. The value of ε is determined based on the user-specified accuracy requirements. While the ε value does not affect the nature of signal, it sets the percentage change in bin distributions from previous estimates that is considered significant for classification. A higher value of ε ignores the small fluctuations in the bin distributions, leading to reduced BN events. Accordingly, higher values of both γ and ε are preferred to optimize the resource utilization on sensor nodes when the accuracy requirements are not rigid. A heartbeat mechanism using a parameter $t_{heartbeat}$ can be used to set the maximum time difference between two consecutive BN events and ensure periodicity of classification checks, especially in case of large decay and threshold values, if required. In [8], IEM has been proposed for activity monitoring and behavior analysis in the context of Precision Dairy Farming. The performance evaluation shows the effect of input parameters on classification accuracy and number of BN events for different mobility patterns. Once the activity state for a BN event is determined, the distance traveled since the previous event is calculated for walking activities using the average pace of the person. The user displacement is estimated with the help of gyroscope data, and the updated state and location is recorded in the flash memory of the device. Fig. 2 shows the state diagram for the on-based analytics on the wearable devices.

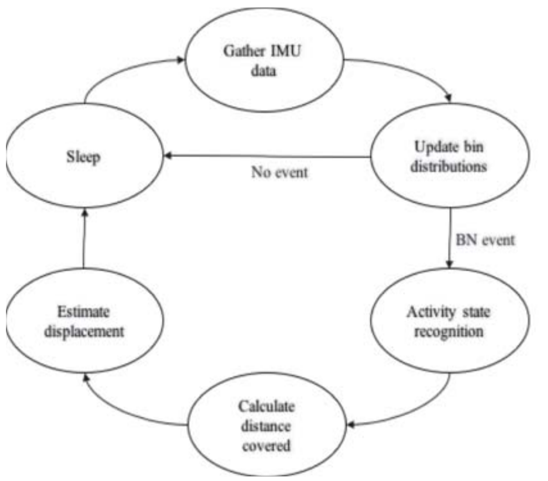


Fig. 2. State diagram for on-board analysis on wearable devices

IV. EVALUATION

We evaluate the performance of IEM in terms of accuracy of classification, cumulative error in distance calculation and reduction in classification frequency (BN events) using acceleration data collected outside our laboratory using the wearable device shown in fig 1(a). The device was hand-held in front of the body and data was collected at a frequency of 10Hz by a single user for a duration of 16 minutes by alternating between walk and stand activities every four minutes. The experiment was repeated for a total of 12 times and the distance covered in each run was 600m (300m*2). We correct the raw data collected during the experiments by removing the offset along each axis and calculate the net acceleration (square root of the sum of squares of each component) to use as input for our algorithm. Of the 12 data sets collected, we reserve 6 files for training the classifier and the other 6 files for testing the model using analysis in R. The mobility traces in training sets are used to calculate the average pace of the user through supervised learning. Moreover, the training data is used to understand the distribution of acceleration values to define bins for the BN algorithm. Fig 3. shows the density distribution of walk and stand activities for one training set. As is evident, there is a significant overlap between walk and stand data that may lead to inaccuracies in the classifier. It is, therefore, imperative to carefully define the bins such that the classification errors and latency in detecting activity state changes is minimized. Since the distribution of stand values is narrow, we use the 68-95-99.7 rule for normal curves and define three bins based on the mean and standard deviation over stand data.

As discussed in section III, while the error threshold ε only regulates the interaction between BN and ClassAct, the decay factor γ also influences the nature of signal distribution. Fig 4. illustrates the smoothing phenomenon, for the same training set as above, across different values of γ. Although the parameter value depends on decay half-life [24], we have chosen three random values to show the change in distributions for a wider range of γ. As expected, the effect of the previous distributions on the current estimate is negligible for very small value of γ (fig. 4(a)), resulting into coarse bin distributions. The smoothness of the distributions increases for higher γ values (fig. 4(b) and 4(c)) due to small changes in bin distributions per sensing cycle. The extent of smoothing affects the frequency of BN events and, in turn, the localization accuracy. We consider the performance of IEM for different γ and ε pairs. We build C5.0 decision-tree classifiers using all data instances from the 6 training files (i.e. ε=0) for five different values of γ. The performance of each classifier is tested with the remaining 6 files using the respective γ values paired with three different ε values. The mean classification accuracy of IEM for walk and stand activities across all test files, for different parameter values, is presented in table I. IEM achieves high accuracy for all γ and ε pairs. The values illustrate the expected drop in accuracy with increased smoothing in the signal distributions. Moreover, while the accuracy is same across all ε values for small γ, it decreases slightly with increase in ε for higher values of γ. This is because the frequency of classification for the former is primarily governed by γ as even the slightest changes in the bin distributions are detected as BN events. An increase in γ value, however, increases the smoothness of the curve and relies on the ε value to capture the significant changes. The effect of γ and ε on the accuracy of distance calculation and classification frequency is shown in fig 5. Instead of calculating the distance travelled for short intervals of walk, we estimate the total distance covered over 8 minutes (4+4mins) using the average pace. The cumulative error over a stretch 600m is shown in fig. 5(a). Our approach performs reasonably well for all different parameter values with the error ranging from 0.4-1%. The error in estimate increases with increase in ε due to latency in detecting state changes and reduced periodicity of distance calculation. Although, the classification accuracy decreases with increasing γ value, a consequent increase in the cumulative error is not recorded. This is because the error is calculated based on the relative time spent in each state which may be identical to the raw data despite the misclassifications. Fig. 5(b) displays the reduction in BN events across different γ and ε values. While the reduction is insignificant for small parameter values, it increases considerably for higher values of both γ and ε as expected. A reduction of 95% is achieved for a γ and ε value of (0.95, 0.7). The values of input parameters can, thus, be chosen to balance the trade-off between accuracy and energy utilization on the sensor nodes according to the user-specified requirements. A heartbeat mechanism can be implemented to ensure periodicity of updates at large γ and ε values, if required.

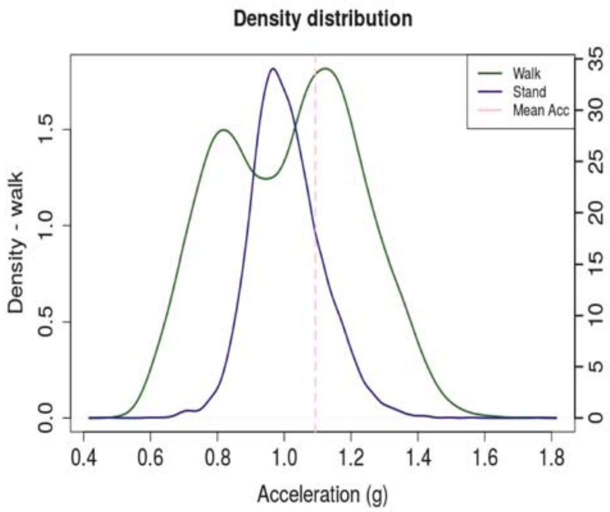
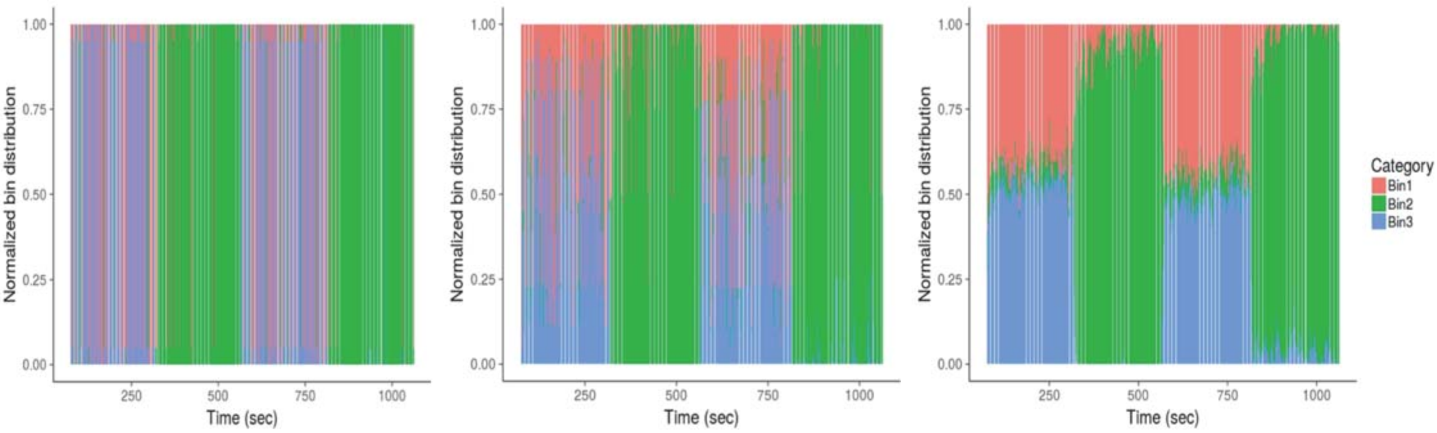


Fig. 3. Density distribution for walk and stand acceleration values

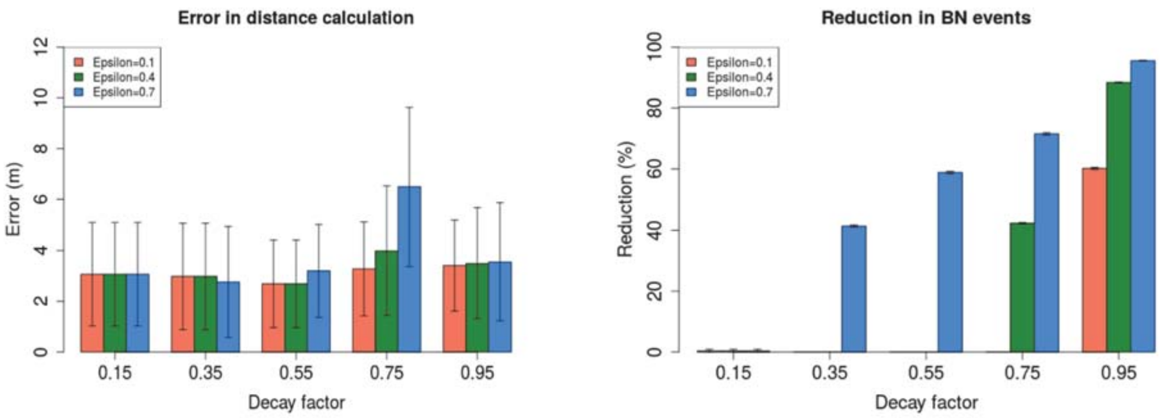
TABLE I. CLASSIFICATION ACCURACY (%)

Error Threshold (ε)	Decay Factor (γ)				
	0.15	0.35	0.55	0.75	0.95
0.1	99.36	99.31	99.17	99.12	99.01
0.4	99.36	99.31	99.17	98.77	98.75
0.7	99.36	99.19	98.75	98.59	97.95



(a) (b) (c)

Fig. 4. Smoothing effect of γ on the signal distributions (a) $\gamma = 0.05$ (b) $\gamma = 0.50$ (c) $\gamma = 0.95$



(a) (b)

Fig. 5. Variation in cumulative error and BN events with 95% confidence interval

V. CONCLUSIONS

In this paper, we present the design of our low-cost WSN system for mobility monitoring and outdoor localization of Alzheimer's patients. Our system consists of an activity tracker and gateway node, and relies on self-tracking and sensor based analytics to perform autonomous, real-time localization as the user moves around an application environment. We discuss our on-board analytics approach along with the IEM algorithm that is used for on-board activity recognition. The mobility traces generated by IEM are used to calculate the distance traveled over short intervals of time to localize the user within the given topology. Moreover, the activity sequence helps in understanding the mobility pattern of the user and enables detection of behavior anomalies to mitigate the risk of wandering. The performance of IEM has been evaluated in terms of accuracy of classification of stand and walk activities, cumulative error in distance calculations and reduction in localization frequency. The results show a classification accuracy above 97.9% and cumulative error percent between 0.4-1 across different values of the input parameters. Although the reduction in localization frequency is negligible for small values of input parameters, reduction up to 95% has been recorded. Fewer calculations can significantly improve the energy profile of sensor devices, especially for a large set of application states. In future, we plan to evaluate the performance of IEM for different mobility patterns and indoor applications. We will also look at how the alerts can be generated and transmitted to the caregivers in case the patients diverge from their normal routes or wander too far.

ACKNOWLEDGMENT

This work has received support from the Science Foundation Ireland (SFI) and the Agriculture and Food Development

Authority, Ireland (TEAGASC) as part of the SFI TEAGASC Future Agri-Food Partnership, in a project (13/IA/1977) titled "Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems". The work has also been promoted under the project CARELINK, AAL-CALL-2016-049 funded by AAL JP, and co-funded by the European Commission and National Funding Authorities of Ireland, Belgium, Portugal and Switzerland.

References

[1] Alzheimer's disease, Alzheimer's Society United Against Dementia, Available at www.alzheimers.org.uk, Last accessed in Apr. 2017.

[2] D. Monekosso, F. Florez-Revuelta, P. Remagnino, "Ambient Assisted Living [Guest editors' introduction]", IEEE Intelligent Systems, vol. 30, no. 4, pp. 2-6, Jul.-Aug. 2015.

[3] J. Wan, C. A. Byrne, M. J. O'Grady, G. M. P. O'Hare, "Managing Wandering Risk in People With Dementia", IEEE Transactions on Human-Machine Systems, vol. 45, no. 6, pp. 819-823, Dec. 2015.

[4] F. Bonomi , R. Milito , J. Zhu, S. Addepalli, "Fog computing and its role in the internet of things", Proceedings of the first edition of the MCC workshop on Mobile cloud computing, Aug. 2012.

[5] Y. Xu, V. Mahendran, S. Radhakrishnan, "Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery", 8th International Conference on Communication Systems and Networks (COMSNETS), Bangalore, pp. 1-6, Jan. 2016.

[6] S. Dey, A. Mukherjee, H. S. Paul, A. Pal, "Challenges of Using Edge Devices in IoT Computation Grids", International Conference on Parallel and Distributed Systems (ICPADS), Seoul, pp. 564-569, Dec. 2013.

[7] E.I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, R. Rednic, "Edge Mining the Internet of Things", IEEE Sensors Journal, vol. 13, no.10, pp. 3816-3825, Oct. 2013.

[8] K. Bhargava, S. Ivanov, C. Kulatunga, W. Donnelly, "Fog-enabled WSN system for animal behavior analysis in precision dairy", International Conference on Computing, Networking and Communications (ICNC), Santa Clara, CA, pp. 504-510, Jan. 2017.

[9] H. Hlavacs, K. A. Hummel, "Cooperative Positioning when Using Local Position Information: Theoretical Framework and Error Analysis", IEEE Transactions on Mobile Computing, vol. 12, no. 10, pp. 2091-2104, Oct. 2013.

[10] V. Moreno, M. A. Zamora, A. F. Skarmeta, "A Low-Cost Indoor Localization System for Energy Sustainability in Smart Buildings", IEEE Sensors Journal, vol. 16, no. 9, pp. 3246-3262, May 2016.

[11] A. Maddumabandara, H. Leung, M. Liu, "Experimental Evaluation of Indoor Localization Using Wireless Sensor Networks", IEEE Sensors Journal, vol. 15, no. 9, pp. 5228-5237, Sep. 2015.

[12] Q. Yuan, I. M. Chen, "3-D Localization of Human Based on an Inertial Capture System", IEEE Transactions on Robotics, vol. 29, no. 3, pp. 806-812, Jun. 2013.

[13] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, "Activity Sequence-Based Indoor Pedestrian Localization Using Smartphones", in IEEE Transactions on Human-Machine Systems, vol. 45, no. 5, pp. 562-574, Oct. 2015.

[14] E. Batista, F. Borras, F. Casino and A. Solanas, "A study on the detection of wandering patterns in human trajectories", 6th International Conference on Information, Intelligence, Systems and Applications (IISA), Corfu, 2015, pp. 1-6.

[15] D. Cook and M. Schmitter-Edgecombe, "Assessing the quality of activities in a smart environment", Methods of Information in Medicine, 2009.

[16] D. Cook, "Learning Setting-Generalized Activity Models for Smart Spaces", IEEE Intelligent Systems, vol. 27, no. 1, pp. 32-38, Jan.-Feb. 2012.

[17] L. Mainetti, L. Patrono, A. Secco, I. Sergi, "An IoT-aware AAL system for elderly people", International Multidisciplinary Conference on Computer and Energy Science (SpliTech), Split, pp. 1-6, Jul. 2016.

[18] F. Franchi, F. Graziosi, C. Rinaldi, F. Tarquini, "AAL solutions toward cultural heritage enjoyment", 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), Valencia, pp. 1-6, Sep. 2016.

[19] P. Barsocchi, S. Chessa, F. Furfari, F. Potortì, "Evaluating Ambient Assisted Living Solutions: The Localization Competition", IEEE Pervasive Computing, vol. 12, no. 4, pp. 72-79, Oct.-Dec. 2013.

[20] P. H. Tsai, Y. J. Lin, Y. Z. Ou, E. T. H. Chu, J. W. S. Liu, "A Framework for Fusion of Human Sensor and Physical Sensor Data", IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 44, no. 9, pp. 1248-1261, Sep. 2014.

[21] G. Serpen, J. Li, L. Liu, Z. Gao, "WSN-ANN: Parallel and Distributed Neurocomputing with Wireless Sensor Networks", The 2013 International Joint Conference on Neural Networks (IJCNN), pp. 1-8, Aug. 2013.

[22] D. Goldsmith, J. Brusey, "The spanish inquisition protocol: Model based transmission reduction for wireless sensor networks", Proceedings of IEEE Sensors 2010, pp. 2043-2048, Nov. 2010.

[23] R. Rednic, E. Gaura, J. Brusey, "Classact: Accelerometer-based real time activity classifier", Proceedings of 2nd WiSIG Showcase, Teddington, UK: Sensors KTN, pp. 2125, Jul. 2009.

[24] E. I. Gaura, J. Brusey, R. Wilkins, "Bare necessities-knowledge driven WSN design", Proceedings of 10th IEEE Sensors, pp. 6670, Oct. 2011.

[25] Advanticsys, MTM-CM5000-MSP, Available at http://www.advanticsys.com/shop/mtmcm5000msp-p-14.html, Last accessed in May 2016.

[26] Invensense, MPU-6000 and MPU-6050 Product Specification Revision 3.4, Invensense Inc., Aug. 2013.

[27] TinyOS Documentation Wiki, Available at http://tinyos.stanford.edu/tinyos-wiki/index.php, Last accessed in Nov. 2015.

[28] Raspberry Pi Foundation, RASPBERRY PI 2 MODEL B, Available at www.raspberrypi.org, Last accessed in May 2016.

Appendix G

A Fog Computing Approach for Localization in WSN

Conference Title:	28th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC 2017)
Article Type	Regular Paper
Complete Author List	Kriti Bhargava and Stepan Ivanov
Status	Published: Oct. 2017

A Fog Computing Approach for Localization in WSN

Kriti Bhargava, Stepan Ivanov
Telecommunications Software & Systems Group
Waterford Institute of Technology, Waterford, Ireland
Email: (kbhargava, sivanov)@tssg.org

***Abstract*—The Fog Computing paradigm proposes an extension of the cloud-based computing to the network edges in the Internet of Things. It facilitates localized analysis closer to the data sources for improved responsiveness of the system as well as cloud-based learning for historical analysis. In this paper, we present our fog-enabled Wireless Sensor Network (WSN) system for activity monitoring and localization in the context of Ambient Assisted Living. Our WSN architecture consists of two types of devices - a wearable sensor device and a cloud gateway node. We discuss our Edge Mining approach for real-time activity classification on the sensor device as well as the Genetic Algorithm used for cloud-based analysis. The design of our analytical framework together with the communication model addresses the challenge of sensor-cloud integration. We evaluate the performance of our system for outdoor localization of the elderly. The analysis is based on acceleration data collected using our wearable device across different activity sequences obtained from the Kasteren dataset.**

I. Introduction

In the past few years, there has been a dramatic improvement in the design of devices that sit at the edge of the Internet of Things (IoT). The devices, today, are equipped with sensor, actuator and communication modules, and posses sufficient computational capabilities to perform certain data analytics apart from the routine data collection and transmission tasks. A new networking paradigm called Fog Computing [1] considers the selective migration of intelligence from the cloud to the edge devices with a view to locally reduce data at the sources. Unlike Edge Computing, the fog works in conjunction with the cloud and addresses the distribution of compute, storage and networking tasks along the thing-cloud continuum. It allows better utilization of the scarce sensor resources by reducing packet transmissions to the cloud, along with a reduction in the cloud storage requirements. Fog nodes have been realized using network components such as switches, gateways or end-user terminals that can lend free computation slots [2][3]. More recently, the role of sensor devices, that constitute the Wireless Sensor Networks (WSN), has been explored for the implementation of data analytic tasks (or sensor analytics). WSN-based localization is a form of sensor analytics that has been proposed to improve the relevance of sensor data. Associating the WSN data with its location facilitates the design of customised context-aware services for better user experience. Localization of the sensor nodes has proven beneficial for various applications in navigation, inventory management, emergency search and rescue, healthcare and Ambient Assisted Living (AAL). AAL suggests the use of Information and Communication Technologies (ICT) solutions to assist people, especially the elderly, with mobility and activities of daily living, and has gained importance over the years in order to improve the quality of life [4]. Activity monitoring and localization of the user are two important factors of AAL systems to understand the individual behaviour and preferences. While numerous approaches have been proposed to locate an individual in a given environment, the deployment of these systems is constrained by the high operational cost incurred due to dependency on expensive hardware such as Global Positioning System (GPS) units or use of dense sensor networks and cloud infrastructure for data sensing and analysis.

978-1-5386-3531-5/17/$31.00

Edge Mining [5] is another dimension of sensor intelligence that suggests the transformation of raw data into contextually relevant information by performing light-weight data mining tasks at the sensor nodes. The localized data processing not only improves the energy efficiency of the network through reduced packet transmissions but also realizes real-time event detection. In this paper, we implement node localization in WSN using our Edge Mining based approach, Iterative Edge Mining (IEM) [6], in the context of AAL for the elderly. We present the design of our WSN system that is based on the Fog Computing architecture and discuss the analytical framework used for localization along with the communication paradigm adopted for sensor-cloud integration. Our system comprises of two nodes - a wearable sensor device and a cloud gateway. IEM is implemented on the wearable device for real-time activity recognition using a decision-tree classifier as a user moves around a given environment. An event is detected if the activity state differs from the previous estimate, and is stored locally in the flash memory of the device. The mobility trace generated by IEM is, in turn, used along with the gyroscope data to estimate the user location in a given topology. Alerts can be generated locally in case the user deviates from normal behaviour or at the occurrence of unexpected events such as a fall. While the data processing is performed autonomously on the sensor device, the results of analysis are transmitted to a cloud gateway, for resource-intensive network learning, in a delay tolerant manner. Cloud-based learning is performed using a Genetic Algorithm (GA) to find the optimal set of values for the input parameters of IEM in order to regulate the on-board analysis according to the application requirements. For instance, while certain applications may demand a high localization frequency to achieve sub-meter position accuracy, the performance requirements of the others may be met with fewer calculations. The GA based learning, accordingly, adjusts the parameter values to manage the localization accuracy as well as the frequency of activity classification on the sensor device. Such flexibility in the system intelligence is an attempt to balance the trade-off between accuracy and resource utilization. The results of cloud-based analysis are, in turn, sent to the device via the gateway node in a delay-tolerant manner. The delay-tolerant communication ensures minimal dependency on the external infrastructure, thereby, reducing the operational cost of the system. We evaluate the performance of our WSN system for outdoor localization using data collected by our wearable device across different mobility patterns extracted from the Kasteren dataset [7].

II. Related Work

Several localization approaches have been proposed, to date. The choice of technology is primarily governed by the application environment and the performance requirements. In this section, we present an overview of some of these techniques and discuss the Edge Mining approach that forms the basis of the IEM algorithm.

A. Localization techniques

GPS based systems are well-known for outdoor localization owing to their high-accuracy positioning and easy integrability of GPS modules with the current IoT devices. The performance of these systems, however, degrades rapidly in unfavorable weather conditions, and crowded and indoor areas due to the absence of GPS signal. Local positioning techniques have been used in conjunction with the GPS systems to improve the prediction accuracy by enabling interaction between mobile nodes in the vicinity of each other [8]. However, the GPS units are energy-intensive and may significantly reduce the lifetime of the IoT devices that are typically battery-powered. WSN systems have also been proposed for localization as mentioned above. Anchor-based techniques that rely on static sensors equipped with GPS modules or previously known locations have been used to estimate the location of mobile nodes with unknown locations. The distance between the anchor and mobile nodes is calculated using either range-based measures such as received signal strength, angle of arrival, time of arrival and time difference of arrival or range-free methods such as hop count and, in turn, converted into global position coordinates. The performance evaluation of these approaches is presented in [9]. Although the time based approaches perform better, they pose the challenge of clock synchronization among all nodes. Additionally, the physical measurements suffer from time-varying errors arising from environment noise and interference. Node localization using inertial sensors embedded in modern smartphones and wearable activity monitors has also been proposed in Pedestrian Dead Reckoning (PDR) systems. The approach is based on self-measurements and estimates the location of mobile nodes using past location and displacement measure. Personalized monitoring with PDR systems allows better understanding of user behaviour and preferences. A few implementations of PDR systems have been proposed to date. In [10], authors propose a blind localization approach to estimate position of user smartphones using acoustic signals and in-built inertial sensors in indoor environments. While PDR systems can operate independently with reasonable accuracy, they often accumulate error over time due to sensor drift. Therefore, their use is often combined with topology information or low-cost beacons that facilitate recalibration as discussed in [11] and [12] respectively.

B. Edge Mining

Edge Mining [5] is defined as the data mining that takes place at the network edges or the data sources (sensor nodes) itself in order to convert the raw signal into contextually relevant information. The approach aims at the efficient utilization of scarce energy resources available at the sensor devices through reduced packet transmissions to the cloud. Furthermore, it facilitates timely response to unexpected changes in the sensor data through real-time event detection. Edge Mining algorithms are based on the Spanish Inquisition Protocol (SIP), discussed in [13]. SIP suggests the transformation of raw data into application specific states that are transmitted to a cloud gateway or sink node only if the current state value cannot be predicted at the sink using an approximation model and past estimates. Three instances of the general-SIP model, namely Linear SIP (L-SIP), ClassAct and Bare Necessities (BN), have been considered for the implementation of Edge Mining. The algorithms differ on the basis of the encoding scheme used for state representation. The application state in L-SIP is defined as a pair of current state value and rate of change. The state is smoothed over the past readings and calculated periodically at the sensor node. Additionally, the expected state value at sink is predicted using the local copy of a shared approximation model and previous state estimate. If the difference between the calculated and predicted value differs by more than a specified threshold, a packet containing the new state is transmitted to the sink. ClassAct is a decision-tree based activity classifier. It reads acceleration data from wearable sensor devices and converts it into a probability distribution over a given set of activities. The state is reduced to the index of the most likely activity and an event is generated if the new state differs from the previously stored estimate. ClassAct significantly reduces the data by storing only the activity states. The BN approach further discards the raw data by maintaining only the summary of data over time. It represents the state as a distribution across non-overlapping bins, where each bin corresponds to a possible application state. An event is detected if the distribution of any bin changes by more than a user-specified threshold. As such, BN estimates the relative time spent in each state. Unlike L-SIP, ClassAct and BN are destructive approaches and, therefore, preferred in applications that do not require signal reconstruction.

III. Proposed Solution

In this work, we propose a WSN-based system for localization of the elderly in order to facilitate autonomous mobilization in a given environment, under the umbrella of AAL. The authors in [14], outline five benchmarks, including accuracy, availability, installation complexity and user acceptance, to assess the usability and validity of AAL solutions. Considering the above metrics, we propose a fog-enabled system for mobility tracking and localization. The solution is based on self-tracking and comprises of two kinds of nodes - a wearable sensor device and a cloud gateway node as shown in Fig. 1. Our system relies on an initial learning phase to understand the user behaviour and preferences to facilitate anomaly detection. It also assumes prior knowledge of the user speed that is considered constant for the duration of the experiment (as expected in the older population). The wearable device can either be worn around the wrist or on the waist and consists of MPU-9255 [15], a 9 degrees of freedom (DOF) Inertial Motion Unit (IMU) that includes a 3-DOF accelerometer, gyroscope and magnetometer, connected to a CM5000 mote [16]. On-board analysis of the acceleration data is performed using IEM to identify the activity state as the user moves around an outdoor environment. If a change in the activity state is detected by IEM, the mobility trace is used along with the user speed to calculate the distance travelled over time. Furthermore, the displacement measure is estimated by associating the distance value with gyroscope readings to localize the user in the given topology. Alerts are generated at the detection of behaviour anomalies such as wandering or fall as illustrated. The results of analysis are stored locally on the device until the user comes in vicinity of the gateway node hosted indoors. The gateway node includes a CM5000 mote connected to a Raspberry Pi. Data from the device is transmitted to the gateway using mote-mote communication and further uploaded on the cloud using Wi-Fi signals. Cloud-based learning is performed using a GA in order to optimize the input parameters for IEM, based on the performance requirements. The analysis considers performance in terms of accuracy of activity classification and distance calculation, as well as the reduction in resource utilization. The modified analytical model is sent back to the device via the gateway node in a delay-tolerant manner to improve the on-board prediction. Our system design, thereby, facilitates the integration of sensors with the cloud along the thing-cloud continuum. Moreover, the delay tolerant communication framework eliminates the need for continuous Internet connectivity and ensures the minimal dependency of our system on external infrastructure.

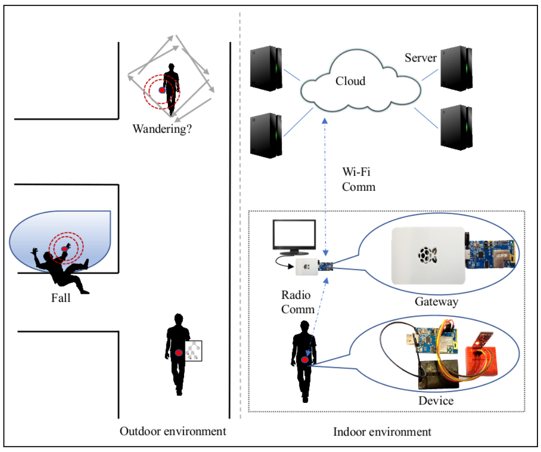


Fig. 1. WSN architecture for user localization

A. *Analytical framework*

Our localization technique depends on device-based activity recognition using IEM as well as the cloud-based network learning via GA. The two algorithms together constitute our analytical framework and are discussed in further detail below.

1) Iterative Edge Mining: IEM forms the fundamental component of our device-based analysis. It is realized by superimposing two Edge Mining algorithms - BN and ClassAct as proposed in [6]. At first, the BN approach is used to convert the raw signal into intermediate states by calculating the distribution of data across different bins. BN events are generated based on the detection of unexpected changes in the signal. Next, the sequence of BN events is fed as input to the ClassAct algorithm for activity recognition. IEM events are generated if the activity state differs from the previous estimate. Unlike ClassAct, the classification in IEM is based on the signal distributions as opposed to a finite set of probabilistic moments represented by feature values. This facilitates the distinction between signals that may follow different distributions but exhibit same feature values. The interaction between BN and ClassAct is governed by three input parameters - decay factor (γ), error threshold (ε) and $t_{heartbeat}$. The decay factor is used for smoothing the bin distributions in BN on the assumption that no sudden changes will occur in the activity state. The value of γ ranges between 0 and 1. A higher value increases the smoothing in the distribution, suggesting a greater impact of the past readings on current values. This reduces the frequency of BN events and, in turn, the classification. While the reduced number of classification improves the resource utilization, it introduces latency in detection of state changes. Unlike γ, the ε parameter does not affect the nature of distribution. It defines the percentage change in bin distributions that is considered significant for the generation of a BN event. A small value of ε results in a higher sensitivity of the algorithm to fluctuations in the bin distributions and causes an increase in the number of classifications. The $t_{heartbeat}$ parameter fixes the maximum time duration between two consecutive BN events and is used to ensure periodicity of classification. The heartbeat mechanism is primarily useful in case of large values of γ and ε. While we propose the implementation of IEM on our prototype device, the algorithm is generic in nature and can be implemented in off-the-shelf activity monitors.

2) Genetic Algorithm: While localization is performed autonomously on the wearable device, GA based network learning is proposed in order to find the optimal set of (γ, ε, $t_{heartbeat}$) values to improve the performance of IEM based localization. GA is the most frequently used class of evolutionary algorithms and is based on the Darwinian theory of survival of the fittest. We prefer the use of GA owing to its ability to generate high quality solutions for a large search space in polynomial time. We consider the implementation of GA on labeled acceleration data and define a fitness function that minimizes the error in distance calculation along with possible reduction in the classification events. Accordingly, our fitness function incorporates the IEM algorithm to build an activity classifier and estimates the values for the above metrics corresponding to each set of (γ, ε, $t_{heartbeat}$). The algorithm randomly generates an initial population of (γ, ε, $t_{heartbeat}$) using the given set of possible values for each parameter. A fitness value is assigned to each member of the population using eq. 1 where differ denotes the difference between actual and estimated value of distance travelled, events is the number of BN events, nR is the total number of raw readings, and error is the percent misclassification. A separate component is used for the misclassification since a poor classification accuracy may still result in accurate localization as our distance calculation is based on the relative time spent in each activity state. A constant of 1 is used for each term to ensure a greater weight is given to the distance term while allowing sufficient impact of the event and accuracy values. Upon calculation of the fitness values, crossover and mutation operators are used on randomly chosen pairs of individuals from the population to generate off-springs that represent different combinations of the parameter values. Fitness values are calculated for the off-springs and a new generation of (γ, ε, $t_{heartbeat}$) is assembled using a selection technique. Fitness based selection allows reduction of the search space in order to converge towards an optimal solution. Moreover, the crossover and mutation probabilities ensure diversity in the population such that the GA traverses the entire search space to reach the global optima. This process is repeated for a specified number of iterations. Once the result from GA is obtained, it is transmitted to the wearable device via the gateway. The performance of the GA is governed by the choice of operators as well as the quality of fitness function. Accordingly, the results of the GA can be modified by adjusting the weights of different components in eq. 1 based on the system requirements.

```
+-----------------------------------+
|         Genetic Algorithm         |
+-----------------------------------+

GA settings:
Type                  =  real-valued
Population size       =  50
Number of generations =  50
Elitism               =  0.02
Crossover probability =  0.8
Mutation probability  =  0.1
Search domain =
     decay threshold heartbeat
Min     0         0        10
Max     1         1       100

GA results:
Iterations             = 50
Fitness function value = -6.030132
Solution =
           decay threshold heartbeat
[1,] 0.4967259 0.7799882  66.77735
```

Fig. 2. Summary of GA

$$fit = -(differ + (events/nR) + error) \quad (1)$$

IV. Evaluation

We evaluate the performance of our system in terms of accuracy of classification and calculated distance, and reduction in events using analysis in R. The evaluation is carried out in two phases - firstly to determine the suitability of GA for historical analysis and secondly to estimate the efficiency of GA-based classifier for the above metrics. We assess the performance of our GA approach using labeled training data. GA is implemented over acceleration data gathered using our wearable device. The device was hand-held in front of the body and data was collected at a frequency of 10Hz for a duration of 16 minutes in an outdoor environment. A single mobility pattern was maintained by the user by alternating between stand and walk activities every four minutes. The experiment was repeated 12 times and a total distance of 600m (2*300m) was covered in each

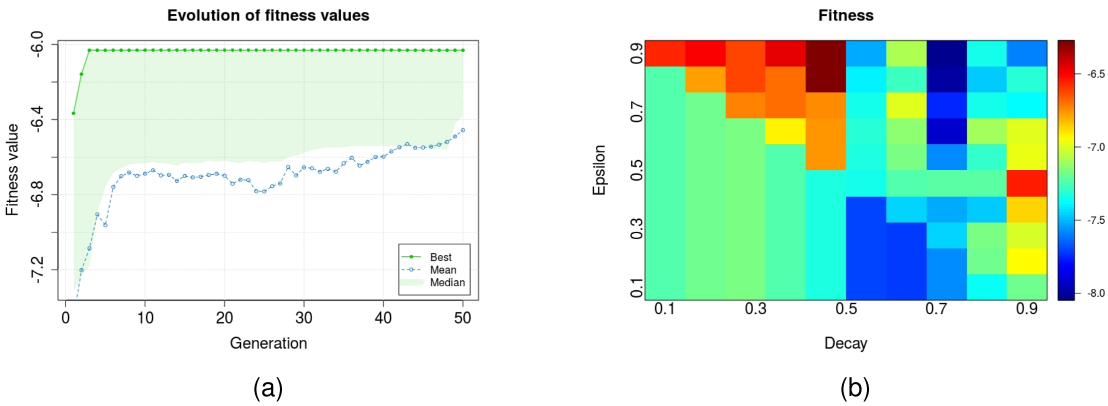


Fig. 3. Performance of GA (a) Evolution of fitness values (b) Image plot for fitness values at constant $t_{heartbeat} = 66.77$

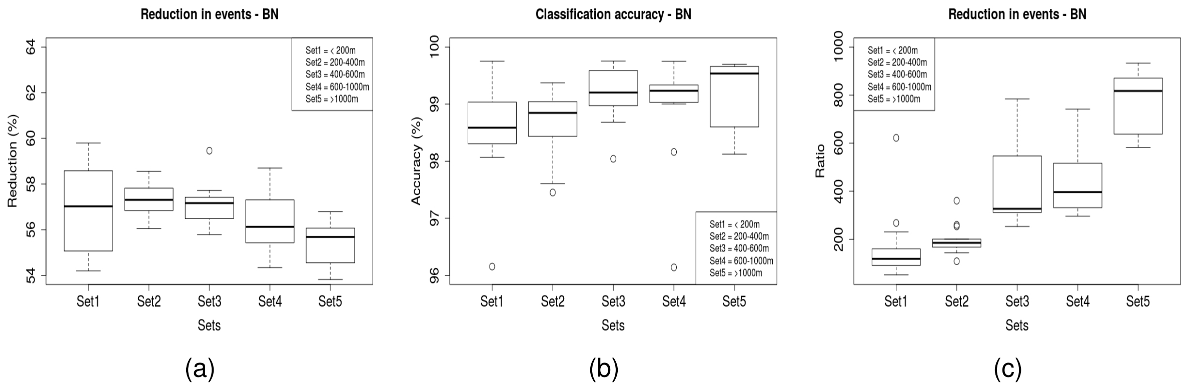


Fig. 4. Performance metrics (a) Reduction in events - BN (b) Classification accuracy of BN events (c) Ratio of BN events

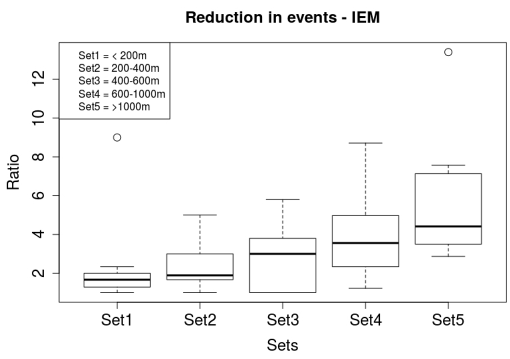
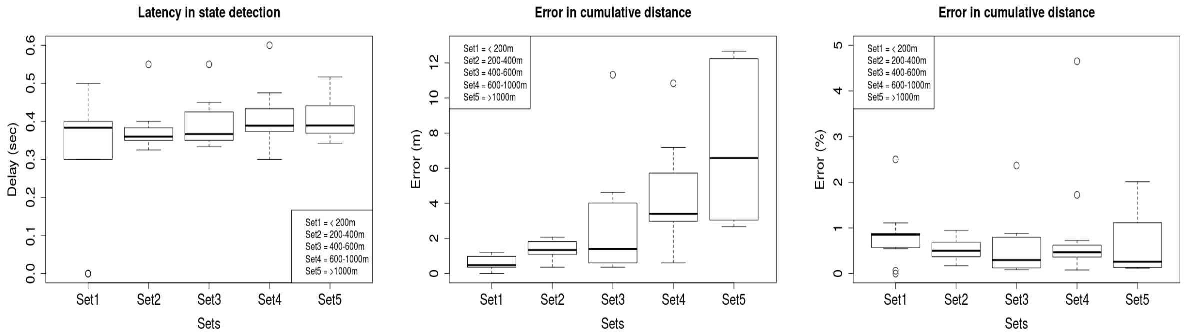


Fig. 5. Ratio of IEM events

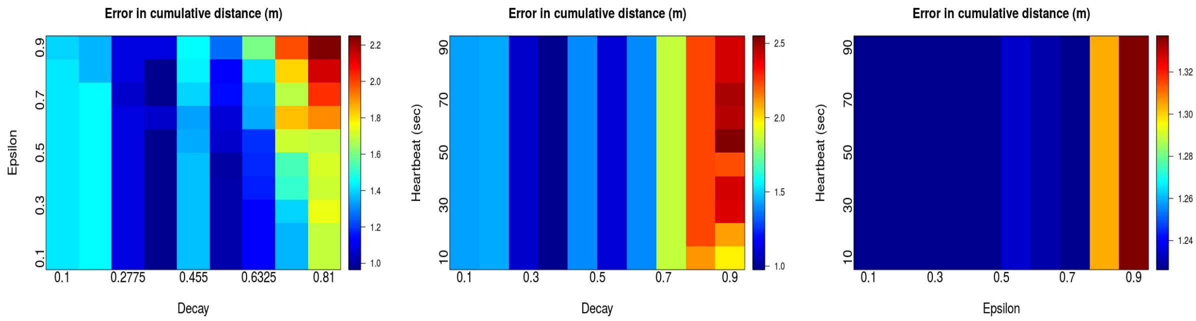
iteration by roughly maintaining a constant speed. The data was corrected to remove the offset along each axis and the distribution of the data was analyzed to define bins for the BN algorithm. We use the Linear Rank Selection, Whole Arithmetic Crossover and Random Mutation operators for the implementation of the GA in R. A summary of the GA settings and results is shown in Fig. 2. An elitism value of 0.02 allows us to keep the best solution across the different generations while the crossover and mutation probabilities ensure diversity in the population. The evolution of fitness value through the generations is illustrated in Fig. 3a. The plot shows the best, mean and median fitness values per generation. It exhibits the speedy convergence of the GA to an optimal value, thereby, suggesting a fast-learning process. Fig. 3b displays the fitness values across different values of γ and ε at constant $t_{heartbeat}$ derived from the GA. The plot confirms that the convergence of the fitness function is not premature and reaches its global maxima. A similar behaviour is observed at constant γ or ε while varying the other 2 parameters.

Whereas the GA is trained using data with planned interactions between walk and stand activities, the mobility pattern of the user could vary each day. Accordingly, we evaluate the performance of localization across different mobility patterns using our GA classifier. For the same, we generate mobility traces by mapping our acceleration data to the activity sequences provided by the Kasteren dataset [7]. The dataset includes the time spent on differ-



(a) (b) (c)

Fig. 6. (a) Latency in state change detection (b) Error in cumulative distance (c) Percent error in cumulative distance



(a) (b) (c)

Fig. 7. Image plots for error in cumulative distance at (a) constant $t_{heartbeat} = 66.77$ (b) constant $\varepsilon = 0.78$ (c) constant $\gamma = 0.49$

ent activities of daily living such as preparing breakfast and dinner, taking a shower, going to bed, getting a drink and leaving the house for work. We consider the sequence of activities as distributions of walk and stand periods, and develop 56 mobility traces with varying intervals of the two modes. We assume the same user speed as training data and calculate the distance covered in each trace. The datasets are then categorized into 5 sets based on the total distance - less than 200m, 200-400m, 400-600m, 600-1000m and greater than 1000m, to evaluate the localization performance for short, medium and long range mobility traces. Fig. 4a and 4b show the average percentage reduction in events achieved by the BN algorithm for all 56 traces and the corresponding classification accuracy respectively. The box plots show the range of values across the files in each set along with the median value. The algorithm achieves >98% accuracy while reducing the number of data points by half. Moreover, the performance is similar across different sets since the metrics are primarily dependent on the values of the input parameters. Although the BN algorithm reduces the memory requirements on-board, it detects events based on small changes in the bin distributions. As a result, the ratio of BN events to actual state changes in the data is rather high as shown in Fig 4c. While the ratio ranges around 200 per state change for smaller data files, it increases up to 900 for Set 5. Given the high classification accuracy of BN events, IEM proposes further reduction in the memory requirements by storing only those instances that correspond to a change in the activity state as opposed to storing all the BN events. The resultant ratio of IEM events to state changes is shown in Fig. 5. As evident, the IEM algorithm significantly reduces the total number of events generated across all five sets. While the median is close to 1 event per state change for smaller files, it increases up to 4 events for Set 5 owing to an increase in the number of misclassification.

Fig. 6a shows the latency incurred by IEM in detecting state changes owing to the smoothing of bin distributions. As the smoothing is controlled by the γ value, the average latency is comparable across the different sets and ranges from 0.3 to 0.5 seconds. This latency, in turn, causes error in the distance calculation and affects the localization accuracy. Fig. 6b illustrates the average value of cumulative error in distance calculation i.e. the error accumulated over the entire trace, across different files. The error is characterized by the speed of the user and averages at 1.33m across all the 56 files. Furthermore, the value increases with an increase in the total distance covered. This is driven by the rise in number of average state changes from 3.77 in Set 1 to 11 in Set 5. The ratio of cumulative error to total distance, however, is similar for both shorter and longer distances as shown in Fig. 6c. The percent error in distance calculation floats between 0-2% for all sets. The results validate the performance of IEM and on-board localization based on our GA classifier across all data files. This suggests that our system performs reasonably well for all mobility patterns despite training the GA using a single activity sequence. The performance of the on-board analysis can be further improved by training the GA classifier using different mobility patterns. The average error in cumulative distance obtained by training and testing the classifier across all the 56 files for different values of input parameters is shown in Fig. 7. The error values in Fig. 7a are calculated for varying γ and ε at constant $t_{heartbeat}$ obtained from our GA. Similarly, the error values in Fig. 7b and 7c are calculated at constant ε and γ respectively, along with the variation in other two parameters. As evident, the error value of our GA classifier is comparable to the values obtained in all three instances. The performance, however, can be improved to reach the optimal value using iterative cloud-based learning for applications that have fairly rigid performance requirements.

V. Conclusion

In this paper, we propose a fog-enabled WSN system to address the localization problem in the context of AAL for the elderly. We present the design of our system and discuss the sensor analytics approach that is used for real-time activity monitoring based localization of the user in a given environment. The localization is performed autonomously on the sensor devices without interaction with any third party components and the results of analysis are communicated to a gateway node in a delay tolerant manner, for cloud-based learning. We describe our GA based learning model that performs the optimization of input parameters used in device-based analysis in order to meet the performance requirements. The reduced dependency of our system on external infrastructure ensures ease of implementation in both indoor and outdoor environments along with a low operational cost. We evaluate our system for outdoor localization using acceleration data gathered by our wearable device. We assess the suitability of our GA for the optimization problem and analyze the performance of the on-board analytics in terms of classification and localization accuracy over different mobility patterns. The results confirm that our system can achieve a high positional accuracy despite a short initial training period and low frequency of data computations on the resource-constrained sensor devices.

Acknowledgment

This work has received support from the Science Foundation Ireland (SFI) and the Agriculture and Food Development Authority, Ireland (TEAGASC) as part of the SFI TEAGASC Future Agri-Food Partnership, in a project (13/IA/1977) titled "Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems".

References

[1] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, *Fog computing and its role in the internet of things*, Proceedings of the first edition of the MCC workshop on Mobile cloud computing, Aug. 2012.

[2] Y. Xu, V. Mahendran, S. Radhakrishnan, *Towards SDN-based fog computing: MQTT broker virtualization for effective and reliable delivery*, 8th International Conference on Communication Systems and Networks (COMSNETS), Bangalore, pp. 1-6, Jan. 2016.

[3] S. Dey, A. Mukherjee, H. S. Paul, A. Pal, *Challenges of Using Edge Devices in IoT Computation Grids*, International Conference on Parallel and Distributed Systems (ICPADS), Seoul, pp. 564-569, Dec. 2013.

[4] D. Monekosso, F. Florez-Revuelta, P. Remagnino, *Ambient Assisted Living [Guest editors' introduction]*, IEEE Intelligent Systems, vol. 30, no. 4, pp. 2-6, Jul.-Aug. 2015.

[5] E.I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, R. Rednic, *Edge Mining the Internet of Things*, IEEE Sensors Journal, vol. 13, no. 10, pp. 3816-3825, Oct. 2013.

[6] K. Bhargava, S. Ivanov, C. Kulatunga, W. Donnelly, *Fog-enabled WSN system for animal behavior analysis in precision dairy*, International Conference on Computing, Networking and Communications (ICNC), Santa Clara, CA, pp. 504-510, Jan. 2017.

[7] T. van Kasteren, A. Noulas, G. Englebienne, B. Kröse, *Accurate activity recognition in a home setting*, 10th international conference on Ubiquitous computing, Seoul, Korea, Sep. 2008.

[8] H. Hlavacs, K. A. Hummel, *Cooperative Positioning when Using Local Position Information:Theoretical Framework and Error Analysis*, IEEE Transactions on Mobile Computing, vol. 12, no. 10, pp. 2091-2104, Oct. 2013.

[9] A. Maddumabandara, H. Leung, M. Liu, *Experimental Evaluation of Indoor Localization Using Wireless Sensor Networks*, IEEE Sensors Journal, vol. 15, no. 9, pp. 5228-5237, Sep. 2015.

[10] D. Ayllón, H. A. Sánchez-Hevia, R. Gil-Pita, M. U. Manso, M. R. Zurera, *Indoor Blind Localization of Smartphones by Means of Sensor Data Fusion*, IEEE Transactions on Instrumentation and Measurement, vol. 65, no. 4, pp. 783-794, Apr. 2016.

[11] B. Zhou, Q. Li, Q. Mao, W. Tu, X. Zhang, *Activity Sequence-Based Indoor Pedestrian Localization Using Smartphones*, in IEEE Transactions on Human-Machine Systems, vol. 45, no. 5, pp. 562-574, Oct. 2015.

[12] Z. Chen, Q. Zhu, Y. C. Soh, *Smartphone Inertial Sensor-Based Indoor Localization and Tracking With iBeacon Corrections*, IEEE Transactions on Industrial Informatics, vol. 12, no. 4, pp. 1540-1549, Aug. 2016.

[13] D. Goldsmith, J. Brusey, *The spanish inquisition protocol: Model based transmission reduction for wireless sensor networks*, Proceedings of IEEE Sensors 2010, pp. 2043-2048, Nov. 2010.

[14] P. Barsocchi, S. Chessa, F. Furfari, F. Potort, *Evaluating Ambient Assisted Living Solutions: The Localization Competition*, IEEE Pervasive Computing, vol. 12, no. 4, pp. 72-79, Oct.-Dec. 2013.

[15] Invensense, MPU-9255 Product Specification, rev. 1.0, InvenSense Inc., Sep. 2014.

[16] Advanticsys, MTM-CM5000-MSP, Available at http://www.advanticsys.com/shop/mtmcm5000msp-p-14.html, Last accessed in May 2016.

Appendix H

Leveraging Fog Analytics for Context-Aware Sensing in Cooperative Wireless Sensor Networks

Journal Title:	ACM Transactions on Sensor Networks
Article Type	Regular
Complete Author List	Kriti Bhargava, Stepan Ivanov, Diarmuid McSweeney and William Donnelly
Status	Published: vol. 15, no. 2, pp. 23:1-23:35, Mar. 2019

Leveraging Fog Analytics for Context-Aware Sensing in Cooperative Wireless Sensor Networks

KRITI BHARGAVA and STEPAN IVANOV, Telecommunications Software & Systems Group, Waterford Institute of Technology, Ireland
DIARMUID MCSWEENEY and WILLIAM DONNELLY, Waterford Institute of Technology, Ireland

In this article, we present a fog computing technique for real-time activity recognition and localization onboard wearable Internet of Things(IoT) devices. Our technique makes joint use of two light-weight analytic methods—Iterative Edge Mining(IEM) and Cooperative Activity Sequence-based Map Matching(CASMM). IEM is a decision-tree classifier that uses acceleration data to estimate the activity state. The sequence of activities generated by IEM is analyzed by the CASMM method for identifying the location. The CASMM method uses cooperation between devices to improve accuracy of classification and then performs map matching to identify the location. We evaluate the performance of our approach for activity recognition and localization of animals. The evaluation is performed using real-world acceleration data of cows collected during a pilot study at a Dairygold-sponsored farm in Kilworth, Ireland. The analysis shows that our approach can achieve a localization accuracy of up to 99%. In addition, we exploit the location-awareness of devices and present an event-driven communication approach to transmit data from the IoT devices to the cloud. The delay-tolerant communication facilitates context-aware sensing and significantly improves energy profile of the devices. Furthermore, an array-based implementation of IEM is discussed, and resource assessment is performed to verify its suitability for device-based implementation.

CCS Concepts: • **Networks → In-network processing**; *Location based services*; • **Computer systems organization → Sensor networks**; *Embedded systems*; • **Information systems →** *Location based services*;

Additional Key Words and Phrases: Fog computing, edge mining, cooperative wireless sensor network, localization, precision farming, testbeds

This work has received support in part from the Science Foundation Ireland (SFI) and the Agriculture and Food Development Authority, Ireland (TEAGASC) under the SFI-TEAGASC Future Agri-Food Partnership, in a project (13/IA/1977) titled "Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems." In addition, this publication has emanated from research supported by a research grant from SFI and the Department of Agriculture, Food and Marine on behalf of the Government of Ireland under grant number [16/RC/3835].

Authors' addresses: K. Bhargava (corresponding author) and S. Ivanov, Telecommunications Software & Systems Group, Waterford Institute of Technology, WIT West Campus, Carriganore, Waterford, Waterford, X91P20H, Ireland; emails: {kbhargava, sivanov}@tssg.org; D. McSweeney and W. Donnelly, Waterford Institute of Technology, Waterford, Ireland; emails: diarmuid.mcsweeney@teagasc.ie, wdonnelly@wit.ie.

1550-4859/2019/03-ART23 $15.00
https://doi.org/10.1145/3306147

ACM Reference format:
Kriti Bhargava, Stepan Ivanov, Diarmuid McSweeney, and William Donnelly. 2019. Leveraging Fog Analytics for Context-Aware Sensing in Cooperative Wireless Sensor Networks. *ACM Trans. Sen. Netw.* 15, 2, Article 23 (March 2019), 35 pages.
https://doi.org/10.1145/3306147

1 INTRODUCTION

With increase in the number of Internet of Things (IoT) applications, localization of IoT devices—such as routers, smartphones, and various wearable technology—has gained significant importance for improving context-awareness and providing Location Based Services (LBS), such as navigation and target tracking [1]. Traditionally, use of Global Positioning Systems (GPS) has been proposed for realization of outdoor LBS. GPS-enabled Ceres tags [2], for instance, have been designed for livestock and farmland management in pursuit of Precision Farming. Besides detecting feeding rates and rumination for individual cows, these ear tags are used for mobility tracking to detect boundary breakouts and alert farmers in case of theft or an ambitious animal. Location awareness can, in turn, be used to control navigation of animals within farms for implementation of virtual fence [3]. While GPS technology is preferred due to ease of integration with IoT devices, majority of modern-day IoT solutions are replacing its use, owing to poor accuracy in bad weather conditions and crowded environments, as well as the energy-hungry nature of GPS receivers that negatively affects the battery-life of the IoT devices. Alternatively, use of Wireless Sensor Networks (WSN) has been proposed for localization [4]. The WSN-based techniques perform triangulation using range-based measures, such as Received Signal Strength (RSS) [5], to estimate the relative distance of mobile nodes from static, anchor nodes for localization. SmartBow [6], for instance, is an ear tag that has been designed to monitor mobility and rumination of dairy cows. The system uses a triangulation algorithm to calculate the x/y/z coordinates of cows with respect to a fixed access point (wallpoint). Although WSN-based techniques are low power when compared to GPS, they require the use of either additional infrastructure deployed on the farm or external cloud resources for data analysis. While the former increases the cost of system deployment and maintenance, the latter requires accessibility to cloud resources, which is typically limited in remote applications, such as in dairy farms. Furthermore, the performance and efficiency of these approaches is often affected by outdoor noise and the need for frequent time synchronization between devices.

Meanwhile, with advances in the design and computational capabilities of IoT edge devices (e.g., smartphones and sensors), localization on-board these devices (using data from built-in inertial sensors) has been suggested under the umbrella of Fog Computing [7]. Fog Computing is a novel paradigm that extends Cloud Computing to the edge of the network and proposes the use of existing compute and networking resources available at the IoT edge devices for real-time data analytics. In doing so, it aims at optimizing resource efficiency of the system while improving responsiveness to alerts through reduced cloud dependency. As such, fog-enabled localization on edge devices can potentially overcome limitations of the WSN-based approaches discussed above. Indoor localization on-board user smartphones, for instance, has been discussed in Reference [8]. The proposed technique detects activity states using inertial data obtained from user smartphones and performs activity-sequence-based map matching (ASMM) using Hidden Markov Model (HMM) to identify special points on the map as the user walks around the given topology. While quite a few smartphone-based localization techniques have been proposed to date (discussed in detail in Section 2), certain IoT applications designed using the WSN technology lack such relatively powerful edge devices. Current animal health monitoring systems in dairy farms, for instance, consist only of low-power animal-wearable sensor devices, such as Moomonitor [9] and HerdInsights [10]. These devices borrow principles of Fog Computing and operate autonomously (without continuous interaction with third-party components, such as gateways/PC or the cloud) to detect small-scale health or behaviour anomalies. These events are stored locally on the collar devices and transmitted to end-users in a delay-tolerant manner. Such devices, however, lack location awareness, owing to inadequate infrastructure in remote farms. To ensure real-time contextualization of sensor data in similar mobility tracking WSN that are deployed in remote applications, there is a need to design novel light-weight localization algorithms suitable for implementation on-board the low-power, resource-constrained sensor devices.

Edge Mining [11] is a Fog Computing approach that proposes implementation of light-weight data mining tasks on sensor devices. The approach aims at improving real-time responsiveness of these devices through on-board detection of application-related events. Furthermore, it improves the energy efficiency of devices through reduced packet transmissions to the cloud. ClassAct, an instance of Edge Mining, has been proposed for sensor-based activity classification. It is a decision-tree-based technique that uses acceleration data from wearable inertial sensors to estimate the user activity state. The activity states can, in turn, be analyzed to determine the location. However, ClassAct bases its prediction on low-order moments, such as windowed mean and variance at fixed time intervals. This limits its use in applications where the acceleration signal comprises of activity states with significant overlap in measurements. As such, while the values may come from different distributions, they exhibit the same feature values and cannot be distinguished from each other. To address this limitation, the authors have previously proposed Iterative Edge Mining (IEM) in Reference [12]. Unlike ClassAct, IEM classifies the activity states based on the histograms of acceleration measurements across multiple bins. It, thus, captures the distribution of signal and is particularly useful in scenarios where the overlap in states is significant and the mixture is imbalanced, i.e., the likelihood of occurrence for a certain activity is significantly higher, compared to the others. The histogram approach, however, incurs additional costs in calculating and maintaining the bins and may affect resource efficiency of the approach.

To overcome this limitation, in this article, we present an extension of the IEM approach, namely IEM2.0. The IEM2.0 algorithm replaces the histograms with Moving Windowed Minimum and Maximum features for analyzing the signal distribution and classification. The adaptation aims at reducing the program size and number of computations for activity classification, while capturing changes in the distribution. In addition, we propose a novel localization technique based on IEM2.0 that is suitable for execution on low-power wearable sensor devices. The technique makes joint use of two light-weight analytic methods—IEM2.0 and Cooperative Activity Sequence-based Map Matching (CASMM). First, the approach performs acceleration-based activity recognition using IEM2.0. The sequence of activities generated by IEM2.0 is then analyzed by the CASMM method to detect the location. CASMM exploits the spatial-temporal coherence of neighboring sensor devices for Cooperative activity-state detection by facilitating exchange of location updates between devices and extends the ASMM approach proposed in Reference [8] to map the resultant sequence of activities to a given topology and determine the location. Furthermore, we exploit the location information of devices and present a context-aware, event-driven communication framework for data transmission to the cloud. The framework is proposed to improve the energy efficiency of the devices by reducing unnecessary periodic transmissions. We illustrate the use of our IEM2.0-CASMM approach for activity recognition and localization of animals in a pasture-based dairy farm. While IEM2.0 is used for classification of high-level activity states of animals, CASMM is used to map the sequence of activities to an outdoor road network and estimate the location. The main contributions of the article can be summarized in the following:

- Adaptation of the IEM approach proposed in Reference [12], namely IEM2.0, for activity classification. IEM2.0 is proposed to reduce the number of on-board computations and improve resource efficiency of devices. It replaces the histogram-based approach with windowed feature analysis to capture the signal distribution while removing unnecessary calculations. The mathematical formulation of IEM2.0 is discussed, and its suitability over ClassAct is demonstrated for naturally occurring mixed Gaussian signals with different mixture proportions.
- Design of an end-to-end WSN system for IEM2.0-CASMM-based context-aware sensing and communication. The system performs activity recognition using IEM2.0 and adapts the existing ASMM technique for cooperative activity sequence-based map matching to allow on-board localization in outdoor environments. We also present theoretical models for calculating communication energy cost incurred by the devices and discuss an event-driven communication framework for optimizing energy consumption of the network.
- An application of our IEM2.0-CASMM approach for high-level activity recognition and localization of animals in a pasture-based dairy farm. An extensive evaluation has been carried out to analyze the accuracy and energy efficiency of our localization approach using real-world animal-mobility data collected during a pilot study in Kilworth, Co. Cork, Ireland. Moreover, a dedicated memory analysis has been carried out to assess the resource requirements of IEM-2.0 to verify its suitability for sensor-based execution.

The remainder of this article is structured as follows: In Section 2, we present the related work. In Section 3, we present our system architecture and discuss the IEM2.0-CASMM-based localization approach. We also describe our context-aware communication framework. In Section 4, we present our case study and the implementation of IEM2.0-CASMM in the context of dairy farming. We also discuss our experimental setup and field study. In Section 5, we present an extensive evaluation of our approach using real animal-mobility data, followed by a resource assessment of IEM-2.0 in Section 6. In Section 7, we conclude the article.

2 RELATED WORK

In this section, we review state-of-the-art IoT-based localization and discuss the recent advances in sensor-based analytics.

2.1 Localization Techniques

Several localization techniques have been proposed, to date, for IoT applications. Traditional IoT-based systems make use of GPS for outdoor localization due to their high accuracy as well as ease of integration of GPS receivers with IoT devices. For instance, GPS units have been used for localization of the elderly for assisted living in Reference [13]. While the approach achieves high accuracy, the system relies on a remote reasoning system for data analysis and may incur delay in getting insights due to the intermittent Internet connectivity. Moreover, the use of GPS receivers coupled with the frequent data transmissions may negatively impact the lifetime of the devices. Alternatively, the use of cellular systems has been proposed for trajectory tracking. In Reference [14], for instance, the system uses cellular technology to estimate the coarse location of mobile devices through signal trilateration. This information is combined with stationary state detection and HMM-based algorithms to decipher the most probable path. The performance of such a system, however, is affected by low sampling frequency and may result in errors ranging to a few kilometers. A digital map-matching system called SnapNet [15] has been proposed to improve the location accuracy of cellular-based systems. The system implements an incremental HMM algorithm to account for the noise in the input data and uses digital map hints to enhance the accuracy of the estimated road segments. The use of such systems, however, is limited to scenarios with reliable cellular networks. In Reference [16], a Wi-Fi-based localization approach has been discussed. The approach uses commodity Wi-Fi (Intel 5300) to estimate the doppler velocity and angle of arrival measures for localization purposes and incurs an error as low as 35cm. The performance of Wi-Fi-based localization systems, however, is usually affected by radio signal noise, making it unsuitable for outdoor environments.

Alternatively, the use of WSN for localization has also been proposed. In Reference [17], for instance, the authors present a light intensity-based indoor positioning system that performs predictions using RSS measures within WSN. Another study in Reference [5] investigates the feasibility of RSS-based sensor node localization in well-defined outdoor topology. Such range-based measures, however, often exhibit a low signal-to-noise ratio, thus affecting the quality of prediction. An experimental evaluation of WSN-based localization has been carried out in Reference [4]. Alternatively, with advances in embedded sensor technology, the use of Pedestrian Dead Reckoning (PDR) systems has been proposed for localization purposes. PDR systems use mobility data (e.g., acceleration,velocity) from built-in inertial sensors in user wearables/smartphones and calculate displacement to get the current location. The authors in Reference [18] present a PDR system that uses 8 Inertial Motion Units (IMU) worn on the body and a force sensor worn under the feet to capture joint movements for user localization. Another instance of a PDR system has been discussed in Reference [19]. The system presents a blind localization algorithm that combines data from built-in inertial and acoustic sensors in user smartphones using a maximum likelihood estimator to gauge the location of the smartphone. Standalone PDR systems, however, often accumulate errors due to drift with walking distance over time. To overcome this issue, assisted-PDR approaches have been proposed. In Reference [20], a PDR system is accompanied by iBeacons and the Kalman-Filter-based calibration algorithm is used to correct the drift. A PDR-based ASMM technique has been proposed for indoor localization in Reference [8]. The system performs low-level activity recognition, such as turning or walking up and down different floors, using built-in inertial sensors in user smartphones as a user walks to special points, such as corners, elevators, escalators, and stairs. The sequence of activities is then used to establish the user's trajectory and, in turn, mapped to an indoor road network for accurate positioning. The ASMM approach presents a cost-effective solution for indoor localization, as it requires minimum interaction with external third-party components.

In this work, we present our IEM2.0-CASMM-based PDR system for real-time localization. The approach takes as input acceleration data from built-in inertial sensors in wearable devices and performs decision-tree-based activity recognition using the IEM2.0 algorithm. As compared to the existing techniques, IEM2.0 is light-weight and suitable for implementation on-board low-cost sensor devices. The sequence of activities generated using IEM2.0 is then analyzed by the CASMM module for localization. CASMM is a cooperative extension of the ASMM approach discussed in Reference [8]. At first, the approach implements cooperative computing via collective participation between co-located devices to improve accuracy of classification on individual devices. Next, if a change in activity state is observed for any device, then ASMM is performed to map the sequence of activities to a given outdoor topology for localization. While HMM is used to implement ASMM in Reference [8], we replace this approach with a light-weight window analysis using a threshold $\mathcal{T}$ to ensure suitability for sensor-based execution. The two techniques are discussed in detail in Section 3.

2.2 Sensor Analytics

With increase in the number of IoT devices, huge amounts of data is periodically created and uploaded on the cloud for analysis. Such data abundance (typically referred to as "big data"), however, burdens the existing cloud resources and causes latency in getting insights into the data. Subsequently, the Fog Computing paradigm has been proposed to shift certain intelligence from the cloud towards the data sources, i.e., the network edge devices [21]. The use of compute and network capabilities available at these devices would allow localized reduction of data within the network to not only optimize the use of existing resources but also improve the responsiveness of the IoT system by reducing dependency on the cloud [22]. As mentioned earlier, while the use of IoT edge devices (e.g., network switches, smartphones) as fog agents has widely been proposed, recent studies have further brought down the computations to sensor devices. Owing to improvements in the computational capabilities of sensor devices conventionally limited to sense and send, the tasks designed for these devices today incorporate certain sophisticated data analytics. For instance, Data Fusion within WSN has been proposed in Reference [23] to reduce redundancy in overlapping data and improve coverage. Another study in Reference [24] suggests the mapping of an Artificial Neural Network (ANN) onto WSN for the design of "Smart Furniture." The authors in Reference [11] propose Edge Mining techniques to perform data mining on-board sensor devices. Edge Mining forms the basis of our activity classification approach, IEM2.0, and is discussed in greater detail below.

Edge Mining [11] is a Fog Computing technique that suggests the implementation of lightweight data mining tasks on sensor devices. It adopts the principles of the *Spanish Inquisition Protocol* (SIP) [25] that proposes transmission of only the unexpected information from the network to a sink (gateway). SIP converts the raw data from sensors into an application relevant state that is considered significant and reported by the sensor only if it cannot be predicted using the past estimates. Three instances of Edge Mining have been discussed based on generalized SIP—Linear SIP (L-SIP), Bare Necessities (BN), and ClassAct. L-SIP defines the application state as the point-in-time value and the rate of change. BN represents the state as a distribution of data across non-overlapping bins where each bin defines a possible outcome [26]. ClassAct is a decision-tree-based classifier. It takes as input raw sensor data and encodes the application state as a probability distribution over a given set of states. The use of ClassAct has been shown for identification of low-level activities, such as sitting, standing, and walking, in Reference [27]. While the system achieves a high classification accuracy, the classification is performed using low-order moments, such as windowed mean and variance at fixed points in time. This approach inevitably leads to classification errors while separating signals (time-variant data reflecting a particular behaviour, such as acceleration while walking and standing) for which measurements have similar mean and variance though come from different distributions. While the use of higher moments (e.g., skewness and kurtosis) may help in identifying the different states, their calculation is computationally complex for the sensor devices.

IEM has been previously proposed by the authors in Reference [12] to overcome the limitation of ClassAct approach. IEM is a decision-tree classifier that is designed as the superimposition of two Edge Mining algorithms—BN and ClassAct. First, IEM runs the BN algorithm to convert raw sensor measurements into a distribution across a set of non-overlapping and exhaustive bins, where each bin represents a range of values that the variable can take. The distribution is smoothed over the past readings using a decay factor γ on account that no sudden changes occur in the activity state. Next, the percentage change in distribution is estimated. If the change exceeds a threshold ε, where $0 < \varepsilon < 1$, the distribution for all bins is fed as input to the ClassAct algorithm for activity-state recognition. By considering the signal distribution as input to the classifier (as opposed to windowed mean and variance), IEM captures the nature of the signal over time and thereby addresses the limitation of ClassAct. The performance of IEM has been evaluated for classifying low-level activities, such as walk and stand in Reference [12]. While IEM is shown to achieve an accuracy of 95% with very low frequency of computations, the histogram-based implementation (inspired

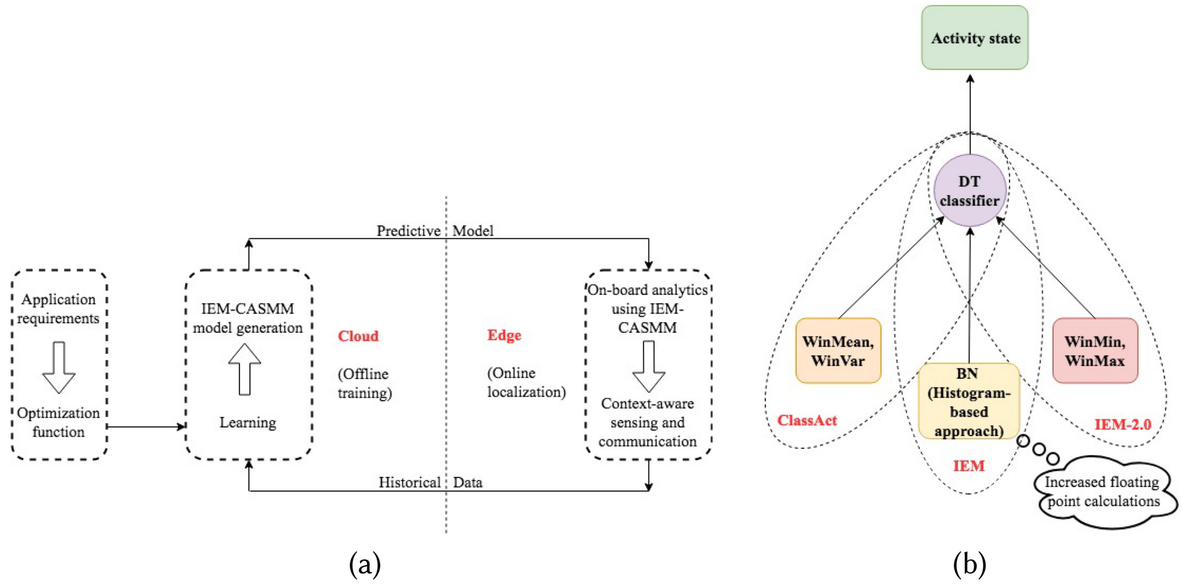


Fig. 1. (a) System architecture (b) ClassAct-, IEM-, and IEM-2.0-based classification.

by the BN algorithm) requires multiple floating-point operations to maintain the bin counts and distribution. Based on the selection of bins, this may negatively affect the resource efficiency of the algorithm for sensor-based execution. In this work, we discuss an adaptation of IEM that is more suited for implementation on sensor devices (Section 3.1) and evaluate its performance for high-level activity recognition for dairy cows. An array-based implementation of the algorithm is also discussed in Section 6 to evaluate the resource requirements.

3 IEM2.0-CASMM FOR ON-BOARD LOCALIZATION

Figure 1(a) illustrates the architecture of our IEM2.0-CASMM-based localization system. As can be seen, the system operates in two phases—offline training phase on the cloud and online localization phase at the edge. While the IEM2.0-CASMM model is light-weight and suitable for sensor-based localization, training the model is a compute-intensive task and is, therefore, carried out offline on the cloud. In the training phase, at first, historical data is collected from in-built inertial sensors in wearable devices and analyzed to extract suitable feature(s) for classification. Then, (un)supervised learning is performed to train and test the IEM2.0 and CASMM models for the given application scenario. IEM-based classifier (*DT*) is generated for different values of input parameters. The *DT* is used to analyze the acceleration data and identify the activity state. The sequence of activities generated by IEM is then analyzed by the CASMM method for map-matching-based localization. CASMM performs cooperative analysis between neighboring devices (considered as a coalition) by allowing exchange of location updates to improve accuracy of individual predictions and maps the updated sequence of activities to a given topology for identifying the location. The performance of CASMM is evaluated for different coalition sizes. Based on the performance evaluation and a given optimization function (e.g., maximizing location accuracy or minimizing energy consumption) that is derived from application requirements, the values for input parameters for IEM2.0-CASMM are fixed (i.e., $windowSize$, ε, and coalition size). The optimal performing model is then transferred onto the sensor devices for on-board analysis. In the online phase, IEM2.0-CASMM is executed to analyze the periodically sensed acceleration data for real-time activity recognition and localization. The estimated location is combined with data from other sensors (such as temperature, humidity) to facilitate context-aware sensing and communication. An instance of this architecture is discussed for localization of dairy cows in Section 4 (depicted in Figure 6). The IEM2.0-CASMM model is suitable for implementation on-board animal-wearable devices and allows for real-time context-aware sensing as the cows move around the farm. We assume prior knowledge of the farm topology for the CASMM module. Furthermore, as CASMM assumes presence of co-located or coherently moving devices for coalition-based cooperation, we consider conventionally milked dairy cows that move together in a herd between parlour and paddocks. Note, however, the CASMM approach extends easily to scenarios where devices move independently: for instance, in case of automatic milking wherein dairy cows may follow different milking cycles, by forming dynamic coalitions on the move (as discussed in Section 3.2). The calculation of overhead while setting up coalitions is beyond the scope of this work. In the remainder of this section, we present a detailed description of the two analytic approaches and our context-aware, event-driven communication framework.

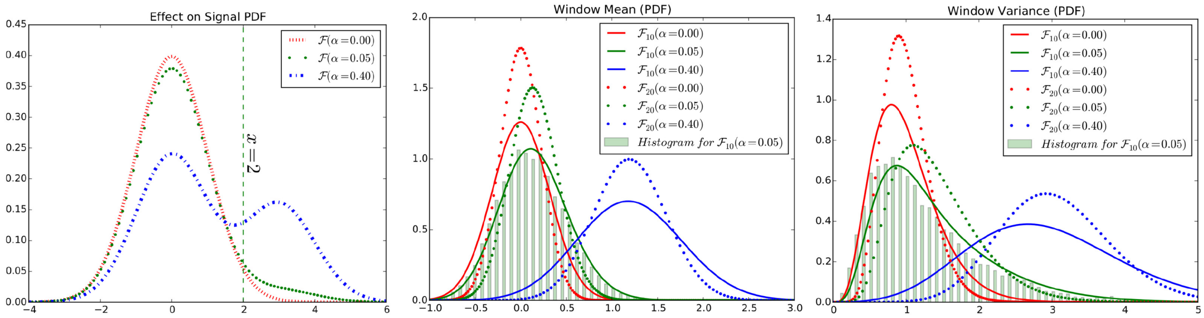
3.1 Iterative Edge Mining (IEM)

IEM2.0 is an adaptation of the IEM approach, which replaces the histogram-based analysis with Windowed Minimum and Maximum ($winMin$, $winMax$) features for activity-state classification. The moving window analysis examines the temporal patterns present within the signal and captures the variability in distribution of values over time. The use of these features ensures sensitivity to minute changes in distribution of sensor measurements while reducing the unnecessary floating-point operations. This, in turn, improves the efficiency of the algorithm, making it suitable for increased range of IoT devices and applications. Here, the window size is an input parameter that accounts for smoothing over the historical data similar to the decay factor γ used in histogram-based IEM (discussed in Section 2.2). Classification is performed only if the percentage change in either of the feature values exceeds the threshold ε, where $0 < \varepsilon < 1$. When it comes to floating-point operations, IEM-2.0 requires only $\geq$ and $<$, as opposed to the floating-point division and multiplication (e.g., histogram estimation, smoothing) that are additional requirements of the previously proposed IEM technique. The difference between ClassAct-, IEM-, and IEM-2.0-based classification is depicted in Figure 1(b). We present the mathematical formulation of IEM2.0 and illustrate its suitability over ClassAct for normal and mixed Gaussian distributions in the next section. We consider these signals owing to the nature of real-world acceleration data collected for different activity states as seen in this study (see Figure 15).

3.1.1 Gaussian Mixtures and Their Impact on ClassAct Classification. Consider signals S_{norm} and S_{mix} for which values are i.i.d. and come from a normal Gaussian distribution $p_{norm}(x) = \mathcal{N}(x, \mu_1, \sigma^2)$ and a two-component Mixed Gaussian distribution, respectively, where x represents sensor measurements. The first component of the mixture follows the same normal distribution as S_{norm}, while the second component follows a normal distribution with the same variance σ^2 but larger expectation $\mu_2 > \mu_1$. The samples x are drawn from the first and second components with probabilities $1 - \alpha$ and α, respectively, where $\alpha < 0.5$ (i.e., dominance of the first component). Accordingly, the distribution of S_{mix} values has the probability density function (PDF) expressed here:

$$p_{mix}(x, \alpha) = (1 - \alpha) \cdot \mathcal{N}(x, \mu_1, \sigma^2) + \alpha \cdot \mathcal{N}(x, \mu_2, \sigma^2). \tag{1}$$

Naturally, both S_{norm} and S_{mix} can be treated as representatives of the same parametric family $\mathcal{F}$ of signals, where values come from distributions with PDF specified by the Equation (1) for different α-values. In a way, α describes the impact of minor component on the overall value distribution. Figure 2(a) illustrates the effect of α on the signal values and their distribution (generated using Equation (1)) for $\mu_1 = 0$, $\mu_2 = 3$, and $\sigma = 1$. As expected, $\mathcal{F}(0.00)$ produces the normal signal S_{norm}. As α increases, the impact becomes more apparent (e.g., $\mathcal{F}(0.05)$) and eventually makes the signal



(a) (b) (c)

Fig. 2. Gaussian Mixture Effect and its impact on (a) Signal distribution (b) Win mean (c) Win variance.

bi-modal (e.g., $\mathcal{F}(0.40)$).[1] This directly impacts the precision with which samples of $\mathcal{F}(0.00)$ (i.e., normal distribution) can be separated (e.g., using windowed mean and variance as in ClassAct) from samples of other, truly Mixed Gaussian elements of $\mathcal{F}$ (i.e., $\alpha > 0.00$). For a fixed $\alpha \in (0, 0.5)$, an arbitrary window $\mathcal{F}_n(\alpha)$ of n consecutive samples from $\mathcal{F}(\alpha)$ will include exactly $n-i$ and i values from the major and minor components with probability

$$P(\mathcal{I}(\mathcal{F}_n(\alpha)) = i) \quad = \quad C_n^i \cdot \alpha^i \cdot (1-\alpha)^{n-i}, \tag{2}$$

where $\mathcal{I}$ is an indicator function that shows the number of values from the minor component of $\mathcal{F}_n(\alpha)$ window. Under the condition $\mathcal{I}(\mathcal{F}_n(\alpha)) = i$, the window can be analyzed as if it consisted of n independent normal variables. Therefore, conditional PDFs for windowed mean and variance of these variables equates to

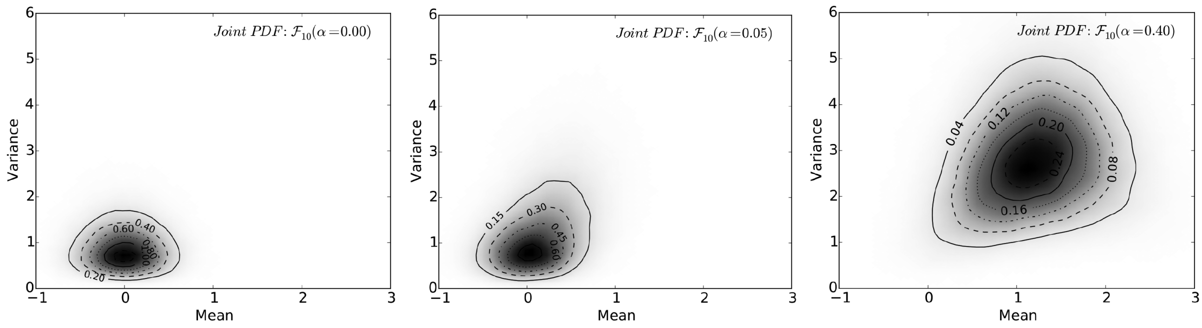
$$\begin{aligned} P(E(\mathcal{F}_n(\alpha)) = x | \, \mathcal{I}(\mathcal{F}_n(\alpha)) = i) \quad &= \quad n \cdot N(n \cdot x,\ (n-i) \cdot \mu_1 + i \cdot \mu_2,\ n \cdot \sigma^2), \\ P(Var(\mathcal{F}_n(\alpha)) = x | \, \mathcal{I}(\mathcal{F}_n(\alpha)) = i) \quad &= \quad n \cdot \mathcal{X}^2(n \cdot x, n, (n-i) \cdot \mu_1^2 + i \cdot \mu_2^2), \end{aligned} \tag{3}$$

where $\mathcal{X}^2$ is a non-central chi-squared distribution. Here, to simplify the formulae, we deliberately make use of the fact that all of the normal variables are uni-variate with $\sigma^2 = 1$ (see Figure 2). Subsequently, using Equations (2) and (3), the overall probability function of windowed mean and variance can be calculated as

$$\begin{aligned} P(E(\mathcal{F}_n(\alpha)) = x) \quad &= \quad \sum_{i=0}^{i=n} P(E(\mathcal{F}_n(\alpha)) = x | \, \mathcal{I}(\mathcal{F}_n(\alpha)) = i) \cdot P(\mathcal{I}(\mathcal{F}_n(\alpha)) = i), \\ P(Var(\mathcal{F}_n(\alpha)) = x) \quad &= \quad \sum_{i=0}^{i=n} P(Var(\mathcal{F}_n(\alpha)) = x | \, \mathcal{I}(\mathcal{F}_n(\alpha)) = i) \cdot P(\mathcal{I}(\mathcal{F}_n(\alpha)) = i). \end{aligned} \tag{4}$$

Note, the above equations (Equation (4)) also hold for windowed mean and variance of normal signals represented by the α-value equal 0.00. These equations particularly help us evaluate the impact of α on the distributions of windowed mean and variance of various signals from $\mathcal{F}$ family. Figures 2(b) and 2(c) show exemplar distributions (generated using Equation (4)) for various window sizes (i.e., 10 and 20) and α-values (i.e., 0.00, 0.05, 0.40). The histograms for mean and variance are generated using simulated data. As shown, signals with $\alpha = 0.00$ and $\alpha = 0.05$ share majority of their windowed mean and variance values, which significantly affects separability of the two cases using traditional ClassAct method. As *alpha* increases, typical windowed mean and variance values move further away from those of $\alpha = 0.00$ and, hence, increase separability. An increase

[1]Figure 2(a) demonstrates that a mixture of multiple components that follow normal distributions may not always follow a normal distribution. The distribution is, in fact, governed by the α factor.



(a) (b) (c)

Fig. 3. Gaussian Mixture Effect on joint windowed mean and variance distribution. (a) No effect ($\alpha = 0.0$) (b) Low effect ($\alpha = 0.05$) (c) Medium effect ($\alpha = 0.4$).

in window size compresses the distributions along the value axis and also aids the separation. A similar behaviour is observed if the windowed mean and variance are used jointly, as shown in Figure 3.

As seen in Figure 2, while $\alpha = 0.05$ increases the ratio of higher values (i.e., above $x = 2$), the increase is not sufficient to warrant a noticeable impact on windowed mean and variance and, therefore, classification of signals. To overcome this, Iterative Edge Mining [12] is based on histogram representation of signal and, therefore, has higher sensitivity to minute changes of signal distributions. Subject to bin selection, the increase in ratio of higher values will be reflected by the histograms, thereby improving the classification. This method, however, comes at a cost where multiple bins need to be continuously maintained and analyzed on-board IoT devices. Accordingly, we discuss an adaptation of IEM that is more suitable for IoT-based execution.

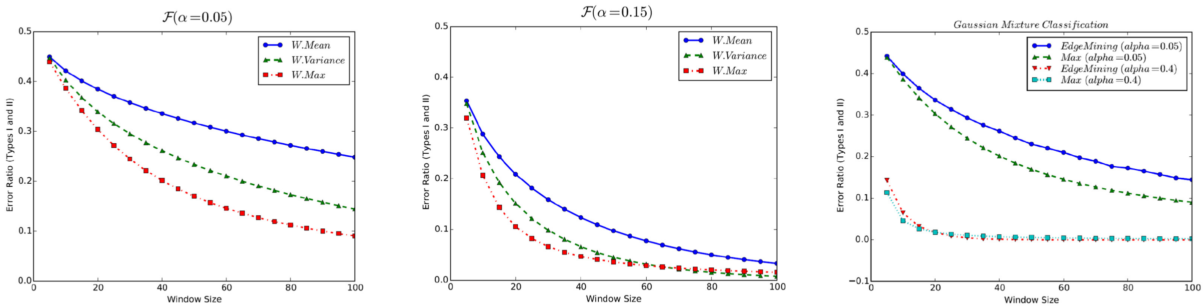
3.1.2 IEM-2.0 for Classification of Mixed-Gaussian Signals. To analyze the predictive capabilities of IEM-2.0, we first evaluate distribution of values for the *winMax* feature for $\mathcal{F}(\alpha)$ Mixed-Gaussian Signals. For brevity, we omit the *winMin* feature, since the analysis for it is a mere adaptation of the analysis presented here. Consider the maximum of an arbitrary window $\mathcal{F}_n(\alpha)$. Similar to Equation (3), under conditions $\mathcal{I}(\mathcal{F}_n(\alpha)) = i$, the Cumulative Distribution Function (CDF) for *winMax* equals to:

$$P(Max(\mathcal{F}_n(\alpha)) <= x|\ \mathcal{I}(\mathcal{F}_n(\alpha)) = i) \quad = \quad N^*(x, \mu_1, \sigma^2)^{n-i} \cdot N^*(x, \mu_2, \sigma^2)^i, \tag{5}$$

where N^* denotes a CDF of normal distribution. Accordingly, the overall CDF of $\mathcal{F}_n(\alpha)$ is:

$$P(Max(\mathcal{F}_n(\alpha)) \leq x) \quad = \quad \sum_{i=0}^{i=n} P(Max(\mathcal{F}_n(\alpha)) \leq x|\ \mathcal{I}(\mathcal{F}_n(\alpha)) = i) \cdot P(\mathcal{I}(\mathcal{F}_n(\alpha)) = i). \tag{6}$$

Now, let us assume that for a particular $n \geq 1$ and $\alpha > 0$, a decision tree is used to separate sequences $\mathcal{F}_n(\alpha)$ from $\mathcal{F}_n(0.00)$, based on a particular m-dimensional feature f that is a function from $\mathbb{R}^n$ onto $\mathbb{R}^m$. While $m = 1$ implies Window Mean, Variance, Maximum, and Minimum are used independently, $m = 2$ implies they are used jointly. Assume that CDF for the possible feature values of $\mathcal{F}_n(0.00)$ and $\mathcal{F}_n(\alpha)$ sequences are known and denoted as $P_{\mathcal{F}_n(0.00)}$ and $P_{\mathcal{F}_n(\alpha)}$, respectively. During the decision-tree analysis, feature values are first derived from the given n signal values and then subjected to a number of threshold assessments, as specified by the decision tree. Going back to the example considered in Figure 2, it is fair to assume that the optimal decision tree will consist of only one node. Sequences for which features exceed the threshold will be classified as $\mathcal{F}_n(\alpha)$, whereas the sequences for which the features are below the threshold will be classed as $\mathcal{F}_n(0.00)$. Subsequently, for a threshold x_{tr}, probabilities of type I and II errors (P_I, P_{II})



(a) (b) (c)

Fig. 4. Gaussian Mixture Effect on single-feature classification for low ($\alpha = 0.05$ (a)) and low-to-medium mixture effects ($\alpha = 0.15$ (b)); Joint windowed mean & variance classification for low and medium mixture effects ($\alpha \in \{0.05, 0.4\}$ (c)).

are equal to:

$$\begin{aligned} P_I(x_{tr}) &= P_{\mathcal{F}_n(\alpha)}(x_{tr}), \\ P_{II}(x_{tr}) &= 1 - P_{\mathcal{F}_n(0.00)}(x_{tr}). \end{aligned} \tag{7}$$

Therefore, the optimal threshold minimizing the error of both types can be calculated as:

$$X_{OPT} \quad = \quad Argmin_{x \in \mathbb{R}^n} \left(Max(P_{\mathcal{F}_n(\alpha)}(x),\ 1 - P_{\mathcal{F}_n(0.00)}(x))\right). \tag{8}$$

As all CDF are continuous, monotonously increasing functions with range between [0,1], it can be shown that X_{OPT} always exists and that $P_I(X_{OPT}) = P_{II}(X_{OPT})$. Thus, where 1-dimensional features (m=1) are concerned, the solution for the problem in Equation (8) can be calculated thusly:

$$P_{\mathcal{F}_n(\alpha)}(x) = 1 - P_{\mathcal{F}_n(0.00)}(x)). \tag{9}$$

For $m \geq 2$, solving Equation (9) will generate a subset $\widetilde{X}$ of the original feature space $\mathbb{R}^m$. Subsequently, the optimization problem can be re-formulated as:

$$X_{OPT} = Argmin_{\widetilde{X}}\ P_{\mathcal{F}_n(\alpha)}(x). \tag{10}$$

Knowing X_{OPT} allows us to further numerically evaluate probabilities of P_I and P_{II} errors for selected features. Figures 4(a) and 4(b) demonstrate results of such evaluation that have been performed using CDF functions for windowed mean, variance, and maximum obtained above (note that for the first two metrics, we derive PDFs that can be easily transformed into CDFs). The evaluation was made for the same set of μ and σ^2 parameters and shows that for windows of low and moderate sizes *winMax* and, therefore, IEM-2.0 has a lower error rate (i.e., better prediction capability) than ClassAct. The advantage of the IEM-2.0 is more apparent for lower α-values (Figure 4(a)) and diminishes as α and/or window-size increase (Figure 4(b)). And, finally, Figure 4(c) demonstrates this effect when Window Mean and Variance are used jointly. While in this work, we do not present analytic formulae for joint CDF for Window Mean and Variance; during the analysis, we interpolate these functions based on results of numeric simulations. As evident, it is particularly beneficial to use IEM-2.0 for classification of signals whose behaviour closely resembles that of $\mathcal{F}_\alpha$ with lower α values. Note that while the *winMax* feature of IEM-2.0 has been deliberately used in this example due to the positive nature of histogram shift (as demonstrated in Section 3.1.1), the shift in histogram is typically non-stationary and may be positive or negative in nature. Therefore, in IEM-2.0, we perform classification based on the joint use of (*winMin*, *winMax*) features.

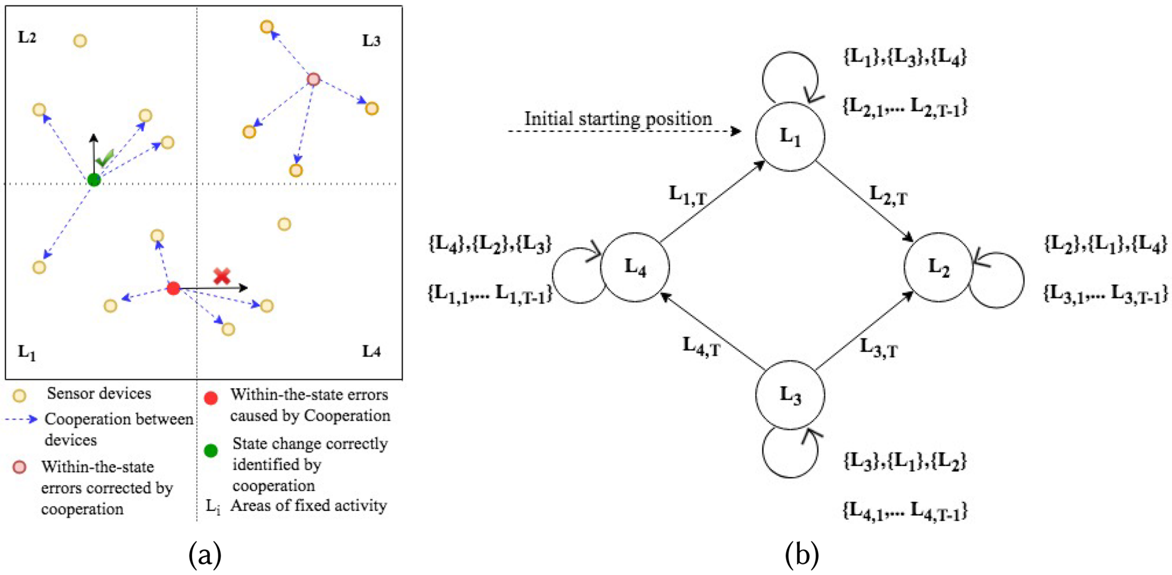


Fig. 5. (a) Effect of cooperation (b) Implementation of ASMM.

3.2 Cooperative Activity Sequence-Based Map Matching (CASMM)

Once the activity state is identified, the sequence of activities generated by IEM[2] is analyzed by the CASMM method for localization. The CASMM method consists of two light-weight computational tasks—Cooperative activity-state detection and ASMM.

Despite the improved classification accuracy of IEM over ClassAct, certain (P_I, P_{II}) errors may persist, owing to strong overlap between signals, especially, for lower α values. These errors may further increase in the presence of >2 signals (i.e., more than two activity states). Now, let us assume that at any given time t, a set of devices N ($|N|$), where each device $n \in N$ runs the IEM algorithm for on-board localization, are located within the same physical area denoted by L_i (Figure 5(a)). The area L_i is defined such that all IoT devices within this area exhibit a common high-level activity state. Therefore, while each node n analyzes individual activity state, it can be argued that analysis on a single node (referred to as the initializing node (IN)) would suffice the activity recognition for all N in L_i. However, we suggest analysis on all $n \in N$ or a subset of N devices as well as cooperation between the neighboring participating devices for exchange of activity-state updates to improve accuracy of individual predictions.

We envisage a set of participating devices N' ($|N'| \leq |N|$) nearest to node n (at any given time) as a coalition that exhibits a common activity state based on the location. Besides the individual predictions, we propose that each participating device maintains a local copy of the shared network state. If a change in activity is predicted by any device $n \in N'$ such that the predicted state differs from the shared network state, then it initiates cooperation with the remaining nodes in N'. We use an equal-weight majority-voting scheme wherein the shared network state is calculated as the mode of the predicted state at each device $n \in N'$. If the majority of devices in N' agree with change in state, then it implies that device has departed from L_i and moved to another area $L_j, i \neq j$ and, therefore, exhibits a different activity state. Otherwise, it is assumed that the device has predicted an untimely change in activity state, and the last updated activity state is maintained. Such cooperation between devices would not only allow detection of misclassified states but also facilitate the timely detection of state transitions. For instance, as shown in Figure 5(a), cooperation between devices facilitates correction of within-the-state errors (in L_3) and timely detection of

[2] All mentions of IEM hereafter refer to IEM-2.0 unless specified otherwise.

change in state as a node moves from $L1 \rightarrow L2$. Depending on the vicinity of the node, however, cooperation may lead to certain errors. As shown in Figure 5(a), a node in L_1 may assume the activity state in L_4, owing to its closeness to the sensor devices. The ASMM module is used to identify such errors and improve accuracy of state detection and, thereby, localization.

Once the cooperation is performed, the sequence of activity states is interpreted by the ASMM module. As mentioned earlier, ASMM is primarily proposed for indoor pedestrian localization [8]. The approach uses activity-related locations (e.g., staircase and corners) within a building as virtual landmarks to determine user trajectory and location. While a large outdoor environment may lack such characteristic landmarks, the ASMM approach can be extended to outdoor IoT-based localization, since the high-level activity and mobility of a user are essentially bounded by the outdoor topology. We, therefore, propose to determine the location of a node by mapping the sequence of activities along with their corresponding duration to a given outdoor map. Such mapping is light-weight and suitable for sensor-based implementation. If a change in state is recorded after cooperation, then the sequence of previously stored activities along with the corresponding duration is fed as input to the ASMM module. The ASMM module accepts the change in state only if it is consistent with the topology (i.e., physically feasible) and has been predicted for a continuous period higher than a given threshold $\mathcal{T}$. The trajectory of motion and location is then determined. Otherwise, the change in state is regarded as a classification error, and the user activity state and location is considered unchanged. For instance, consider that a user (sensor device) in Figure 5(a) can only move in a clockwise direction from $L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow L_4 \rightarrow L_1$, as shown in Figure 5(b). Given the initial reference point (L_1), a node can either remain in the same activity state (and location) or move to L_2. Therefore, any changes in state corresponding to locations L_3 and L_4 are discarded by the ASMM module. Moreover, location is updated to L_2 only if the corresponding state is predicted for a duration of $\mathcal{T}$. Similar behavior is implemented for all state transitions. While such an approach may increase the delay in detecting state transitions (depending on the value of $\mathcal{T}$), it reduces the incoherent and untimely changes in activity that may be predicted after cooperation (e.g., error in L_1 in Figure 5(a)).

Our IEM2.0-CASMM-based localization approach is summarized in Algorithm 1. The algorithm takes as input acceleration data at time t (acc_t), parameters $windowSize$, ε, and decision-tree DT for IEM-based classification, set of nodes N, coalition N'_t at time t, threshold $\mathcal{T}$ and $roadMap$ for CASMM, and returns two vectors containing the sequence of activities ($actVector$) and locations ($locVector$). First off, the distribution of acc values is estimated using `DIST` function that calculates the $winMin, winMax$ features. If the percentage change in either of the features exceeds the threshold ε, then DT is used to classify the activity state ($state$). If the predicted state differs from the last updated device state ($lastUpdatedState$) as well as the last stored network state ($lastNetworkState$), then the change in activity may suggest a change in the device location. Subsequently, cooperation between N'_t neighboring devices is performed to obtain the majority voted activity state. If the $networkState$ is not in harmony with the $state$ value, then the change in activity is considered as a classification error and discarded. Otherwise, if the change in state persists for a period $\mathcal{T}$, then ASMM is performed to validate the change in activity and estimate location of the device. If the change in state is inconsistent with the given topology map ($roadMap$), then the prediction is discarded and a $NULL$ value is returned. Else, the location of the device is returned and the activity and location vectors are updated.

3.3 Context-Aware Event-Driven Communication

As mentioned above, the optimal IEM2.0-CASMM model is determined based on the localization accuracy as well as an optimization function. The function is designed to meet the application requirements of the WSN-system and sets the criterion for selecting values of the input

ALGORITHM 1: IEM2.0-CASMM-based localization

```
Input: acc_t, windowSize, ε, DT, N, N'_t, 𝒯, roadMap
Output: actVector, locVector
repeat
    Read sensor for acc_t
    accVector ← APPEND(accVector, acc_t)
    (winMin, winMax) ← DIST(accVector, windowSize) #Evaluating distribution
    if ((|winMin − lastUpdatedMin| ≥ ε ∗ lastUpdatedMin) ∨ (|winMax − lastUpdatedMax| ≥
    ε ∗ lastUpdatedMax))
    then
        lastUpdatedMin ← winMin
        lastUpdatedMax ← winMax
        state ← PREDICT(DT, winMin, winMax) #Classification
        if ((state ≠ lastUpdatedState) ∧ (state ≠ lastNetworkState))
        then
            networkState ← MODE(lastUpdatedState[1 : N'_t − 1]) #Cooperation
            if (networkState[𝒯 − t + 1 : t] == state[𝒯 − t + 1 : t])
            then
                location ← ASMM(roadMap, actVector, state) #ASMM
                if (location ≠ NULL)
                then
                    lastUpdatedState ← state
                    lastNetworkState ← state
                    actVector ← APPEND(actVector, state)
                    locVector ← APPEND(locVector, location)
                end
            end
        end
    end
until Offload data to gateway

Function DIST(accVector, windowSize)
    return (min(accVector[(t − windowSize + 1) : t]), max(accVector[(t − windowSize + 1) : t]))
```

parameters. In this work, we consider minimization of the device energy consumption and determine the appropriate IEM2.0-CASMM model for sensor-based execution.

A vast majority of WSN-based systems are deployed to monitor remote areas that stretch over several kilometers. As such, communication of data packets from sensor devices to a cloud gateway is the most energy-intensive task performed by these devices. Continuous packet transmissions to the gateway can significantly reduce the operational time of these battery-operated devices. However, most sensor data is not time sensitive enough to maintain continuous real-time Internet connectivity. Accordingly, we propose a context-aware event-driven communication approach to transfer data from WSN to the gateway. We exploit the location information of devices obtained from IEM2.0-CASMM-based analysis and transmit data to the gateway only at the occurrence of a change in location. The delay-tolerant approach would not only improve energy efficiency of the devices through reduced packet transmissions but also reduce the operational cost of the system by eliminating the need for continuous Internet connectivity. As such, accuracy of localization has a direct impact on the energy consumption of the devices. The energy cost incurred in sending a data packet to the cloud can be calculated as shown below [28]:

$$E_{CL} = (e + \beta \cdot d^2) \cdot bits. \tag{11}$$

E_{CL} is the energy consumed by a node for sending a packet containing $bits$ number of bits to the gateway over a distance d. The variable e denotes the energy cost of transceiver for receiving and transmitting unit data (hardware dependent) and β is a constant $[J/bit.m^2]$.

As discussed previously, the CASMM method can help improve accuracy of classification of IEM (via cooperation between devices) and, in turn, the accuracy of localization. The cooperation itself, however, incurs a communication overhead in sending and receiving cooperation requests and location updates. These costs can be estimated using the following equations:

$$\begin{aligned} E_{CO} &= q_n \cdot ((2e + \beta \cdot d'^2) \cdot bits' \cdot (N' - 1) + E_{agg}) + (1 - q_n) \cdot ((2e + \beta \cdot d'^2) \cdot bits'), \\ E_{LO} &= p_n \cdot (e + \beta \cdot d'^2) \cdot bits' \cdot (N - 1) + (1 - p_n) \cdot (e \cdot bits'), \\ E_C &= \sum_{t=1}^{\tau} (r_t \cdot \sum_{n=1}^{N} (E_{CL} + E_{LO}) + s_t \cdot \sum_{n=1}^{N} E_{CO}). \end{aligned} \tag{12}$$

E_{CO} is the energy consumed by node n per cooperation between N' nodes, d' is the distance between the participating devices N', $bits'$ represents the number of bits per packet, and E_{agg} is the energy cost for aggregating the location data of N' nodes. The decision variable q_n takes a value of 1 if the cooperation is initiated by node n and 0 if it receives a request from another node. E_{LO} is the energy consumed by node n per distribution of location updates among N devices. The decision variable p_n assumes 1 if node n predicts the change in location and disseminates packets to other nodes and 0 if it receives a packet from another node. Note that the value of $bits' < bits$, as the packet sent to the gateway contains accumulated sensor data over time while the packet sent locally among devices contains just the state information. Moreover, $d' < d$, as the packet sent to gateway is over a longer distance than device-to-device communication. The overall communication energy consumed by N devices over a planning time horizon τ then equates to E_C. The variable r_t takes a value 1 if a change in location is predicted at time t and 0 otherwise. Similarly, the variable s_t takes a value of 1 if a cooperation is initiated at time t and 0 otherwise. We study the effect of $windowSize$, ε, and coalition size $|N'|$ on the energy consumption of the network in Section 5.

4 EXPERIMENTAL DESIGN

In this section, we present an application of our IEM2.0-CASMM system for animal localization in dairy farms. We describe our application scenario and discuss the implementation of IEM2.0-CASMM on-board animal-wearable sensor devices, followed by the design of our WSN-based prototype and the pilot study.

4.1 Animal Activity Monitoring and Localization

Real-time activity monitoring and localization of livestock is strongly advocated for on-farm LBS, such as behavior analysis, virtual fencing, and feed management under the umbrella of Precision Dairy Farming. Today, animal-wearable sensors are widely used to facilitate continuous monitoring of the physiological state of the cows for early diagnosis and treatment of diseases [29]. Enriching the results of health monitoring with animal-mobility data will allow for better understanding of animal behavior and well-being [30]. Combined analysis of both physiological and behavioral data with respect to location of the animal has been shown to provide vital insights into the farm processes and help improve their overall efficiency [31].

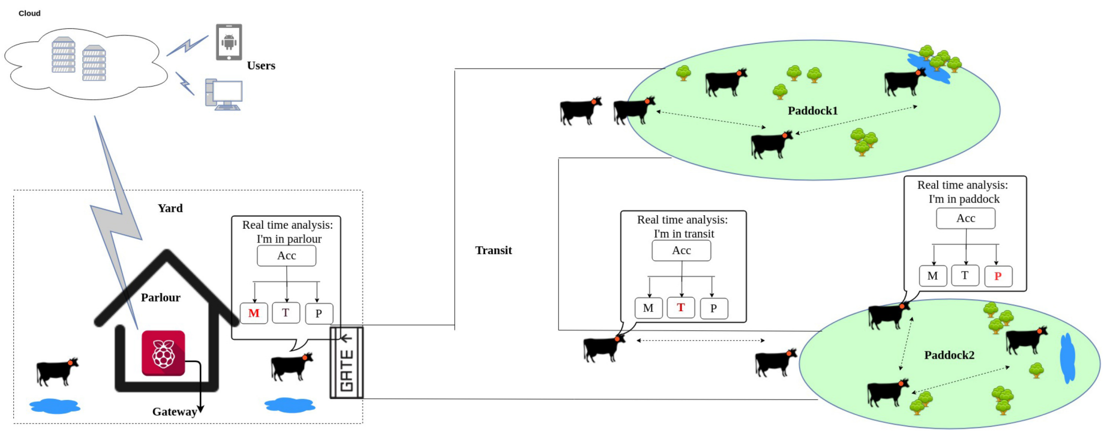


Fig. 6. IEM2.0-CASMM-based animal localization in dairy farms.

4.1.1 Application Scenario. Figure 6 depicts our application scenario. Our WSN system consists of animal-wearable sensor devices and a cloud gateway and allows for location-aware data collection for livestock management. The animal wearable is an extensible sensor device that consists of sensors to monitor the physiological state as well as the mobility of cows. We propose the implementation of IEM on-board the collar devices to predict the activity state of cows as they move around a farm. Furthermore, device-to-device communication is proposed to allow cooperation between the cows and perform ASMM to estimate the location as they predict changes in activity state. A gateway node is installed within the farm (hosted inside the parlour in Figure 6) to collect location-enriched data from sensor devices and upload it onto the cloud for future analysis. Since a typical farm spans across a large area and the majority of the data relating to the farm processes is delay-tolerant, we adopt the event-driven communication approach discussed in Section 3.3. Accordingly, sensor data combined with location information is stored locally on the collar devices as cows move around the farm, and the data is transmitted to the gateway once a change in location is predicted. This eliminates the need for continuous Internet connectivity within a farm, which is particularly important in rural deployments. Whereas the existing animal-wearable technologies such as RumiWatch [32] also follow a delay-tolerant communication approach, sensor data is transmitted to the cloud every 15 minutes, as the devices incorporate very little intelligence and rely on external (e.g., cloud-based) analysis for localization and behavior modelling. Implementation of IEM2.0-CASMM is expected to reduce the frequency of packet transmissions and improve the energy efficiency of the device operation. Moreover, real-time localization on-board collar devices could potentially allow timely detection of behavior anomalies in cows that may be indicative of stress and other health-related issues. Our WSN-based approach, thus, lays the foundation for future smart livestock farming.

4.1.2 IEM2.0-CASMM Approach for Animal Localization. In Reference [12], we evaluate the performance of IEM (histogram-based approach) for classification of low-level activities, such as standing and walking. Since the mobility of a cow is random, identification of such low-level activities is unnecessary and irrelevant for localization. Rather, we model our IEM (v. 2.0) classifier to predict the coarse location of cows—parlour (M), paddock (P), and transit between parlour and paddocks (T) within a farm, as shown in Figure 6. These locations span the entire farm topology and correspond to the three primary activities performed by a cow—milking, grazing, strolling around a farm, respectively.[3] The IEM-based classification, thus, helps identify the high-level activity state and location of cows.

[3]Note, we identify the entire yard as parlour, since the primary activity associated with cows within a yard is milking.

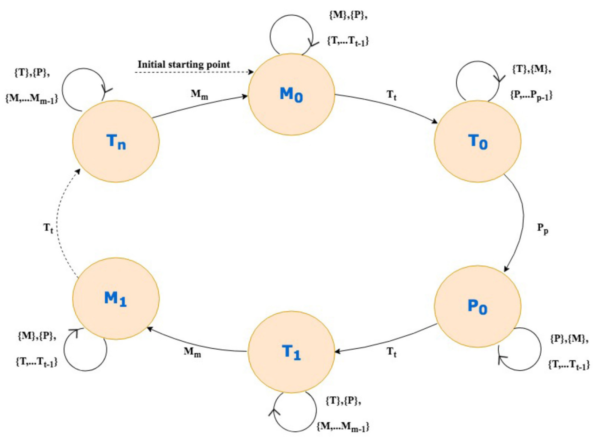


Fig. 7. Behavioural state transitions using ASMM.

Furthermore, as cows move in a herd, we exploit their spatial-temporal coherence for CASMM. Consider a herd of size N such that all cows $n \in N$ are equipped with a collar device and move together from one state to another. As such, a single cow or subset of N cows can suffice localization for the entire herd. We envisage the set of participating devices $N' \subseteq N$ within a herd to form a coalition that exhibits a common high-level activity based on location of the herd at any given time. If any device $n \in N'$ predicts a change in the activity state that differs from the network state, then it initiates cooperation between the participating devices to allow exchange of state information. Based on majority voting, the device updates its prediction and performs ASMM if required. Any change in location is disseminated to all N devices. The cooperation, thus, ensures a consistent activity state across the herd and is expected to reduce classification errors as cows replicate low-level mobility patterns from one activity state to another. For instance, CASMM may help fix errors in prediction when the classifier identifies a transit state while cows walk to a water trough within a paddock, owing to the similarity in behavior.

In Reference [8], while the route chosen by a user is unknown, the ASMM approach is used to establish the user's trajectory based on low-level activities, as the user follows a fixed mobility pattern on each route. On the contrary, in a dairy-farming scenario, the cows follow designated routes between the parlour and the paddocks due to the restricted topology of the farm. However, as mentioned above, they perform random low-level activities (e.g., walking, standing, and sitting) while moving along these routes and grazing within the paddocks. However, the cows follow a fixed sequence of the high-level activities (e.g., milking, transit, and grazing). The cows are brought into the parlour for milking. Once milked, they transit through the pathways to a paddock. After grazing, the cows leave the paddock and transit back through the same path to the parlour, and so on. Accordingly, we propose an adaptation of the ASMM approach to estimate the animal location based on the sequence of these high-level activities generated by IEM, as shown in Figure 7. The monitoring of cows commences at the milking parlour, location M_0, on day 1. At M_0, the cows can either remain within the parlour or enter into the pathways, i.e., transit state T_0. Therefore, any state changes to paddock predicted after cooperation can be ignored. If a change in state to transit is predicted for a continuous period of $\mathcal{T}$ (denoted as $T, ..T_t$ in Figure 7), then it is considered feasible and the location is changed to T_0. At T_0, the cows can either remain in transit state (i.e., stroll along the pathways) or enter into the paddocks. Any state changes to parlour can, therefore, be ignored. Moreover, continuous change in state to paddock denoted by $P, ..P_p$ is accepted and location is changed to P_0. Similar logic is followed to change location from P_0 to T_1 as cows return to the parlour (M_1) for milking and so on. Since the farmers follow a specific sequence of paddocks to

(a) (b) (c)

Fig. 8. (a) and (b) Animal-wearable collar devices (c) Cloud gateway.

Table 1. Implementation Details

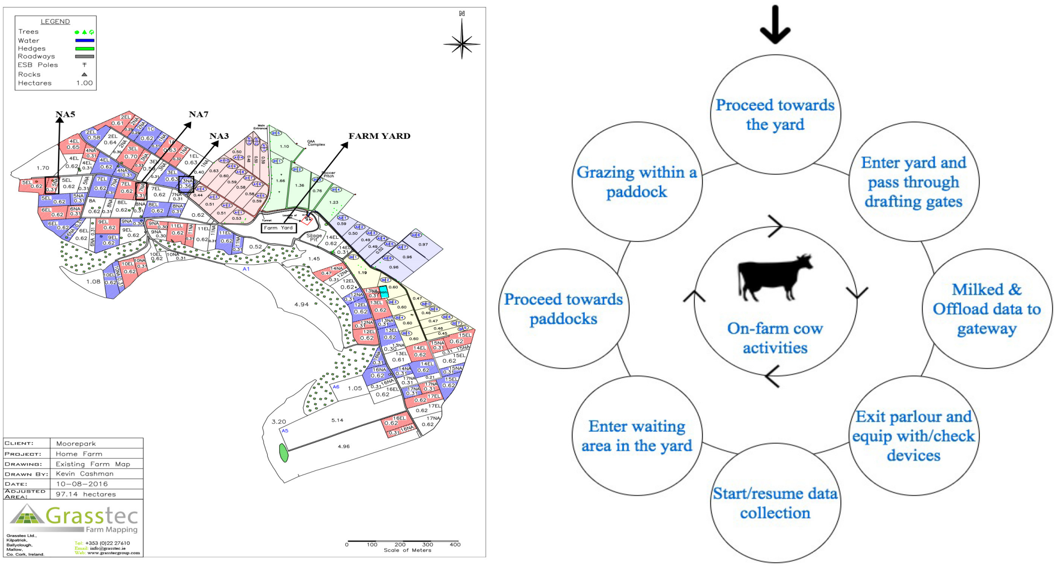
Device type	Characteristic	
Collar device	Components	CM5000 mote [33], MPU9255 Inertial Motion Unit (IMU) [34]
	Memory	48KB program flash and 1MB non-volatile flash for data storage
	Battery	2xAA batteries
	Operating system	TinyOS [35]
Gateway device	Components	CM5000 mote, Raspberry Pi (v. 2B) [36], Wi-Fi dongle

be grazed, the state transitions along with the sequence numbers 1..*n* detected by IEM2.0-CASMM can be used to determine which paddock the cows must be headed to after milking. Based on the selection of paddock, the pathway can be determined and the time elapsed in transit state can be used to estimate the exact location along the pathway.

4.2 Field Experiment

As mentioned earlier, our WSN prototype consists of two types of devices—wearable collar devices and a cloud gateway, as shown in Figure 8. The design details of the two devices are given in Table 1. While collar devices are responsible for data collection and on-board analysis of animal health and mobility, the role of gateway is to collect sensor data from the collar devices (via mote-to-mote communication) and upload it onto the cloud for future analysis. We deployed our prototype in a Dairygold-sponsored farm located in Kilworth, Co. Cork, Ireland (Latitude: 52.168096, Longitude: -8.24206) (Figure 9(a)). The farm is operated by TEAGASC, the Agriculture and Food Development Authority of Ireland. The experiment was conducted on 5 Holstein Friesian cows (using five collar devices) selected randomly from a herd of 46 cows over a period of five days in June 2017. For the purpose of this study, we programmed the collar devices for collecting raw acceleration data of cows at a frequency of 1Hz for a 10h duration per day (in accordance with the daytime milking cycle). The data was used to examine the behaviour of cows within the milking parlour, transit, and paddock and build the IEM2.0-CASMM model to evaluate its performance in a real-life scenario.

A LELY collar is used to place the device around a cow's neck, as shown in Figure 8(b). An additional weight is attached to the collars to keep the device stable. The ideal orientation of the accelerometer axes is as follows: *y-axis towards the front of the cow, z-axis was out on the side, and x-axis was downwards.* The cows follow a fixed milking cycle, as shown in Figure 9(b). They are brought into the yard for milking in the morning. Once the milking is complete, cows exit the parlour and proceed to the waiting area, as shown in Figure 10(a). Once the entire herd is milked, the cows are released towards the paddocks (Figure 10(b)). Figure 10(c) shows two of the



(a) (b)

Fig. 9. (a) Map of the Dairygold farm in Kilworth, Co. Cork, Ireland (b) Milking cycle followed by the cows.

(a) (b) (c)

Fig. 10. Dairy cows during the pilot study (a) In yard (b) In transit (c) In paddock.

experimental cows inside a paddock. A single paddock is assigned to the herd per day. During the experiment, the herd was taken to paddock NA7 on days 1 and 2, NA5 on day 3, and NA3 on days 4 and 5 (earmarked in Figure 9(a)). In the evening, the cows are brought back into the yard for milking. For this study, the gateway node was hosted inside the milking parlour, and data from the devices was transmitted to the gateway once the cows enter the parlour in the evening. The time corresponding to changes in location (parlour → transit → paddock → transit → parlour) is recorded using manual observations for annotating the data with ground-truth locations, i.e., parlour, transit, and paddock. These observations are made by qualified TEAGASC technicians who handle the herd for ensuring animal safety. Since we study high-level localization of animals, the use of these timestamps along with start and end time of experiment suffice the labelling of raw acceleration data. In addition, the system time corresponding to the receipt of the first data packet from each node is maintained at the gateway. The recorded time is compared with clock on collar device to assess drift in clock speed, as discussed in Section 5.1. For the purpose of CASMM, a simple topology map is required that illustrates the relative position of parlour and different paddocks with respect to each other. In this study, we obtained an existing map of the Dairygold farm depicting the various paddocks (designed by Grasstec, as shown in Figure 9(a)) from TEAGASC.

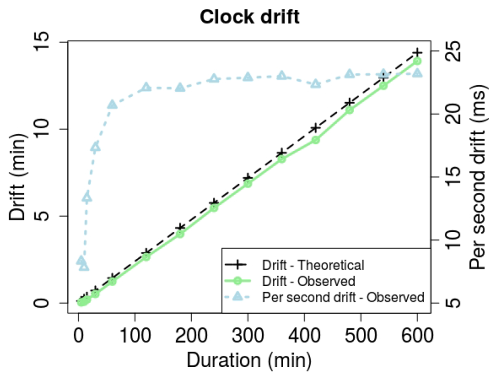


Fig. 11. Clock drift incurred by a collar device over time.

5 EVALUATION

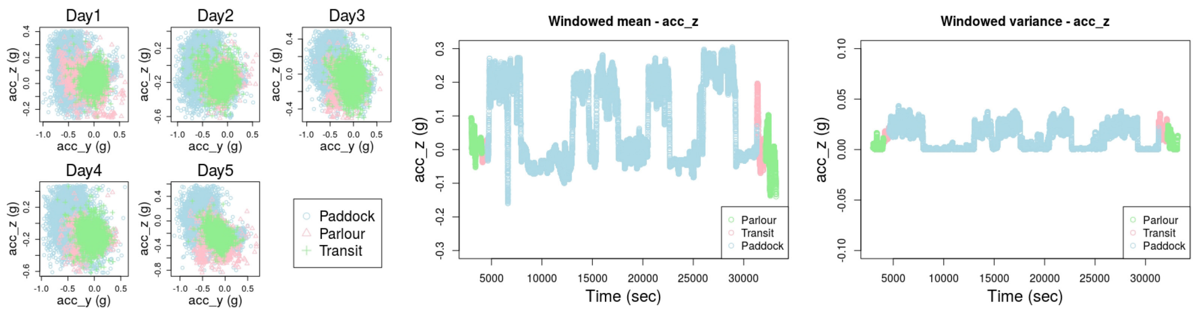
In this section, we evaluate the performance of our IEM2.0-CASMM approach using the animal-mobility data collected during our pilot study described in Section 4.2. We discuss our data exploration and feature selection approaches used for IEM-based classification, followed by the supervised learning and performance analysis of IEM2.0-CASMM for different values of input parameters. All analysis is performed using R programming.

5.1 Data Exploration and Feature Selection

Prior to training the IEM classifier, we analyze the acceleration data for necessary pre-processing and feature extraction. First off, we annotate the raw data with location (i.e., parlour, transit, and paddock) using the recorded timestamps. A positive clock skew is observed on comparing time of transmission of the first packet on the sensor devices with the corresponding system time (recorded by the Raspberry Pi). That is, the devices associate with the gateway node prior to expiration of the 10h duration. This is because a skew of 24ms per second has been noted for TelosB devices [37], owing to the software implementation of device clock in TinyOS. Furthermore, this value is affected by environmental factors, such as temperature, humidity, and vibration. The theoretical and observed drift is illustrated in Figure 11. As can be seen, the observed drift maps closely to the theory but is slightly less than the expected values. A skew of roughly 14min is incurred over the 10h period and must be accounted for to correctly annotate the acceleration readings. We also calculate the per-second drift for different time duration, as shown in Figure 11. Whereas the value increases initially, it stabilizes for longer duration. We model the linear dependency between the drift and the time duration using the *lm* function in R, as shown below. We then calculate the value of drift until each state transition and label the data accordingly:

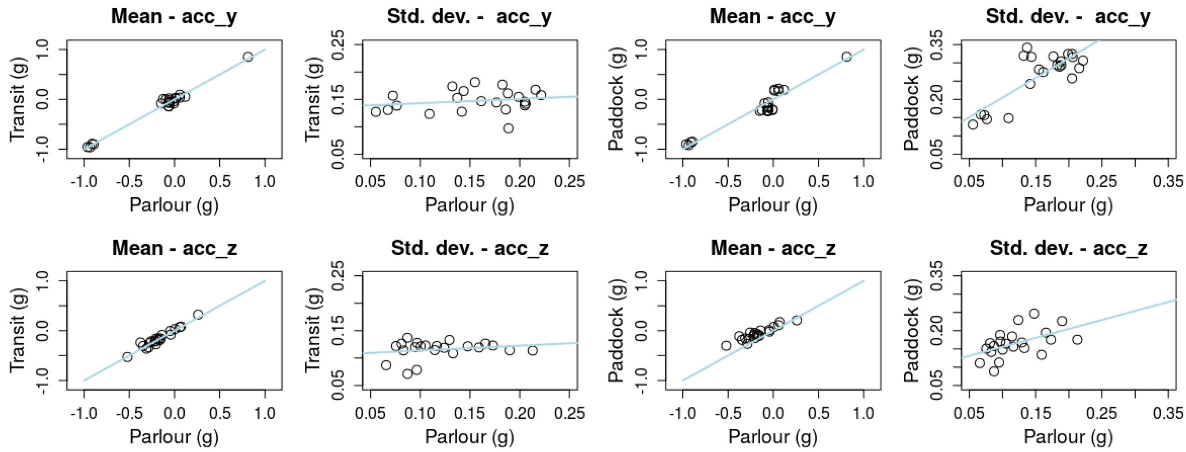
$$drift\ (min) = -0.158 + 0.023 * duration\ (min).$$

Next, we examine the raw data for outliers. Figure 12(a) shows the acceleration of a cow in the plane of movement after removal of the outliers. As can be seen, distribution of values in each state (i.e., parlour, paddock, and transit) varies across the five days. This is due to environmental factors, such as weather conditions and the quality of grass in the paddocks that affect behavior of the cow. We recalibrate the acceleration data to reduce the effect of the environment on the performance of the classifier. As evident from Figure 12(a), there is a significant overlap in the acceleration measurements of the three states. Figures 12(b) and 12(c) illustrate the windowed mean and variance of z-axis acceleration for all states. We use the Spearman's Correlation Coefficient to measure the correlation between the mean and standard deviation of parlour and transit, and parlour and paddock data along the y and z axis, i.e., plane of movement. The test suggests a



(a) (b) (c)

Fig. 12. (a) Acceleration of a cow during different activity states in the y-z plane (b) Windowed mean of acc_z (c) Windowed variance of acc_z at $windowSize = 60$.

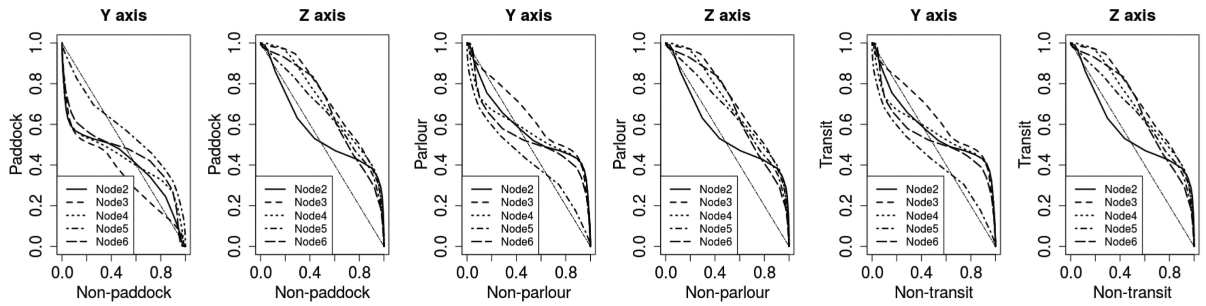


(a) (b)

Fig. 13. Linear dependence between mean & std dev. of (a) parlour & transit (b) parlour & paddock values.

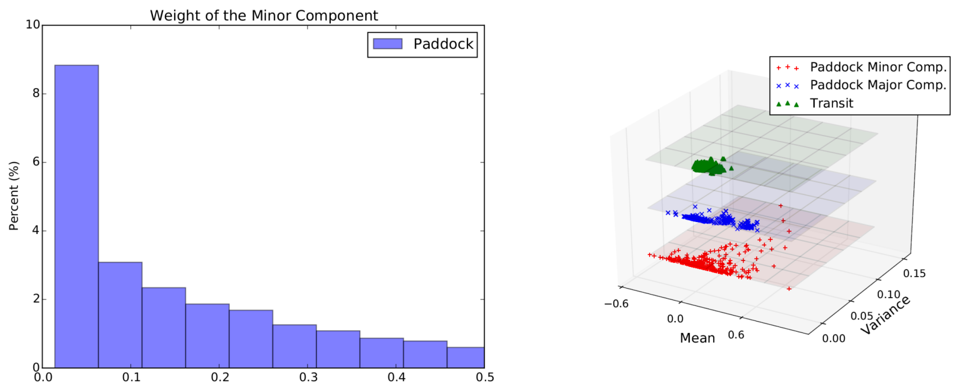
moderate correlation between the states. Accordingly, we derive the linear dependency between mean and standard deviation of y and z-axis acceleration in the parlour and transit and paddock data across the entire dataset, as shown in Figure 13. The mean of parlour is then set to zero, and the linear models are used to recalibrate the data for all three states.

Thereafter, we direct our attention to feature selection for classification. We use the Receiver Operating Characteristic (ROC) criterion to test the diagnostic ability of x-axis acceleration (acc_x), y-axis acceleration (acc_y), z-axis acceleration (acc_z), and net acceleration ($\sqrt{acc_x^2 + acc_y^2 + acc_z^2}$) for different cut-off values. Since we have a multiclass problem, we carry out a pairwise comparison (one state vs. all other states). While the acc_x and net acceleration do not capture clear distinction between the three states, acc_y and acc_z achieve a reasonable quality of separation for all nodes, as shown in Figure 14. The area under curve for the z-axis is greater than the y-axis for all nodes, thereby suggesting a better classification performance. Accordingly, we base our IEM implementation on feature values derived from acc_z measurements. The z-axis reflects horizontal movement of a cow's neck. The difference in behaviour between the states is potentially caused by the movement of cows as they graze within the paddocks and eat fodder during milking. Figure 15 provides further insights into the acceleration data from paddock and transit states across the entire dataset. While Figure 15(a) shows prevalence of two-component Gaussian mixtures with lower α values for windowed measurements in the paddock state, Figure 15(b) illustrates the similarity between



(a) (b) (c)

Fig. 14. ROC curves to evaluate the performance of acc_y and acc_z for classifying (a) Paddock (b) Parlour (c) Transit states.



(a) (b)

Fig. 15. Mixture effect in animal-mobility data at $windowSize = 60$ (a) Ratio of two-component mixtures within paddock state (b) Mixture fitting of transit and two-component paddock values.

parameters of major components of those two mixtures and one-component mixtures ($\alpha = 0.00$) prevalent during transit. The dominance of major component in the mixture and the significant overlap between the measurements highlights the need to use IEM-2.0 rather than ClassAct for animal-activity classification. Accordingly, we use *winMin*, *winMax* features for classification and study the performance of the IEM2.0-CASMM approach for different values of input parameters.

5.2 Supervised Learning

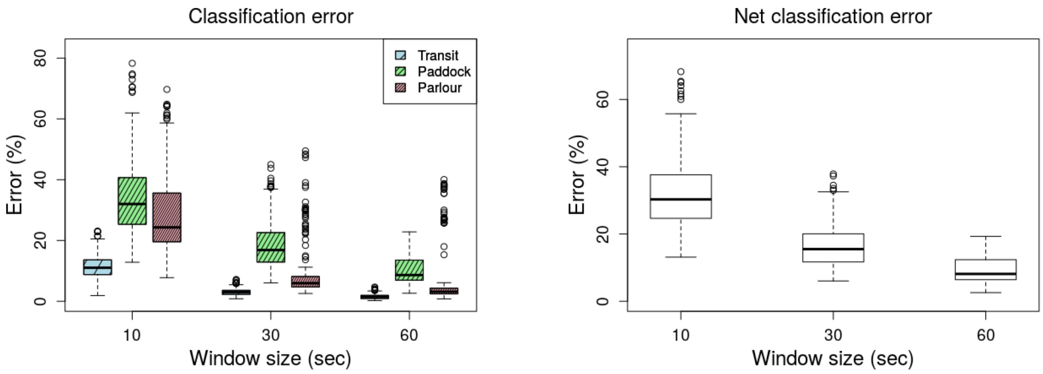
Once the classification features are selected, we train and test the IEM2.0-CASMM model for different sets of parameter values. We start by analyzing the effect of *windowSize* and ε on the performance of IEM, followed by the effect of coalition size on the performance of CASMM.

The accuracy of IEM is primarily governed by the input parameters *windowSize* and ε. The window size affects the calculation of min and max values and, thus, characterizes the signal distribution. While a small window may not capture the local min and max in close vicinity, a large window will increase the impact of historical data and may miss the small fluctuations that reflect actual state changes. As a result, increase in window size may cause a reduction in within-the-state classification errors at the expense of increasing cross-state errors around state transitions. To analyze the effect of *windowSize*, we train the IEM classifier *DT* for each device across three window sizes: 10s, 30s, 60s. First, we calculate the *winMin* and *winMax* pairs for each trace per window size. Next, we combine the data files from all five days per device and *windowSize*, and generate training sets using stratified sampling. Each training set consists of 10% of the total samples with an equal number of parlour, paddock, and transit measurements. This is done to ensure that the classifiers are fairly trained for all three states and the dominance of paddock data does not conceal the behavior in other states. Thus, we generate three training sets corresponding to the three window sizes for each of the five nodes. The sampled data is then fed to the $C5.0$ classifier to build the decision trees. We assign $\varepsilon = 0$ and study the effect of window size on the classification accuracy. The performance is evaluated per data trace (file) for all five days using appropriate DT (per device and *windowSize*). A value of 0 for ε allows us to evaluate the classifier for all possible distributions for a given *windowSize*. The training process is iterated ten times, i.e., 10 DT are generated for each node and window size, for performance validation.

Next, we introduce the ε parameter and study its effect on the performance of IEM. The value of ε controls the frequency of classification. Whereas a small ε will feed even the slightest changes in distribution to the classifier, a large ε value will accommodate significant changes in the distribution without presuming change in the activity state. Accordingly, while a large ε may improve the energy profile of the system through reduced classifications, it may increase the errors due to delay in detecting state transitions. Moreover, an error within the state persists longer due to infrequent classifications. We evaluate the impact of ε on the number of classifications as well as the classification accuracy across three values: 0.2, 0.4, 0.6, which correspond to 20%, 40%, and 60% change in distribution of the signal, using the DT trained above. While *winMin* and *winMax* are calculated per acc_z reading, classification is performed only if the difference between the updated values and the previous estimates exceeds ε. The cows are considered to be in the same activity state as the last identified state until the next classification. Furthermore, as we adopt an event-driven communication approach, we study the effect of *windowSize* and ε on total number of packet transmissions to the cloud by the network (P_{CL}) and resultant E_C prior to applying the CASMM.

Finally, we evaluate the performance of CASMM for localization. As discussed in Section 3.2, we use an equal-weight majority voting scheme for cooperative activity-state detection. Accordingly, we estimate the shared activity state per day and per ε for window size 60s for different coalitions. The performance of cooperation varies with the coalition size, i.e., the number of participating devices. Since we have a total of five nodes, we analyze the effect of cooperation on accuracy of state detection for four different coalition sizes—$N' = 2/3/4/5$. Moreover, we study its impact on P_{CL}, total number of packet transmissions within the network for collaboration (P_{CO}) and dissemination of updates (P_{LO}), and the resultant communication energies (E_{CL}, E_{CO}, E_{LO}, and E_C). Once the appropriate coalition size is selected, we evaluate the performance of ASMM for localization. The effect of ASMM is governed by the threshold parameter $\mathcal{T}$. To set the value of $\mathcal{T}$, we evaluate the distribution of the errors within each state. We use the eighth decile value as the threshold for each state. We then implement ASMM (as shown in Figure 7) each time a change in state is observed after cooperation. We assess the effect of ASMM on accuracy of localization, P_{CL}, and E_C for different ε.

5.2.1 Effect of Window Size. To test the performance of IEM for different window sizes, we predict the activity state for each (*winMin*, *winMax*) pair across the entire dataset using the appropriate DT. The error in classification is calculated by comparing the predicted states against the observed states for each activity as well as net trace per data file for all days. This evaluation is repeated over ten iterations using the 10 DT models generated above. Figure 16 illustrates the classification errors for all traces over the ten iterations. The errors per activity state are shown in Figure 16(a). An overall reduction in error of each state is observed with an increase in the window size from 10s to 60s. While a median error of 11% is incurred for transit states at $windowSize = 10$, the value reduces to 3% and 1.5% at $windowSize = 30$ and $windowSize = 60$, respectively. Similarly,



(a) (b)

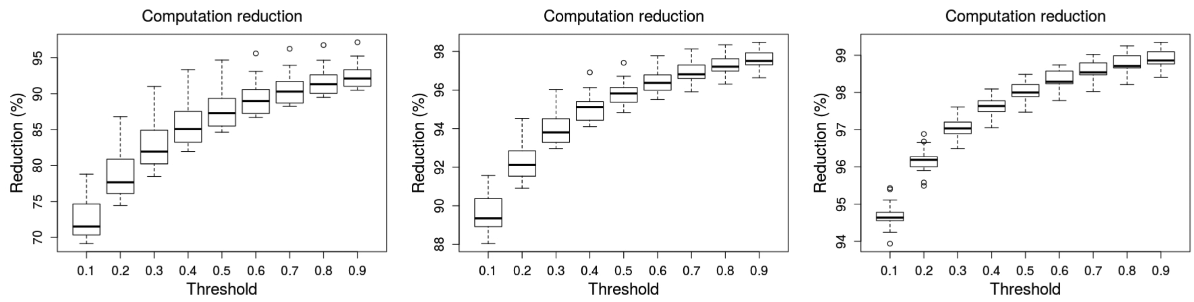
Fig. 16. Effect of *windowSize* on (a) Classification error per activity state (b) Net classification error.

the median errors for paddock and parlour states reduce from 32% and 24.3% at $windowSize = 10$ to 16.8% and 5.9% at $windowSize = 30$, and 8.6% and 3.1% at $windowSize = 60$, respectively. This is because a small time window is too narrow to correctly capture the local min and max values. As such, the calculated distribution of the signal misses the short-lived fluctuations in the vicinity and, in turn, affects the classification accuracy. While the median errors are low for transit and parlour at $windowSize = 60$, the median error for paddock states is slightly high with error for certain traces as high as 31.7%. On examining the traces, we observe that most of these errors are caused by misclassifications within an activity state, as opposed to misclassifications due to delay in detecting state transitions. This is because the classifier is unable to separate certain instances of stationary behavior and long walks within the paddocks (e.g., if a cow walks to and from a water trough located in one corner of the paddock) with the mobility patterns that are mainly observed in parlour and transit activity states, respectively. Figure 16(b) depicts the net error (all three states) for all traces. A median error of 30.2% is incurred at $windowSize = 10$, and the value decreases to 15.5% and 8.1% with increase in the window size to 30s and 60s, respectively. Furthermore, as is seen, the net error closely resembles the error in paddock as they constitute majority of the data points in any trace. The results suggest that while a $windowSize = 10$ is too narrow to capture the activity state of animals, a $windowSize = 60$ (i.e., 60 sensor readings) is capable of identifying the behavior with an accuracy over 90%. A window of 60s implies a set of 60 readings, as we collect data with a very low frequency of 1Hz. However, a window size of 10s presents the lower boundary of our analysis wherein classification is performed based on ten readings. It represents an extreme case and has been included in the analysis to illustrate the scope of our technique. The analysis shows that, despite a small set of readings, our technique can correctly classify 70% of the observations. However, the use of larger window sizes (i.e., 30s and 60s) is preferred for further analysis and CASMM-based localization. Since typical activity classifiers use high-frequency inertial data (usually 10Hz), we believe that our approach would work well with the commercially available activity trackers for the different window sizes.

5.2.2 Effect of Epsilon. As mentioned above, the value of ε controls the frequency of classification. It sets the threshold for change that is acceptable in the distribution of signal assuming the same activity state. We study the effect of ε on the frequency and accuracy of classification for all three window sizes. A summary of the analysis results is shown in Table 2. As expected, the number of classifications (computations) as percentage of the total number of readings per trace reduce with increase in the ε value for a constant window size. The median value of reduction percent increases from 77.5% at $\varepsilon = 0.2$ to 89% at $\varepsilon = 0.6$ for $windowSize = 10$; that is, only 11%

Table 2. Performance Summary of IEM (without Collaboration) for $\varepsilon \neq 0$

Metric		Window = 10s			Window = 30s			Window = 60s		
		$\varepsilon = 0.2$	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.2$	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.2$	$\varepsilon = 0.4$	$\varepsilon = 0.6$
Comp reduction (%)		77.5	85.0	89.0	92.1	95.0	96.2	96.2	97.5	98.2
Error (%)	T	11.9	12.0	12.0	4.4	6.1	7.0	2.5	3.4	7.5
	P	31.9	32.5	32.5	17.2	18.6	19.1	9.7	11.5	12.4
	M	27.1	27.6	28.9	9.0	12.6	13	7.6	11.3	14.1
P_{CL} - P1		8,438	6,350	4,963	2,693	1,835	1,355	998	643	450
Net E_{CL} (J) - P1		1.31	0.98	0.77	0.42	0.28	0.21	0.15	0.10	0.07
Net E_{LO} (J) - P1		$0.70e^{-3}$	$0.53e^{-3}$	$0.41e^{-3}$	$0.22e^{-3}$	$0.15e^{-3}$	$0.11e^{-3}$	$0.08e^{-3}$	$0.05e^{-3}$	$0.04e^{-3}$
P_{CL} - P5		7,577	5,715	4,503	2,451	1,701	1,244	957	619	432
Net E_C (J) - P5		1.17	0.89	0.70	0.38	0.26	0.19	0.15	0.09	0.07

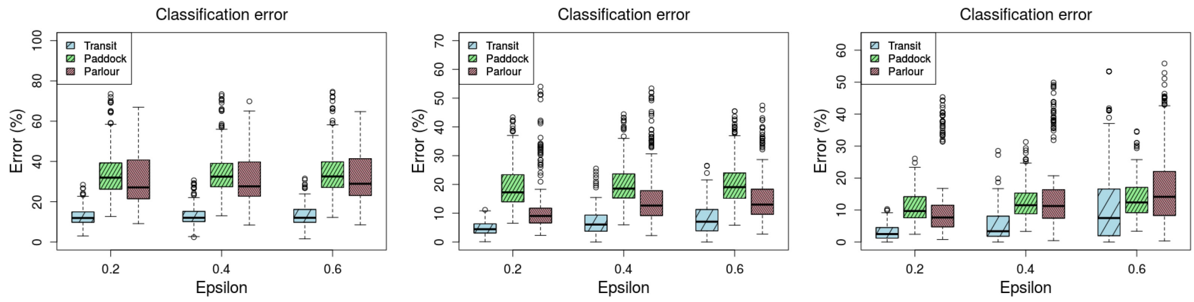


(a) (b) (c)

Fig. 17. Effect of ε on reduction of IEM classifications for *windowSize* (a) 10s (b) 30s (c) 60s.

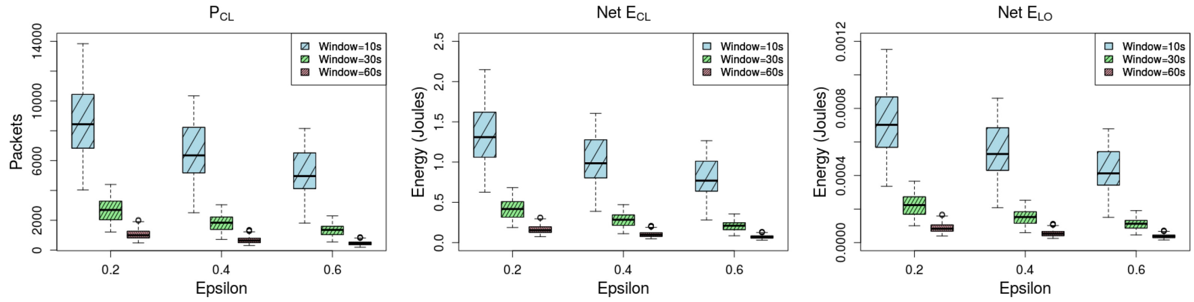
of the data traces are classified if a change in signal distribution ≥60% is considered significant for classification. A similar trend in reduction percentage is observed for window sizes 30s and 60s. Moreover, the value of reduction is higher for larger window sizes, as the smoothing in data is increased such that the small fluctuations in the signal are concealed, resulting in fewer changes in the distribution that exceed the threshold. Figure 17 illustrates the trend in computation reduction for different values of ε and *windowSize*. The reduction in classification not only improves the memory usage by storing fewer readings in the flash but can also improve energy profile of the devices. This could, in turn, result in an increase in the operational time of the wearable devices.

Next, we examine the effect of ε on the classification accuracy of the three activity states: transit (T), paddock (P), and parlour (M), for the three window sizes. We calculate error as the percentage of misclassified states per trace across all ten iterations, as shown in Figure 18. For a small window of 10s, the value of ε has very little impact on the classification accuracy. The median error of transit, for instance, increases from 11.9% at $\varepsilon = 0.2$ to 12% at $\varepsilon = 0.4$ and $\varepsilon = 0.6$, as shown in Figure 18(a); that is, an approximate increase of 1%, compared to the resultant error at $\varepsilon = 0$ (see Figure 16(a)). We associate the small changes in the median errors with the nature of smoothing in the data. For a small window, the smoothing is very low, such that the slightest change in the distribution exceeds the threshold value. As such, a small ε filters out the redundant data (see Figure 17) and maintains the quality of the results. The effect of ε is more prominent for larger window sizes (30s and 60s) due to increased smoothing, as shown in Figures 18(b) and 18(c). An increase in error is observed with increase in the value of ε. Moreover, the change in error with ε value is greater, compared to the change in errors for *windowSize* = 10. However, an overall



(a) (b) (c)

Fig. 18. Effect of ε on classification error of IEM for *windowSize* (a) 10s (b) 30s (c) 60s.



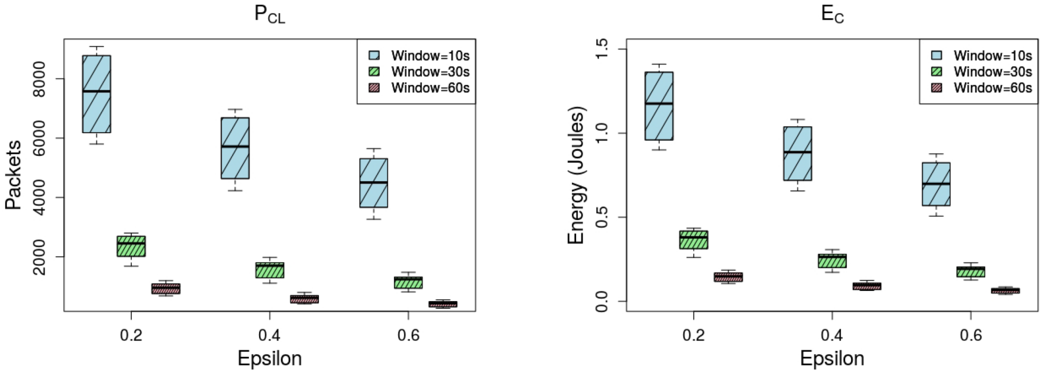
(a) (b) (c)

Fig. 19. Effect of *windowSize* and ε on (a) P_{CL} (b) Net E_{CL} (c) Net E_{LO} for scenario P1.

reduction in the error values is observed for all states with increase in the *windowSize* as local min and max in the signals are accurately captured.

As mentioned earlier, the accuracy of IEM affects the frequency of packet transmissions to the cloud and, in turn, the communication cost incurred by the sensor nodes. We assume that each node sends a single packet to the cloud per state change. The energy cost incurred by each node in sending one packet to the cloud (E_{CL}) is calculated in Equation (11). For our analysis, we set the constants $e = 50x10^{-9}J$ and $\beta = 10^{-11}J/bit \cdot m^2$ [28], $d = 120m$ (maximum radio range of CM5000 motes for outdoor), and $bits = 800$ (maximum payload of 802.15.4 packets). Ideally, it suffices to run the IEM algorithm on one node (IN) to localize a given herd within the farm (denoted as scenario P1 in Table 2). Each time the IN predicts a change in its activity state, it assumes the same change in state across the entire herd and forwards the location update to the remaining nodes within the herd. All nodes then transmit their sensor data along with the location information to the cloud gateway. The energy cost incurred by each node due to the local communication between nodes (E_{LO}) is calculated in Equation (12). We set $d' = 20m$ (usual maximum distance between neighboring cows within a herd) and $bits' = 1$ (payload required for sending location update). We calculate the total number of packets sent by all nodes to the cloud (P_{CL}) per day and resultant net E_{CL} and E_{LO} values for the network by considering each node as IN for different values of *windowSize* and ε. Since each node has a different prediction accuracy, the value of P_{CL}, and net energies also varies. The median values for all nodes over ten iterations are listed in Table 2.

Despite the increase in classification error, the value of P_{CL} reduces with increase in the value of ε for a fixed window size (Figure 19(a)). This is attributed to the significant drop in the number of classifications at higher ε values that results in fewer predictions and, in turn, a lesser number of state changes. Note, however, the error in classification is higher, owing to the prolonged effect of a misclassified state and delay in detecting state changes. The value of P_{CL} further reduces



(a) (b)

Fig. 20. Effect of *windowSize* and ε on (a) P_{CL} (b) E_C for scenario P5.

with increase in *windowSize*, owing to better smoothing in the signal that reduces within-the-state misclassifications and thereby prevents untimely state-change predictions. A similar trend is observed in the values of resultant communication energies E_{CL} and E_{LO} with changes in the input parameters, as shown in Figures 19(b) and 19(c). Whereas $E_{CL} = 1.31J$ for $windowSize = 10$ and $\varepsilon = 0.2$, it reduces to 0.77J with increase in ε to 0.6, and further reduces to 0.07J with increase in the value of *windowSize* to 60s. Similarly, $E_{LO} = 0.70e^{-3}$ for $windowSize = 10$ and $\varepsilon = 0.2$, and reduces to $0.41e^{-3}$ with increase in ε to 0.6, and further to $0.04e^{-3}$ for $windowSize = 60$ and $\varepsilon = 0.6$. As is evident, the energy cost incurred by the local communications is significantly lower, compared to energy spent in the long-range communication to the cloud gateway. The network communication energy, in this case, is calculated as the summation of net E_{CL} and E_{LO}. Furthermore, we consider the scenario where each node runs the IEM algorithm and predicts its activity state in isolation (denoted as scenario P5 in Table 2). That is, the nodes do not communicate locally with each other and directly send data packets to the cloud at the occurrence of individual state changes. We calculate the total packets sent by all five nodes to the cloud per day (P_{CL}) and resultant energy cost E_{CL} for different values of *windowSize* and ε over ten iterations (as illustrated in Figure 20). The value of P_{CL} and, thereby, E_{CL}, follows the same trend with increasing *windowSize* and ε values as P1. Moreover, the median values for P_{CL} and net E_{CL} in P5 are lower than the corresponding values in P1, as shown in Table 2. This is because, whereas in P5 the packet transmissions are governed by a node's own accuracy, transmissions in P1 are guided by the accuracy of one node. As a result, the number of packets increase across all nodes if the IN has poor accuracy.

5.2.3 Effect of Coalition Size. As discussed above, the performance of CASMM is primarily governed by the coalition size. Given that our pilot study includes five nodes, we consider four possible scenarios based on the coalition sizes 2,3,4, and 5 and evaluate the performance for each coalition group shown in Table 3 for a fixed $windowSize = 60$. $N' = 2$, for instance, represents the scenario where two of the five nodes form a coalition and participate in the analysis. We study the effect of coalition size on classification accuracy, net packet transmissions for cooperation (P_{CO}), local communication (P_{LO}) and cloud communication (P_{CL}), and the resultant energy E_C that comprises of E_{CL}, E_{LO}, and E_{CO}. The median values for the above metrics across traces for all ten iterations are summarized in Table 4.

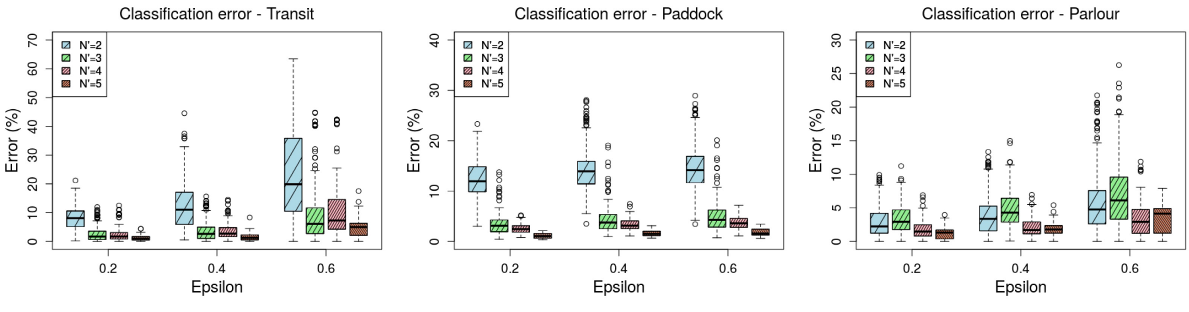
We calculate the classification error for each state, considering the network as a whole by comparing the shared network state with the observed states. The error values are affected by both coalition size and selection of nodes (as participating nodes have different accuracy), as depicted in Figure 21. As can be seen, the error varies for a particular value of N' (owing to the selection

Table 3. Coalition Groups

$N' = 2$	$N' = 3$	$N' = 4$	$N' = 5$
{N2,N3}	{N2,N3,N4}	{N2,N3,N4,N5}	{N2,N3,N4,N5,N6}
{N2,N4}	{N2,N3,N5}	{N2,N3,N4,N6}	
{N2,N5}	{N2,N3,N6}	{N2,N3,N5,N6}	
{N2,N6}	{N2,N4,N5}	{N2,N4,N5,N6}	
{N3,N4}	{N2,N4,N6}	{N3,N4,N5,N6}	
{N3,N5}	{N2,N5,N6}		
{N3,N6}	{N3,N4,N5}		
{N4,N5}	{N3,N4,N6}		
{N4,N6}	{N3,N5,N6}		
{N5,N6}	{N4,N5,N6}		

Table 4. Performance Summary of IEM (with Collaboration) for $windowSize = 60$

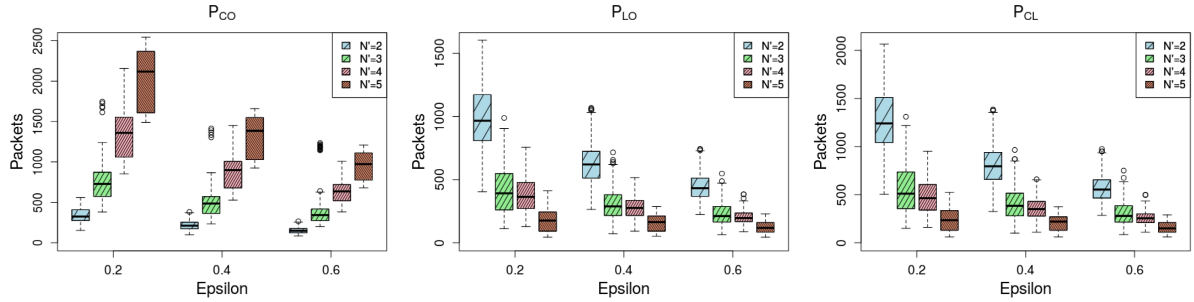
		$N' = 2$			$N' = 3$			$N' = 4$			$N' = 5$		
Metric		$\varepsilon = 0.2$	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.2$	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.2$	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.2$	$\varepsilon = 0.4$	$\varepsilon = 0.6$
Error(%)	T	8.1	11.0	19.8	1.7	2.6	6.1	1.7	2.7	7.3	1.0	1.2	5.0
	P	12.0	13.9	14.1	3.1	3.8	4.3	2.4	3.1	3.6	1.0	1.5	1.6
	M	2.2	3.4	4.7	2.9	4.3	6.1	1.5	1.7	2.9	1.3	1.8	4.1
Packets	P_{CO}	325	211	149	728	486	342	1,361	900	636	2,118	1,386	974
	P_{LO}	966	620	432	392	288	212	364	276	196	176	164	118
	P_{CL}	1,240	795	553	510	385	280	463	350	255	235	220	150
Energy (J)	E_{LO}+E_{CO}	$0.17e^{-3}$	$0.11e^{-3}$	$0.08e^{-3}$	$0.19e^{-3}$	$0.13e^{-3}$	$0.10e^{-3}$	$0.33e^{-3}$	$0.22e^{-3}$	$0.15e^{-3}$	$0.46e^{-3}$	$0.31e^{-3}$	$0.21e^{-3}$
	E_{CL}	0.19	0.12	0.09	0.08	0.06	0.04	0.07	0.05	0.04	0.04	0.03	0.02
	E_C	0.19	0.12	0.09	0.08	0.06	0.04	0.07	0.05	0.04	0.04	0.03	0.02



(a) (b) (c)

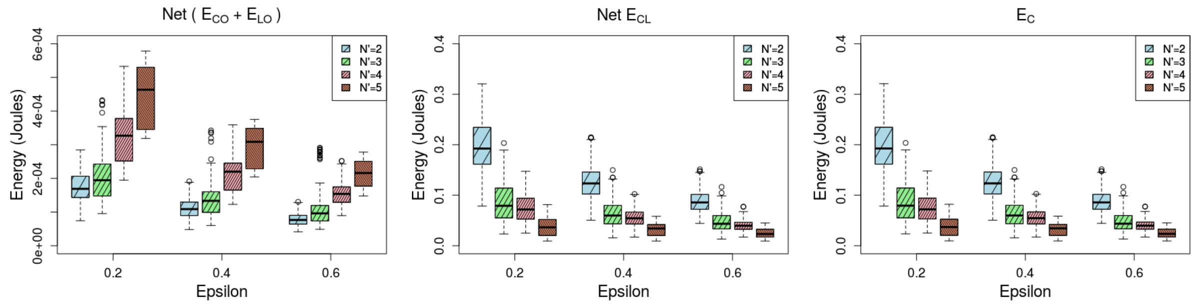
Fig. 21. Effect of coalition size on classification error for (a) Transit (b) Paddock (c) Parlour states at $windowSize = 60$.

of nodes) and decreases with increase in N' from 2 to 5. This decrease in error is achieved, as classification errors of a node with low accuracy are masked by the accurate classification of other nodes via majority voting. Moreover, the error values are lower when compared to Table 2, with the exception of $N' = 2$, wherein majority implies the vote of one node against the other. As expected, the errors increase with increase in the ε value due to reduced frequency of classification. For $N' = 5$, the cooperation achieves an accuracy >98% for all three states at $\varepsilon = 0.2$ and ≥95%



(a) (b) (c)

Fig. 22. Effect of coalition size on (a) P_{CO} (b) P_{LO} (c) P_{CL} at $windowSize = 60$.



(a) (b) (c)

Fig. 23. Effect of coalition size on (a) Net local energy ($E_{CO} + E_{LO}$) (b) Net E_{CL} (c) E_C at $winsdowSize = 60$.

at $\varepsilon = 0.6$, with a corresponding reduction in classifications by 96.2% and 98.2%, respectively (see Table 2). Although it is feasible to further reduce the number of computations by increasing the ε value, it will adversely affect the system accuracy. The values of the input parameters should, therefore, be chosen such that they balance the trade-off between the number of classifications and accuracy to meet the application requirements.

Next, we consider the impact of N' on P_{CO}, P_{LO}, and P_{CL}. The values depict net packet transmissions for the network of five nodes, as shown in Figure 22. We observe an increase in the value of P_{CO} with an increase in the value of N' (Figure 22(a)). This is due to an increase in the number of participating nodes that are polled during cooperation. On the contrary, a decrease in the values of P_{LO} and P_{CL} is observed with increase in N' (Figures 22(b) and 22(c)), owing to the improved accuracy. For a fixed N', the values of P_{CO}, P_{LO}, and P_{CL} decrease with increase in the ε value, due to reduced number of classifications on each node. Moreover, the packet transmissions are lower, compared to P1 and P5 scenarios discussed earlier, with the exception of $N' = 2$, which has lower accuracy. Similar trends are observed in the resultant communication energies, as shown in Figure 23. Figure 23(a) illustrates the net energy cost for local communication between devices, i.e., net $E_{CO} + E_{LO}$. While an increase in E_{CO} and decrease in E_{LO} is expected with increase in N', we observe a net increase in the local communication energy due to higher impact of E_{CO} (as $P_{CO} > P_{LO}$). On the contrary, a drop in E_{CL} is observed with increase in N', owing to improved accuracy and fewer packet transmissions to the cloud (Figure 23(b)). The net communication energy (E_C) is then calculated using Equation (11) and depicted in Figure 23(c). Since the magnitude of local communication cost is significantly lower when compared to the cost for cloud communication, the value of E_C mimics the value of E_{CL}. Moreover, the value decreases with increase in the coalition size N', thereby improving the network efficiency. Similar to the packet transmissions, the energy costs further decrease with increase in the ε value.

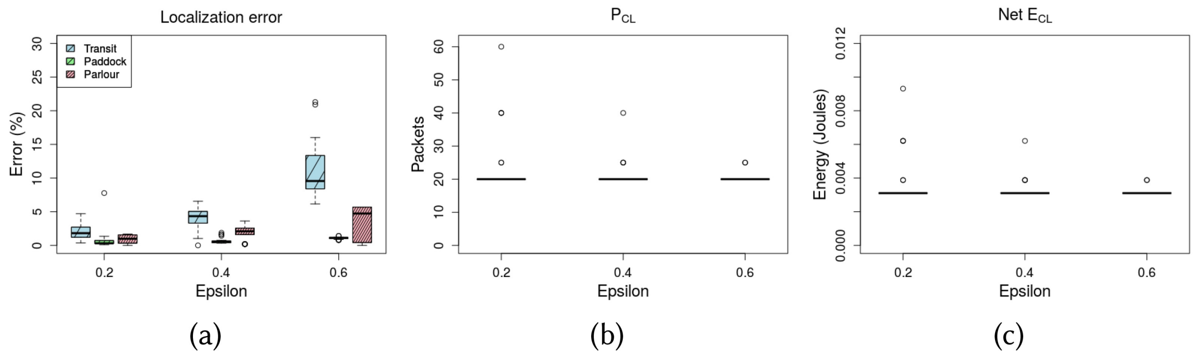


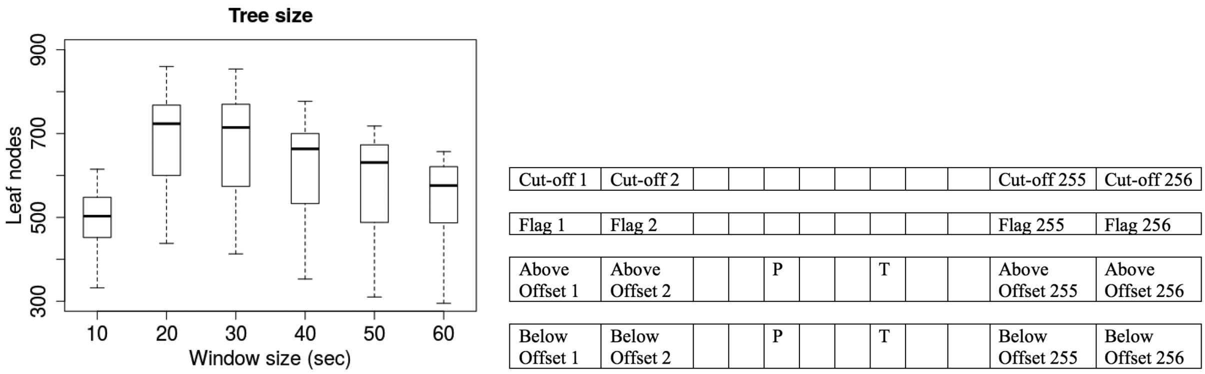
Fig. 24. Effect of ASMM on (a) Localization error (b) P_{CL} (c) Net E_{Cl} for $windowSize = 60$ and $N' = 5$.

While the cooperation significantly improves the classification accuracy, the number of state transitions detected by the system (resulting into P_{CL}) is quite high. As such, our pilot study incorporates four state changes ($M_0 \to T_0 \to P_0 \to T_1 \to M_1$) and should result into exactly four packet transmissions to the cloud. This implies that our system detects untimely state changes (within-the-state errors) that are short-lived (suggested by high accuracy) but occur frequently (suggested by value of P_{CL}). We expect the ASMM approach to address such errors by mapping the sequence of activities to the farm topology and reducing the within-the-state errors. We consider the coalition groups for $N' = 5$ for the analysis, as they allow highest accuracy of classification along with minimum E_C. As mentioned above, we use the eight decile value based on the distribution of errors to calculate threshold $\mathcal{T}$ for each state (depicted as T_t, P_p, and M_m in Figure 7). ASMM accepts a change in state detected by the cooperation only if it is consistent with the topology (follows the state transition diagram) and continues for a period assigned by $\mathcal{T}$. Figure 24 shows the effect of ASMM on the location accuracy, P_{CL}, and E_{CL}.[4] As can be seen, the accuracy for all three states does not alter significantly (Figure 24(a)) and closely resembles the values achieved after cooperation (see Figure 21). Note that the localization accuracy is calculated in terms of percentage, as we consider high-level localization of cows in three discrete regions. On the contrary, the median of number of packets transmitted to the cloud reduces remarkably to 20, i.e., 4 packets per node as desired (Figure 24(b)). That is, ASMM eliminates all the untimely state transitions. Resultantly, it leads to a significant drop in the value of E_{CL}. As shown in Figure 24(c), the value of E_{CL} drops to less than 10%, compared to Figure 23(b), i.e., a reduction of 90%. The error in classification can be explained by early or delayed detection of state changes, owing to the use of $\mathcal{T}$ parameter.

6 DISCUSSION AND FUTURE WORK

In the previous section, we evaluated the performance of the IEM2.0-CASMM model for different values of the input parameter. The analysis shows that while the stand-alone IEM classifier can achieve a reasonable level of accuracy (>90% for $windowSize = 60$ and $\varepsilon = 0.2$) for all three activity states along with very low frequency of classifications (a reduction of >96% for $windowSize = 60$ and $\varepsilon = 0.2$), it results in a considerable number of unnecessary and expensive packet transmissions to the cloud. The CASMM method improves the accuracy of IEM-based classification (~99% for $N' = 5$, $windowSize = 60$, and $\varepsilon = 0.2$) through cooperation between devices with very low overhead energy costs of the order of 10^{-4} and facilitates accurate localization via ASMM. The ASMM eliminates the unnecessary packet transmissions to the cloud, thereby improving the overall energy efficiency of the WSN operation by 90%. The analysis, thus, confirms the suitability of

[4] ASMM has no effect on the local communication between the devices.



(a) (b)

Fig. 25. (a) Effect of *windowSize* on the size of *DT* (b) Array-based implementation of DT.

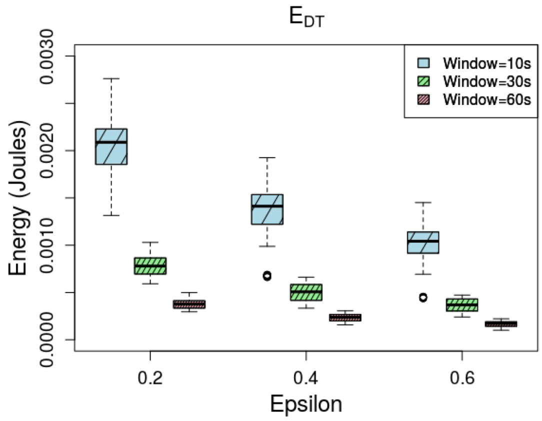
using the IEM2.0-CASMM approach for activity recognition and localization of the cows. In this section, we assess the feasibility of implementing the IEM classifier *DT* on-board the sensor devices. We discuss an array-based implementation of the IEM algorithm and present a memory analysis for the same. In comparison to *DT*, implementation of CASMM only requires a few variables, such as N', state vector, and $\mathcal{T}$ to be maintained by the device. In addition, we evaluate the energy cost associated with the *DT*-based classification (E_{DT}) for different values of the input parameters. Last, we present the proposed future work.

6.1 Memory Analysis

Figure 25(a) shows the effect of *windowSize* on the size of *DT* in terms of number of leaf nodes. As can be seen, the number of nodes increase as we move from *windowSize* 10 to 20s and follows a downward trend thereafter with further increase in the *windowSize*. Accordingly, while a median value of ~700 is obtained for $windowSize = 30$, it decreases to 580 for $windowSize = 60$. However, the number of leaf nodes is as high as 850 nodes for certain cases with $windowSize = 20/30$. We use this upper case to calculate the memory requirements for IEM-based *DT* and verify its feasibility for sensor-based execution. We present an array-based implementation of a *DT* with 850 leaf nodes, as shown in Figure 25(b). We require four arrays of length 850 each. The first array holds the cut-off values used at the decision nodes in *DT* to split the data into two subsets. As mentioned in Section 4.2, the range of the acceleration values of cows is −2g to +2g. We scale down the measurements such that they range between −1g and +1g. The cut-off values can then be represented as $0.int$ and would require 2 bytes per reading; that is, a total of $850 * 2$ bytes is required for the first array. The second array is used to store flags that indicate whether the cut-off value sets a constraint on the windowed min or max. Each flag requires 1 bit, adding up to $850 * 1/8$ bytes. The third and the fourth arrays provide link to the child nodes—left child and right child. If the child is a leaf, a label "P" for paddock, "T" for transit, or "M" for parlour is assigned to the appropriate index variable. Otherwise, the variable contains an offset value for the pointer to the first array for subsequent decisions along the *DT*. Each entry in both arrays requires 1 byte to store the value and totals to $850 * 1 * 2$ bytes for both arrays. The net memory required for the IEM implementation, thus, equals 3.4KB ($1KB = 1024bytes$). The CM5000 mote used in our prototype, for instance, features a program flash memory of 48KB. This analysis, thus, validates the suitability of IEM for on-board implementation on the resource-constrained sensor devices. Furthermore, the generic nature of the implementation suggests that IEM can also be incorporated in the commercially available wearable sensor devices such as RumiWatch [32] and SmartBow [6].

Operator	Data type	E_{op} (mean)	E_{op} (variance)	E_{op} (error)
$<$	float	4.23E-8	8.17E-17	2.92E-17
$\geq$	float	4.08E-8	5.35E-17	0.0
assign	float	8.27E-8	5.29E-18	2.30E-9

(a)



(b)

Fig. 26. (a) Energy consumption per operation [41] (b) Effect of *windowSize* and ε on E_{DT}.

6.2 Computation Energy Cost

While we discuss the optimization of network communication cost in Section 5, it is important to evaluate the energy consumed by on-board analysis to ensure that it does not significantly impact the sensing and communication tasks. The conventional approach to evaluate the power consumption involves periodic measurement of remaining battery level on physical hardware, as presented in Reference [38]. Although this approach provides accurate analysis, it has several limitations, including potential hardware and human failures, complexity and size of WSN, as well as inherent dynamism of the environment. Alternatively, the use of modelling has been proposed to evaluate the power consumption of WSN applications. In Reference [39], for instance, the authors use Colored Petri Nets (CPN) tools to automatically generate consumption models for given NesC [40] (programming language used in TinyOS) operators, structures and functions to, in turn, estimate the energy cost of an entire application. While this approach may have slightly less accuracy, it provides flexibility and agility to evaluate energy consumption in complex application scenarios in a timely and cost-effective manner. We, therefore, adopt the approach presented in Reference [39] for calculating the energy cost associated with DT-based classification.

Using the CPN tools, DT-based classification can be modelled as a sequence of relational operations, i.e., $\geq$ comparisons. The power consumption for each classification can, thus, be calculated as the product of the total number of operations to traverse the DT (N_{op}) and energy consumed per operation (E_{op}). The value of N_{op} is governed by the tree size and is typically calculated as the log base 2 of the total number of nodes in a tree (see Figure 25(a)). To estimate E_{op}, the CPN models discussed for NesC operators in Reference [39] make use of an auxiliary function, namely *addEnergy*. The function is assumed to follow a normal distribution and generates a random value for each instruction's power consumption using given energy mean and variance values. The values of mean and variance are specific to each operator and have been estimated using measurements. We obtain these values for relational and assignment operations from a Github repository [41], as shown in Figure 26(a). The net energy for classification (E_{DT}) per node is then calculated using the following equation:

$$E_{DT} = (N_{op} \cdot E_{op}) \cdot N_{class},$$

where N_{class} is total number of classifications on a given node. The value of N_{class} can be calculated as the percent readings that are classified from each trace (see Figure 17). Figure 26(b) illustrates the effect of *windowSize* and ε on E_{DT}. As expected, the energy consumption decreases with increase in values of both window size and ε, owing to fewer classifications. Furthermore, the value of E_{DT} is of the order of $10^{-3} J$ for different values of the input parameters. For a $windowSize = 60$, the median value of E_{DT} is below 0.0005, thus, validating the suitability of IEM (v. 2.0) for sensor-based execution.

6.3 Future Work

In this work, we present proof-of-concept for our WSN-based localization approach. In the future, we intend to deploy the trained IEM2.0-CASMM model on wearable sensor devices to test the approach in real time. Moreover, we wish to address the scalability of our approach across a larger set of devices. We also plan to assess the impact of CASMM on the response time of the system. Since the initiating node in a coalition waits for a response from all the participating devices before making a decision, a large coalition size may lead to an increase in response time. In this case, a deadline by which all responses must be received may be used to meet the application response time requirements. A trade-off between the quality of result and application deadline should, thus, be considered. Furthermore, since accuracy of individual nodes affects the combined performance of a coalition, we wish to study the effect of selection of nodes for forming a coalition. In addition, we wish to design handover of the analysis to other nodes in the vicinity as the energy level of participating nodes depletes below a given threshold.

7 CONCLUSION

In this article, we show the suitability of using the IEM-2.0 approach for classifying Mixed Gaussian signals (especially with unequal distributions) and analyze the performance of our IEM2.0-CASMM-based localization approach for animal-activity recognition and localization in dairy farms. The performance evaluation is based on real-world mobility data of cows and shows that the IEM2.0-CASMM approach can achieve a localization accuracy of 99% with very low frequency of classifications. With such high accuracy of localization, a location-aware event-driven communication approach is used to transfer sensor data to the cloud. Such an approach consumes energy of the order 10^{-4} and significantly improves the energy efficiency of the WSN operation. Furthermore, memory analysis for the approach shows that it requires only 3.4KB of the program flash and is suitable for implementation on wearable sensor devices. On-board implementation of IEM2.0-CASMM on animal wearables would allow uninterrupted context-aware sensing in Cooperative WSN, as cows move around a farm despite the lack of continuous Internet connectivity. This, in turn, would facilitate real-time LBS within the farm as well as early detection of behavior anomalies that may indicate health-related issues. As IEM is applicable for classification of generic Mixed Gaussian signals, our approach can be extended to different WSN applications.

REFERENCES

[1] T. Wang, Z. Peng, J. Liang, S. Wen, Md Z. A. Bhuiyan, Y. Cai, and J. Cao. 2016. Following targets for mobile tracking in wireless sensor networks. *ACM Trans. Sen. Netw.* 12, 4 (2016). DOI: http://dx.doi.org/10.1145/2968450

[2] CERES TAG. 2018. What is ceres tag. Retrieved from cerestag.com.

[3] T. Wark, D. Swain, C. Crossman, P. Valencia, G. Bishop-Hurley, and R. Handcock. 2009. Sensor and actuator networks: Protecting environmentally sensitive areas. *IEEE Perv. Comput.* 8, 1 (2009), 30–36. DOI: http://dx.doi.org/10.1109/MPRV.2009.15

[4] A. Maddumabandara, H. Leung, and M. Liu. 2015. Experimental evaluation of indoor localization using wireless sensor networks. *IEEE Sens. J.* 15, 9 (2015), 5228–5237.

[5] T. Stoyanova, F. Kerasiotis, C. Antonopoulos, and G. Papadopoulos. 2014. RSS-based localization for wireless sensor networks in practice. In *Proceedings of the 9th International Symposium on Communication Systems, Networks Digital Sign (CSNDSP'14)*. 134–139.

[6] SmartBow. 2014. User manual: E093 SmartBow eartag. Retrieved from smartbow.com. (accessed Oct. 2017).

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. 2012. Fog computing and its role in the internet of things. In *Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing*. 13–16. DOI: http://dx.doi.org/10.1145/2342509.2342513

[8] B. Zhou, Q. Li, Q. Mao, W. Tu, and X. Zhang. 2015. Activity sequence-based indoor pedestrian localization using smartphones. *IEEE Trans. Hum.-Mach. Syst.* 45, 5 (2015), 562–574. DOI: http://dx.doi.org/10.1109/THMS.2014.2368092

[9] DAIRYMASTER. 2013. What is moomonitor+. Retrieved from moomonitor.dairymaster.com.

[10] HerdInsights. 2015. How HerdInsights works. Retrieved from herdinsights.com.

[11] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic. 2013. Edge mining the internet of things. *IEEE Sens. J.* 13, 10 (2013), 3816–3825. DOI: http://dx.doi.org/10.1109/JSEN.2013.2266895

[12] K. Bhargava, S. Ivanov, C. Kulatunga, and W. Donnelly. 2017. Fog-enabled WSN system for animal behavior analysis in precision dairy. In *Proceedings of the International Conference on Computing, Networking and Communications (ICNC'17)*. 504–510. DOI: http://dx.doi.org/10.1109/ICCNC.2017.7876180

[13] L. Mainetti, L. Patrono, A. Secco, and I. Sergi. 2016. An IoT-aware AAL system for elderly people. In *Proceedings of the International Multidisciplinary Conference on Computer and Energy Science (SpliTech'16)*. 1–6. DOI: http://dx.doi.org/10.1109/SpliTech.2016.7555929

[14] X. Huang, Y. Li, Y. Wang, X. Chen, Y. Xiao, and L. Zhang. 2018. CTS: A cellular-based trajectory tracking system with GPS-level accuracy. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 140 (2018), 140:1–140:29 pages. DOI: http://dx.doi.org/10.1145/3161185

[15] R. Mohamed, H. Aly, and M. Youssef. 2017. Accurate real-time map matching for challenging environments. *IEEE Trans. Intell. Transport. Syst.* 18, 4 (2017), 847–857. DOI: http://dx.doi.org/10.1109/TITS.2016.2591958

[16] X. Li, D. Zhang, Q. Lv, J. Xiong, S. Li, Y. Zhang, and H. Mei. 2017. IndoTrack: Device-free indoor human tracking with commodity Wi-Fi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 72 (2017), 72:1–72:22 pages. DOI: http://dx.doi.org/10.1145/3130940

[17] B. Xie, K. Chen, G. Tan, M. Lu, Y. Liu, J. Wu, and T. He. 2016. LIPS: A light intensity–based positioning system for indoor environments. *ACM Trans. Sen. Netw.* 12, 4 (2016). DOI: http://dx.doi.org/10.1145/2953880

[18] Q. Yuan and I. M. Chen. 2013. 3-D localization of human based on an inertial capture system. *IEEE Trans. Robot.* 29, 3 (2013), 806–812. DOI: http://dx.doi.org/10.1109/TRO.2013.2248535

[19] D. Ayllón, H. A. Sánchez-Hevia, R. Gil-Pita, M. U. Manso, and M. R. Zurera. 2016. Indoor blind localization of smartphones by means of sensor data fusion. *IEEE Trans. Instr. Meas.* 65, 4 (2016), 783–794. DOI: http://dx.doi.org/10.1109/TIM.2015.2494629

[20] Z. Chen, Q. Zhu, and Y. C. Soh. 2016. Smartphone inertial sensor-based indoor localization and tracking with iBeacon corrections. *IEEE Trans. Industr. Inform.* 12, 4 (2016), 1540–1549. DOI: http://dx.doi.org/10.1109/TII.2016.2579265

[21] OpenFog Consortium Architecture Working Group. 2017. OpenFog reference architecture for fog computing. Retrieved from openfogconsortium.org.

[22] S. Sarkar, S. Chatterjee, and S. Misra. 2015. Assessment of the suitability of fog computing in the context of internet of things. *IEEE Trans. Cloud Comput.* PP, 99 (2015). DOI: http://dx.doi.org/10.1109/TCC.2015.2485206

[23] R. Tan, G. Xing, B. Liu, J. Wang, and X. Jia. 2012. Exploiting data fusion to improve the coverage of wireless sensor networks. *IEEE/ACM Trans. Netw.* 20, 2 (2012), 450–462.

[24] S. Gomes Soares, A. F. da Rocha, T. M. G. de A. Barbosa, and R. A. de Matos Araújo. 2010. Embedding a neural network into WSN furniture. In *Proceedings of the 10th International Conference on Hybrid Intelligent Systems*. 219–222.

[25] D. Goldsmith and J. Brusey. 2010. The Spanish Inquisition protocol: Model-based transmission reduction for wireless sensor networks. In *IEEE Sensors*. 2043–2048. DOI: http://dx.doi.org/10.1109/ICSENS.2010.5690285

[26] E. I. Gaura, J. Brusey, and R. Wilkins. 2011. Bare necessities—Knowledge-driven WSN design. In *Proceedings of the IEEE Concerence on Sensors* 66–70. DOI: http://dx.doi.org/10.1109/ICSENS.2011.6127213

[27] R. Rednic, E. Gaura, and J. Brusey. 2009. ClassAct: Accelerometer-based real-time activity classifier. In *Proceedings of the Sensors & Instrumentation KTN: Wireless Sensing Demonstrator Showcase (WiSIG'09)*. 1–6.

[28] M. Abo-Zahhad, O. Amin, M. Farrag, and A. Ali. 2014. Survey on energy consumption models in wireless sensor networks. *Open Trans. Wireless Comm.* 1, 1 (2014), 63–79.

[29] C. J. Rutten, A. G. Velthuis, W. Steeneveld, and H. Hogeveen. 2013. Invited review: Sensors to support health management on dairy farms. *J. Dairy Sci.* 96, 4 (2013), 1928–1952. DOI: http://dx.doi.org/10.3168/jds.2012-6107

[30] M. Alsaaod, C. Römer, J. Kleinmanns, K. Hendriksen, S. Rose-Meierhöfer, L. Plümer, and W. Büscher. 2012. Electronic detection of lameness in dairy cows through measuring pedometric activity and lying behavior. *Appl. Animal Behav. Sci.* 142, 3 (2012), 134–141. DOI: http://dx.doi.org/10.1016/j.applanim.2012.10.001

[31] M. Trotter et al. 2012. Monitoring and managing landscape variability in grazing systems. In *Proceedings of the 15th Precision Agriculture Symposium*. 42–48.

[32] J. Werner et al. 2017. Evaluation of the RumiWatchSystem for measuring grazing behaviour of cows. *J. Neurosci. Meth., Elsevier* (2017). DOI: http://dx.doi.org/10.1016/j.jneumeth.2017.08.022

[33] Advanticsys. 2017. MTM-CM5000-MSP. Retrieved from advanticsys.com.
[34] Invensense. 2014. MPU-9255 product specification (rev. 1.0). Retrieved from stanford.edu.
[35] Wiki. 2015. TinyOS documentation wiki. Retrieved from tinyos.stanford.edu/tinyos-wiki.
[36] Raspberry Pi Foundation. 2016. Raspberry Pi 2 model B. Retrieved from raspberrypi.org.
[37] Md. B. Uddin and C. Castelluccia. 2010. Toward clock skew–based wireless sensor node services. In *Proceedings of the 5th ICST Wireless Internet Conference (WICON'10)*. 1–9. DOI: http://dx.doi.org/10.4108/ICST.WICON2010.8503
[38] K. Shinghal, A. Noor, N. Srivastava, and R. Singh. 2011. Power measurements of wireless sensor network node. *Int. J. Comput. Eng. Sci.* 1, 1 (2011), 8–13.
[39] Antônio Dâmaso, Davi Freitas, Nelson Rosa, Bruno Silva, and Paulo Maciel. 2013. Evaluating the power consumption of wireless sensor network applications using models. *Sensors* 13, 3 (2013), 3473–3500. DOI: http://dx.doi.org/10.3390/s130303473
[40] David Gay, Philip Levis, Robert von Behren, Matt Welsh, Eric Brewer, and David Culler. 2003. The nesC language: A holistic approach to networked embedded systems. *SIGPLAN Not.* 38, 5 (May 2003), 1–11. DOI: http://dx.doi.org/10.1145/780822.781133
[41] Antônio Dâmaso, Davi Freitas, and Nelson Rosa. 2013. nesc2cpn. Retrieved from github.com/sensor2model-group/nesc2cpn.

Received July 2018; revised November 2018; accepted January 2019

Appendix I

Cooperative In-network Computation in Energy Harvesting Device Clouds

Journal Title:	Sustainable Computing: Informatics and Systems
Article Type	Journal
Complete Author List	Chamil Kulatunga, Kriti Bhargava, Dixon Vimalajeewa and Stepan Ivanov
Status	Published: vol. 16, pp. 106-116, Dec. 2017
Contribution	My contribution to this research article was identification of state-of-the-art in cooperative computing within WSN, recognizing limitations of existing techniques and modelling energy consumption of computation and communication tasks. I also took part in writing some parts of the paper and proof-reading.

Sustainable Computing: Informatics and Systems 16 (2017) 106–116

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Cooperative in-network computation in energy harvesting device clouds

Chamil Kulatunga*, Kriti Bhargava, Dixon Vimalajeewa, Stepan Ivanov

Telecommunications Software and Systems Group, Arclabs Research and Innovation Centre, Waterford Institute of Technology, Carriganore, Waterford, Ireland

ARTICLE INFO

Article history:
Received 10 July 2017
Accepted 9 October 2017
Available online 16 October 2017

Keywords:
In-network analytics
Cooperative computing
Computation offloading
Energy harvesting
Low-latency applications
Fog computing

ABSTRACT

The Internet of Things paradigm is creating an environment where the big data originators will be located at the edge of the Internet. Accordingly, data analytic infrastructure is also being relocated to the network edges, to fulfill the philosophy of data gravity, under the umbrella of Fog computing. The extreme edge of the hierarchical infrastructure consists of sensor devices that constitute the wireless sensor networks. The role of these devices has evolved tremendously over the past few years owing to significant improvements in their design and computational capabilities. Sensor devices, today, are not only capable of performing sense and send tasks but also certain kinds of in-network processing. As such, triple optimization of sensing, computing and communication tasks is required to facilitate the implementation of data analytics on the sensor devices. A sensor node may optimally partition a computation task, for instance, and offload sub-tasks to cooperative neighbouring nodes for parallel execution to, in turn, optimize the network resources. This approach is crucial, especially, for energy harvesting sensor devices where the energy profile and, therefore, the computation capability of each device differs depending on the node location and time of day. Accordingly, future in-network computing must capture the energy harvesting information of sensor nodes to jointly optimize the computation and communication within the network. In this paper, we present a theoretical model for computation offloading in micro-solar powered energy harvesting sensor devices. Optimum data partitioning to minimize the total energy consumption has been discussed based on the energy harvesting status of sensor nodes for different scenarios. The simulation results show that our model reduced both energy losses and waste due to energy conversion and overflows respectively compared to a data partitioning algorithm that offloads computation tasks without taking the energy harvesting status of nodes into consideration. Our approach also improves energy balance of a WSN which is an important factor for its long-term autonomous operation.

1. Introduction

With a growing number of devices in the Internet of Things (IoT) and high adopt-ability of cloud-based Big Data analytic platforms, the centralized cloud computing architecture has been recently challenged within the Internet community. Conventional cloud computing had been designed for monolithic applications assuming high availability of resources at large data centres. It saved CPEX for SMEs, particularly, the overall energy consumption of maintaining an Information and Communication Technologies (ICT) infrastructure. Furthermore, centralized clouds optimized resource utilization by statistically multiplexing peak-loads to avoid over-provisioning. This architecture functioned well until IoT devices generated some large datasets in remotely connected application domains such as smart agriculture [9] and Industry 4.0 [1]. Fog computing [26], is a new computing paradigm, that proposes the analysis of data (before aggregating it into big data sets) in a hierarchical and scalable way closer to the data sources. Although the term was coined by Cisco in 2012, the philosophy of data gravity where computation moves towards the data sources as far as they can, had been presented by Dave McCrory in 2010. Harnessing the computational power of the network devices for data processing has the potential to not only reduce the data in the backhaul network and, in turn, the latency experienced by the end users but also improve the overall energy consumption of the IoT platforms [10].

* Corresponding author.
E-mail addresses: ckulatunga@tssg.org (C. Kulatunga), kbhargava@tssg.org (K. Bhargava), dvimalajeewa@tssg.org (D. Vimalajeewa), sivanov@tssg.org (S. Ivanov).

https://doi.org/10.1016/j.suscom.2017.10.006
2210-5379/

This is particularly useful for applications in rural agriculture and Industry 4.0 where backhaul connectivity is limited between the remote rural farms/factories and the cloud [7].

A number of interpretations of Fog nodes have been proposed, to date. Authors in [2], for instance, discusses Mobile Edge Computing where mobile operators leverage resources of the edge devices in 5G rather than the centralized servers used in cloud computing for data processing. Several forms of ad-hoc cloudlets (micro-clouds) have been proposed in [4,18]. Certain studies have also extended the concept of Fog computing towards the extreme edge of the IoT in the private, enterprise, and community domains. This is primarily due to the design of pervasive low-power wireless technologies like ULP-PAN and LP-WAN as well as the tremendous improvement in computation capabilities of small devices (as mini-servers) such as CCTV cameras, mobile phones, and more recently, sensor devices that constitute Wireless Sensor Networks (WSN) [8]. In-network processing within WSN (referred here as in-network analytics) has been performed using different techniques such as data fusion, aggregation, compression and feature extraction [25,21].

It is of particular importance in latency-sensitive applications such as object tracking, intrusion detection, monitoring structural and machine failures, where the result of the processing may not be useful at all times, the response time at event detection is of the order of fraction of a second. As a result, while numerous studies in the past have focused on optimizing sensing and networking tasks to improve the energy efficiency of WSN, attention is being drawn towards triple optimization that includes on-board computation given the increased capabilities of sensor nodes. Maximizing computation within WSN through resource optimization is more desirable as future sensor nodes will be powered via energy harvesting, for continuous use, from background sources such as solar, wind, vibration and radio frequency [15].

Cooperative computing via computation offloading has been suggested for maximizing the use of in-network computational resources. In computation offloading, a device can select (sometimes in an opportunistic way [5,16]) a proximate infrastructure edge device (gateway) or another stationary or mobile device as an offloadee for parallel execution of tasks at different participating nodes [19]. Collaborative computing within WSN can enhance the capabilities of the resource constrained environment towards effective cyber-foraging approaches as shown in [20]. Multi-objective intelligent decisions can be made to optimize Fog computing resources and their application performance. The decision of how to optimally partition a task and where to offload given a completion time is an important research question which has not been much investigated in the literature. An analytical model for application partitioning in battery-powered WSN environment has been presented in [20]. An initiating node (IN) that is responsible for sensing data is designed that offloads partial computation to a neighbouring node known as the cooperating node (CN) such that the given task completion deadline is met while optimizing the energy resources of the network.

In this work, we consider in-network computation in WSN [14] and extend the cooperative computing approach discussed in [20] for different scenarios in an energy harvesting WSN. While in conventional WSN, the IN offloads less computation to CN owing to high communication energy, in case of energy-harvested nodes, the partitioning must be based on the level of stored energy as well as the current state of the device that determines the level of harvested energy. This is important to avoid over-flow of harvested energy (hence an energy waste) when battery is fully charged or energy conversion efficiency (75–65%) incurred by storing harvested energy into battery. Accordingly, we develop models for task partitioning to reduce the overall energy consumption of the network under different scenarios for latency-sensitive applications. Furthermore, we aim at improving the fairness within the network to ensure energy balancing. Our model and the simulation results show that our approach enables optimization of computation and communication for future energy harvested WSN and ensures sustainable operation.

2. Computational policies for clean energy

A node in a conventional sensor network forwards data without changing the payload. Instead, in-network processing allows a Fog node to not only function as a data source or merely relay a data chunk but also perform some computation on the data. In the early days of in-network processing, researchers were limited to a particular application within a sensor network such as calculation of average humidity or identifying a location of an event based on statistically correlated data aggregation. However, this is changing to embed more generic computational functionalities in WSN.

2.1. In-network cooperative computing in wireless sensor networks

In-network processing has been applied for data aggregation, fusion, compression and feature abstraction in WSN to save energy by reducing the number of bits and, in turn, data packets transmitted to a centralized server. Computations are performed at specific aggregation nodes (cluster heads) along the path to the destination node (gateway or server). Offloading decisions are, therefore, simple and based on the forwarding algorithm used such as LEACH to answer the question of where rather than what. This has progressed recently to use a swarm of heterogeneous nodes (such as sensors, actuators, robots, smart phones, drones, cameras) that collectively form an in-network analytic platform and requires specification of where as well as what to send. Authors in [11] propose for instance a new in-network computation algorithm based on channel fading to improve the reliability of aggregation function compared to simultaneously sending all or only one sensor reading.

Computation offloading is a useful distributed computing paradigm at different levels of network resources from large data centres to implanted nano-sensors. Highly available cloud computing provides VM/container level computing resources to the users to perform computation tasks in geographically distributed data centres. Mobile edge computing brings cloud resources into the edge of the operator-managed network to reduce core network traffic of the operator and provide low-latency for the users. Enterprise and community-cloud allow the installation of micro data centres that execute micro-services at the proximity of a company office or a community. The concept of cloudlets proposes the use of a set of mobile devices (different users) that collectively form an ad-hoc cloud [13]. Mobile computation offloading, for instance, can facilitate the execution of compute intensive tasks either on a nearby mobile (in terms of annotations) or on an infrastructure node (e.g. Androidx86).

Computation offloading in WSN is becoming increasingly important as the sensor devices exhibit improved capabilities in terms of computation power and reduced communication energy consumption. In conventional networks, sensor nodes transmit raw data to the sink node where some processing is performed and the results are communicated to the remote cloud. As a result, sensor nodes have prior knowledge of where and what to communicate. Moreover, the energy optimization is included in the algorithms. In modern-day WSN, sensor nodes can make on-the-fly decisions of where and what to compute under a subjected application completion deadline and, in turn, optimize energy usage. Therefore, the pre-designed computation offloading algorithms must be modified to make on-the-fly decisions. Accordingly, energy harvesting and in-network processing can be combined to develop a sustainable and autonomous network operation.

2.2. Heterogeneity in energy harvesting sensor nodes

Computational sensor nodes, in future, will be powered using diverse natural energy harvesting sources such as solar, wind, radio-frequency, thermal, vibration or piezoelectric [22]. Such energy sources demonstrate random spatial–temporal generation patterns leading to heterogeneity in stored energy between sensor nodes in both outdoor and indoor environments. Changes in the temporal patterns might be significant only on a macro time scale. For instance, while weather may differ from one city to another on a single day at a given time, a sensor network on a smart farm will experience the same effect at the same time. On the contrary, spatial variations among co-located mobile sensor nodes may be obtained due to different orientations and obstacles, for e.g., presence of IMU and GPS modules [3] for animal mobility and location tracking under direct sunlight vs shadows. This heterogeneity will be higher, particularly, in outdoor WSN such as those used in agricultural practices for pasture-based dairy farming (e.g. laying animals with solar-covered tags), site-specific irrigation in cultivation (e.g. leaves may grow into or fall onto the senor nodes) and soil monitoring (e.g. shadows of the plants may cover the soil monitoring sensors).

Optimal energy management in such environments has been proposed using adaptive duty cycling, adaptive communication strategies, routing decision making and application policy management. Authors in [27], for instance, propose optimization of the duty cycle to maximize the common active time based on unpredictable heterogeneity of energy harvesting nodes. The authors propose both online and offline algorithms based on the probability of the harvested energy obtained using a real deployment environment.

We consider cooperation between such sensor nodes to collectively perform computation tasks under a heterogeneous energy harvesting environment. For example, each sensor node in such a scenario could partially perform some pre-processing or basic functional tasks such as averaging or compressing data. Balancing energy usage with computation offloading is important in such a Fog resource pooling environment due to three perspectives.

(a) Energy harvesting incurs a significant conversion loss while storing energy into a storage device like a battery or a capacitor. It accounts for about 25–35% of the total energy in battery storage and even higher for capacitors [27]. It is, therefore, preferable to use harvested energy directly whenever possible so as to minimize the conversion losses. Accordingly, any computation offloading to a node which is currently on solar power has a safe margin to use some energy to compensate for the communication overheads.

(b) If the amount of harvested energy is low, the system cannot perform both the charging and direct energy use operations together. That is, when the amount of harvested energy (E) is below a threshold (θ) a node must decide to either store the energy or use it directly but not both. Usually, in such situations, the most appropriate action is to store the harvested energy and consume the required energy from the battery. Therefore balancing stored energy within the nodes of a WSN is highly advantageous.

(c) Rechargeable batteries are a costly unit for energy harvesting sensor nodes. Therefore, they may have some limited capacity. Cooperative computing between the sensors is critical in such networks to optimize the energy usage via load balancing and avoid overflow of energy on nodes that are fully-charged with no computation task or energy deficit for others. Therefore, balancing energy consumption without using high capacity batteries is a positive trend in future WSN using energy harvesting.

2.3. Related work

Mobile computation offloading has been widely researched in the recent years with varied objectives such as energy saving, transparent code migration and scalability. An optimal technique for application partitioning and fair node selection between two homogeneous nodes has been discussed in [24]. Computation offloading in WSN, however, did not gain much attention until Sheng et al. [20] proposed optimal application partition and cooperation between two nodes to minimize overall energy consumption. Their work is based on cooperation between battery-powered homogeneous sensor nodes and assumes no selfish node behaviour. A cooperating node selection strategy that balances trade-off between fairness and energy consumption has been discussed.

Meanwhile, energy harvesting sensor nodes are becoming widely deployed and several studies discuss the heterogeneity in harvesting energy [15]. Dang et al. [6] presents predictive solar energy models for spatial–temporal weather conditions. Authors in [27] propose a stochastic duty cycling approach to minimize energy consumption by taking into account the heterogeneous energy harvesting sensor networks. In [25], authors discuss the importance of triple optimization of sensing, networking and in-network data processing based on energy harvesting. The authors have implemented an optimization algorithm to recycle wasted energy due to battery overflow in an energy harvesting WSN. In this paper, we extend the work done by [20] and propose an approach to balance the energy in computational sensor network using cooperative computing in energy harvesting networks. We apply this approach for the scenario where certain solar powered sensor nodes are under sunlight while others are obstructed by shadows for a certain duration within a day.

3. Modelling for cooperative computation

In this section, we present our application model, computation and communication energy consumption models, and the micro-solar based energy harvesting model.

3.1. Application model

In this work, we consider a lightweight analytic application that consists of a set of independent processing tasks to be computed cooperatively between two peer sensor nodes. We use the canonical model used in [28] to capture the essential characteristics of such a task-oriented application. Such tasks are normally arranged in a computational work-flow using a Dynamic Acyclic Graph (DAG) to be scheduled for execution in a distributed computing environment. A single processing task (A) is modelled with input data size (D) and a deadline for application completion (T). The Initiating Node (IN), which may be responsible for sensing the data, divides a single task into two sub-tasks for partial offloading to a target remote peer, referred to as the Cooperating Node (CN). The amount of processing data at the local node is denoted by L and the amount of data that is offloaded to the CN is denoted as R, where $D = L + R$. We assume there are no dependencies between the sub-tasks. For instance, in case of calculating average for a sensing variable, L and R may consist of n_L and n_R samples respectively. Note that, only R amount of input data is offloaded to the CN with no extra amount of code. We also assume that the response or the outcome of the processing sub-task at each node is negligible or locally consumed by another process. In the mentioned average calculation example, the local node will transmit only two values, which is the local average (A_L) and n_L, while CN will transmit its own local average (A_R) and n_R. An aggregation or the destination node will then calculate the overall average using the two responses from IN and CN.

3.2. Computation energy model

The energy consumption in embedded processors is dominated by dynamic power and can be regulated by the clock frequency using dynamic voltage and frequency scaling (DVFS) technique. Several attempts have been made to develop a simple and general computation energy estimation model for mobile and embedded processors. According to the literature, the computational energy consumption is proportional to the CPU load of a processor i.e. the number of CPU cycles required. Most of the work, therefore, considers the trade-off between energy (E) and task completion time (T) such that $E.\,T^{\alpha}$ is a constant for some values of α. In [24], the energy consumption for computing a task locally is calculated using Eq. (1), where K (in the order of 10^{-11} starting from ARM to Intel) is called the computation coefficient. The value of K depends on the effective switched capacity (determined by the chip architecture and the clock-frequency), the processing capability of the node, and the application completion probability used in the model in [28]. As evident in Eq. (1), a node consumes more energy for short completion deadlines T. A sensor node may, therefore, prefer more delay-tolerant tasks for local computation and offload tasks with large L and small T to a peer sensor node.

$$E_C = \frac{KL^3}{T^2} \tag{1}$$

3.3. Communication energy model

When a task is offloaded to another node, the energy used for communication depends on the number of bits transmitted [17]. This is energy consumed by the electronics in the physical layer and depends on the state of nodes – idle, transmit and receive. According to IEEE 802.15.4, energy consumption in the idle state can be neglected and, therefore, total energy consumption depends on the transmission of the number of bits at the sender and the reception of the same bits at the receiver which are equal in value but belong to two different nodes. A task can be scheduled for transmission to another node within one or more time-slots. This scheduling has been modelled using the Markov process based on whether the Additive White Gaussian Model (AWGN) channel state is good or bad. The energy used to communicate b bits within a time-slot t to another computational node depends on the path condition and the distance between the two nodes (represented as channel gain g) and is given by the following equation.

$$e = \frac{(2^b - 1)}{g}$$

According to one-shot channel allocation policy to transmit data task within a single time-slot, the scheduler must send L bits within one time-slot T. This is the simplest case in which all the data is sent within a single time-slot of communication window and the energy consumed is represented by a convex-monomial function as shown in Eq. (2).

$$E_t = \rho\frac{L^n}{g} \tag{2}$$

Here ρ is the communication coefficient of the link between the offloader and the offloaded and $g[0\cdots 1]$ is the channel gain of the link that is calculated proportional to $1/d^2$ according to AWGN in free-space propagation where d is the distance between the two nodes. According to [20], transmission in one-shot policy ($n=1$) only depends on the channel state and it is the most optimal approach for latency-sensitive applications. It also minimizes the time shift between local and remote computation since it assumes a negligible delay in over-the-air transmission. Moreover, it saves energy that is otherwise incurred by overhead scheduling due to data split across multiple time-slots.

3.4. Total energy requirement calculation per task

The total energy consumption owing to computation and communication during processing a task between two nodes can be calculated as the summation of four components as shown in Eq. (3). In [20], authors present the energy consumption for different input data sizes from 512 to 2048 bits. Here the job completion deadline is set to 20 ms, $K=5\times10^{-11}$ and $\rho=0.05$. For large data sizes, the gain in energy consumption is much better in case of using cooperative computing and varies with the values of the computation and communication coefficients. After a distance of 5 m, however, cooperative computing is not effective and localized computation becomes the preferred mode for the entire task according to their analysis.

$$\begin{aligned}\text{Total Energy}(E) &= \text{IN}\{\text{Computation} \quad L + \text{Transmission} \quad R\}\\ &+\text{CN}\{\text{Reception} \quad R + \text{Computation} \quad R\}\end{aligned} \tag{3}$$

In this paper we extend this approach by taking into account the energy harvesting state of the IN and CN nodes and also the energy conversion efficiency. We estimate the required equivalent energy (E) (i.e. before the conversion) from the energy harvesting source within the optimization algorithms.

3.5. Micro-solar energy harvesting model

We selected a latitude of 52° and longitude of −8° where the experimental smart farm for the project is located in Moorepark, Co. Cork, Ireland. We chose April 1st as the representative date of neither a winter day nor a summer day for the solar energy harvesting model. We model the solar energy harvesting pattern as a Gaussian curve (Fig. 2) with 8 h (T) clear sunlight from 8.00 am to 4.00 pm according to astronomical model developed by [23,12]. We consider a discrete time model with a time-slot of 1 min. A solar energy density of 15 mW cm^{-3} is assumed for 5 cm × 3 cm area on a micro-solar panel associated with a sensor node. This implies 735 μJ energy can be generated by a sensor node on a day without any clouds and obstacles shadowing it. We also modeled a shadow of 4 h which will randomly cover sensor nodes within the field. Micro-solar panel inclination was set to 90° and orientation to 45° in our model.

4. Energy-aware task partitioning

The aim of this work is to find the optimal data size for a task that is suitable for local computation (L) and remote computation (R) based on the state of harvested energy (under shadow, under sunlight with energy stored being under-flown and under sunlight with energy stored being over-flown) on both IN and CN. While we discuss the energy-aware application partitioning by IN and CN selection (in the following section), the energy state interchanges among the nodes using a distributed or a centralized approach is beyond the scope of this paper. The Lagrange Multiplier is used to solve the equal constrained optimization problem with an objective to minimize total required energy (E) from the solar panel at both nodes. When a task is to be processed at any given time, IN and CN may be in different states as shown in Table 1, resulting into different E_L and E_R values compared to non-energy harvesting-aware partitioning approach proposed by [20]. We calculate E accordingly as the summation of E_L and E_R values. We consider an energy gain factor λ as the reciprocal of the energy conversion efficiency in the equations for simplicity of deriving equations. For instance, λ = 1.54 represents 65% efficiency (Fig. 1) and implies that if a task consumes 10 μJ stored energy from the battery when the node is under a shadow, the value of E will be 20 μJ.

Table 1
Different energy harvesting states at IN and CN and the amount of total required energy in μJ using our energy-harvesting-aware task partitioning at $T = 20ms$. Local computing respectively consumes and computes 41.3 μJ (1024 bit).

			Initiating Node (IN)		
			Shadow	Light	
				Underflow	Overflow
Cooperating Node (CN)	Shadow		12.1 (526)	9.8 (469)	0 (1024)
	Light	Underflow	9.8 (472)	7.8 (526)	0 (1024)
		Overflow	1.6 (122)	1.1 (122)	0 (1024)

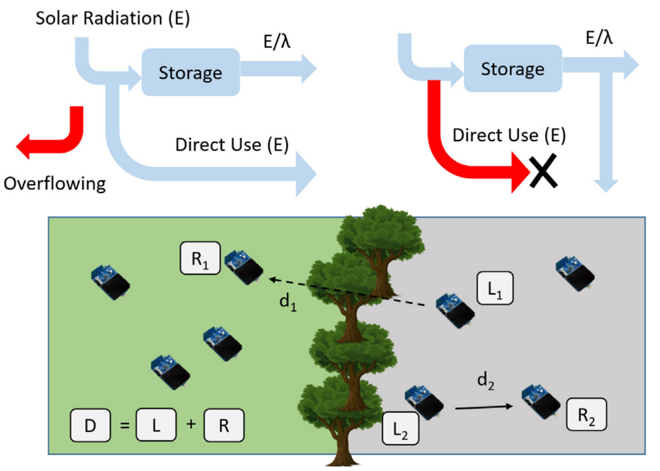


Fig. 1. Heterogeneity of energy harvested will be captured by an appropriate data partitioning and in-network computation offloading.

$$\cos\theta = \cos\alpha_p.\cos\alpha_s + \sin\alpha_p.\sin\alpha_s.\cos(\beta_p - \beta_s)$$

$$\cos\alpha_s = \sin\delta.\sin L + \cos\delta.\cos L.\cosh \qquad \sin\beta_s = -\cos\delta.\sinh/\sin\alpha_s$$

$$x = 2\pi n/365, \qquad h = 15(t-12)$$

$$\delta = 0.302 - 22.93\cos x - o.229\cos 2x + 0.243\cos 3x + 3.851\sin x + 0.002\sin 2x - 0.055\sin 3x$$

Fig. 2. The set of used energy harvesting astronomical modelling equations.

In the following sub-sections, we discuss the optimal task partitioning in terms of number of bits and the total energy required at both IN and CN to execute the task in μJ under the different IN and CN states (Table 1). The data size (D) is set to 1024 bits and the task completion deadline is changed from 5 to 100 ms. The channel gain between IN and CN is set to 0.9 and the values of K and ρ are 10^{-11} and 10^{-3} respectively. Energy gain factor $\lambda = 1.54$.

4.1. Shadow–shadow

When IN and CN are under shadow, both nodes consume energy from the stored battery power for task processing. Such a scenario does not incur any waste from the harvested energy. In this case, E can be calculated as the sum of local energy requirement E_L at IN (for computation of local task L and transmission of data R to CN) and remote energy requirement E_R at CN (for reception of data R from IN and computation of data R).

$$E = E_L + E_R = \{\alpha L^3 + \beta R\}\lambda + \{\beta R + \alpha R^3\}\lambda \tag{4}$$

On solving Eq. (4) using Lagrange constraint optimization in order to minimize E subjected to the constraint $L + R = D$, we obtain the values for L and R.

$$L = \frac{D}{2} + \frac{\beta}{3\alpha D} \text{ and } R = \frac{D}{2} - \frac{\beta}{3\alpha D}$$

Even though the amount of task partition is the same as in the non-energy harvesting case, the energy requirement is multiplied by the energy gain factor λ when we calculate the amount of surplus energy to be stored at each node. Fig. 3 shows that cooperative computing gains with low energy and the amount of the locally computed data increase with the task completion deadline. After a certain time of completion deadline, however, IN processes all the data locally and does not achieve any advantage by cooperating with a CN.

4.2. Shadow–light

In this case, the CN is under sunlight while energy is being harvested during the task processing. Therefore, remote computation R tends to be larger than in the previous case since energy required at the CN can be consumed directly from the energy harvesting source without incurring any conversion loss, if the battery is underflow (not charged up to the full capacity). Furthermore, it can use abundant energy if the battery overflows (battery fully charged and harvesting energy being wasted). Accordingly, we analyze this case separately for the two scenarios as the amount of L and R will be different.

Energy under-flowing: In this scenario, the energy is directly used from the solar panel at CN through the input regulator without incurring battery conversion loss. However, any surplus harvested energy can be stored in the CN battery without contributing towards energy waste as the battery is not charged to the full capacity. Therefore, E can be calculated as follows.

$$E = E_L + E_R = \{\alpha L^3 + \beta R\}\lambda + \{\beta R + \alpha R^3\} \tag{5}$$

On solving Eq. (5) to minimize E, we obtain the values for L and R as given below, where the value of A is obtained by solving the quadratic equation $aA^2 + bA + c = 0$ (see Appendix A) such that $L < D$.

$$L = \sqrt{\frac{A}{3\alpha\lambda)}} \text{ and } R = \sqrt{\frac{A-(1+\lambda)\beta}{3\alpha)}}$$

Furthermore, values of a, b and c are calculated as follows.

$$a = (1-\lambda)^2$$
$$b = 2\lambda(1+\lambda)\{(1-\lambda)\beta - 3\alpha D^2\}$$
$$c = \lambda^2[9\alpha^2 D^4 + \beta(1+\lambda)\{6\alpha D^2 + (1+\lambda)\beta\}]$$

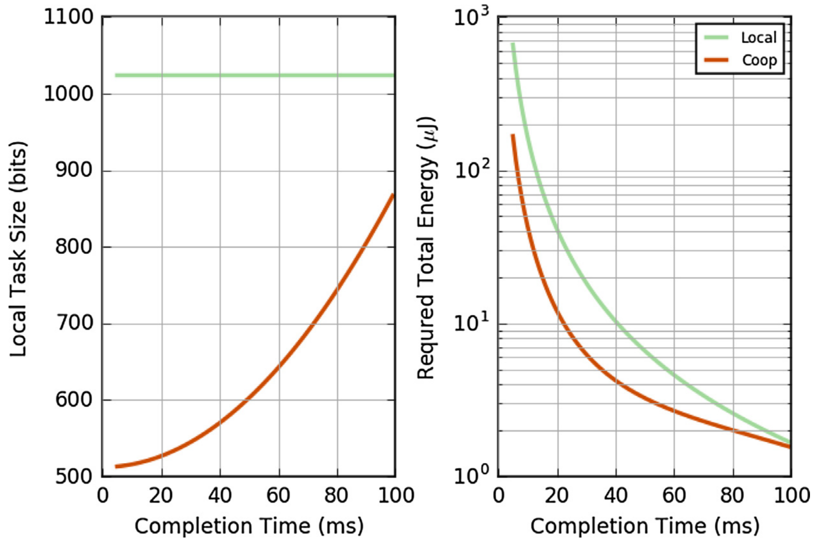


Fig. 3. Cooperative computing gains with low energy when both nodes are under shadows. However, it does not gain any energy saving when completion deadline is larger than 100 ms.

Energy over-flowing: If the battery at CN is fully charged, the energy required at CN is not considered for the total energy requirement calculation since CN in this case is wasting the harvested energy. However, transmission energy used for offloading data R to CN should be considered in the energy consumed at IN, which prevents offloading all the data D to CN.

$$E = E_L + E_R = \{\alpha L^3 + \beta R\}\lambda + \{0\} \tag{6}$$

On solving Eq. (6), we obtain $L = \sqrt{(\beta/3\alpha)}$ which is a trade-off between the required computation and communication energy at IN, and $R = D - L$. This shows that even though harvested energy at CN is wasted, IN cannot offload all the task to CN unless the completion deadline is very low.

As illustrated in Fig. 4, IN offloads more data to the CN when CN is under sunlight. We can see that if CN is overflowing, more computation can be offloaded than in the case of CN under-flowing. In case of the former, significant energy gain is observed for lower task completion deadlines when compared to the local computation only.

4.3. Light–shadow

When IN is under sunlight, the size of local computation L tends to be larger than in the previous case. This is because energy consumed at the IN can be used directly from the energy harvesting source without incurring conversion loss or from the energy being wasted according to the level of charge of the battery (similar to the previous case). Therefore, this case is also investigated under two scenarios where the amount of L and R is different.

Energy under-flowing: In this scenario, energy is directly used without conversion loss but harvested energy can be stored in the IN battery rather than being wasted. Therefore, E can be calculated as follows.

$$E = E_L + E_R = \{\alpha L^3 + \beta R\} + \{\beta R + \alpha R^3\}\lambda \tag{7}$$

On solving the optimization problem, we obtain the values for L and R as under.

$$L = \sqrt{\frac{A}{3\alpha}} \text{ and } R = \sqrt{\frac{A-(1+\lambda)\beta}{3\alpha\lambda}}$$

The value of A can be obtained by solving the quadratic equation $aA^2 + bA + c = 0$ such that $L < D$ using the following values of a, b and c.

$$a = (1-\lambda)^2$$
$$b = (1+\lambda)\{(\lambda-1)2\beta - 6\alpha\lambda D^2\}$$
$$c = 9\alpha^2\lambda^2 D^4 + (1+\lambda)\beta\{6\alpha\lambda D^2 + (1+\lambda)\beta\}$$

Energy over-flowing: In this scenario, the energy required at IN is not considered for the total required energy calculation since the node is wasting the harvested energy. Furthermore, all the computation is done locally at IN rather than offloading partial computation to CN. Accordingly, $E = E_L + E_R = 0 + 0$ and we obtain $L = D$ and $R = 0$. Fig. 5 shows that cooperative computing gains when IN is under sunlight.

4.4. Light–light

This case results in three possibilities for deciding the values of L and R. The calculation of the total required energy for each scenario is explained below.

Both nodes energy under-flowing: When both IN and CN are under sunlight without energy over-flowing, nodes can consume energy directly from the energy source and store surplus energy in the battery without any waste. In this case, E can be calculated as shown in Eq. (8), and the values of L and R can be calculated as in the shadow-shadow scenario in Section 4.1 (however the energy required at each node will be differed by a factor of λ).

$$E = E_L + E_R = \{\alpha L^3 + \beta R\} + \{\beta R + \alpha R^3\} \tag{8}$$

On solving Eq. (8) to minimize E subject to the condition $L + R = D$, we can obtain the values for L and R as under.

$$L = \frac{D}{2} + \frac{\beta}{3\alpha D} \text{ and } R = \frac{D}{2} - \frac{\beta}{3\alpha D}$$

IN energy over-flowing: In this scenario, all the processing takes place locally at the IN irrespective of the CN state and the energy required at IN is not considered for the total energy calcu-

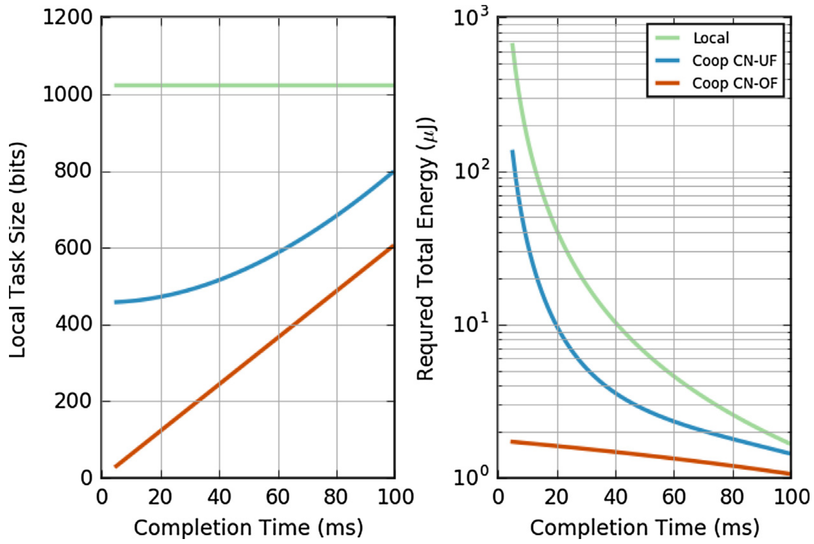


Fig. 4. IN offloads more data to CN when it is under sunlight. CN overflowing can achieve much lesser total energy consumption than underflowing scenario.

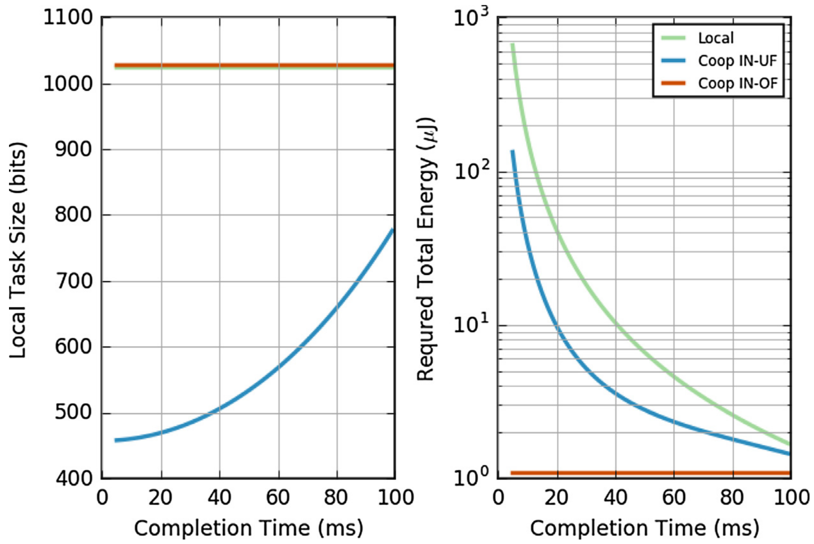


Fig. 5. Overflowing IN does not offload any data to a CN. However, underflowing IN offloads data in cooperative computing.

lation. Therefore, total energy is calculated as $E = E_L + E_R = 0 + 0$ and we obtain the $L = D$ and $R = 0$.

IN under-flowing and CN over-flowing: If the battery at CN is fully charged, the energy required at the CN is not considered for the total energy (E) calculation since CN, in this scenario, will waste the harvested energy. However, energy used for offloading data R to CN must be considered as the energy consumed at IN. Accordingly, total energy is calculated as given in Eq. (9).

$$E = E_L + E_R = \{\alpha L^3 + \beta R\} + \{\theta\} \tag{9}$$

We then obtain the value of $L = \sqrt{\beta/3\alpha}$ which is a trade-off between the required computation and communication energy at IN, and $R = D - L$. This shows that again even though harvested energy at CN is wasted, IN cannot offload all the data to CN. Also, the energy required by IN does not incur any conversion loss. Fig. 6 shows the gain in cooperative computing in these scenarios. As in the previous case, IN does not offload any data to CN in case of energy over-flowing whereas it offloads a considerable amount of data to CN when CN is over-flowing.

5. Energy-aware node selection strategy

The CN selection strategy must also be modified to make it suitable for our application model compared to non-energy harvesting scenario. In case of a non-energy harvesting environment, the minimum total energy strategy (MES), where the CN with minimum total cooperative energy cost is selected among the set of neighbouring nodes. This strategy does not consider past energy consumption (i.e. utilization). The drawback of it is that some nodes are overused due to cooperation and may lead to reduced battery lifetimes or many dead batteries, which affect the long-term autonomous functioning of the WSN. For example, a node in close proximity to a computationally-intensive node may cooperate heavily and may, therefore, be overused unfairly than what they save from cooperative computing.

In this work, CN selection is performed based on utility function as in [20], where authors define a utility function (U) based on the energy saved from cooperative computing compared to executing the complete data task locally at an IN. Our simplified utility function incorporating with the energy gain factor is given as follows.

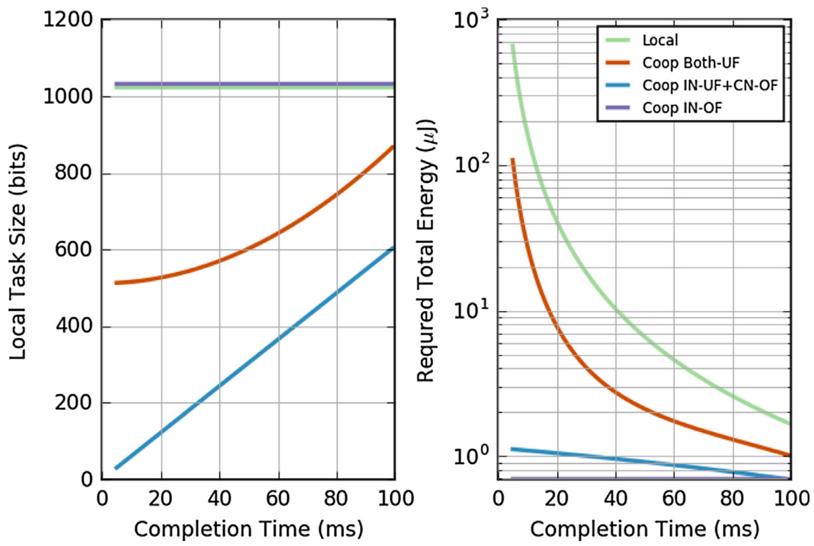


Fig. 6. IN does not offload any data when it is overflowing. However, IN does not offload all the data when CN is overflowing due to communication energy used at the IN.

$$U = \begin{cases} E_{LO} - E_L & \text{if IN} \\ -\gamma E_R & \text{if CN} \end{cases}$$

Here $E_{LO} = \frac{KD^3}{T^2}$ and $\gamma = 1$ if the CN is under sunlight and $\gamma = \lambda$ if CN is under shadow and under-flowing. The value of $\gamma = 0$ if the CN is over-flowing energy. Utility of IN will not change as the impact of the sunlight is already calculated in the required energy optimization. A Cooperation Index (CI) is then defined based on the cumulative utility as given below for $t = 0$ to $t - 1$ same as in [20]. A node can be used as a CN at time t if and only if the value of CI is positive.

$$CI = \begin{cases} 1 & \text{if } U(0:t-1) \geq 0 \\ 0 & \text{if } U(0:t-1) < 0 \end{cases}$$

This strategy is called positive utility strategy (PUS) [20]. Larger utility will have a higher chance to be selected as a CN. Designing an algorithm for this process based on the harvested energy (either in the past or predicted) is beyond the scope of this paper and remains as our future work.

6. Performance evaluation

We simulated our energy harvesting-aware computation offloading algorithm (e-COFF) with 30 energy harvesting sensor nodes using the *SimGrid* simulator.[1] Nodes were randomly located within a 10 m × 10 m geographical space. We selected latitude of 53°, where the project site is located and day of the year as 91 (01st April) in the micro-solar energy harvesting model, which harvested energy in a sinusoidal pattern within a day. We used randomly distributed obstacles for shadowing for a duration of 4 h. The size of the solar panel at a node was selected as 5 cm × 3 cm, which determined the multiplication factor of the sinusoidal harvesting pattern. We update the stored and wasted energy at each node per minute based on the harvested and the consumed energy during that period. We compared our results with non-energy harvesting-aware data offloading algorithm (COFF).

A computational task was created every 2 s randomly by a selected senor node in the WSN with a size (D) of 1024 bits. We used a maximum capacity for a battery storage of a sensor node as 2000 mAh and set it to its half at the start of the day. Harvested (E_H), required to consume (E_C), stored (E_S) and wasted (E_W) energy at the end of 24 h duration from 6.00 am were measured. Task completion time (T), harvesting energy gain factor (λ), K and ρ were set respectively as 20 ms, 1.54, 10^{-11} and 0.001 unless otherwise changed in some sections. We calculated channel gain (g) according to the

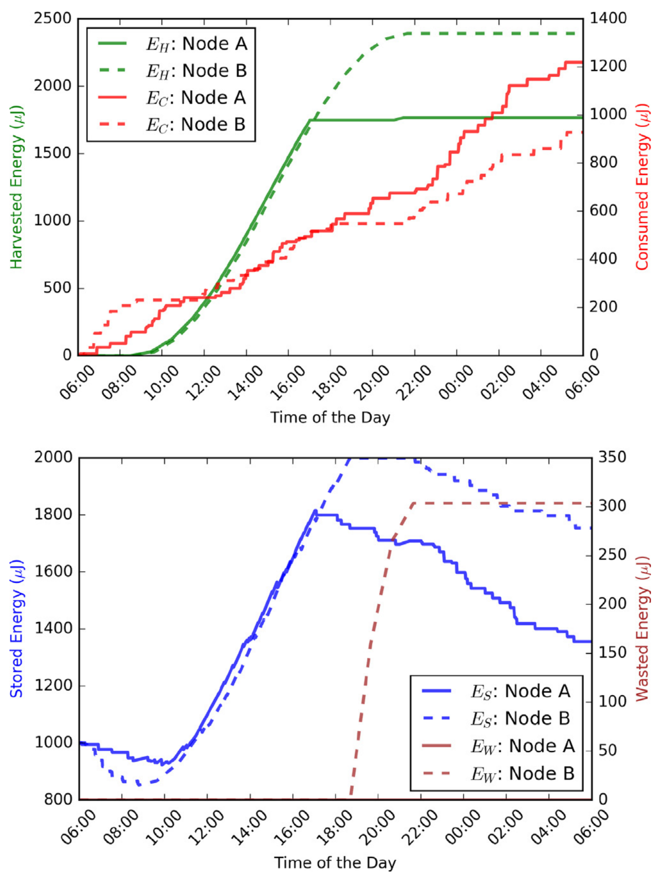


Fig. 7. The amounts of measured energy performance parameters at two different energy harvesting sensor nodes for duration of 24 h. Full battery capacity = 2000 mAh.

[1] http://simgrid.gforge.inria.fr.

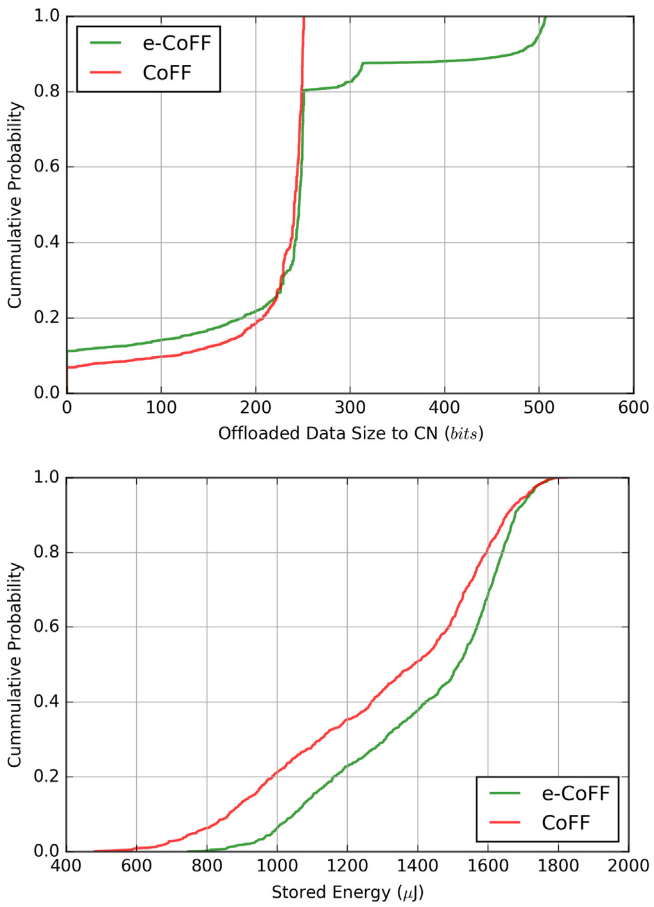


Fig. 8. Top: CDF of the sizes of data chunks being offloaded to a remote CN (R). Bottom: CDF of the stored energy (E_S) at the end of the day. $K = 10^{-11}$, $\rho = 0.001$, $D = 1024$ bits, $T = 20$ ms.

free-space wave propagation of AWGN as,

$$g = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-d^2/2\sigma^2}$$

where we selected σ as 8 in our simulations and d was calculated in the units of m.

As shown in Fig. 7, the harvested energy (E_H) of Node B does not experience any shadow while Node A experiences shadow during the day. Moreover, Node A demands slightly more energy (i.e. required energy (E_C) for task executions either as an IN or CN before being converted) than Node B. As we can see in the bottom graph, Node B saturated with stored energy (E_S) from 6.00 pm to 8.00 pm resulting in a waste of energy (E_W). Node A's battery capacity does not overflow at any given time and therefore does not experience any waste of energy. This validates our chosen relative values of energy performance parameters in order to fulfill a requirement of self-sustainability of the wireless sensor network.

We then observe the probability distribution of the offloaded task sizes to a CN (R) and the end of the day stored energy (E_S) for the two algorithms; e-COFF and COFF. The top and the bottom graphs of Fig. 8 shows the cumulative probability densities of R and (E_S) respectively with 30 different seed values set in the simulator. As we can see e-COFF offloads more data to a CN than the COFF algorithm does. The second figure shows COFF leaves with more sensor nodes towards lower energy levels at the end of the day while e-COFF leaves more stored energy towards higher energy levels.

Fig. 9 shows the difference between the consumed energy of COFF and e-COFF (E_C of COFF – E_C of e-COFF). We have changed

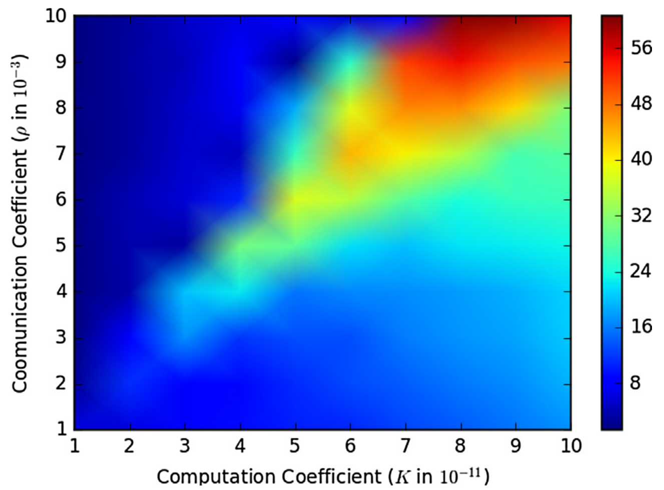


Fig. 9. The difference of consumed energy (E_C) in mJ between the energy-unaware (COFF) and our energy-aware (e-COFF) data partitioning and computation offloading algorithms (task completion deadlines = 20 ms). Both used Positive Utility Strategy (PUS) in selecting a CN.

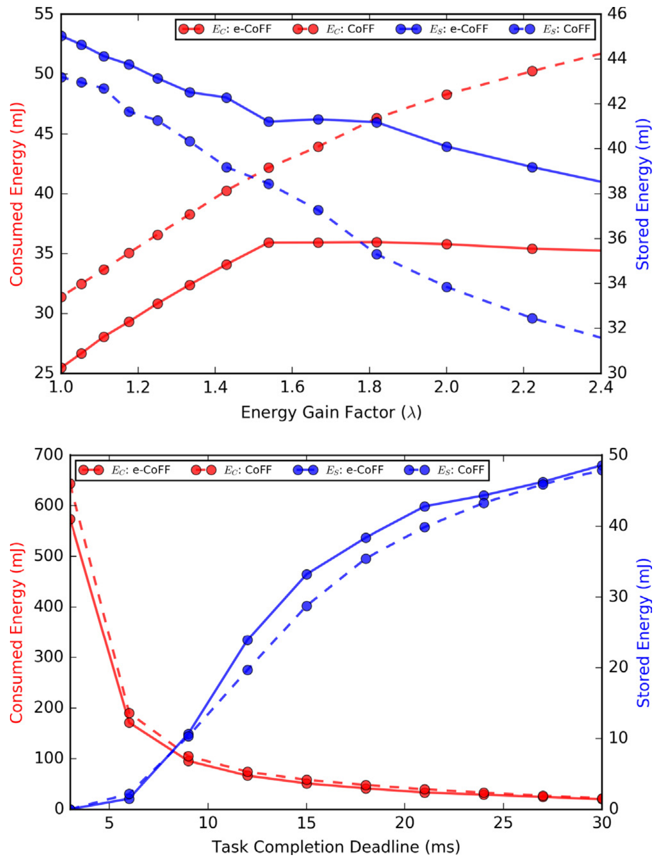


Fig. 10. Consumed and stored energy of the two algorithms for different energy gain factors (λ) when $t = 20$ ms (top) and for different task completion deadlines when $\lambda = 1.54$ (bottom).

the computation coefficients (K) in the range of 10^{-11}–10^{-10} and the communication coefficient (ρ) in the range of 0.01 and 0.001 both with a step size of 0.1. According to the figure, the performance improvements of the e-COFF is apparent for all the values of computation and communication coefficients since all the values in figure are positive. When both K and ρ are higher (top-left corner), performance improvement is significant.

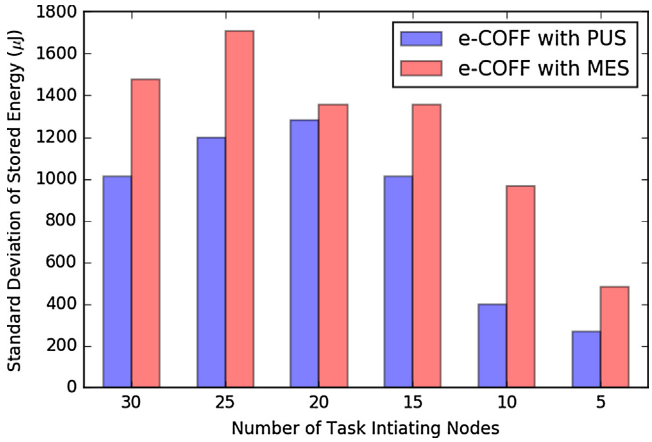


Fig. 11. The standard deviation (STD) of the stored energy (E_S), where a smaller STD indicates a better energy balance, of 30 micro-solar energy harvesting sensor nodes at the end of the day (completion deadline = 20 ms and λ = 1.54).

Next, we change (top graph in 10) the energy gain factor (λ) from 2.5 to 1.0 (i.e. energy conversion efficiency from 0.4 to 1.0) with a step of 0.5 while keeping T at 20 ms. In another experiment we also change task completion deadline (bottom graph in 10) from 5 ms to 30 ms with a step size of 5 ms while keeping λ at 1.54. Figures show the consumed energy during the day and the stored energy at the end of the day. According to the figure at the top, e-COFF shows lesser (E_C) than the COFF. Our algorithm also shows that stored energy performance is also higher compared to COFF. Performance improvement of e-COFF is much better when energy gain factor λ is low. However, the performance improvement is not very apparent for the changing range of task completion deadline (T).

We then localize the task generations only to a subset of sensor nodes to investigate the adverse impact of the overuse of energy at a CN. In this case, we reduced the number of task originating nodes from 30 (all, which is the same as before) to 5 with a step size of 5. We used two CN selection strategies; MES and PUS, with our e-COFF algorithm. Fig. 11 shows the standard deviation of the end of the day stored energy E_S, which is lower with the PUS strategy. It shows that the impact using the utilization factor in micro-energy harvesting where, if energy level of a node is low, becoming a CN persistently is critical. According to the figure use of CI solves the problem of overuse of CNs by INs in a computation intensive hotspots.

7. Conclusions

Energy-aware cooperative computing is a key technology that will benefit from energy harvesting in Fog computing applications. It is particularly important when the energy harvesting patterns and obstructions are dynamic, thereby, creating spatially heterogeneous energy sources. In this paper, we extend the optimal data partitioning algorithms developed for computation offloading by taking into account the state of the energy being harvested at the heterogeneous nodes. We evaluate our e-COFF algorithm under different scenarios and compare with COFF algorithm. Our results illustrate that overall energy consumption can be improved in a WSN by minimizing energy losses due to a poor energy conversion efficiency and waste due to energy overflows under constrained energy storage capacities. Our algorithm preformed the optimized data portioning with a positive utility cooperating node selection strategy, which balances the stored energy of the sensor nodes at the end of a day, which is useful concern for the sustainability of a WSN using micro-scale energy harvesting sources.

Acknowledgements

This work has received support from the Science Foundation Ireland (SFI) and the Agriculture and Food Development Authority, Ireland (Teagasc) as part of the SFI-Teagasc Future Agri-Food Partnership, in a project [13/IA/1977] titled "Using precision technologies, technology platforms and computational biology to increase the economic and environmental sustainability of pasture based production systems".

Appendix A

In this appendix, we discuss the optimal data partitioning for a scenario where the Initiating Node (IN) is under shadow while the Cooperating Node (CN) is under sunlight. The energy required by IN is obtained directly from the harvested energy whereas the energy required by CN is obtained from the battery. The total energy consumed is calculated as follows.

$$E = E_L + E_R = \{\alpha L^3 + \beta R\} + \{\beta R + \alpha R^3\} + A\{D - L - R\}$$

Using gradient optimization with partial derivatives, we get

$$\frac{\partial E}{\partial L} = 3\alpha L^2 - A \rightarrow L^2 = \frac{A}{3\alpha}$$
$$\frac{\partial E}{\partial R} = \beta + \lambda\beta + 3\alpha R^2 - A \rightarrow R^2 = \frac{A - (1+\lambda)\beta}{3\alpha}$$

After solving the equation $(L+R)^2 = D^2$, we get a quadratic equation $aA^2 + bA + c = 0$ to find the roots for A.

References

[1] E. Baccarelli, P. Naranjo, M. Scarpiniti, M. Shojafar, J. Abawajy, Fog of everything: energy-efficient networked computing architectures, research challenges, and a case study, IEEE Access 5 (2017) 9882–9910.
[2] S. Barbarossa, S. Sardellitti, P.D. Lorenzo, Communicating while computing: distributed mobile cloud computing over 5G heterogeneous networks, IEEE Signal Process. Mag. 31 (6) (2014) 45–55.
[3] K. Bhargava, S. Ivanov, C. Kulatunga, W. Donnelly, Fog-enabled WSN system for animal behavior analysis in precision dairy, IEEE International Conference on Computing, Networking and Communications (ICNC) (2017).
[4] M. Chen, Y. Hao, Y. Li, C.F. Lai, D. Wu, On the computation offloading at ad hoc cloudlet: architecture and service modes, IEEE Commun. Mag. 53 (6) (2015) 18–24.
[5] M. Conti, M. Kumar, Opportunities in opportunistic computing, IEEE Comput. Mag. 43 (1) (2010).
[6] N. Dang, E. Bozorgzadeh, N. Venkatasubramanian, QuARES: a quality-aware renewable energy-driven sensing framework, Sustain. Comput.: Inf. Syst. 2 (4) (2012) 171–183.
[7] M. Eto, R. Katsuma, M. Tamai, K. Yasumoto, Efficient coverage of agricultural field with mobile sensors by predicting solar power generation, 2015 IEEE 29th International Conference on Advanced Information Networking and Applications (2015) 62–69.
[8] A. Giridhar, P.R. Kumar, Toward a theory of in-network computation in wireless sensor networks, IEEE Commun. Mag. 44 (4) (2006) 98–107.
[9] A. Grogan, Smart farming, IET Eng. Technol. Mag. 7 (6) (2012).
[10] F.J.K. Hinton, R. Ayre, T. Alpcan, R.S. Tucker, Fog computing may help to save energy in cloud computing, IEEE J. Sel. Areas Commun. 34 (5) (2016) 1728–1739.
[11] S.-W. Jeon, B.C. Jung, Opportunistic function computation for wireless sensor networks, IEEE Trans. Wirel. Commun. 15 (6) (2016).
[12] J. Jeong, D. Culler, A practical theory of micro-solar power sensor networks, ACM Trans. Sens. Netw. 9 (1) (2012) 1–36.
[13] S. Jeong, O. Simeone, J. Kang, Mobile cloud computing with a UAV-mounted cloudlet: optimal bit allocation for communication and computation, IET Commun. 11 (7) (2017) 969–974.
[14] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, P. Polakos, Wireless sensor network virtualization: early architecture and research perspectives, IEEE Netw. 29 (3) (2015) 104–112.
[15] M.L. Ku, Y. Chen, K.J.R. Liu, Data-driven stochastic models and policies for energy harvesting sensor communications, IEEE J. Sel. Areas Commun. 33 (8) (2015) 1505–1520.
[16] C. Kulatunga, L. Shalloo, W. Donnelly, E. Robson, S. Ivanov, Opportunistic wireless networking for smart dairy farming, IEEE IT Prof. Mag. 19 (2) (2017) 16–23.

[17] J. Lee, N. Jindal, Energy-efficient scheduling of delay constrained traffic over fading channels, IEEE Trans. Wirel. Commun. 8 (4) (2009) 1866–1875.
[18] G.A. Lewis, S. Echeverría, S. Simanta, B. Bradshaw, J. Root, Cloudlet-based cyber-foraging for mobile systems in resource-constrained edge environments, Companion Proceedings of the 36th International Conference on Software Engineering (2014).
[19] A. Mtibaa, K.A. Harras, K. Habak, M. Ammar, E. Zegura, Towards mobile opportunistic computing, IEEE Conference of Cloud Computing (2015).
[20] Z. Sheng, C. Mahapatra, V. Leung, M. Chen, P. Sahu, Energy efficient cooperative computing in mobile wireless sensor networks, IEEE Trans. Cloud Comput. (2015) 1.
[21] I. Stojmenovic, Machine-to-machine communications with in-network data aggregation, processing, and actuation for large-scale cyber-physical systems, IEEE Internet Things J. 1 (2) (2014) 122–128.
[22] S. Sudevalayam, P. Kulkarni, Energy harvesting sensor nodes: survey and implications, IEEE Commun. Surv. Tutor. 13 (3) (2011) 443–461.
[23] J. Taneja, J. Jeong, D. Culler, Design, Modeling, and Capacity Planning for Micro-Solar Power Sensor Network, 2008.
[24] L.M. Vaquero, L. Rodero-Merino, Finding your way in the fog: towards a comprehensive definition of fog computing, ACM SIGCOMM Comput. Commun. Rev. 44 (25) (2014).
[25] S. Yang, Y. Tahir, P.Y. Chen, A. Marshall, J. McCann, Distributed optimization in energy harvesting sensor networks with dynamic in-network data processing, IEEE INFOCOM 2016 – The 35th Annual IEEE International Conference on Computer Communications (2016) 1–9.
[26] B. Zhang, N. Mor, J. Kolb, D.S. Chan, K. Lutz, E. Allman, J. Wawrzynek, E. Lee, J. Kubiatowicz, The cloud is not enough: saving IoT from the cloud, 7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15) (2015) 12–21.
[27] J. Zhang, M. Wang, Z. Li, Stochastic duty cycling for heterogeneous energy harvesting networks, 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC) (2015) 1–9.
[28] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, D.O. Wu, Energy-optimal mobile cloud computing under stochastic wireless channel, IEEE Trans. Wirel. Commun. 12 (9) (2013) 4569–4581.

Chamil Kulatunga received BSc in Electronics and Telecommunication Engineering degree from University of Moratuwa, Sri Lanka in 1999, MSc in Telecommunications from Waterford Institute of Technology (WIT), Ireland in 2003 and PhD in Internet Engineering from University of Aberdeen, UK in 2009. He is currently working as a post-doctoral researcher in Telecommunications Software and Systems Group (TSSG), WIT, Ireland. His research interests include Distributed Data Analytics, Fog Computing, Internet QoS and Smart Agriculture.

Kriti Bhargava received Bachelor's-Master's of Technology degree in Computer Science and Engineering from LNM Institute of Information Technology, Jaipur, India, in 2014. She is currently pursuing PhD in Science at TSSG, WIT, Ireland. Her research interests include Internet of Things, Fog Computing and Data Mining.

Dixon Vimalajeewa received Bachelor's of Science degree in Mathematics and Statistics from University of Ruhuna, Sri Lanka, in 2012 and Master's of Science degree in Computational Engineering from Lappeenranta University of Technology, Finland, in 2015. He is currently pursuing PhD in Science at TSSG, WIT, Ireland. His research interests include Distributed Machine Learning Algorithms, Data Analytics and Mathematical Modelling.

Stepan Ivanov received Diploma (with honours) in Applied Mathematics and Informatics from Moscow State University, Russia, in 2007 and PhD in Science from WIT, Ireland, in 2013. He is currently working as a post-doctoral researcher in TSSG, WIT, Ireland. His research interests include Wireless Communications, Internet of Things, Fog Computing and Edge Analytics.