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.

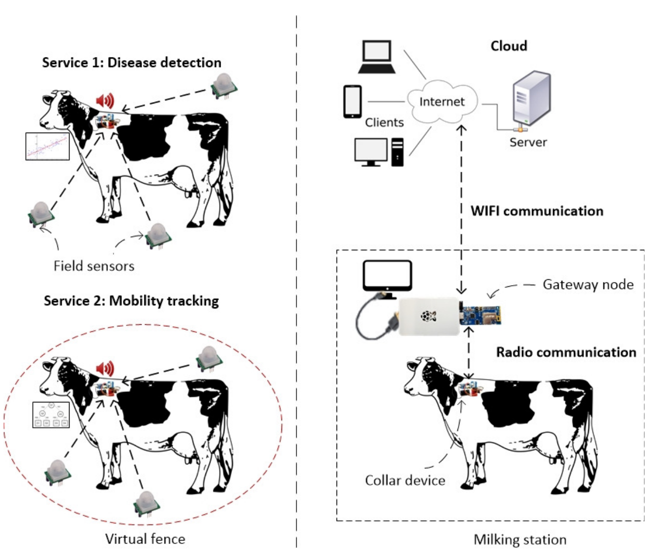

Fig. 1. Delay-tolerant networking framework for precision dairy farming

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

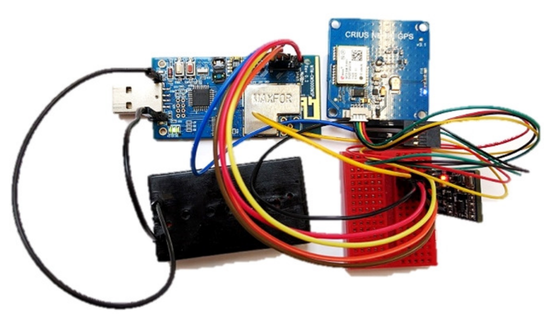

Fig. 2. Collar device comprising of CM5000 mote connected externally to a 10 DOF IMU and ublox NEO6 GPS receiver

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 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.

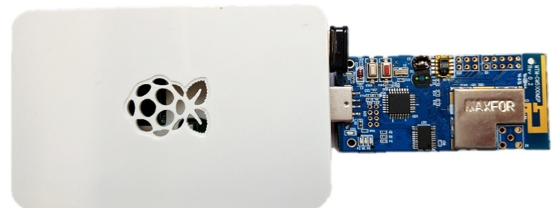

Fig. 3. Cloud gateway consisting of a CM5000 mote connected to a Raspberry Pi (model 2B)

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

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:

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}

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

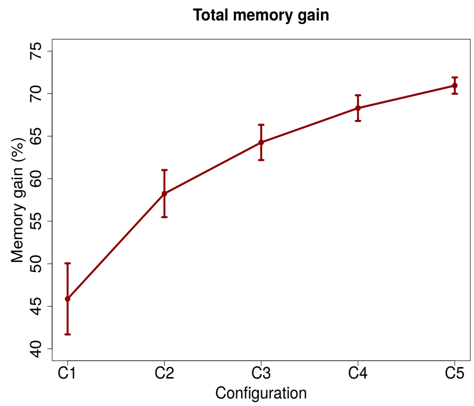

Fig. 4. Total memory gain averaged over 8 iterations for different configurations

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

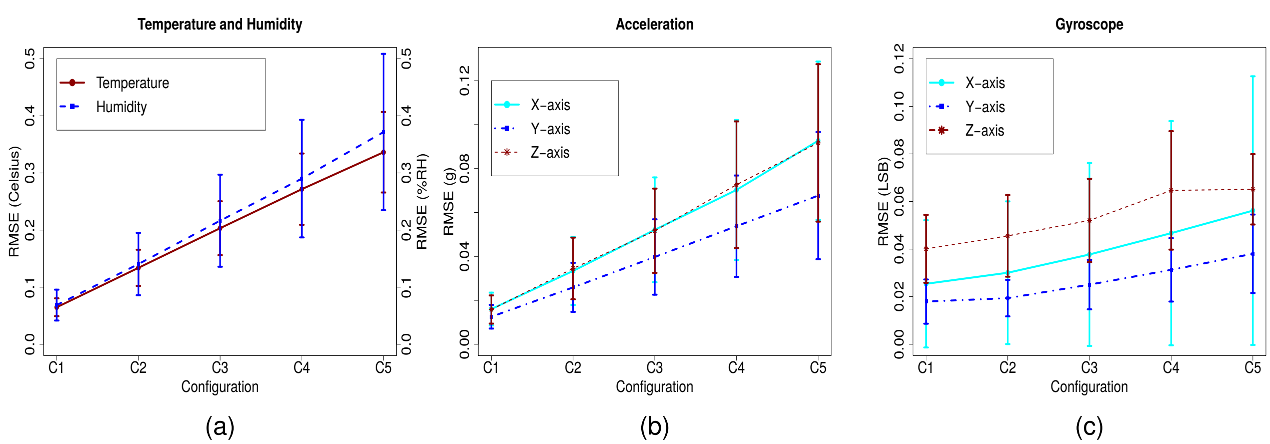

Fig. 5. RMSE for all signals averaged over 8 iterations for different configurations

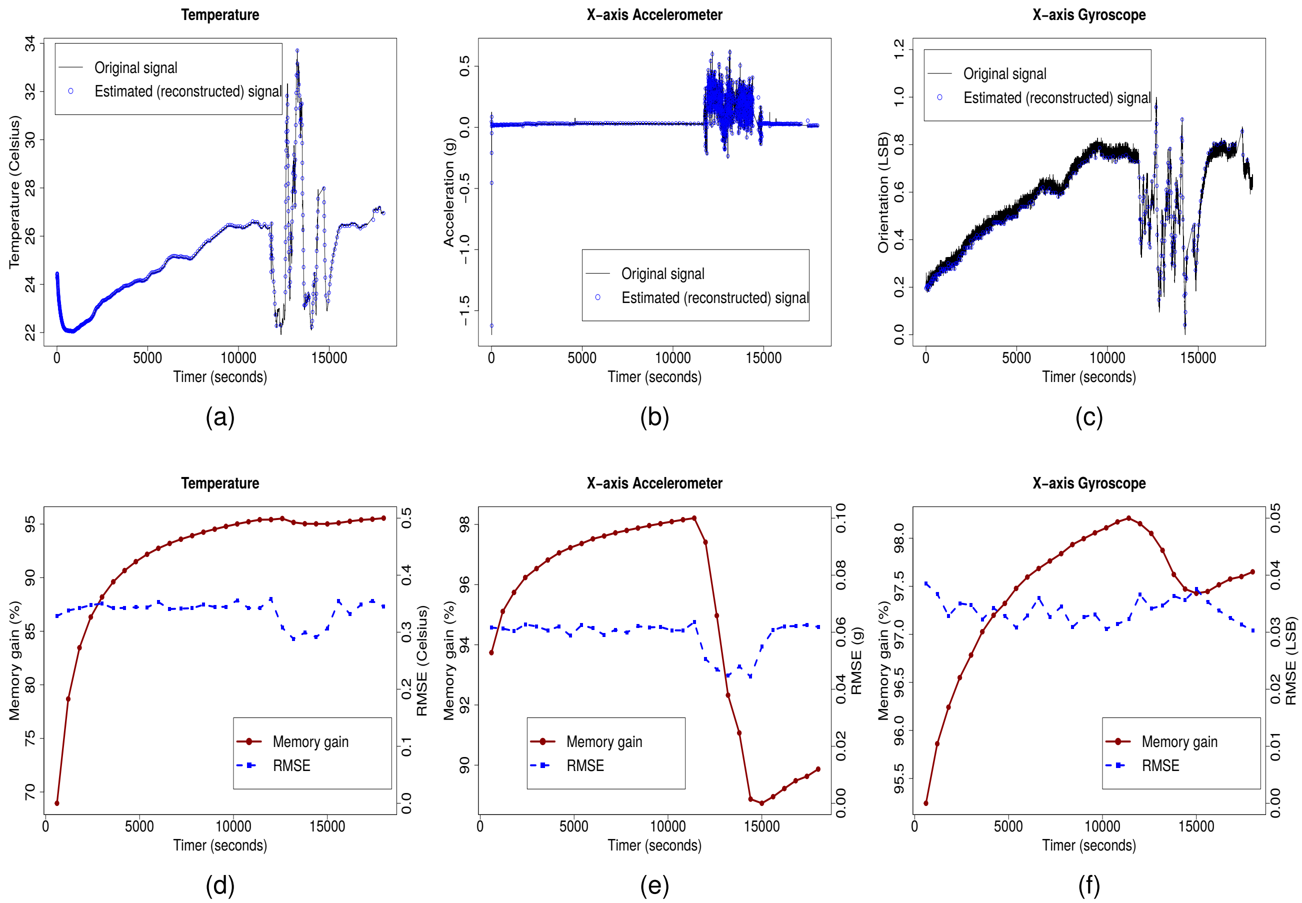

Fig. 6. Estimated signals superimposed onto the original signals along with the cumulative memory gain and RMSE for ε values from C3

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.