Proceedings of the ASME 2017 International Technical Conference and Exhibition on Packaging and Integration of Electronic and Photonic Microsystems
InterPACK2017
August 29-September 1, 2017, San Francisco, California, USA

IPACK2017-74304

EDGE COMPUTING AND CONTEXTUAL INFORMATION FOR THE INTERNET OF THINGS SENSORS

Lavanya Turlapati
IBM TJ Watson Research Center
Yorktown Heights, NY, USA

Nigel Hinds
IBM TJ Watson Research Center
Yorktown Heights, NY, USA

Dhruv Nair
IBM TJ Watson Research Center
Yorktown Heights, NY, USA

Levente J. Klein
IBM TJ Watson Research Center
Yorktown Heights, NY, USA
Email:kleinl@us.ibm.com

ABSTRACT

Interpreting sensor data require knowledge about sensor placement and the surrounding environment. For a single sensor measurement, it is easy to document the context by visual observation, however for millions of sensors reporting data back to a server, the contextual information needs to be automatically extracted from either data analysis or leveraging complimentary data sources. Data layers that overlap spatially or temporally with sensor locations, can be used to extract the context and to validate the measurement. To minimize the amount of data transmitted through the internet, while preserving signal information content, two methods are explored; computation at the edge and compressed sensing. We validate the above methods on wind and chemical sensor data (1) eliminate redundant measurement from wind sensors and (2) extract peak value of a chemical sensor measuring a methane plume. We present a general cloud based framework to validate sensor data based on statistical and physical modeling and contextual data extracted from geospatial data.

NOMENCLATURE

API Application program interface
HART Highway addressable remote transducer
HTTP Hypertext transfer protocol
ID Identification Document
LoRa Long range, low power wireless platform
M2M Machine to Machine
MQTT Message Queue Telemetry Transportation
NBIoT Narrow Band Internet of Thing
RPi Raspberry Pi

INTRODUCTION

It is projected that there will be more than 30 billion devices connected to the internet by 2020 (1). Many of the devices will generate structured data from the physical world (temperature, relative humidity, etc) and unstructured data in form of videos, text, and machine logs (2). It is argued, that most of the interaction will be a direct Machine to Machine (M2M) communications and would require automatic data verifications on data streams. Automatic validation is even more important if action is implemented on the data and control is taken on devices or infrastructure. Determining the right data acquisition frequency require decisions to be taken by devices in real time, in response to requests from other devices.

For automatic data curation, contextual data needs to be integrated along with the sensor data to validate measurements. Geospatial data in form of satellite, aerial or drones are one candidate in addition to other type of survey data like topography, and land cover (3,4). Access to big data platform that integrate heterogeneous data sources and can filter data based on flexible query (3) are a pre requisite to enable the automatic extraction of contextual data.

One area where contextual data can be leveraged is in remote operations of wireless sensor networks. In the last decade, wireless sensor networks found applications in building management, data center operation, precision agriculture, and environmental monitoring. Wireless sensor network fulfill the need for spatially dense and very high frequency data acquisition. Large scale adoption of wireless sensing technology is still limited by the robustness of current systems that needs to stay operational for 5+ years. In the case of outdoor sensor network access to low cost and reliable communication paths that allow data transmission from the site to a cloud platform are still limited in coverage.

Power availability, is another main limitation of outdoor wireless sensor networks. Power harvesting may enable long term deployment of sensor networks but currently most of the wireless sensor networks rely on batteries that may be supplemented by the solar panels, piezoelectric power generation, vibration and or wind energy generation (5). The size of solar panel and battery required for multi day operation, to maintain operation when solar energy may not be available, can increase packaging from a matchbox size to shoe box size. Even with the dramatic drop in solar panel cost, deployment of solar power generation and batteries carry a burden on sensor network and long term operation cost.

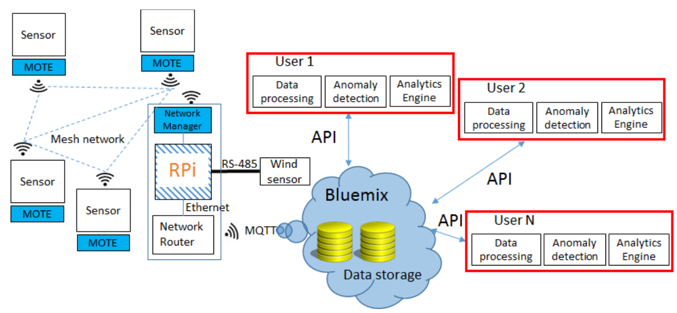
Intelligence built into the sensor network can enable sensing and operation of the devices only when power is available. Dynamically managing the power and communication path can make systems adaptable to their environment based on constrained resources. Ongoing research is assessing the tradeoff between cost effective wireless sensing solution, that can be quickly deployed, and the possibility to operate such network reliably. Innovation both in hardware, software and analytics is required for such solutions to become viable commercial solutions.

Currently, communication links between sensor networks and cloud platform is achieved using cellular link, Long Range and Low Power network(LoRa), Narrow Band Internet of Thing (NBIoT)), or satellite communication. Indoor sensor network can leverage Ethernet connection to transmit all the acquired data to a network server for data processing and analytics. For outdoor sensor networks the communication bandwidth is limited; minimizing the amount of transmitted data while maintaining the information content is a requirement.

Specifically for environmental monitoring application in the oil and gas industry where well pads are remotely located, when chemical plumes needs to be identified from multiple potential leaks, there is a requirement to sample the chemical plumes at the highest possible frequency in order to quickly detect chemical that could have detrimental effect on human health and climate (6). For example, in case of an environmental monitoring setup containing a wind sensor and 10 volatile organic sensor that are queried every second, there are more than 1 million data point generated daily. Acquiring data at this high frequency improves statistical analysis by increasing the number of detected events under different wind conditions but transmitting all data points to a cloud server will strain communication bandwidth. High frequency sampling is even more important when detecting plumes dispersion under turbulent wind condition caused by the infrastructure and surrounding vegetation (7).

As the size of the sensor network is increased, the volume of data scales accordingly and quickly data collision and data packet lost starts to affect the network stability. Since most of the sampled data may just detect background chemical levels with little relevance to the plume dispersion a large volume of data points can be eliminated. Typically, each sensors will generate a

Figure 1 General architecture of data acquisition from a wireless sensor network and data storage in the cloud. Users access the raw data and run their analytics on the data.



packet of data that contains; sensor ID, timestamp and the sensor value. If all the data is transmitted back to a central server for storage and processing, sending sensor ID with each data packet is much of the information like sensor ID is redundant across the acquisition period and reduce the communication bandwidth.

Multiple data minimization approaches like triggered detection, data compression or censoring/optimization have been investigated to minimize data transmission volume (8). A current area of research is edge computing where most of the data analytics is carried out at the detection point and only aggregated data is transmitted back to the servers.

Contextual data

Our study focuses on detection of methane plumes using a wireless sensor network with integrated wind and methane sensors where few components have integrated GPS localization while other components can be located through triangulation and time of flight measurements between radios. Localization is carried out automatically by the sensor network and if one sensing node is changed the new location for all other sensors is automatically recalculated. Placing these sensors on a map may require the knowledge of infrastructure location on the site. Detailed site description may exist but they tend to became obsolete as new construction or addition may happen on the well pads. Contextual information can be extracted from satellite, aerial or camera imagery to reconstruct oil and gas well pad sites (4). Images from satellite/drone are processed in image recognition software to identify features and classify land use. Our previous studies demonstrated that drone based imagery may be a flexible way to reconstruct buildings or any type of infrastructure in 3D (4). The advantage of drone image based

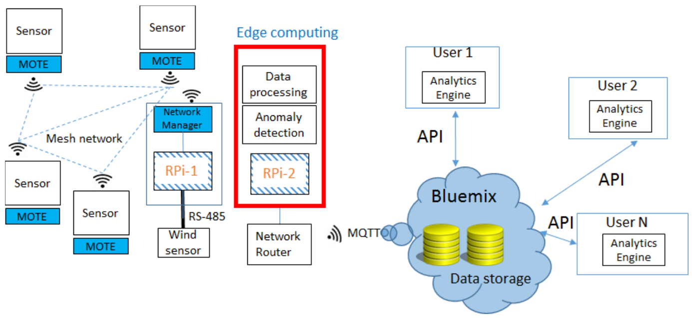


Figure 2 General architecture of edge computing where data from a wireless sensor network is pre-processed and users retrieve the data that contain information of interest.

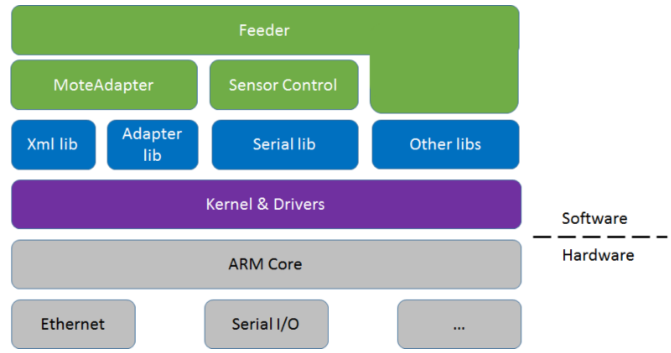


Figure 3 Feeder hardware and software stack used for sensor data transmission from point of detection to cloud platform.

reconstruction is the possibility to acquire imagery and carry out reconstruction on the fly. Currently multiple nanosatellite are operated that can provide daily coverage of any location on the Earth surface. These data streams can be leveraged to observe change in every location and business activity.

Additional contextual information can be assessed by analyzing anomalies, outliers and pair wise correlation of signal from neighbor sensors. For sensors readings that shows little correlation with neighbor sensors, while they are placed in similar locations, indicates that sensor performance may be affected by environmental conditions. Each of the above data processing steps needs to be aware of sensor placement and surrounding to validate sensor data.

Wireless sensor network

The sensing solutions rely on a wireless sensor network uisng low power time synchronized radios based on HART communication protocol. Each radio is connected to a microprocessor and sensors; due to their small size the sensing nodes are called motes. The radios automatically create a mesh network and send data back to the network manager. The mote manager aggregates the data from all sensors and communicates with a Raspberry Pi (RPi) that can handle data formatting and run the feeders that send data back to network servers. The choice for RPi is triggered due to its low cost and functionality in networking, communication and data processing and support available from a large development community. RPis additionally supports multiple network connections; Serial Peripheral Interface (SPI), Inter- Integrated Circuit (I2C), WiFi, and Bluetooth. The RPi can run Python and image processing software like OpenCv making then easily adaptable to quick analytics implementation (10).

Multiple data protocols exist to configure the data message that is transmitted from the sensor to the cloud platform. The MQTT (Message Queue Telemetry Transportation) (9) protocol is used for this implementation. MQTT is a publish-subscribe-based "lightweight" messaging for use on top of the TCP/IP protocol. It is designed for connections with remote locations where a "small code footprint" is required or the network bandwidth is limited.

In the conventional wireless sensor network approach (Fig 1) all data is sent back to a network server (Bluemix) where data is organized, analyzed and stored. For running analytics, each user will submit a query and will pull a large amount of data that is processed and analyzed locally on the user computer. The drawback of this approach is that large amount of data that is moved across the network and user analytics may be different based on interpretation and completeness of retrieved data. This approach require continuous network connections such that data can be moved from network server to user end device.

Edge Computing

Transmitting massive amounts of raw data over a network strains network resources. It is much more efficient to process data near its source and send only the aggregated/analyzed data over the network to a cloud server. One huge benefit of this approach is the reduced network traffic (10). Data can be analyzed up front at the edge of network and processed data sent back to the network server. The name "edge" in edge computing is derived from network diagrams; typically, the edge is the point at which traffic enters or exits the network. The edge is also the point at which the underlying protocol for transporting data may change. For example, a smart sensor might use a low-latency protocol like MQTT to transmit data to a message broker located on the network edge, and the broker would use the hypertext transfer protocol (HTTP) to forward the data from the sensor to a remote server over the Internet (2).

One implementation of edge computing architecture is shown in Fig 2. The system rely of two RPis running Linux operating system. RPi-1 is the main data processing unit that fulfills the same functions as was presented in the previous paragraphs. RPi-1 can have one or more wind sensors connected via serial RS-485 links. RPi-1 collects and formats the data packages and forward to the network server (see Figure 3). The RPi-1 can also stream the data to RPi-2 (acting as a local MQTT broker). RPi-2 is used mainly to store a copy of sensor data and run real time analytics. The analytics running on RPi-2 results

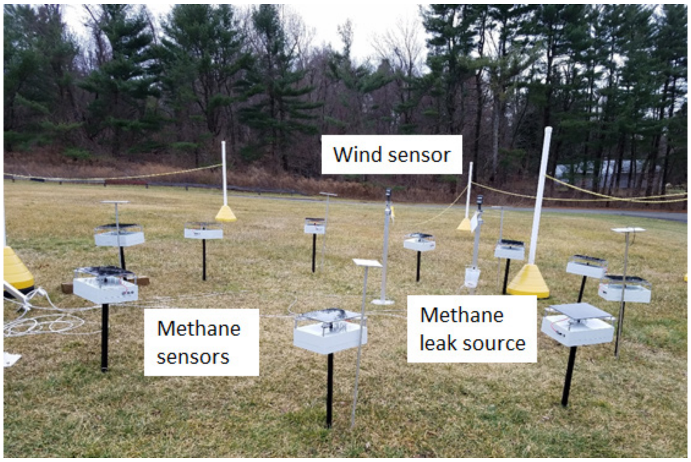


Figure 4 Experimental setup of the wireless sensor network with a wind sensor in middle and surrounded by methane sensors.

in new data stream which can be fed back to RPi-1 for transmission to a remote MQTT broker, or the new data is sent directly to a remote broker.

The two RPis are operating independently in order to separate workloads. The separation is necessary as analytics can be demanding on resources like memory and CPU operations. One RPi will run data collection while the second RPi will run the analytics. Overlap in tasks can jam the data orchestration on the RPis and to overcome this issues require extensive task scheduling and prioritization.

Data compression

Data compression is the first choice in reducing data size that is transmitted to the network (8). As a test case, the potential to reduce wind sensor data is investigated. The wind sensor is connected to RPi-1 through an RS485 serial link (Fig 2) and is sampled every second. The feeder software is installed on RPi1 while the Listener and Broker (Mosquitto Subscriber and publisher) are running on RPi-2. A Python script on RPi-2 is processing the real time wind data and carry out the signal compression. Once the wind data is run through the analytics engine, it is then published into IBM's Bluemix using the MQTT broker.

Wind data has a persistence in both wind speed and direction; e.g. where consecutive measurements of wind direction and speed do not change in value for consecutive points (Fig 4). If no change is detected in wind data, than the latest data point is not reported in the data stream. We note that the value can be reconstructed in analytics once the data is retrieved from the cloud platform. In case a data point is not acquired, the missing data is flagged to ensure that no data generation occur on the server side. The method transmits data only when a change is detected while eliminating redundant data transmission (12).

Moving average is the simplest technique to attenuate additive noise (12). It is based on the assumption that independent noise is not going to change the underlying structure

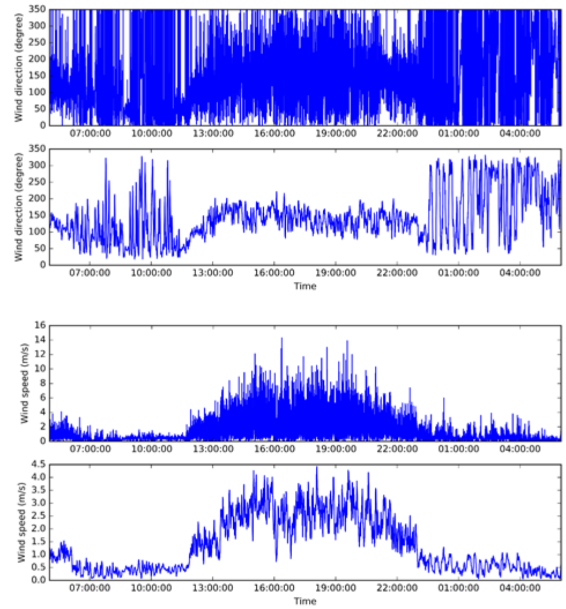


Figure 5 (A) Wind direction sampled at 1Hz and averaged across 100 data points using a rolling mean average,(B) wind speed sampled at 1 Hz and averaged across 100 data points

of the signal. If this is true, averaging few points should attenuate the contribution of the noise. In this technique, for each signal point, an average of the neighboring points is calculated. The neighboring points considered during averaging is called the window size. Increasing the window size reduces the effect of the added noise, but it is also likely to cause an excessive smoothing of the original signal. This technique works well for signals that are continuous and varying on a daily time scale. When big fluctuations are present, this filtering technique is likely to alter the original signal more than the noise itself (12).

The wind data show the typical daily patterns with higher wind speeds during daytime and reduced speed during night. The wind direction is more stable during daytime and will be more turbulence during night (Fig 5). Signal averaging can vary between daytime and nighttime.

For instantaneous wind direction and speed the unfiltered data will show very little persistence due to measurement noise and changing wind conditions. Very few points can be eliminated from the transmitted stream. Once the averaging window is increased to 100 data points (example shown in Fig 5) the amount of data sent to the cloud server can be decreased by 5%. If the average window is increased to 600 data points (10 min averaging window) the transmitted data is reduced by 18%. Averaging 1200 data point (20 min window) the data is reduced by 25%. As more averaging is carried out on the data, the final result will be smoother and larger compression can be achieved.

Chemical plumes peak detection

The sensors in the wireless sensor network are continuously monitoring the methane plumes released from a

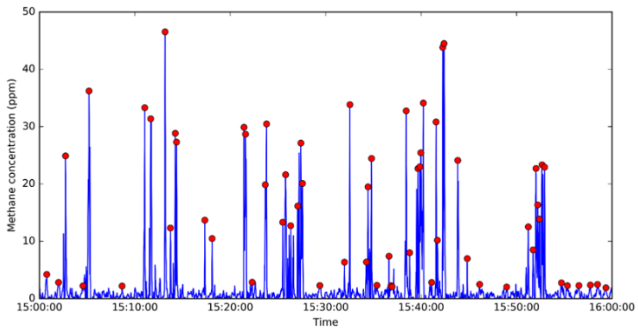


Figure 6 Peak detection in methane sensor signal measurement where only the event peaks carry information required for analytics.

well control source (Fig 4). Majority of the time, the sensor will measure just the background methane level fluctuations. When the methane plume crosses the sensor, the sensor signal increases up to 10 folds. Methane plumes are detected as narrow peaks in the methane sensor signal (Fig 6). Our analytics rely on extracting peaks magnitude, peaks width and timestamp. The background data can be ignored as it carry very little information required for advanced analytics to detect leaks location and magnitude.

Peak detection algorithms can be run on real time data to identify peak magnitude and their distribution. The simplest peak detection algorithm will look for change in slope while the signal is above a certain threshold level. Many more advanced algorithms are already implemented that can carry out peak analysis on the fly (13).

In Fig 6 the results of peak detection algorithm is shown for a time series acquired by methane sensors. The peaks are identified with the red dots and those represent the data points that carry information of value. If only those data points are sent back to the cloud server the overall amount of data from methane sensors is reduced up to 99%.

In many of the sensor data, periodic outliers and also data anomalies are measured. These points needs to be eliminated in a general framework where the sensors data are bound using physical knowledge about the surroundings and the contextual data (14,15). The methane plume is moved by wind and the wind distribution is modeled using computational fluid dynamics taking into account the infrastructure location and sensor position. Since the wind turbulence will be driven by the infrastructure on the site recognizing infrastructure from external data sources can improve the correctness of the simulations.

CONCLUSION

With large scale sensor networks the capabilities of communication networks to transmit every single data point to cloud servers will be limited. Processing of the data streams close to the sensing "edge" point and transmitting only the relevant data back to the cloud server is investigated in this study. In order for machines to validate measurements, contextual data is required for anomaly detection and outlier identification. Such contextual databases are developed to offer support for machines to validate sensor results and automatically determine the right sampling frequency and data that needs to be recorded. Machines needs to be aware of power availability and communication bandwidth to adjust dynamically to data acquisition and analytics. Fine tuning data compression can reduce by an order of magnitude the amount of recorded data without losing information content. We demonstrated a preliminary study in reducing the number of data points that need to be transmitted while preserving the information carried by these data streams.

FUTURE WORK

Edge or fog computing will ultimately require dynamic or elastic components. That is, an infrastructure where analytic components can be defined without knowledge of where the component will be executed. If we model the system as a data pipeline, then this will require mechanisms for describing required data input and output streams, mechanisms for moving computation components to/from various locations in the pipeline (e.g. edge Feeders and Cloud Virtual Machines) while maintain the necessary data stream connections. Such an architecture would facilitate workload re-distribution that could respond to differences in application processing requirements.

ACKNOWLEDGMENTS

The work presented herein was funded in part by Advanced Research Projects Agency Energy, U.S. Department of Energy Award No. DE-AR0000540. This report was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represented that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or any agency thereof.

We acknowledge fruitful discussion with Hendrik Hamann, Norma Sosa, and Theodore van Kessel.

REFERENCES

1. Gartner, "Gartner Says the Internet of Things Installed

Base Will Grow to 26 Billion Units by 2020", December 12, 2013, http://www.gartner.com/newsroom/id/2636073 accessed April 8, 2017.
2. Gubbi, Jayavardhana, et al. "Internet of Things (IoT): A vision, architectural elements, and future directions." Future generation computer systems 29.7 (2013): 1645-1660.
3. Klein, Levente J., et al. "PAIRS: A scalable geo-spatial data analytics platform." Big Data (Big Data), 2015 IEEE International Conference on. IEEE, 2015.
4. Renwick, Jason D., Levente J. Klein, and Hendrik F. Hamann. "Drone-based reconstruction for 3D geospatial data processing." Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on. IEEE, 2016.
5. Roundy, Shad, Paul Kenneth Wright, and Jan M. Rabaey. Energy scavenging for wireless sensor networks. Norwell, 2003
6. Venkatram, Akula, et al. "Modeling dispersion at distances of meters from urban sources." Atmospheric Environment 38.28 (2004): 4633-4641.
7. Hanna, Steven R., Rex Britter, and Pasquale Franzese. "A baseline urban dispersion model evaluated with Salt Lake City and Los Angeles tracer data." Atmospheric Environment 37.36 (2003): 5069-5082.
8. Kimura, Naoto, and Shahram Latifi. "A survey on data compression in wireless sensor networks." Information Technology: Coding and Computing, 2005. ITCC 2005. International Conference on. Vol. 2. IEEE, 2005.
9. Hunkeler, Urs, Hong Linh Truong, and Andy Stanford-Clark. "MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks." Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on. IEEE, 2008.
10. Rabinovich, Michael, Zhen Xiao, and Amit Aggarwal. "Computing on the edge: A platform for replicating internet applications." Web content caching and distribution. Springer Netherlands, 2004. 57-77.
11. Bradski, Gary, and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008.
12. Downey, Allen B. Think DSP: digital signal processing in Python. " O'Reilly Media, Inc.", 2016.
13. Du, Pan, Warren A. Kibbe, and Simon M. Lin. "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching." *Bioinformatics* 22.17 (2006): 2059-2065.
14. Hayes, Michael A., and Miriam AM Capretz. "Contextual anomaly detection framework for big sensor data." *Journal of Big Data* 2.1 (2015): 2.
15. Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." *ACM computing surveys (CSUR)* 41.3 (2009): 15.

