24th IMEKO TC4 International Symposium
22nd International Workshop on ADC and DAC Modelling and Testing
IMEKO TC-4 2020
Palermo, Italy, September 14-16, 2020

A new Fog-enabled Wireless Sensor Network architecture for Industrial Internet of Things applications

Zakaria Benomar[1], Giuseppe Campobello[1], Francesco Longo[1], Giovanni Merlino[1], Antonio Puliafito[1]

[1]*Dipartimento di Ingegneria, Università di Messina, Italy, {zbenomar,gcampobello,flongo,gmerlino,apuliafito}@unime.it*

Abstract – **Wireless Sensor Networks (WSNs) have gained considerable popularity in industrial applications and are widely used nowadays in several industrial monitoring and control systems. To process and store the huge amount of data WSNs generate, most of the solutions opt for the Cloud approach as it provides suitable capabilities to deal with 'Big Data' scenarios. However, as discussed in this paper, Cloud-based solutions have several drawbacks, e.g. latency and security, to name just a few. To overcome these drawbacks, we propose a system meant for monitoring of industrial assets based on the Fog computing paradigm. As shown in this paper, the proposed solution provides better performance in terms of reliability/accuracy and allows full control over the sensor nodes. The system provides also the ability to rapidly program the Fog nodes and update their business logic. Finally, the advantages of the proposed Fog-based solution are discussed considering a real-world industrial application scenario.**

I. INTRODUCTION

Nowadays, the manufacturing sector is undergoing a new age of transformation given a number of substantial changes on multiple fronts. Following the current trend for digitalization, companies are taking inspiration from the Internet of Things (IoT) to develop and enhance their manufacturing processes. As a growing subcategory of IoT tailored for the industry, the Industrial Internet of Things (IIoT) leverages smart sensors and actuators in order to interconnect industrial machines/assets to the Internet and thus making any user or machine around the world able to communicate, sense or interact with factory-local resources or surrounding environments. Consequently, smart machines become able to generate useful data that can be afterward analyzed/processed to boost the industrial processes and induce meaningful production optimizations [1].

For industrial applications, the placement of a sensor or a control device (e.g., actuator) is critical. Indeed, they should usually be deployed in harsh environments (e.g., attached to rotating equipment). To this end, Wireless Sensors Network (WSN) technologies have promoted the use of monitoring systems in the manufacturing field, therefore giving birth to a new paradigm meant specifically for the industry, namely Industrial Wireless Sensor Network (IWSN) [2]. Such a kind of networks provides a great level of flexibility and placement freedom, makes systems' deployment and upgrade easier, as well as reduces operating and infrastructure costs [3]. Since IWSN deployments may consist of a myriad of nodes with sensors on-board, and with a view to make monitoring solutions cost-effective, in terms of network architecture the system designer opts for inherently resource-constrained nodes with very limited storage, processing, and communication bandwidth capabilities. Under these circumstances, IWSN nodes are ill-equipped to cope with data processing/management and relatively resource-intensive tasks such as data analytics. With the recent rapid adoption of IWSN solutions, another information revolution related to the ability to create, store, and process data gathered from the sensors is pushing for new management and processing methods. The Cloud computing paradigm, in this context, arises as a suitable approach to deal with WSNs' data management/processing duties. On the one hand, with the huge amount of resources (i.e., compute and storage) it has, the Cloud is inherently meant to deal with big data management. On the other hand, the Cloud provides, for its users, flexible deployments able to scale up/down, hinging on the resources required while exempting users from buying, configuring, and over-provisioning the infrastructure. However, considering the huge amount of data a WSN deployment may generate and the latency the (distant) Cloud may inflict, Cloud-based solutions can bear several issues (e.g., bandwidth consumption, data storage cost, high delays). As a new approach introduced to deal with such limitations in the IoT landscape, the Fog computing paradigm [4], by extending the Cloud capabilities towards the network edge and close to end devices, is poised to address most of the issues brought about by the Cloud.

Following this trend, we are introducing in this paper a Fog-based system meant for industrial machines monitoring. As a use case, we focus on induction motors monitor-

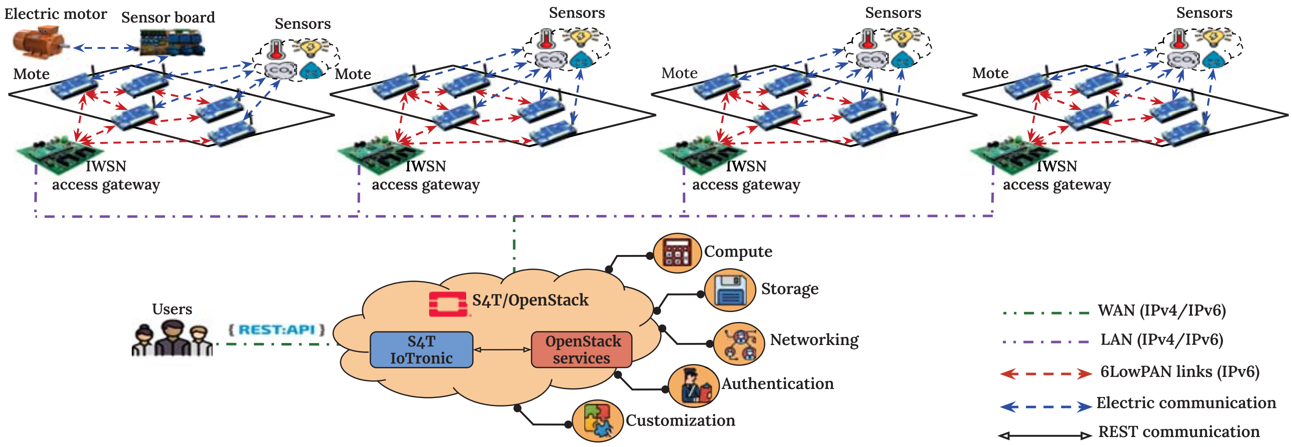

Fig. 1. Proposed Architecture

ing. Through our platform, an administrator can monitor different parameters of the motors such as current, vibration, etc. Besides, the system provides users the ability to manage Fog nodes, (re)program and access them remotely.

The contributions in this paper are listed in the following: i) an outline of a Fog-based system for industrial machines monitoring; ii) usage of an IoT resource management system, Stack4Things, to manage the Fog layer; iii) design of a suitable sensing board to monitor induction motors health. The paper is organized as follows. Section II discusses the impact of Fog computing in WSN/IWSN deployments. Section III gives an overview of our system architecture, and hardware devices used to conceive the system. Section IV discusses the monitoring use case, for induction motors, based on our solution. Finally, Section V closes the paper with conclusions and future work.

II. FOG COMPUTING IN WSN/IWSN

Thanks to their flexibility and ease of implementation, WSN-based monitoring systems have gained a lot of interest in the last years, either for limited or large scale monitoring (e.g., urban air monitoring) [5]. Considering the colossal amount of data that WSNs may generate, the Cloud paradigm has been promoted as a suitable solution for WSN-originated data management. Yet, with the explosive growth of deployments around the world, and the emergence of time sensitive-applications, the Cloud paradigm alone is not able anymore to deal with further requirements. On the one hand, the huge amount of data generated can cause several issues even before getting to the Cloud, for instance, the bandwidth consumption at the level of the local-area network. Moreover, forwarding the whole raw data generated to the Cloud, either for storage or processing, may become prohibitively expensive. Additionally, critical, time-sensitive applications in industrial fields, such as alert systems can not always rely on the Cloud, due to the high latency it may cause. Indeed, the Fog computing paradigm was promoted as a relevant approach to solve the aforementioned issues. Following the trend of combining WSN with the Fog Computing paradigm, a number of solutions were introduced in the literature. Authors in [6] propose a WSN-based system for air quality monitoring. The Fog nodes in this solution preprocess the data at the network edge before forwarding it to the Cloud. Specifically, the Fog layer reduces the amount of data to be forwarded to the Cloud using an aggregation algorithm. Another Fog architecture for IWSN is proposed in [7]. In the same spirit, [8] presents a system that uses the Fog paradigm to provide services in a Wireless Sensors and Actuators Network (WSAN) context. However, these papers did not address the remote (re)programmability and management of the Fog layer. Specifically, such deployments are lacking in terms of flexibility and plurality. They are mainly designed to achieve particular tasks, and the management of the lifecycle for the logic they may host is not addressed. Indeed, WSN/WSAN are highly dynamic environments were administrators may need frequent modifications to the network, by adding or removing nodes to/from the deployments.

III. A FOG-BASED ARCHITECTURE FOR ASYNCHRONOUS ELECTRIC MOTORS MONITORING

The system we are introducing in this paper is meant for monitoring industrial machines/assets (see Fig. 1). As hardware specification for the devices used in the IWSN, we opted for IRIS motes which are based on the broadly used IEEE 802.15.4 protocol [9]. As operating system for these motes, TinyOS [10] was used, to make applications' portability easier, with only a few changes required in case other commercial platforms, e.g., Advanticsys and Zolertia motes, are to be used. About the rest of the protocol stack, we opted for IPv6 over Low-Power Wireless Personal Area Networks (6LowPAN) [11], with the Routing

Protocol for Low-Power and Lossy Networks (RPL) [12] as routing protocol among the motes, and the UDP protocol for the transport layer. In our scenario, the constrained WSNs motes are used only to forward sensed data to the IWSN(s) access gateway(s) (i.e., Fog nodes). To get data from a motor, the mote is connected to a sensor board that we conceived specifically for this use case. More in detail, this sensor board can interface 3.3V IoT devices with a three-phase motor. We report the sensor board block diagram in Fig. 2. The sensor board includes Hall-effect transducers for measuring voltages in the range 10-500V and currents up to 50A, an optical encoder for motor speed measurements, and temperature and accelerometer sensors for measuring temperature and mechanical vibrations as well.

To make the IWSNs gateways capable of executing (relatively) complex computing tasks, the system uses the low-cost Arancino[1] Single Board Computers (SBCs) commercialized by the academic spin-off company smartme.IO (see Fig. 1). These devices are equipped with both microprocessor (MPU) and micro-controller (MCU) units, hence they are capable of hosting a minimal Linux distro (e.g., OpenWRT), including different programming runtimes and a set of Linux-based tools. On the other hand, being MCU-powered makes such devices suitable to be used, at the same time, as sensing/actuation devices, and to interact with the resources they may host in real-time. To enhance the capabilities of the monitoring system, we integrated it with our Cloud-based IoT resources management platform, namely Stack4Things (S4T) [13]. S4T is an OpenStack-based platform which extends Cloud-level capabilities to deal with IoT deployments. A deployment of S4T consists of a datacenter, where a subsystem called IoTronic is hosted, and a number of geo-distributed IoT devices hosting the S4T device-side Lightning-Rod (LR) agents. To cope with the unique constraints of IoT deployments, the status and *presence* of IoT devices (i.e., LR agents) is relayed to the Cloud-side (i.e., IoTronic) using suitable mechanisms based on WebSockets, with a reverse tunneling approach to bypass NATs/firewalls. Being IoTronic fully compliant with OpenStack makes deep integration of IoTronic with other subsystems in the OpenStack ecosystem possible, and thus the platform can provide advanced user-facing features such as users authentication/authorization, virtual networking and more. Furthermore, S4T provides users/administrators the ability to remotely (re)program IoT nodes, even at runtime, through the injection of customized user-defined functions. Specifically, a user/administrator can thus define the behavior for a remote device by developing (a unit of) business logic, upload it to the Cloud and afterward inject it on a device (or group of devices), still according to authorization and privacy policies. S4T actually provides Node.js and Python

[1]https://arancino.cc/

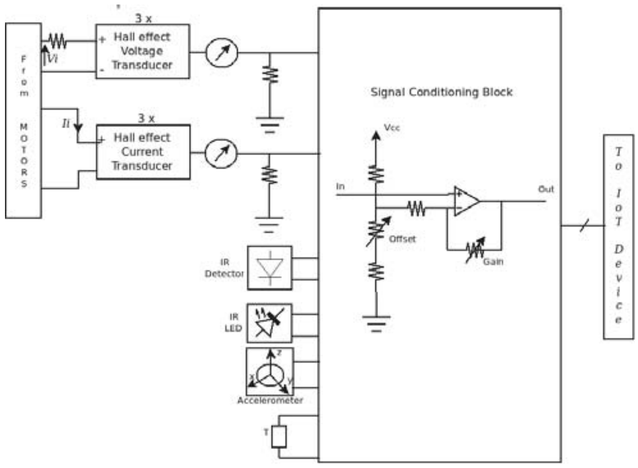

Fig. 2. Sensor Board

as runtime environments for user-defined functions. On the IoT node-side, the main components involved in the management of functions contextualization are a function management library together with a default wrapper used for isolation between functions as well as the control of their life cycle: each S4T function runs as an independent (either synchronous or asynchronous) process spawned by the LR agent. Consequently, an injected function can be considered as any other process running in parallel while executing a specific task: for example, a basic behavior we can consider may be an ongoing sampling of metrics coming from different sensors, and computing statistical indexes (higher-complexity operations can be considered, see Sec. iv.)

IV. EXPERIMENTAL RESULTS

To illustrate advantages of the proposed Fog-based architecture in an industrial context, we consider in this section a real-world industrial scenario, where 20 three-phase induction motors, all distributed across the same factory, must be monitored in real-time. We assume that each motor is coupled with an IRIS mote specifically programmed with the aim of measuring amplitude and frequency of currents in the stator. We suppose that power supply is rated at nominally 400V 50Hz and that currents are sampled with a sampling frequency of 1,000 samples per seconds (sps), that is chosen with the aim of detecting possible harmonic distorsions up to the 9th order. According to the maximum resolution of analog-to-digital converters (ADC) of IRIS motes, each current sample is represented with $n = 10$ bits. Under the above mentioned assumptions, a data stream of $R_b = 3nf_s = 30,000$ bits per second (bps) must be transmitted by each mote, i.e. for each of the 20 three-phase induction motor. In particular, we consider that 60 samples are transmitted for each packet, corresponding to one period for each of the three

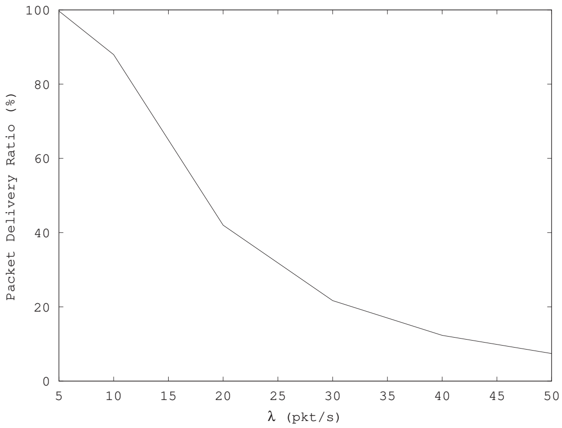

Fig. 3. Packet Delivery Ratio (PDR) for different traffic generation rates λ in the case of a classical single-gateway WSN with 20 nodes.

stator currents. We assume that each Fog node (i.e., IWSN gateway) is the cluster head of a few IRIS motes, as highlighted in Fig. 1 and compare the proposed solution with a classical WSN architecture where all the 20 motors are managed by the same gateway connected to the Cloud.

We have simulated the above scenarios considering the model developed in [14] for multi-hop WSNs based on the beaconless mode of the 802.15.4 MAC protocol. For sake of completeness, we briefly summarize here the model and our assumptions. Basically, we simplified the model proposed in [14] to consider a simple star network topology where sensor nodes generate fixed-length packets and send them to a gateway at a constant rate λ (here expressed in packets per second, pk/s). Note that, considering a bit-rate equal to 30kbps and a packet length of $60 \times 10 = 600$ bits (75bytes), it follows that $\lambda = 50$pk/s is the traffic generated by each node.

We further assume that each sensor node transmits packets according to the 802.15.4 MAC protocol and the following MAC parameters: macminBE=3, macmaxBE=5, macMaxCSMABackoffs=4, and aMaxFrameRetries=3. Each time a node needs to transmit, it waits for a random backoff period and then performs the Clear Channel Assessment (CCA) to determine whether the channel is idle. If the CCA succeeds, the node starts transmitting on the channel, otherwise it waits another random backoff period up to a maximum number of successive CCA attempts specified by macMaxCSMABackoffs. Note that, according to the 802.15.4 standard, macMaxCSMABackoffs does not include the first backoff, therefore the actual maximum number of consecutive backoffs for the same packet is macMaxCSMABackoffs+1, i.e. 5. As a consequence, by indicating with α the probability of CCA failure, the probability that a node may discard a packet due to consecutive CCA failures is given by α^5. When a packet is transmitted it can fail to be received due to collisions or noise or other impairments introduced by the channel. Henceforward we indicate with p the probability of packet collision, i.e., the probability that two or more nodes, in the same carrier sensing range, transmit simultaneously, and with l the packet error rate due to the channel. In general, l is related to the signal-to-interference-plus-noise ratio (SINR) and is a time-varying quantity. However, for sake of simplicity, we considered that all nodes have the same distance to the sink and that links are symmetric and not time varying. Under the above assumptions l is constant and the overall packet failure probability γ due to both collisions and channel impariments can be evaluated as $\gamma = p+(1-p)\cdot l$. For simulation purpose we assumed $l = 0.01$.

Finally, given α and γ, the packet delivery ratio can be evaluated as $PDR = 1 - \delta$, where δ is the probability of discarding a packet due to either consecutive CCA failures or successive failed retransmissions and is given by

$$\delta = \alpha^5+(1-\alpha^5)\gamma[\alpha^5+(1-\alpha^5)\gamma[\alpha^5+(1-\alpha^5)\gamma[\alpha^5+(1-\alpha^5)\gamma]]], \tag{1}$$

In Fig. 3 we reported the PDR achieved with the above model for different values of the packet generation rate λ in the case of a classical single gateway WSN with 20 nodes all in the same carrier sensing range. As it is possible to observe, the PDR is near to 8% when $\lambda = 50$pk/s. According to simulation results, a PDR greater than 90%, as usually requested in many IWSN applications, can be obtained only if λ is reduced at least by a factor of 5, i.e. from 50pk/s to 10pk/s. We were able to solve this problem by exploiting the Fog layer to implement a data aggregation technique based on Singular Value Decomposition (SVD) [15]. The proposed technique is described below and is able to improve both the PDR and accuracy of measurements.

Basically, SVD allows to rewrite a $N \times M$ real matrix A as the product of three matrices, U, Σ and V, such that $A = U\Sigma V^T$. In particular, Σ is a diagonal matrix whose diagonal elements, σ_i with $i \in [1, ..., N]$, are referred to as singular values. From SVD theory it is well known that the matrix A can be approximated considering only a few singular values as $A \approx \tilde{A} = U_k \Sigma_k V_k^T$ where $k < N$ is the number of singular values exploited for the approximation and U_k, Σ_k and V_k are submatrices of, respectively, U, Σ and V, obtained considering only the first k columns of U and V and the first k diagonal elements of Σ. It is worth noting that the overall number of nonzero elements in the submatrices U_k, Σ_k and V_k are at most $P = N \times k + k + M \times k = k(N + 1 + M)$. This number of elements can be less than the original number of elements of the matrix A. In particular, $P < N \cdot M$ holds every time that the condition $k < \frac{NM}{N+M+1}$ is satisfied. Therefore, by transmitting the matrices U_k, S_k and V_k instead of the original matrix A, it is possible to reduce the amount of data that must be sent in the network.

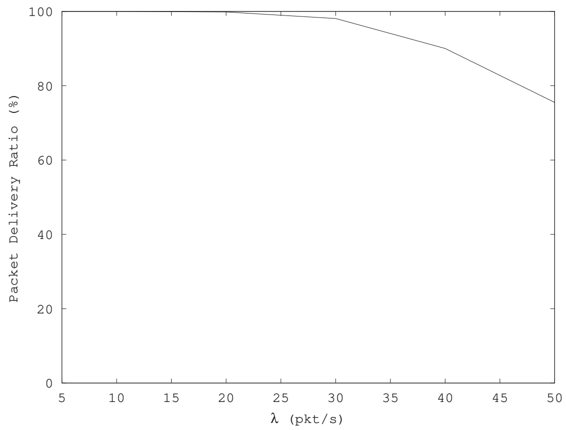

Fig. 4. Packet Delivery Ratio (PDR) for different traffic generation rates λ in the case of a cluster with 5 nodes.

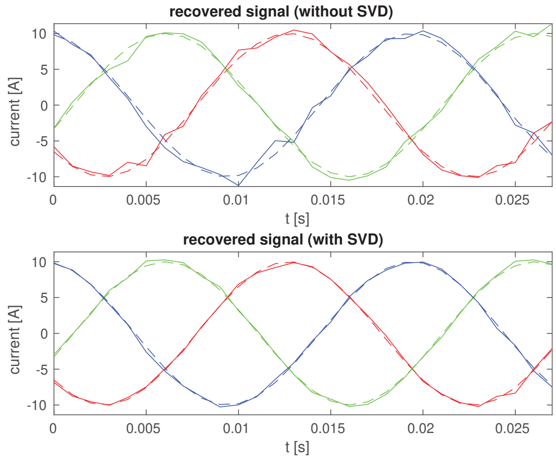

Fig. 5. Reconstructed voltage signals with and without SVD.

This obviously reduces network load and thus the number of packets lost. In particular, a compression factor equal to $F_c = \frac{NM}{k(N+1+M)} \approx \frac{N}{2k}$ can be achieved when $M \approx N$.

Now let us consider a Fog layer with $N_G = 4$ gateways so that each gateway coordinates a cluster of 5 IRIS motes. We assume that each gateway receives packets from motes belonging to their respective clusters and rearranges collected data, i.e., 5 packets representing 60 current samples each, into a matrix A of 15×20 elements. In this case by fixing $k = 2$ it is possible to achieve a compression factor $F_c = \frac{15 \cdot 20}{2(15+1+20)} \approx 4.17$. This compression factor largely reduces the amount of data that must be transmitted and thus reduces packet collisions and increases the PDR.

Note that, considering the SVD, the traffic that must be forwarded by each gateway is $\lambda_g = \frac{5\lambda}{F_c} \approx$ 60pk/s instead of 250kbps. However, in the proposed architecture, this traffic is sent through high-speed LAN or WLAN networks, therefore we can assume that related packets do not collide with packets generated by motes. Furthermore, when more gateways are used, it is possible to reduce transmission power of all sensor nodes and thus to consider that sensor nodes belonging to different clusters do not interfere each other. Therefore the PDR in the case of the introduction of the Fog layer almost coincides with the PDR of a single cluster with 5 motes.

In Fig. 4 we reported the PDR obtained with our model in the case of a WSN cluster with 5 motes. As it is possible to observe, in this case the PDR is near the 75% when $\lambda =$ 50pk/s and further increase to 90% when $\lambda =$ 40pk/s. Note that λ can be reduced from 50pk/s to 40pk/s by simply reducing the sampling frequency f_s from 1KHz to 800Hz, that is enough to detect possible harmonic distorsions up to the 7th order.

Furthermore, the SVD impacts on measurement accuracy as well by reducing noise effects. In fact, as highlighted in Fig. 5 where we reported current signals reconstructed with and without SVD, SVD also improves the signal-to-noise ratio (SNR).

Table 1. Average percentage errors obtained from samples of stator currents with and without data aggregation for different values of SNR

SNR	aggregation	$\bar{err}_{I_{max}}$	$\bar{err}_{I_{RMS}}$	$\bar{err}_f$
10	NO	26.90%	4.58%	0.0075%
10	YES	7.89%	0.78%	0.0010%
20	NO	6.34%	0.49%	0.0008%
20	YES	1.56%	0.11%	0.0002%
30	NO	1.22%	0.05%	0.0001%
30	YES	0.09%	0.01%	0.0000%

To better highlight this advantage of SVD, in Tab. 1 we reported average percentage errors achieved on measurements of maximum values (I_{max}), root mean square (rms) values (I_{rms}) and frequency (f) related to stator currents, obtained with and without data aggregation, i.e., with and without SVD, for different values of SNR. It is worth mentioning that to simulate different SNR values we considered original measurements corrupted by synthetic Additive White Gaussian Noise (AWGN).

Relative errors have been evaluated for each motor and each quantity $x \in \{I_{max}, I_{rms}, f\}$ as $err_x = 100 \cdot \frac{|\hat{x} - x_{true}|}{|x_{true}|}$ where x_{true} is the true (noise-free) value and $\hat{x}$ is the corresponding measured/estimated value obtained from $N_s = 20$ samples, i.e., I_i with $i \in \{1, 20\}$. In particular, we obtained maximum and rms estimation as $\hat{I}_{max} = \texttt{max}\{I_i\}$ and $\hat{I}_{rms} = \sqrt{\frac{\sum_{i=1}^{N_s} I_i^2}{N_s}}$, respectively, instead frequency $\hat{f}$ has been estimated with the frequency estimation algorithm recently proposed in [16]. Finally, average percentage errors $\bar{err}_x$ have been obtained by averaging measurement errors err_x over 1000 transmissions. As it is possible to observe data aggregation improves accuracy for almost all measured quantities.

In summary, it is worth reiterating that SVD cannot be implemented directly by IRIS or other commercial motes commonly used in WSNs, therefore a Fog Layer is mandatory for the proposed SVD-based data aggregation.

Finally, it is worth mentioning that Fog gateways can be remotely reconfigured in order to adapt parameters of SVD, i.e., N, M and k, to other possible application scenarios, with different number of nodes and/or different kind of signals to be handled. Moreover, gateways can be reprogrammed to support other aggregation techniques based on distributed source coding [17] and/or IoT-specific compression algorithms [18].

V. CONCLUSIONS AND FUTURE WORK

In this paper, we introduced a Fog-based system meant for industrial equipment monitoring and outlined its advantages. As future work, we expect to integrate actuators in the use case to exercise the capabilities of the Fog layer even in the case of real-time control systems and time-sensitive applications. Moreover, by exploiting the reconfigurability of the Fog layer, we will evaluate experimentally the trade-off between accuracy, compression and energy consumption of different aggregation techniques.

REFERENCES

[1] R. Beudert, L. Juergensen, and J. Weiland, "Understanding smart machines: how they will shape the future," *Schneider-Electric, white paper*, 2015.

[2] G. Campobello, M. Castano, A. Fucile, and A. Segreto, "WEVA: A complete solution for industrial internet of things," in *Ad-hoc, Mobile, and Wireless Networks - 16th Int. Conf. on Ad Hoc Networks and Wireless, ADHOC-NOW 2017, Messina, Italy, September 20-22, 2017, Proceedings* (A. Puliafito, D. Bruneo, S. Distefano, and F. Longo, eds.), vol. 10517 of *Lecture Notes in Computer Science*, pp. 231–238, Springer, 2017.

[3] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on industrial electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.

[4] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.

[5] B. Rashid and M. H. Rehmani, "Applications of wireless sensor networks for urban areas: A survey," *Journal of network and computer applications*, vol. 60, pp. 192–219, 2016.

[6] V. Mihai, C. Dragana, G. Stamatescu, D. Popescu, and L. Ichim, "Wireless sensor network architecture based on fog computing," in *2018 5th International Conference on Control, Decision and Information Technologies (CoDIT)*, pp. 743–747, IEEE, 2018.

[7] M. Lavassani, S. Forsström, U. Jennehag, and T. Zhang, "Combining fog computing with sensor mote machine learning for industrial iot," *Sensors*, vol. 18, no. 5, p. 1532, 2018.

[8] W. Lee, K. Nam, H.-G. Roh, and S.-H. Kim, "A gateway based fog computing architecture for wireless sensors and actuator networks," in *2016 18th International Conference on Advanced Communication Technology (ICACT)*, pp. 210–213, IEEE, 2016.

[9] J. Zheng and M. J. Lee, "A comprehensive performance study of ieee 802.15. 4," *Sensor network operations*, vol. 4, pp. 218–237, 2006.

[10] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, *et al.*, "Tinyos: An operating system for sensor networks," in *Ambient intelligence*, pp. 115–148, Springer, 2005.

[11] G. Mulligan, "The 6lowpan architecture," in *Proceedings of the 4th workshop on Embedded networked sensors*, pp. 78–82, 2007.

[12] T. Winter et al., "Rpl: Ipv6 routing protocol for low-power and lossy networks.," *rfc6550*, pp. 1–157, 2012.

[13] F. Longo, D. Bruneo, S. Distefano, G. Merlino, and A. Puliafito, "Stack4things: a sensing-and-actuation-as-a-service framework for iot and cloud integration," *Annals of Telecommunications*, vol. 72, no. 1-2, pp. 53–70, 2017.

[14] R. Srivastava, S. M. Ladwa, A. Bhattacharya, and A. Kumar, "A fast and accurate performance analysis of beaconless ieee 802.15.4 multi-hop networks," *Ad Hoc Networks*, vol. 37, pp. 435 – 459, 2016.

[15] G. W. Stewart, "On the early history of the singular value decomposition," *SIAM Review*, vol. 35, no. 4, pp. 551–566, 1993.

[16] G. Campobello, A. Segreto, and N. Donato, "A new frequency estimation algorithm for iiot applications and low-cost instrumentation," in *IEEE International Instrumentation and Measurement Technology Conference, I2MTC 2020, Dubrovnik, Croatia, May 25-28, 2020*, pp. 1–6, IEEE, 2020.

[17] G. Campobello, A. Segreto, and S. Serrano, "Data gathering techniques for wireless sensor networks: A comparison," *Int. J. Distributed Sens. Networks*, vol. 12, pp. 4156358:1–4156358:17, 2016.

[18] G. Campobello, A. Segreto, S. Zanafi, and S. Serrano, "RAKE: A simple and efficient lossless compression algorithm for the internet of things," in *25th European Signal Processing Conference, EUSIPCO 2017, Kos, Greece, August 28 - September 2, 2017*, pp. 2581–2585, IEEE, 2017.