



The 2nd International Conference on Emerging Data and Industry 4.0 (EDI40)
April 29 - May 2, 2019, Leuven, Belgium

Towards Distributed IoT/Cloud based Fault Detection and Maintenance in Industrial Automation¹

Apostolos Xenakis^a, Anthony Karageorgos^a, Efthimios Lallas^a, Adriana E. Chis^b,
Horacio González-Vélez^b

^aUniversity of Thessaly, 37 Glavani – 28th October Str, Volos, 38221, Greece

^bNational College of Ireland, Cloud Competency Centre, Dublin 1, Ireland

Abstract

Industrial Internet of Things (IIoT) automation should be based on a framework that guarantees flexible and energy efficient monitoring and control, without the need for frequent human intervention. The ability to analyse and process machine faults in real time is vital, however it poses many technical difficulties and challenges, mainly for industrial application environments. In our paper, we propose a novel, energy efficient, IoT and Cloud based decentralised framework for real time machine condition monitoring (MCM) and fault prediction, where computational demanding tasks are distributed across fog nodes and decision fusion rules are set and controlled by the Cloud. In particular, data acquisition phase is done by sensors distributed across machines, feature extraction and health condition classification is done by fog nodes, after receiving data and instructions as processed by the Cloud node. Our framework is based on collaboration and information flow among IoT, Fog and Cloud layers. To this purpose, we formulate a global consensus cross layer optimisation problem, concerning industrial healthy status monitoring, and we solve it in a distributed manner by applying asynchronous altering direction method of multipliers (ADMM) algorithm.

© 2019 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Industrial Internet of Things; Fault Detection; Industrial Automation; ADMM Optimisation

1. Introduction

In the Industry 4.0 era, manufacturing companies generate tremendous amounts of industrial data based on advanced sensor technology and the internet of things (IOT) among others [1]. In doing so, IoT with its industrial wireless sensor networks may be used for machine condition monitoring (MCM) purposes [2].

* Corresponding author. Tel.: +30-6977562678

E-mail address: axenakis@uth.gr

In an IoT environment, multiple autonomous, tiny, low cost and low power sensor nodes, gather and sense data and collaborate with each other to forward these data to centralised backend units called cloud server stations for further processing. In time simple data sensing, collection and forwarding is not enough and hence IoT environments need be empowered with high-level intelligence and cognitive capabilities giving rise to cognitive IoT (CIoT) [1]. In CIoT settings, produced data are mostly heterogeneous and IoT nodes generally communicate in an ad-hoc, and decentralised manner, while the classic centralised data processing, is carried out by high performance cloud servers. Fog computing comes as an intermediate layer and interface between distributed IoT edge computing and centralized Cloud computing, to fill the gap as far as concerns mobility support, location awareness, and low latency requirements [3][4]. In this paper we propose a novel and energy efficient IoT, Fog and Cloud based decentralised framework for real time MCM, where classification tasks are distributed across fog nodes, and final decision / fusion rules are set and controlled by the Cloud node (i.e. Master node). In particular our contributions rely to the following: 1) We propose a cross optimisation problem and solve it with the ADMM distributed optimization method [8] so as to maintain the energy efficient information flow among Fog and Cloud layer. 2) Feature extraction and health condition classification is done by Fog nodes 3) Fusion rules are set by the Cloud node in order to decide whether machines reveal non-healthy status. The rest of the paper is organized as follows: in section 2 we present our system model and describe the information flow layers and the MCM formulations. In section 3 we present our problem formulation and solution, while in section 4 we discuss results. Finally, in section 5 we conclude our paper.

2. Approach

In this section our approach methodology is presented. In particular, we present the proposed system model, we define the MCM formulations, and we describe the vibrations data format.

2.1. System Model

Our system model is presented in Fig. 1. We assume that the industrial production line is comprised of M machines. Each machine is characterised by two states: *healthy* and *non-healthy*. Healthy state means that machine is functioning properly and thus no maintenance is needed, and non-healthy otherwise. Following, our model is comprised of three distinctive layers: *IoT layer*, *Fog layer* and *Decision layer*. The functionality of each layer is as follows:

- *IoT layer*: This layer consists of N , $N > 0$ in total sensor nodes (depicted as blue dots) which perform the following undemanding computationally tasks: 1) Raw data measurements (e.g. vibration signals) from each machine 2) Send the data to their gateway nodes. Therefore, each machine has $\frac{N}{M}$ nodes mounted randomly all over it. These nodes communicate only with their gateway nodes in wireless mode (e.g. wifi connectivity) and have limited computational capabilities and energy reserves. We assume that due to small distance between sensor and gateway nodes, the signal quality is always good.
- *Fog layer*: This layer consists of M gateway nodes (or fog nodes or workers), each of which is mounted on the top of every machine. These nodes are more powerful as compared to the sensors and perform the following tasks: 1) Combining and aggregating the classification results from IoT layer, 2) Perform feature computation and classification and 3) Send and receive data/instructions from the Cloud server (i.e. Master node). The data exchange instructions are set by ADMM algorithm.
- *Decision layer*: This layer consists of the Cloud server (Master node) which is responsible for all decision making, handling demanding computational tasks and applying our proposed optimisation method to determine the proper machine status result. In particular, Cloud node does: 1) Receives aggregated classification data from fog nodes 2) Perform fusion and yields decision results 3) Iteratively updates primal and dual optimisation variables as defined by ADMM steps [8]. We assume that information reaches Cloud node via the fog nodes.

2.2. Channel Model

We are interested in smooth and energy efficient information flow between fog nodes and the cloud, due to the fact that fog nodes aggregate decision features concerning a machine's state, and thus the information sent is vital. On the

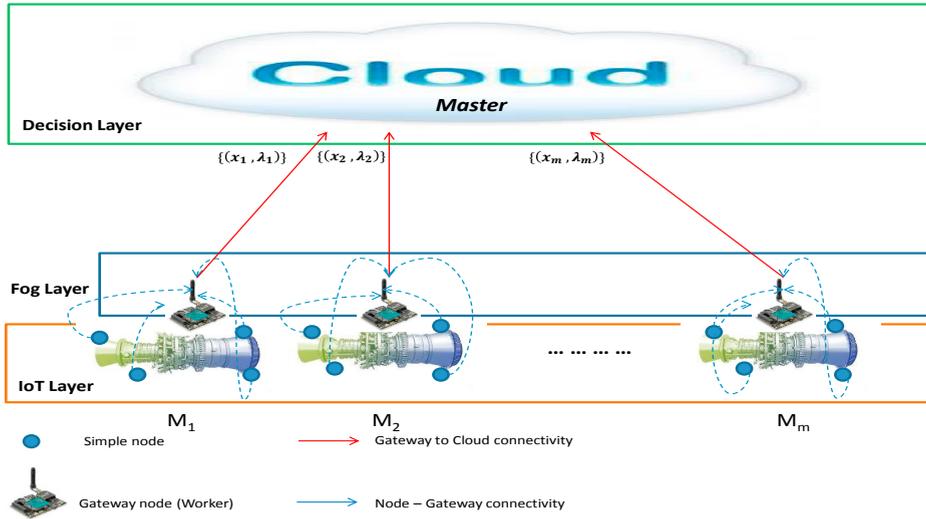


Fig. 1. System Model

other hand, we may relax the previous assumption regarding information flow from IoT to Fog layer, due to the fact that sensor nodes close proximity and the high data correlation among raw measurements, may allow for some data loss.

We assume log-normal shadowing path loss channel model. This model can be used for large and small coverage systems. Empirical studies [9] have shown that log-normal shadowing model provides more accurate multi-path channel models than Nakagami and Rayleigh for a deployments prone to *noise*, such as the industrial. The path loss $PL(d_{ij})$ is given by:

$$PL(d_{ij}) = PL(d_0) + 10\beta \log\left(\frac{d_{ij}}{d_0}\right) + X_\sigma \tag{1}$$

where d_{ij} is the distance between transmitter i and receiver j , d_0 is a reference distance, β is the path loss exponent, usually $2 \geq \beta \leq 4$ and X is a zero-mean Gaussian Random variable in dBm with standard deviation σ for shadowing effects. Assuming the transmission power P_t in dBm , the received power P_r is given by:

$$P_r = P_t - PL(d_{ij}) \tag{2}$$

and SNR at distance d_{ij} is given by:

$$SNR(d_{ij}) = P_r - P_n \tag{3}$$

where P_n is the *noise* power in dBm .

2.3. Machine Control Monitoring Formulations

Our work on an industrial MCM use case, in which the monitoring and classification operation is divided into the following phases [5]:

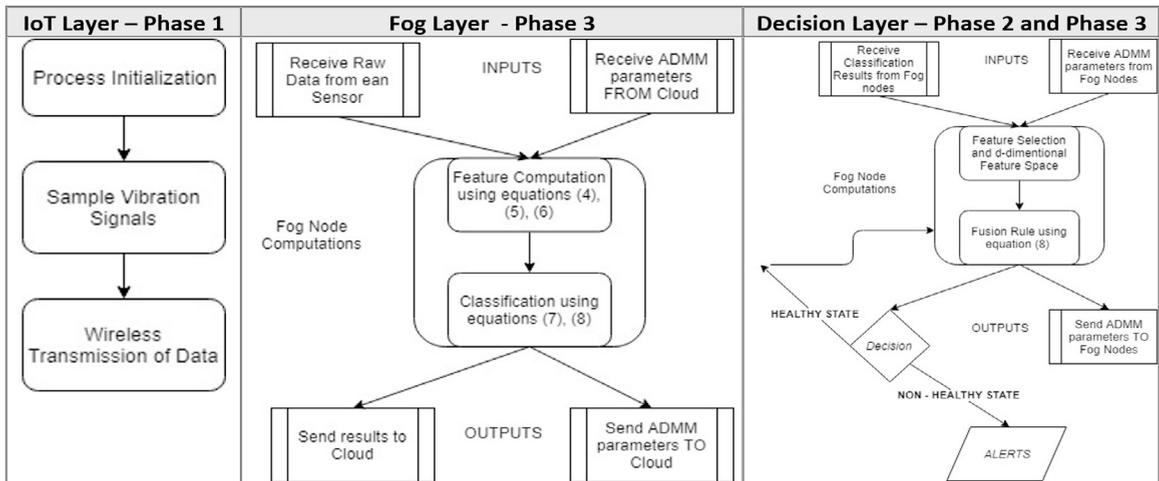


Fig. 2. Phases of Operation per Layer

- **Phase 1: Data Sampling:** Sensors from IoT layer take samples regarding vibration signals, with sampling rate at 7.7 KHz (assuring the Nyquist limit of 3.85 KHz), which satisfies the requirements for vibration signals in the range of 0 - 2 KHz. The data is forwarded to the fog nodes.
- **Phase 2: Feature Selection:** The Cloud node selects the appropriate features, which are then used for classification purposes by the fog nodes. The set of d selected features creates a $d - dimensional$ feature space, $x \in R^d$. The classifier (fog node) decides whether a given sample belongs to the target class or is an outlier. The model function could be chosen by density methods. Once the model has been chosen, the training is initiated with x features as an input. To this end, the process results in parameters μ and C of the model $f(x; \mu, C)$.
- **Phase 3: Monitoring and Classification:** This is the real-time monitoring and classification stage. Each node (IoT layer) periodically samples and sends raw data to fog nodes (Fog layer). Fog nodes compute features and classify according to the rules set by the Cloud. Afterwards the results from each of M machines are send to the Cloud node (i.e. Master node), which *fuses* the results, providing an early warning about the plant status.

In Fig. 2 we depict the phases of operation per layer and the interactions among them for the MCM use case.

2.3.1. Feature Selection

The features are in time domain. These are[6][7]:

- **RMS:** The RMS equation measures the power content in the vibration signature and describes the overall vibrations and the noise level. RMS detects out-of-balance of a rotary machine. The equation is the following:

$$RMS = \sqrt{\frac{1}{N} \sum_{n=1}^N s_i^2} \quad (4)$$

where s_i is the i_{th} node sample and N is the total samples taken.

- **Crest:** The crest factor focuses on impulsive vibration sources, such as gear tooth damage. The ratio of the peak (max) level to RMS is given by:

$$crest = \frac{PEAK}{RMS} \quad (5)$$

- **kyrtosis:** Kyrtois is the fourth statistical moment and indicates major peaks in the time samples, which are related to an increase vibration level. The equation is the following:

$$kyrtosis = \frac{\sum_{n=1}^N (s_i - \bar{s})^4}{N\sigma^4} \quad (6)$$

where \bar{s} is the mean value of s and σ is the standard deviation.

2.3.2. Classification Rules

The classifier's function derives from the density method [10], in which there is a requirement for large number of training samples. We assume that in our use case the distribution of the features follows normal distribution. In order to determine if a tested feature is a target or an outlier, we use the *Mahalanobis* distance $D(x)$ as follows:

$$D(x) = (x - \mu)^T \times C^{-1} \times (x - \mu) \quad (7)$$

where x is the classified object in the feature domain, μ is the vector of mean values and C is the covariance matrix. Each classifier creates a continuous boundary around the training objects in the feature domain and the determination of target versus outlier x is given by:

$$CL(D(x)) = \begin{cases} target & \text{if } D(x) \leq \theta \\ outlier & \text{if } D(x) > \theta \end{cases} \quad (8)$$

where the boundary threshold th is given as input, e.g. a 5% threshold tightens the surface around the training dataset, therefore a 5% of the data is classified as outliers. According to our system model in Fig. 1, each gateway node sends the classification result to the Cloud node. There, all the results get *fused* according to the classifier combining method, which allows the combination of classifiers. The following equation is based on the *mean weighted vote rule*:

$$r(x) = \frac{1}{M} \sum_{k=1}^M [\tau \times CL(D(x) \geq \theta_k) + (1 - \tau) \times (CL(D(x) < \theta_k))] \quad (9)$$

where $n \in M$ is the total number of fog nodes, k denotes the k_{th} fog node, θ_k is the k_{th} node's classification threshold, constant $\tau = 0.95$. The value of $f(x)$ is between 0 and 1, where 0 responds to a *fault* state and 1 responds to a *healthy* state. The threshold for the final decision k_{th} is set to 0.5

2.4. Vibrations Data Format

In order to evaluate our proposed model, we use bearing test ring data, generated by the NSF I/UCR Center for Intelligent Maintenance Systems [11]. Four bearings were installed on a shaft. The rotation speed was kept constant at 2000 RPM by an AC motor coupled to the shaft via rub belts. A radial load of 6000 lbs is applied onto the shaft and bearing by a spring mechanism. All bearings are force lubricated. Concerning the data structure, three (3) data sets are available. Each data set describes a test-to-failure experiment. Each data set consists of individual files that are 1-second vibration signal snapshots recorded at specific intervals. Each file consists of 20,480 points with the sampling rate set at 20 kHz. The file name indicates when the data was collected. After having processed these files, we have (3) tests which include: 1st: 2156 measurements, 2nd: 984 measurements and 3rd: 4448 measurements, all in total: 7588 measurements.

3. Problem formulation and Solution Approach

In an Industrial scenario, approaches to maintenance management can be grouped into three (3) categories: 1) *Function to Failure*: In this case maintenance interventions happen only after the occurrence of machine failure. Therefore real time machine monitoring is useless. 2) *Preventive Maintenance*: In this case maintenance policies are applied according to a planned and periodic schedule, either the machine is in failure state or not. 3) *Predictive Maintenance*: In this case there is an ongoing estimation process of the health status of a machine, based on feature classification and fusion results. This approach is the most cost effective, in which we apply in our work. We are also estimating the plant’s *operation costs* when applying the 3rd policy. The operation cost minimisation concerns the following metrics: 1) Unexpected Failure Frequency (UF): which is the percentage of failures (non - healthy states) not prevented, i.e. Cloud node considers a healthy state instead of non-healthy. 2) Unexploited Operation (UO): which is the average functional operation time that could have run, but instead Cloud node stops machine as it classifies its state as non - healthy.

Moreover, fog nodes operate in an industrial environment, where signal may get easily blocked or their energy reserves go low. So, a problem of energy efficient power transmission allocation is raised, especially because Cloud node raises the probability of valid healthy state estimation if more results enter the fusion process. Different costs can be associated with *UF* and *UO*, which are c_{UF} and c_{UO} respectively, with $c_{UF} \geq c_{UO}$. In this direction, we formulate an optimisation problem for the MCM use case, regarding the optimal power allocation $P(1, 2, \dots, M)$ for every fog node and the minimisation of the operational costs OC_{tot} . It is not possible to simultaneously minimise both metrics, rather an optimal trade-off is sought through cross layer minimisation process. The considered optimal power allocation policy is accomplished using the ADMM, providing thereby a more efficient and robust problem decomposition. The problem is formulated as:

$$\begin{aligned}
 \min_{P_1, \dots, P_M} \quad & OC_{tot} = \sum_{i=1}^M (UF_i \times c_{UB} + UO_i \times c_{UO}) \quad (10) \\
 s.t. \quad & \sum_{i=1}^M P_t_i \leq P_{t_{tot}} \\
 & P_r_i \geq P_{r_{thres}} \\
 & E_i \geq E_{min} \quad (11)
 \end{aligned}$$

where the constraints guarantee that each fog node the minimum (optimal) required P_t to send data to the Cloud, the P_r power at Cloud node is required for the data to be collected and each fog node accomplishes communication according to its energy reserves E_i . The method of ADMM combines the principles of the dual decomposition using also the augmented Lagrangian tool for gradual learning. We start from the Lagrangian:

$$L(P_{t_1}, \dots, P_{t_M}, \lambda_1, \dots, \lambda_M) = OC_{tot} + \lambda_1 (P_{t_1} - P_{r_{th}}) + \dots + \lambda_M (P_{t_M} - P_{r_{th}}) + \frac{\rho}{2} \left((P_{t_1} - P_{r_{th}})^2 + \dots + (P_{t_M} - P_{r_{th}})^2 \right) \quad (12)$$

To designate the dual decomposition as follows:

$$\min_{\lambda_1, \dots, \lambda_M} \{ \max_{P_1, \dots, P_M} OC_1 + \dots + OC_M + \lambda_1 (P_{t_1} - P_{r_{th}}) + \dots + \lambda_M (P_{t_M} - P_{r_{th}}) + \frac{\rho}{2} \left((P_{t_1} - P_{r_{th}})^2 + \dots + (P_{t_M} - P_{r_{th}})^2 \right) \} \quad (13)$$

The resulting ADMM updates that can be done in parallel are the following:

$$P_{t_1}^{k+1} = \arg \min_{P_1, \dots, P_M} L(P_1, \dots, P_M, \lambda_1^k, \dots, \lambda_M^k) = OC(1) + \lambda_1 (P_1^k - \bar{P}_1) + \rho \left((P_{t_1}^k) - P_{r_{th}} \right)^2 + \dots + \left(P_{t_M}^k - P_{r_{th}} \right)^2 \quad (14)$$

<p>Algorithm 1 Synchronous - ADMM: Processing by Master node</p> <ol style="list-style-type: none"> 1. initialize: $k = 0$, convergence value ϵ 2. Loop (repeat): 3. wait; 4. until receive $P_{l_1}, P_{l_2}, \dots, P_{l_M}$ 5. Calculate OC from eq. 10 and send the values back to fog nodes 6. Set $k \rightarrow k + 1$; 7. Stop once $\lambda_j^{k+1} - \lambda_j^k \leq \epsilon, \forall l \in M$ simultaneously, where ϵ is the convergence rule. 	<p>Algorithm 2 Synchronous - ADMM: Processing by Worker node</p> <ul style="list-style-type: none"> • Parameters: constant step size α and constant convergence value ϵ • Loop (repeat): • Initialize: variables $\lambda_1^k = 1, \dots, \lambda_M^k = 1$ <ol style="list-style-type: none"> 1. The Lagrangian dual problem is solved locally by the fog nodes using eq. 14 - 17 2. Calculate λ values using eq. 18 - 19: $\lambda_l^{k+1} = \lambda_l^k - \alpha \left(\sum_l P_{l_i}(\lambda_l) - Pr_{th} \right) \quad (18)$ and broadcasts the new prices $\lambda_1^{k+1}, \dots, \lambda_M^{k+1}$. 3. Wait until receive updates from the Master 4. Set $k \rightarrow k + 1$ and go to step 1) until satisfying termination criterion. 5. Stop once $\lambda_j^{k+1} - \lambda_j^k \leq \epsilon, \forall l \in L$ simultaneously, where ϵ is the convergence rule.
--	--

Fig. 3. ADMM Algorithms.

$$P_M^{k+1} = \arg \min_{P_M} L(P_1^{k+1}, \dots, P_M, \lambda_1^k, \dots, \lambda_M^k) = OC(M) + \lambda_M (P_M^k - Pr_{th}) + \rho \left((P_1^{k+1}(h_1) - Pr_{th})^2 + \dots + (P_M^k - Pr_{th})^2 \right) \quad (15)$$

To solve the outer problem using the sub gradient updates, we use the following equations:

$$\lambda_1^{k+1} = \lambda_1^k + \rho(P_{l_1}^{k+1} - Pr_{th}) \quad (16)$$

⋮

$$\lambda_M^{k+1} = \lambda_M^k + \rho(P_{l_M}^{k+1} - Pr_{th}) \quad (17)$$

where the λ_l are the Lagrangian multipliers, $\rho > 0$ is a penalty parameter. The above update steps can be easily implemented in a distributed fashion with one Master node (cloud) and M workers, i.e. fog nodes. ADMM can be viewed as a version of the method of multipliers where a special joint optimisation is accomplished. The update steps of eq. 14 - 17 can be carried out independently and in parallel in every fog node (i.e. *Worker*), while the steps of eq. 18 - 19 is performed at a fusion centre, i.e. the *Master* node, which is the cloud server. We now present the Algorithms (algorithms 1 and 2 respectively) depicting the operation of fog nodes and Cloud respectively, executing ADMM's steps.

4. Results and Discussion

We proceed with our simulations using Matlab. In Fig. 3 we depict the total operation cost OC_{tot} as a function of cost ratios $\frac{C_{UO}}{C_{UF}}$ for two scenarios. The first (blue line) is a result of the optimal fog node power allocation due to ADMM and the second (red line) is the total cost with non optimal power allocation. Clearly, as ratios get higher cost function gets higher, but in our case of ADMM optimisation we succeed in average 18% less total operation costs. This is because optimal power allocation guarantees that almost all information flows towards the Cloud, better decisions are reached and thus the costly $UF \leq UO$.

5. Conclusions

In this paper we presented a framework for MCM for an Industrial IoT scenario. We proposed a layered System Model in which instrumentation, decisions and more computationally demanding tasks are performed by the Cloud

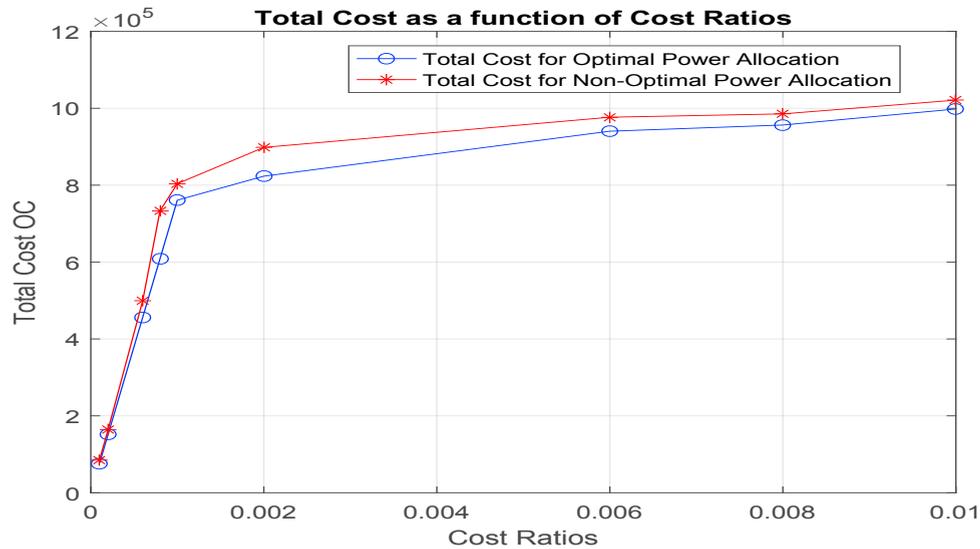


Fig. 4. Total Cost as a function of Cost Ratios for Optimal and Non optimal power allocation

and some data filtering and less demanding estimation operations are performed in parallel by Fog nodes. In order to minimise total plant operational costs, we formulated a cross layer optimisation problem for efficient energy consumption communication among layers and solve it using ADMM algorithms.

Acknowledgements

This research was supported in part by the European Cooperation in Science and Technology (COST Action IC 1406 - Grand: 39320).

References

- [1] Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang and K. Long. (2014) "Cognitive Internet of Things: A New Paradigm Beyond Connection." *IEEE Internet of Things* **1** (2): 129–143.
- [2] J. Yan, Y. Meng, L. Lu and L. Li. (2017) "Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance" *IEEE Access* **5**: 23484–23491.
- [3] I. Stojmenovic and S. Wen (2014) "The Fog Computing Paradigm: Scenarios and Security Issues" *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems (FedCSIS)*: 1–8.
- [4] S. N. Shirazi, A. Gouglidis, A. Farshad and D. Hutchison (2017) "The Extended Cloud: Review and Analysis of Mobile Edge Computing and Fog From a Security and Resilience Perspective" *IEEE Journal on Selected Areas in Communications* **35** **11**.
- [5] G. Antonio Susto, A. Schirru and S. Pampuri (2015) "Machine Learning for Predictive Maintenance: A Multiple Classifier Approach" *IEEE Transactions on Industrial Informatics* **11** **3**: 812–820.
- [6] M. Abu Alsheikh, S. Lin, D. Niyato and H. Tan (2014) "Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications" *IEEE Communications Surveys and Tutorials* **16**.
- [7] J. Neuzil, O. Kreibich and R. Smid (2014) "A Distributed Fault Detection System Based on IWSN for Machine Condition Monitoring" *IEEE Transactions on Industrial Informatics* **10** **2**: 1118–1123.
- [8] R. Zhang and J. Kwok (2014) "Asynchronous Distributed ADMM for Consensus Optimization" *ICML'14 Proceedings of the 31st International Conference on International Conference on Machine Learning* **32**.
- [9] H. Nikoogar and H. Hashemi (1993) "Statistical modeling of signal amplitude fading of indoor radio propagation channels" *2nd International Conference on Universal Personal Communications* **1**:84–88
- [10] K. HempstalkEibe and F.H. Witten (2008) "One-Class Classification by Combining Density and Class Probability Estimation" *Lecture Notes in Computer Science book series (LNCS)* **5211**
- [11] Center for Intelligent Maintenance Systems (IMS) "NASA Prognostics Center of Excellence (PCoE), source: <https://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/>"