

Information Processing and Data Visualization in Networked Industrial Systems

Pavol Mulinka*, Charalampos Kalalas*, Merim Dzaferagic†, Irene Macaluso†, Daniel Gutierrez Rojas‡, Pedro Juliano Nardelli‡, and Nicola Marchetti†

*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona, Spain

†Trinity College Dublin, Ireland

‡LUT University, Finland

Emails: {pmulinka, ckalalas}@cttc.es, {merim.dzaferagic, macalusi, nicola.marchetti}@tcd.ie, {daniel.gutierrez.rojas, pedro.nardelli}@lut.fi

Abstract—Networked industrial systems capitalize on recent advancements in sensing, communications, computing and storage to improve productivity, operational and cost efficiency. The proliferation of effective techniques for knowledge extraction drive a paradigm shift in industrial environments and provide a fertile ground for enhanced process monitoring and control capabilities. In an effort to shed light on industrial data management operations, this paper presents two different approaches for dealing with information processing tasks of aggregated sensor measurements. Such tasks constitute part of an end-to-end process monitoring solution which is implemented in an open-source platform following a modular, scalable and interpretable procedure. A mapping of the industrial data processing components to the operational principles and architecture of a cyber-physical system reveals useful insights for an automated supervision of critical processes and workflows.

Index Terms—Industrial Internet of Things, data imputation, fault detection, classification, cyber-physical system, visualization

I. INTRODUCTION

In recent years, key technological advancements in the areas of sensing, computing and wireless connectivity have remarkably transformed existing industrial setups towards fully integrated, automated and interconnected systems [1]. Real-time condition monitoring of assets, identification/classification of abnormal system behavior and predictive maintenance constitute representative use cases of this new Industrial Internet-of-Things (IIoT) paradigm. The staggering volume of measurement data streams, driven by the widespread sensor deployments, in conjunction with expanded computational resources offer enhanced monitoring capabilities and unlock unprecedented application scenarios. Building on this massive data availability, data-driven knowledge-extraction techniques are able to extract hidden patterns, unknown correlations and actionable intelligence with minimum human intervention.

The pervasive industrial modernization heavily relies on efficient data acquisition and ubiquitous connectivity provided by fifth-generation (5G) communication systems [2]. Besides their ability in addressing stringent requirements in terms of latency, reliability and node density, 5G-based solutions allow the expansion of digital operations through agile network deployments. The installation and maintenance cost associated

with fixed-network technologies can thus be significantly reduced. Nevertheless, the integration of advanced wireless connectivity enablers comes inadvertently with challenges which involve distortions and missing data owing to the inherently shared wireless medium [3]. In addition, industrial plants are typically characterized by harsh propagation environments due to the presence of large metallic objects, mobile units and scattering waves which produce rich multi-path components and may cause connectivity links to be in outage [4].

The massive number of sensors and the heterogeneity of aggregated measurements, such as sensor data, equipment status reports and logistic information, pose new challenges for efficient data fusion and analytics. In this context, emerging fog/edge computing architectures aim at storing, processing, analyzing and responding to data close to the acquisition sensors, enabling dramatically faster processing times and localized decision-making [5]. Locally fused data at the edge nodes combine information from the ambient measurement space to extract the underlying dynamics of the industrial process and correct problematic/noisy data, among other functionalities. Industrial data processing tasks typically include classification, clustering, dimensionality reduction, imputation, prediction, anomaly detection, etc. Such tasks rely on statistical analysis, machine learning (ML) techniques and knowledge rules [6].

Besides information processing, data visualization renders knowledge extraction and understanding of complex large-scale industrial systems much simpler and interpretable. Visualization tools allow a system operator to monitor the infrastructure conditions via key performance indicators (KPIs) and perform informed actions in real-time [7]. Measurement trends and cross-correlations, which otherwise would remain unexplored, can be displayed in a perceptible and intuitive manner. In addition, highly customized dashboards support tailored queries and provide a wide range of charting capabilities, e.g., trajectory graphs and trend maps, for analyzing and presenting the monitoring information.

Contribution: This paper delves into the data-driven operational principles of networked industrial systems with a particular emphasis on information processing and visualization of monitoring data. In particular, we present two different approaches dealing with the key tasks of data imputation,

compression and classification at the level of a fusion center. These tasks constitute part of an end-to-end monitoring solution which is implemented and visualized in an open-source platform. We further provide architectural considerations by highlighting the similarities of industrial process monitoring with the key operational pillars of a cyber-physical system.

Organization: Material in this manuscript is organized as follows. Section II presents two different approaches for dealing with data imputation, compression and classification tasks at the level of a fusion center. The key principles pertaining to the implementation and visualization of our end-to-end monitoring solution are outlined in Section III. Section IV describes the mapping between the industrial data processing flows and the operational principles of a cyber-physical system. Section V provides our concluding remarks.

II. AGGREGATED INFORMATION PROCESSING TASKS

One of the innate challenges for efficient data fusion in networked industrial systems refers to the emergence of missing information in the aggregated measurement streams. In this context, data *imputation* techniques aim to obtain accurate estimates of incomplete sensor trajectories whose missing values can be attributed to hardware malfunctions, imperfect connectivity, security attacks, etc. In addition, *compression* tasks are necessary to ensure scalability and data reduction in large-scale industrial setups. Compression should be efficiently performed to keep the reconstruction error at minimum levels. Finally, *classification* techniques are able to identify common patterns among measurement streams and enhance the process of anomaly detection in the aggregated data.

In what follows, we present two different approaches for dealing with the aforementioned information processing tasks at the level of a fusion center.

A. Dynamical systems for imputation and compression

Dynamical systems offer an interpretable mathematical framework to i) learn the hidden patterns of time-series sensor data which exhibit high spatiotemporal correlation and ii) mine their underlying dynamics to gain insight into the evolution of the process being monitored. As such, dynamical systems provide an effective means for imputation of missing data and compression of the aggregated content at a fusion center [3]. In linear dynamical systems, the dynamics governing the sensor measurements, z_t , are captured by latent variables, h_t , through linear mappings, i.e.,

$$h_{t+1} = Fh_t + e_h, \quad e_h \sim \mathcal{N}(0, \Lambda), \quad (1)$$

$$z_t = Ch_t + e_z, \quad e_z \sim \mathcal{N}(0, \Sigma). \quad (2)$$

Eq. (1) expresses temporal correlations among latent variables of measurement streams, while Eq. (2) captures spatial interactions among different measurements in the same time-step. Gaussian transition and observation noises are further assumed.

In the presence of missing entries in the aggregated measurements, learning parameters $\theta = \{F, \Lambda, C, \Sigma\}$ can be achieved through the maximization of the expected

TABLE I: Imputation performance for varying percentage of missing measurements

	Missing values % ($\times 100$)				
	0.1	0.2	0.3	0.4	0.5
RMSE	0.146	0.188	0.201	0.227	0.239

log-likelihood of the observation sequence by means of Expectation-Maximization (EM) algorithm [8]. The EM algorithm follows an iterative coordinate descent procedure for obtaining the maximum likelihood estimates of θ from incomplete data by successively maximizing the expected log-likelihood of the observation sequence. Upon EM convergence, missing measurements can be computed from the estimation of the latent variables.

An alternative procedure can be followed based on Bayesian updates using sampling by setting conjugate prior distributions over all parameters [9]. This method provides the added benefit of uncertainty quantification based on computed position densities over the parameter space. The aforementioned computation is carried out by Gibbs sampling, which constitutes an iterative Markov chain Monte Carlo (MCMC) scheme [10]. Missing values can be iteratively imputed by computing their conditional expectation with respect to the values of observed measurements, the posterior expectations of latent variables and the updated parameter values. Table I shows the imputation performance of the latter approach, in terms of root mean squared error (RMSE), for randomly missing values among measurement streams and time-steps. A power system synchrophasor dataset with intrinsic spatiotemporal structure has been considered for the evaluation [11]. As expected, imputation performance registers a decline with increasing rate of missing entries, albeit not at prohibitive levels.

At the fusion center, it is also desirable to achieve compression of the aggregated measurements due to storage limitations. The compression should be characterized by a balanced approach to the tradeoff between compression ratio¹ and reconstruction error. In this case, instead of directly storing sensor observations, compression can be accomplished by means of temporally downsampling latent variables h_t and further storing learned parameters in matrices F and C , as well as noise standard deviations Λ and Σ . The decompression error can be computed by the ℓ_2 -norm of mismatch between observed measurements and their estimated counterparts.

The time points at which latent variables are retained can be deduced by two compression strategies: i) latent variables are stored every m -th time step (i.e., *Strategy 1*); ii) latent variables, as well as their temporal location in the buffer of the fusion center, are retained for the time steps at which the decompression error exhibits values above a predetermined threshold (i.e., *Strategy 2*). The two strategies are compared in Table II with a baseline approach that combines Singular Value Decomposition (SVD) and linear interpolation for compression of synchrophasor measurements. It can be observed

¹The compression ratio is defined as the ratio of the total uncompressed space over the size of compressed data.

TABLE II: Performance comparison in terms of RMSE for three different compression techniques

	Compression ratio				
	50	100	150	200	250
Baseline	0.164	0.193	0.225	0.317	0.499
Strategy 1	0.164	0.192	0.218	0.266	0.378
Strategy 2	0.164	0.189	0.212	0.24	0.327

that the proposed strategies outperform the baseline, especially for high compression ratios. In addition, Strategy 2, which selectively stores the hidden variables based on the resulting error, achieves superior compression performance.

B. GAN-based approach for imputation and classification

A Generative Adversarial Network (GAN) constitutes an ML framework designed to create new data instances that resemble the training dataset [12]. This is achieved through the iterative simultaneous training of two neural networks, called the *generator* and the *discriminator*. The generator learns the distribution of the input dataset and attempts to generate data instances that resemble the distribution as closely as possible. On the other side, the discriminator learns to distinguish true data from the output of the generator. The generator is trained to generate data that can fool the discriminator; the discriminator is trained to maximize the probability of correctly labeling training data and generated data. Such learning framework can be applied in an IIoT scenario to learn the distribution of the sensor measurements, allowing for the imputation of missing data.

Our approach builds on the work in [13], which adapted the original GAN framework to the data imputation problem. However, we embed the GAN-based data imputation module within an IIoT monitoring system (see Fig. 1). In fact, as detailed in [14], we propose to validate the GAN performance by assessing the impact of the generated data on the fault detection and classification modules. This way, the GAN hyperparameter optimization is driven by the resulting repercussions of using imputed data on the industrial monitoring system. In this context, this type of feedback is more informative and easier to understand than metrics traditionally used to assess an ML model, e.g., the RMSE.

We now give a brief overview of the GAN-based imputation module. The training process starts with the optimization of the discriminator D using mini-batches of size K_D . The generator G is kept fixed during the optimization of the discriminator. Eq. (3) is used for the training of D , i.e.,

$$\min_D - \sum_{j=1}^{k_D} \mathcal{L}_D(\mathbf{m}(j), \hat{\mathbf{m}}(j), \mathbf{b}(j)), \quad (3)$$

where

$$\mathcal{L}_D(\mathbf{m}, \hat{\mathbf{m}}, \mathbf{b}) = \sum_{i:b_i=0} [m_i \log(\hat{m}_i) + (1 - m_i) \log(1 - \hat{m}_i)], \quad (4)$$

and $\mathbf{m}(j)$ defines whether the j -th sample in the mini-batch is missing or not, while $\hat{\mathbf{m}}(j)$ denotes the prediction made

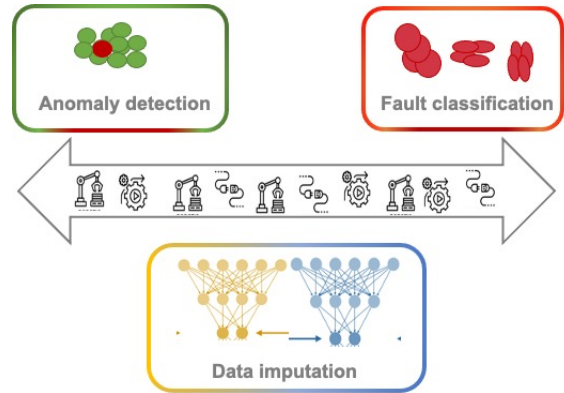


Fig. 1: Hyperparameter tuning for the GAN.

by D . The $\mathbf{b}(j)$ is an N -dimensional vector whose elements are all equal to 1 except for one element which is 0, i.e., this corresponds to the element of the mask $\mathbf{m}(j)$ that is not provided as input to D . After the training process for D , the training of the generator G starts according to Eq. (5), i.e.,

$$\min_G \sum_{j=1}^{k_G} \mathcal{L}_G(\mathbf{m}(j), \hat{\mathbf{m}}(j), \mathbf{b}(j)) + \alpha \mathcal{L}_M(\mathbf{x}(j), \hat{\mathbf{x}}(j)), \quad (5)$$

with mini-batches of size K_G . Eqs. (6) and (7) denote the two components of the cost function for G . In particular, Eq. (6) applies to the missing sensor measurements while Eq. (7) applies to the observed measurements, i.e.,

$$\mathcal{L}_G(\mathbf{m}, \hat{\mathbf{m}}, \mathbf{b}) = - \sum_{i:b_i=0} (1 - m_i) \log(\hat{m}_i), \quad (6)$$

$$\mathcal{L}_M(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{i=1}^d m_i (\hat{x}_i - x_i)^2. \quad (7)$$

It is worth noting that while in [14] the fault detection and classification modules are implemented as an autoencoder and a deep neural network respectively, other options are viable and can be integrated with the GAN. The fault detection module can be implemented through any anomaly detection technique (e.g., clustering, autoencoder, principal component analysis). Similarly, the fault classification module can be implemented via any classification method (e.g., deep neural network, support vector machine, decision tree).

III. VISUALIZATION OF END-TO-END MONITORING SOLUTION

In networked industrial systems, data visualization tools offer valuable insights for the real-time performance of monitoring processes. In this context, we introduce a modular and containerized end-to-end solution for the purpose of evaluation, implementation and integration of data processing and ML methods into an IIoT ecosystem.

Our scalable testbed platform is implemented in Docker² containers and it is publicly available on GitHub [15]. Due to

²Docker [Online]: <https://www.docker.com>.

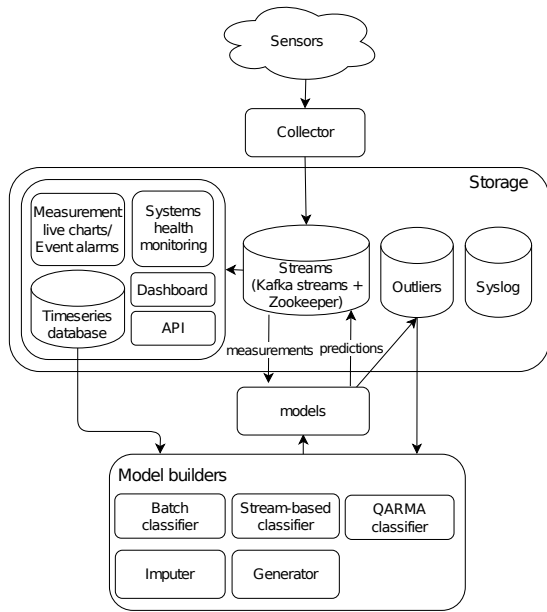


Fig. 2: Building blocks of our modular monitoring solution where the involved interactions (data flows) between individual components are displayed.

repository space restrictions, we have included only a portable UCI Spambase dataset [16] in the testbed. However, incorporation of large-scale industrial datasets including normal and anomalous system behavior is also possible. Fig. 2 illustrates the monitoring solution design and the interactions between the incorporated modules. In particular, the key building blocks of our end-to-end solution include:

- **Generator:** A Python script sending rows from the UCI Spambase dataset to a collector using the HTTP POST method. This script mimics the communication between sensors and the fusion center.
- **Collector:** A RESTful application program interface (API) implemented in Python Flask web framework used to authenticate connecting clients and validate received data.
- **Storage:** A module including databases and applications for data storage (database and queue) and visualization.
- **Models:** A module representing a trained ML model currently used for classification of events.
- **Model builders:** A framework for tracking of the ML model experiments.

To support communication, we adopt (i) HTTP POST method with enabled authentication for connectivity between the sensors and the monitoring solution; and (ii) Kafka³ streams, with a role of distributed message queue and storage of intermediate results, for communication between applications. The **Storage** block constitutes the epicenter of the proposed monitoring solution and further consists of the following components:

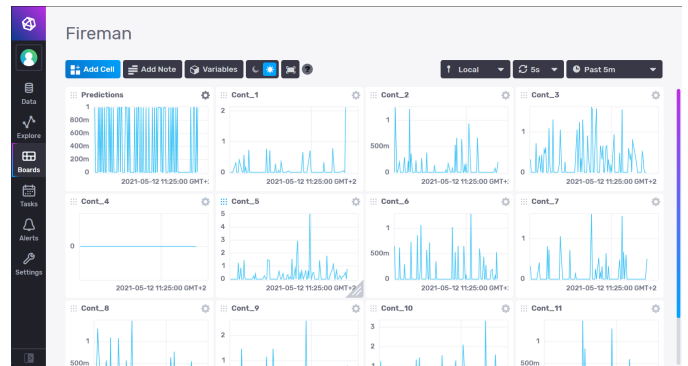


Fig. 3: Dashboard with live charts in InfluxDB, where binary classification predictions are illustrated with live charts for dataset features.

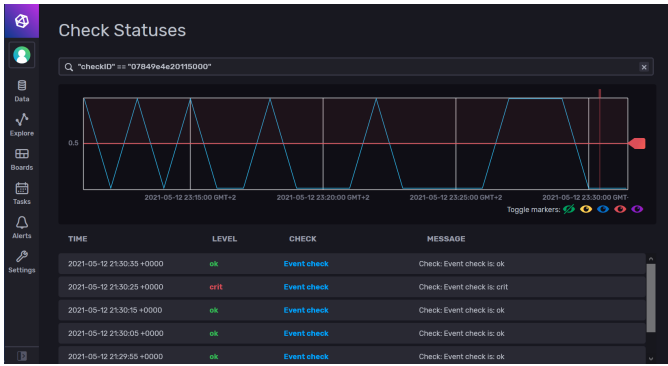
- **Syslog:** A container or localhost for aggregation and storage of all log files.
- **Outliers:** A database to store detected outliers. As the outliers accumulate over time, we periodically check if they can form a new class or cluster in the measurements and use them in the training of the ML models.
- **Streams:** This block consists of Kafka brokers and Zookeeper to maintain and coordinate the distributed Kafka streams in a reliable manner.
- **Timeseries** and corresponding components: In our case, we use the InfluxDB 2.x time series database⁴ and corresponding components, i.e., Telegraf, dashboard and alerts, to monitor sensors and Docker containers health. Telegraf is a plugin-driven server agent for collecting and reporting KPIs, and it can be jointly used with Kafka consumer client to forward the data from Kafka streams to time series database and Influxdb dashboard (shown in Fig. 3) with alerts for measurement live charts and notifications (shown in Fig. 4).

For data prediction tasks, we incorporate a set of models built for the ML pipeline, as illustrated in Fig. 5. Individual data analysis components are publicly available in the form of Jupyter notebooks on GitHub [17]. In particular, we have so far incorporated diverse approaches for event detection in data streams which can be further used for IIoT anomaly detection scenarios, i.e.,

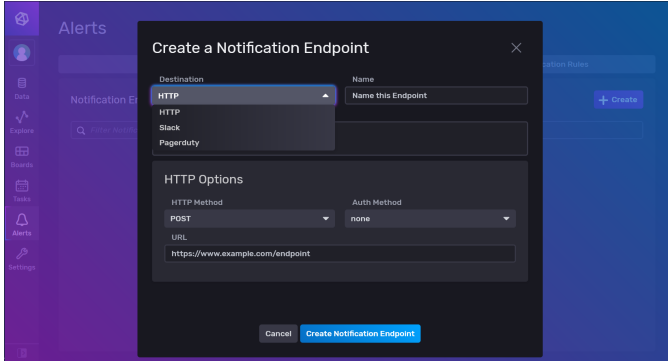
- **Batch classifier:** Traditional ML models trained on fixed size datasets.
- **Stream-based classifier:** Stream-based ML models continuously adapting to data streams.
- **QARMA** [18]: Classifier based on quantitative association rules mining.
- **Imputer:** Methods and models for the imputation of missing measurements.
- **Generator:** Methods and models for the generation of new data to deal with imbalanced datasets.

³Kafka [Online]: <https://kafka.apache.org>.

⁴InfluxDB [Online]: <https://www.influxdata.com>.



(a) Alert definition.



(b) Notification endpoint.

Fig. 4: InfluxDB allows the definition of threshold values for alerts and notification endpoints. We can easily define values of interest in predictions, e.g., 0.5 in case of binary classification for no event vs event occurrence.

In addition, the core ML pipeline used for the predictions of events consists of the following blocks:

- **Normalization/standardization:** Scaling of the dataset.
- **Imputer:** Imputation of the missing data.
- **Feature engineering:** Analysis, selection of features and reduction of feature space.
- **Outlier detection:** Specific methods and models for outlier detection.
- **Dataset balancing:** Oversampling/undersampling synthetic data generation methods for imbalanced dataset.
- **Classification, evaluation, tuning:** Evaluation and tuning of classification algorithm for best results.

In our end-to-end platform, we have selected the combination of Python libraries, RESTful API, Kafka streams and InfluxDB time series database to keep the solution open-source, modular, scalable and robust. All discussed components constitute pieces of publicly available open-source projects and can be easily interchangeable or removable.

IV. ARCHITECTURAL CONSIDERATIONS FOR IIOT

Despite its generality, the proposed information processing and data visualization framework is designed to improve the operation of industrial plants. The modular incorporation of networking, computing and decision-making components aims

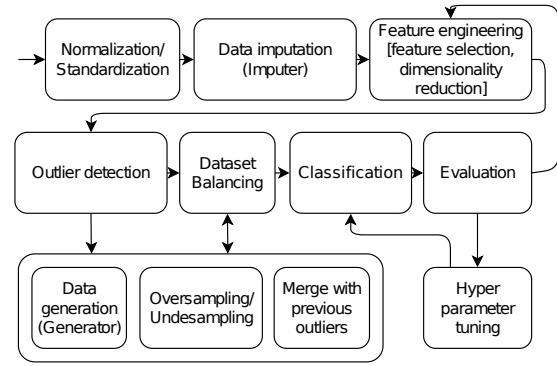


Fig. 5: ML pipeline used for data processing tasks, e.g., predictions of events.

at bridging the gap between the physical and digital worlds, and extends the preliminary conceptualization introduced in [19], as illustrated in Fig. 6. A networked industrial environment can be mapped to a cyber-physical system whose prototype architecture comprises three layers, namely physical, data and decision layers.

The physical layer refers to the specific industrial process being monitored, specifying the data acquisition and transmission methods. Sensors constitute the elements that perform the first mapping of the physical processes into the cyber domain which are generally characterized by the key conceptual traits of high-order, physical context and temporal smoothness, which may render conventional data processing methods inefficient. The signals generated by the sensors correspond to spatially co-evolving time-series with underlying physical meaning (i.e., semantics), rendering techniques, as the ones presented in Section II, suitable for information processing.

At the next stage, aggregated information can be directly or indirectly used for downstream tasks, e.g., impute missing measurements or detect and classify faults in the process monitoring data. The data layer is mainly constructed upon logical (and not physical) relations and involves fusion, data prediction, classification, etc. This architectural layer constitutes the necessary mediation between the physical process and the informed decision-making for possible supervisory actions.

At the final stage (i.e., decision layer), situational awareness is achieved for effective decision-making. Based on the knowledge extracted by the data layer and with the help of appropriate visualization tools and platforms (as discussed in Section III), instructive and actionable insights can be derived in real-time towards an enhanced end-to-end performance, e.g., flag whether a fault has happened and diagnose its type in event-detection operations.

The three architectural layers of an IIoT system are abstractions but can be easily particularized by defining the boundary conditions of the actual use case to be studied, as carefully described in [19]. The idea is that the data flows from physical processes through a well-defined acquisition method to produce a dataset (including fused, aggregated

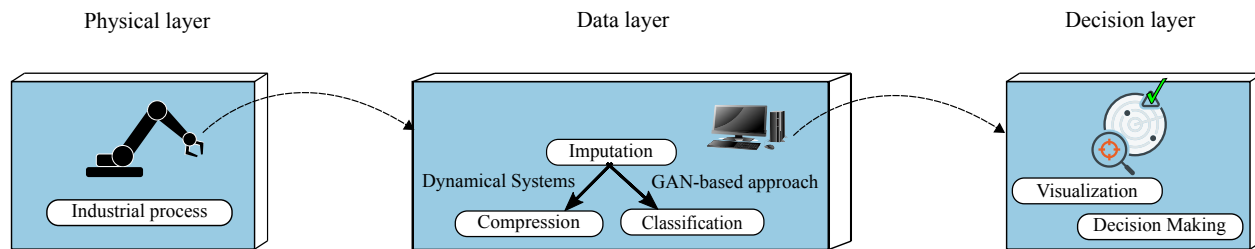


Fig. 6: Mapping of the three-layer architecture of cyber-physical systems to the different industrial data processing and visualization stages.

and/or imputed measurements) that will be used by a decision-making process related to possible actions to be taken for the operation of industrial plants. What is important to note is that the proposed solution assumes the availability of a scalable and reliable underlying communication system capable of supporting machine-type data transmission in large amounts towards computing units, being them cloud- [20] or edge-based [2].

V. CONCLUSIONS

This paper deals with information processing and visualization of monitoring data in networked industrial systems. We present two different approaches for data imputation, compression and classification tasks at the level of a fusion center. These tasks constitute part of an end-to-end monitoring solution of industrial processes. The modular composition of our solution facilitates interpretability and allows the derivation of useful insights related to the state of the monitored process. Finally, a mapping of the different data processing stages to the operating principles and architecture of a cyber-physical system is performed, highlighting the similarities between the two frameworks.

In the path forward, we will direct our efforts towards the applicability of the proposed information processing and data visualization framework in predictive maintenance use cases related to automotive manufacturing.

ACKNOWLEDGMENT

This work is partly supported by the FIREMAN project CHIST-ERA-17-BDSI-003 funded by the Spanish national foundation (PCI2019-103780), by the Irish Research Council (IRC) and Academy of Finland (AKA; n.326270). This work is also partly supported by AKA: (a) ee-IoT n.319009, and (b) EnergyNet n.321265/n.328869, by JAES Finnish Foundation via STREAM project, and by the Generalitat de Catalunya under Grant 2017 SGR 891.

REFERENCES

- [1] R. Alur, *Principles of Cyber-Physical Systems*. The MIT Press, 2015.
- [2] A. Narayanan, A. S. De Sena, D. Gutierrez-Rojas, D. C. Melgarejo, H. M. Hussain, M. Ullah, S. Bayhan, and P. H. Nardelli, "Key advances in pervasive edge computing for industrial internet of things in 5g and beyond," *IEEE Access*, vol. 8, pp. 206734–206754, 2020.
- [3] C. Kalalas and J. Alonso-Zarate, "Sensor data reconstruction in industrial environments with cellular connectivity," in *2020 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE PIMRC '20)*, 2020.
- [4] Z. Kun, L. Liu, T. Cheng, Y. Ze, and Z. Jianhua, "Channel Measurement and Characterization for Industrial Internet of Things," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–5, 2019.
- [5] W. Dai, H. Nishi, V. Vyatkin, V. Huang, Y. Shi, and X. Guan, "Industrial Edge Computing: Enabling Embedded Intelligence," *IEEE Industrial Electronics Magazine*, vol. 13, no. 4, pp. 48–56, 2019.
- [6] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: an overview of methods, challenges, and prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.
- [7] A. R. Jadhav, S. Kiran M. P. R., and R. Pachamuthu, "Development of a Novel IoT-Enabled Power-Monitoring Architecture With Real-Time Data Visualization for Use in Domestic and Industrial Scenarios," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–14, 2021.
- [8] S. Roweis and Z. Ghahramani, "A Unifying Review of Linear Gaussian Models," *Neural Computation*, vol. 11, no. 2, pp. 305–345, 1999.
- [9] A. Wills, T. B. Schön, F. Lindsten, and B. Ninness, "Estimation of Linear Systems using a Gibbs Sampler," *IFAC Proceedings Volumes*, vol. 45, no. 16, pp. 203–208, 2012. 16th IFAC Symposium on System Identification.
- [10] R. E. Kass, B. P. Carlin, A. Gelman, and R. M. Neal, "Markov Chain Monte Carlo in Practice: A Roundtable Discussion," *The American Statistician*, vol. 52, no. 2, pp. 93–100, 1998.
- [11] *EPFL Smart Grid*. [Online]. Available: <http://nanotera-stg2.epfl.ch/data>. Accessed: 12 February 2021.
- [12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," 2014.
- [13] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International Conference on Machine Learning*, pp. 5689–5698, PMLR, 2018.
- [14] M. Dzaferagic, N. Marchetti, and I. Macaluso, "Fault detection and classification in Industrial IoT in case of missing sensor data," *IEEE Internet of Things Journal*, 2021. preprint on TechRxiv at <https://doi.org/10.36227/techrxiv.14540061.v1>.
- [15] *FIREMAN monitoring solution repository*. [Online]. Available at: https://github.com/5superpalo/FIREMAN-project_frontend.
- [16] D. Dua and C. Graff, "UCI machine learning repository," 2017.
- [17] *FIREMAN analysis scripts and documentation*. [Online]. Available at: <https://github.com/5superpalo/FIREMAN-project>.
- [18] I. T. Christou, E. Amolochitis, and Z.-H. Tan, "A parallel/distributed algorithmic framework for mining all quantitative association rules," *arXiv preprint arXiv:1804.06764*, 2018.
- [19] D. Gutierrez-Rojas, M. Ullah, I. T. Christou, G. Almeida, P. Nardelli, D. Carrillo, J. M. Sant'Ana, H. Alves, M. Dzaferagic, A. Chiumento, and C. Kalalas, "Three-layer Approach to Detect Anomalies in Industrial Environments based on Machine Learning," in *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*, vol. 1, pp. 250–256, 2020.
- [20] M. Ullah, P. H. Nardelli, A. Wolff, and K. Smolander, "Twenty-one key factors to choose an iot platform: Theoretical framework and its applications," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10111–10119, 2020.