



This is an electronic reprint of the original article. This reprint may differ from the original in pagination and typographic detail.

Kleyko, Denis; Osipov, Evgeny; Papakonstantinou, Nikolaos; Vyatkin, Valeriy Hyperdimensional computing in industrial systems

Published in: IEEE Access

DOI: 10.1109/ACCESS.2018.2840128

Published: 28/05/2018

Document Version Publisher's PDF, also known as Version of record

Published under the following license: CC BY

Please cite the original version:

Kleyko, D., Osipov, E., Papakonstantinou, N., & Vyatkin, V. (2018). Hyperdimensional computing in industrial systems: the use-case of distributed fault isolation in a power plant. *IEEE Access*, *6*, 30766-30777. https://doi.org/10.1109/ACCESS.2018.2840128

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.



Received April 13, 2018, accepted May 17, 2018, date of publication May 28, 2018, date of current version June 26, 2018. *Digital Object Identifier* 10.1109/ACCESS.2018.2840128

Hyperdimensional Computing in Industrial Systems: The Use-Case of Distributed Fault Isolation in a Power Plant

DENIS KLEYKO^{^[D]}, EVGENY OSIPOV^[D], NIKOLAOS PAPAKONSTANTINOU², AND VALERIY VYATKIN^[D],³, (Senior Member, IEEE) ¹Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden

¹Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, 971 87 Luleå, Sweden
²VTT Technical Research Center of Finland, 02150 Espoo, Finland
³Department of Electrical Engineering and Automation, Aalto University, 02150 Espoo, Finland

Corresponding author: Denis Kleyko (denis.kleyko@ltu.se)

ABSTRACT This paper presents an approach for distributed fault isolation in a generic system of systems. The proposed approach is based on the principles of hyperdimensional computing. In particular, the recently proposed method called Holographic Graph Neuron is used. We present a distributed version of Holographic Graph Neuron and evaluate its performance on the problem of fault isolation in a complex power plant model. Compared to conventional machine learning methods applied in the context of the same scenario the proposed approach shows comparable performance while being distributed and requiring simple binary operations, which allow for a fast and efficient implementation in hardware.

INDEX TERMS Vector symbolic architectures, Holographic Graph Neuron, distributed representation, complex systems, distributed fault isolation, hyperdimensional computing, machine learning.

I. INTRODUCTION

Flattening the management pyramid by distributing the intelligence across networked hardware devices and thus forming Cyber-Physical Systems (CPS) is one of the main current trends in the development of industrial automation towards industrial processes which are intelligent and flexible. Several standards and technologies currently empower the distribution of the intelligence on different levels.

Intelligent maintenance and condition monitoring of complex industrial systems will be important functions of future automation systems, with the growing usage of methods from machine learning and artificial intelligence. Enabling interpretation and on-line learning of heterogeneous data streams is one of the main current trends in machine learning. The main principle of on-line learning is a dynamic formation of a process model through the observation of data streamed from several sources (controllers, sensors, actuators, etc.) over time. The model changes with the evolution of the modeled process. Thus, the on-line learning addresses an important challenge for the modern control systems, the obsolescence of models of system processes. This should increase the efficiency of automation of industrial systems.

Another challenge in CPS is a transformation of raw measurements observed by heterogeneous sources into knowledge. This knowledge has an ultimate importance for the efficient decision support in CPS. The traditional approach to this problem is centralized. Huge amounts of data are normally aggregated in powerful computers where models are built. The questions of distributed in-network processing are so far given limited attention. This is due to a high computational complexity and the centralized nature of the processing algorithms not suitable for low-power and lowperformance devices. On the other hand, in-network processing could dramatically increase the level of intelligence and autonomy in industrial systems.

This article considers an important automation application of fault management. In particular, a practically implementable, supervised data-driven approach to distributed fault isolation in an industrial system using readings from heterogeneous sensors is proposed. The proposed approach is based on the principles of hyperdimensional computing [1], [2], which operates with vectors of very high dimension (i.e., vectors of several thousand elements) and allows the implementation of sophisticated reasoning based on very simple binary operations. Thus, the article demonstrates the feasibility of implementing complex functionality using simple operations on imprecisely encoded data.

Specifically, the recently proposed method called Holographic Graph Neuron [3], [4], which first maps low-dimensional data into the format suitable for the hyperdimensional computing, and then performs pattern processing operations in a high-dimensional space. One of the advantages of the applied approach is that it might implement the on-line learning paradigm. The IEC 61499 distributed automation architecture [5], which is increasingly used in various distributed automation applications, is adopted to support this paradigm. It provides an executable software model of the implementation that is ready for a distributed deployment. The distributed implementation of the proposed approach will deploy an algorithm in each control node, which learns from sensory data produced by the plant during its faulty and/or stable operation modes. It is also notable that in the proposed approach, models created in this way can be kept up-to-date via the on-line learning as the industrial system evolves.

The main contribution of this article is in the demonstrated feasibility of hyperdimensional computing for the problem of distributed fault isolation in a complex industrial system where the system is exemplified on a generic model of a Nuclear Power Plant. The results are compared with the performance of conventional machine learning methods: Artificial Neural Networks, k-nearest neighbors, decision trees, and Sequential minimal optimization for the same problem. The proposed approach demonstrates the accuracy (up to 95%) comparable to the results of conventional methods. An additional advantage of hyperdimensional computing is its feasibility of on-line learning as explained in Section IV.

The preliminary results of this work were presented in [6]. The work there covers the centralized version of the proposed approach and its comparison with the conventional machine learning methods (4 out 5 methods used in this article, see details in Section VI-G). Results reported in this article also include the distributed version of the approach (Section IV) and its performance for two different configurations in the considered fault isolation scenario (Section VI-H).

The article is organized as follows. System model and problem definition are described in Section II. Section III overviews related approaches to fault diagnosis and isolation as well as related works in the area of hyperdimensional computing. Section IV outlines the encoding of sensory data as well as the training and operational phases of the Holographic Graph Neuron for the considered fault isolation application. Section V introduces the essentials of hyperdimensional computing and presents the details of the Holographic Graph Neuron approach in the context of fault isolation in industrial systems. Section VI presents the case study scenario and the results of the performance evaluation of the proposed approaches and comparison with the conventional methods. Section VII concludes the article.

II. SYSTEM MODEL AND PROBLEM DEFINITION

This section discusses the complications of the conventional data-driven approaches to fault isolation as the main motivation for the proposed solution. Further, it describes the system model, notations used throughout the article and formulates the considered problem.

A. MOTIVATION FOR THE PROPOSED ARCHITECTURE: CLASSIC DATA-DRIVEN APPROACH TO FAULT ISOLATION AND ITS SHORTCOMINGS

The conventional supervised machine learning based datadriven fault isolation approach consists of the following phases: a collection of labeled training sensory data; training of a model for fault isolation using the labeled training data; and the operational phase. In the operational phase, it is necessary that a fault is first detected from the sensory measurements by either the trained model or some additional fault detection mechanism. Next, the trained model is provided with the sensory measurements in order to classify (isolate) the detected fault. In such architecture, the fault isolation layer observes the process "through the eyes" of the distributed automation infrastructure and informs the latter of possible faults. In the classic centralized approach, the fault isolation is implemented at one dedicated computational resource. An alternative to that is distributed implementation, where the process can be decomposed into several components deployed directly to the automation (control) nodes.

During the data collection phase, a training batch of sensory measurements from different system components is gathered and labeled, where a component can be a sensor, an actuator, a controller, etc. Usually, raw sensory measurements are used to extract features that are then used by machine learning methods. Several features can be extracted from a single sensory signal. For example, in the case study below, three statistical features (mean, max, and min) were extracted from a signal measured by each sensor over a predefined time interval. Labels in the training batch associate set of features extracted from the sensory measurements with the corresponding state of the system, e.g., with the component, which is faulty. The training batch can be created from either system simulations or historical logs. During the training phase a chosen machine learning algorithm is run on the training batch labeled with the corresponding faults. Because in many practical cases when working with complex systems the interdependencies between different components are not known or require sophisticated analysis, all available sensory signals are used during the training phase for all the methods except the distributed version of the proposed approach.

The choice of the use of measurements from all components implies two main shortcomings. First, most of the conventional machine learning methods have limitations on the number of features in the data set. For example, when the number of features is increasing linear methods such as logistic regression demonstrate degraded convergence and computational performance. In the knowledge-based and specifically in the pattern matching based classification approaches (e.g., Artificial Neural Networks (ANNs)) with the increased number of features the size of the training data set should be enlarged in order to avoid the overfitting problem. The second shortcoming comes from the inherent centralized nature of the conventional machine learning methods. Indeed, in order to classify a previously unseen input pattern of features (i.e., sensory measurements from different components), the method needs to be presented with a full pattern. If only a partial pattern is presented, it will substantially decrease the accuracy unless it was used during the training phase.

B. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a generic CPS (see Section VI for the model of the system considered for the use-case). The distributed control and automation infrastructure consists of k components equipped with various function-specific sensors and actuators. An example of a component in an industrial system could be an automated valve or a pressure sensor connected to a Supervisory control and data acquisition (SCADA) system. Importantly, some components are equipped only with actuators and lack sensory outputs. Functional components with sensors have a signal processing block, which performs initial pre-processing of the raw signal (e.g., denoising, envelope extraction) and extraction of statistical features (e.g., mean, max, and min). Note that alternative feature extraction is not the focus of this article but this part of the data set preparation process is still a very important step and there are methods which can be applied on the acquired input time-series. For further discussion, it is assumed that each of k functional components with sensors exposes n features. Therefore, a system-wide state at time T is jointly characterized by N = kn features. Further, each feature takes values in a discrete range $[1, M_i]$, where M_i is the number of quantization levels for feature *i*. It is important to note that the number of quantization levels and the quantization steps can be different for different features. All N features act as an input to a knowledge management system for modeling system-level faults.

The distributed control and automation infrastructure interacts with the process. It is enhanced with knowledge management abilities that allow learning and conceptualization of new trends into knowledge and then using this knowledge for decision making by distributed control nodes. The IEC 61499 is used as software reference architecture to represent the entire system model. The proposed approach aims at enforcing the emergent distributed automation infrastructures with knowledge management and reasoning.

Formally, the problem is formulated as to isolate a system fault, i.e., the type of the fault and its location, given that the fault was already detected. Such formulation mainly concerns faults in the components, which are not equipped with sensors; therefore, fault isolation is based on the observed system-wide state.

III. RELATED WORKS

This section contains a summary of the state-of-the-art in the fault management and the relevant work in the hyperdimensional computing.

A. SUMMARY OF THE STATE-OF-THE-ART IN THE FAULT MANAGEMENT

This article adopts the generally accepted taxonomy, where the fault diagnosis process consists of two steps: fault detection and fault isolation. Fault detection indicates whether there is a fault in the system. If the fault is detected, fault isolation determines the type of the fault and its location. The process of fault identification determines the size of the fault and the time of onset of the fault.

In general, methods for fault diagnosis are divided into data-driven (agnostic towards the topology of the target system) and model-based methods (a method takes into account information about the structure and/or behavior of the target system). As this article proposes the data-driven approach, this section mainly describes methods, which consider the target system as a black box.

The efforts in this research domain are extensive, and many alternative fault diagnosis techniques and hybrid configurations have been proposed.

The machine learning research domain focuses on the development of algorithms, which enable computers to learn from data. Even though, it has been an active research area since the development of the first artificial intelligence applications; the interest to it has been boosted due to the increased computational resources and the development of new machine learning methods [7] with many engineering applications.

The plethora of research in the fault detection and isolation domain has motivated survey papers overviewing the state of the art. The trilogy of publications [8]-[10] contains a summary of fault diagnosis and identification literature and classifies the research efforts into three categories, depending on the algorithm that is used. The "quantitative modelbased" methods are based on system models. These models analytically identify differences between the expected and the actual behaviors. Decision rules are then applied to perform fault detection and isolation [8]. The "qualitative models and search strategies" category includes methods based on qualitative system models, like topographic templates which are created using the expert knowledge and fault-tree analysis [9]. The last category of fault diagnosis and identification methods is the "process history based methods" which do not utilize any knowledge about the system's structure but rely on data sets of simulated or real process data as input to quantitative (e.g., ANNs) or qualitative (e.g., expert systems) methods [10]. In [11] a literature review, which focuses on model-based methods, is presented; the [12] contains an overview of artificial intelligence based methods.

ANN-based systems were explored for the development of quantitative fault diagnosis and identification systems as well as for safety critical applications [13]. Especially in the nuclear power generation domain, ANNs have been proposed for: fault isolation, accident isolation, transient diagnosis, and condition monitoring.

Hybrid systems involving ANNs and other methods have also been suggested for fault isolation applications. An ANN and an analytical method were combined into a hybrid diagnostic system applied on a nuclear power plant case study. A dynamic neuro-fuzzy network and a dynamic ANN are used in [14] to create an advisory system for the accident diagnosis of a nuclear power plant. Simple time-series and data preprocessed with Fast Fourier Transformation are used in [15] to train ANNs for the improved fault isolation and tolerance against measurement drifts.

Another quantitative method used for fault diagnosis is based on decision trees. Fault isolation systems, which use decision trees, have been applied on AC transmission lines, photovoltaic arrays, power systems [16]. Migration paths for fault diagnosis and identification from fault-trees to decision trees have been proposed for the International Space Station.

Dynamic case-based reasoning, Independent Component Analysis, hidden Markov models, data-driven modeling [17], optimized fuzzy clustering and residual space analysis were also proposed for fault diagnosis and identification.

A decentralized approach to fault diagnosis based on the usage of multiblock kernel partial least squares is presented in [18]. Another distributed approach is described in [19]. It allocates the computationally demanding tasks between different sensor nodes.

B. HYPERDIMENSIONAL COMPUTING

The hyperdimensional computing (it is also referred as Vector Symbolic Architectures, VSAs [20]) is widely used for computer-based semantic reasoning and bio-inspired representations of structured knowledge [1]. There are two main components in the hyperdimensional computing. First, it operates with vectors of very high dimension, which are also referred as a distributed representation of data or HD vectors. Second, there is a set of arithmetic operations that are applied to distributed representations in order to create new structures. The cognitive capabilities achievable using distributed representations on them have been exemplified by a system, which was solving problems in the form of Raven's progressive matrices [21], [22].

The development of VSAs was stimulated by studies on brain activity that showed that the processing of even simple mental events involves simultaneous activity in many dispersed neurons [1]. Information in VSAs is similarly represented in a distributed fashion: a single concept is associated with a pattern of activation of many neurons. This is achieved by using a vector with very large dimensions. Several different types of VSAs have been introduced, each using different representations (see [1]–[4] and references therein).

Recent years reveal a rising interest in applying the principles of hyperdimensional computing to sensory data of technical and biological systems. For example, works [23], [24] are using VSAs for the modeling of statistical dependencies in temporal sequences of heterogeneous measurements. The proposed methods were exemplified via applications in the following areas: a human activity recognition using the accelerometer data and predictions (e.g., the next app to be loaded) using real-life mobile phone user data. Other applications of hyperdimensional computing include natural language processing [25], and numerous applications to classification of EEG error-related

potentials [27], and modality classification of medical images [28]. These results provide the empirical evidence that hyperdimensional computing has a potential to become a powerful tool for the analysis of complex dependencies between a large number of features.

Different approaches to the encoding sensory data into a high-dimensional space and further processing it there can be found in [3], [4], and [29]. The distributed approach to fault isolation presented in this article is based on the Holographic Graph Neuron (HoloGN) method from [3].

IV. THE OUTLINE OF THE APPROACH

This section presents a high-level overview of the proposed approach. The approach can be implemented either in a centralized manner or in a distributed manner. Both versions are introduced where the centralized version is used for benchmarking with the conventional methods. The highlevel principle of the proposed approach to fault isolation is illustrated in Fig. 1.



FIGURE 1. Notations and a graphical representation of the principles used when forming a partial pattern of system's state using HoloGN approach.

The organization and interconnection flow between the important steps are illustrated in Fig. 2.



FIGURE 2. A high-level overview of the proposed approach.

Each cell in the grid in this model (Fig. 1) denotes a value of the system's feature, which is extracted from a particular functional component. Because the HoloGN method applied in this article requires a finite alphabet of symbols, each feature is quantized into a finite number of levels. A quantization scheme can be unique for each feature, and a number of possible states (symbols) per scheme can also be different. Thus, at the particular moment in time, the system is characterized by a pattern of symbols, where a position in the pattern corresponds to a particular feature while a symbol in this position is an ordinal number of the current feature's value (i.e., $1, 2, \dots, M$).

The proposed approach is designed bearing in mind a distributed implementation deployed on top of a modular distributed industrial automation system. Therefore, it is supposed that each component is linked to the corresponding control node observing and acting based on the component's behavior.

For performance benchmarking purposes a centralized architecture is used where a central processing node, which has its own central control node, is introduced. The centralized approach functions using the same phases as in the conventional machine learning based fault isolation solutions. The training phase could either be performed off-line, using, e.g., facilities for system simulations, or through online learning storing faulty situations in an interaction with a SCADA system and a human operator. It should be mentioned that for the purpose of the fair comparison and demonstration of the feasibility of the proposed approach, all results presented in this study were obtained for the case of the off-line training. We kindly refer readers interested in details of an on-line training of hyperdimensional computing methods to the results in [4] and [26].

During the training phase, the central control node takes the training data set containing a collection of patterns of systemwide states associated with specific faults; then it represents all patterns into HD vectors using the HoloGN encoding (see Section V for details). This forms a knowledge-base of the central control node. In the operational phase, the central control node first collects data from all components' control nodes in order to create the current pattern of the system. Next, the pattern is encoded by HoloGN. Finally, the central control node uses its knowledge-base to find the fault which is the most similar to the current state of the system.

In the distributed version, the isolation of a fault is done in two steps. At the first step, the isolation is done independently by each component's control node in a manner similar to the centralized approach. For this, each control node collects only a partial pattern of system's state from several randomly selected components. When the control node constructs its partial pattern, it first gets the current values (i.e., at the moment of the construction) of the features of its component. Next, it also collects the current states from several neighboring control nodes. Finally, the partial pattern is encoded via HoloGN method as an HD vector. This encoding and the associated reasoning based on the principles of hyperdimensional computing result in autonomous accurate fault isolation. It makes the proposed approach novel and original.

Denote the number of neighboring control nodes by K. Anchoring the behavior of a certain component to the behavior of neighboring control nodes is a known technique, which has biological roots [30] and currently used in the context of intelligent swarms as well as in the bio-inspired analysis of complex systems. The selection of neighboring control nodes could be governed by the information about the interconnection of the components of the system [31]. When such information is limited or unavailable as for the system studied in this article, the random selection of neighboring control nodes is adopted. The number of neighboring control nodes is an adjustable parameter of the approach.

During the training phase in the distributed version of the approach, each control node should acquire its own knowledge-base. In the simplest case control nodes have the same knowledge-base, which is formed using all available features, i.e., in the same way as for the centralized version. Alternatively, each control node can form the knowledgebased using only its own features and features of its neighboring control nodes. The operational phase in a control node is the same as for the centralized version: the control node first forms its current partial pattern. Next, the partial pattern is encoded as an HD vector. Finally, the HD vector and control node's knowledge-base are used to find the most similar fault.

At the second step of the operational phase of the distributed approach, control nodes should reach a consensus using their local predictions of the system-level fault. In the distributed version, the consensus is achieved using a majority voting procedure.

V. DETAILS OF THE HOLOGN

This section introduces the principles of hyperdimensional computing and in a concise form provides the algorithmic steps of the HoloGN method in the context of fault isolation. A detailed description of the method and its theoretical characteristics are described in [3].

A. THEORETICAL PRELIMINARIES

Vector Symbolic Architectures are a class of connectionist models that use high-dimensional vectors to encode structured information as distributed or holographic representation. A distributed representation of data structures is an approach actively used in the area of cognitive computing for representing and reasoning upon semantically bound information [1], [2]. In the distributed representation, all entities are represented by HD vectors. In particular, binary HD vectors are utilized in this article. High-dimensionality refers to the fact that in HD vectors, several thousand positions (of binary numbers) are used for representing a single entity; Kanerva [1] theoretically motivated the use of vectors of 10, 000 binary elements. Such entities have the following useful properties.

1) RANDOMNESS

Randomness means that the values at each position of a HD vector are independent of each other, and 0 and 1 components are equally probable. In very high dimensions, the distances from any arbitrary chosen HD vector to more than 99.99% of all other vectors in the representation

space are concentrated around 0.5 normalized Hamming distance.

2) SIMILARITY METRIC

The similarity between two binary representations (A and B) is characterized by Hamming distance $\delta_H(A, B \text{ normalized})$ to the dimensionality (denoted as d) of the HD vectors. Hamming distance (for two vectors) measures the number of positions in which they differ.

3) GENERATION OF HD VECTORS

Binary HD vectors with the described properties can be obtained from one such vector via the cyclic shift operation [32]. Using this operation, a sequence of R vectors, which are dissimilar to a given initial random HD vector A (i.e., the normalized Hamming distance between them equals approximately 0.5), can be generated by cyclically shifting A by i positions, where $1 \le i \le R < d$. The operation is denoted as Sh(A, i).

4) BUNDLING OF HD VECTORS

Joining several entities into one structure is done with the bundling operation. Bundling is implemented via a majority sum of the HD vectors representing the entries. A bitwise majority sum of *n* vectors issues 0 when n/2 or more operands are 0, and 1 otherwise. In the case of an even number in sum, ties are broken at random, which is equivalent to adding an extra random HD vector. The operation is denoted as [A + B + C]. The majority sum operation possesses the following properties:

- For any number of operands, the result is an HD vector, with the number of 1 components being approximately equal to the number of 0 components.
- All components included in the sum are similar to the result;
- The more HD vectors that are involved in a majority operation, the closer the normalized Hamming distance between the resultant HD vector and any HD vector component is to 0.5.

B. HOLOGRAPHIC GRAPH NEURON

HoloGN [3] is an approach for pattern recognition and matching. It is based on a one-shot learning associative memory. The approach is an abstract model memorizing patterns of heterogeneous sensory data for later similarity analysis. In the context of this article HoloGN encodes patterns of system states.

Then the HD index for the current state of the feature *i* is derived as: $E_{(i,j)}^{HD} = \text{Sh}(IV_i, j)$. Let *N* be the number of features in the system. When

Let *N* be the number of features in the system. When HoloGN is exposed to a pattern characterizing the system, the distributed representation of the pattern is formed as: $P^{HD} = [\sum_{i=1}^{N} E^{HD}_{(i,j)}]$, where $E^{HD}_{(i,j)}$ is the HD index of the current state (with ordinal number *j*) of feature *i*; the square brackets [*] denote the majority sum operation which is used on the HD indices.

VI. PERFORMANCE BENCHMARKING WITH CONVENTIONAL MACHINE LEARNING METHODS

A. FORMATION OF THE KNOWLEDGE-BASE BY ENCODING (PARTIAL) SYSTEM'S STATE WITH HOLOGN

In the training phase HoloGN encodes the training data set in order to create the knowledge-base, which is used for the fault isolation, inside a control node. First, referring to Fig. 1, each feature i is assigned a unique initialization HD vector IV_i . There are different possible configurations of the knowledge-base; in this article a matrix D^{HD} is formed. The matrix has d columns and s rows, where s is the number of entries (i.e., patterns of system-wide states) in the training data set. Each row of the matrix is associated with the particular fault. The matrix is populated with the distributed representations of the patterns from the training data set, which are formed by HoloGN as defined above using IV_i .

B. FAULT ISOLATION IN THE PROPOSED APPROACH

The process of fault isolation is based on the mathematics the random HD vectors, which are used to encode the patterns of the system-wide state, and the similarity preserving property of the bundling operation (see [33] for a theoretical analysis of the bundling operation).

The fault isolation is, therefore, performed by estimating the similarity of the distributed representation of the currently observed pattern $P_{current}^{HD}$ to all possible faults using the knowledge-base. The similarity to fault *x* is calculated as the average Hamming distance Δ_x between $P_{current}^{HD}$ and all *g* entries in the knowledge-based corresponding to fault $x: \Delta_x = \sum_{i=1}^g \delta_H(P_{current}^{HD}, D_{i_x}^{HD})/g$, where $D_{i_x}^{HD}$ is the distributed representation of a pattern corresponding to fault *x* in the knowledge-base. The closer Δ_x to 0.5 for the fault *x* the smaller is the likelihood that the current faulty situation was caused by fault *x*.

The purpose of the case study is to demonstrate the feasibility of the proposed fault isolation approach and to compare it with other state-of-the-art data-driven methods. The case study is performed using an accurate generic nuclear power plant model (Fig. 3) provided by the industrial partner Fortum Power and Heat, a power utility with nuclear power plant operation license in Finland. The model is run using the Apros 6 process simulator. Apros 6 is a dynamic process simulator owned by the VTT Technical Research Centre of Finland and Fortum. The power is generated using the nuclear energy. The main process includes two loops: the primary and the secondary circuits.

The primary circuit (Fig. 3) contains the reactor vessel (the central part in Fig. 3) and the nuclear fission within the fuel that generates thermal energy. The generated energy heats the water in the vessel. The coolant pumps (the central part in Fig. 3) in the primary circuit circulate water through the steam generators (the left and the right parts in Fig. 3) and the reactor vessel. In this way, thermal energy is transferred from the primary to the secondary circuit. The pressurizer (the upper part in Fig. 3) is also a part of the primary circuit.



FIGURE 3. Generic nuclear power plant model (for the primary circuit) in the industrial grade simulation environment.

It is a vessel partially filled with water. Its main purpose is to regulate the pressure via heaters and water-sprays.

The secondary circuit (not depicted in Fig. 3) is connected to the primary one through steam generators. The heat from the primary circuit converts the water, flowing into the secondary parts of steam generators, into steam. Turbines use this high-pressure steam flow to drive electric generators. Condensers are placed next to turbines to convert the lowpressure steam back to water.

C. DATA SOURCES AND PREPROCESSING

116 automation components as the potential sources of hardware faults in the nuclear power plant model were analyzed with the functional failure identification and propagation framework [34]. Valve actuators and pumps were the most common types among these components. Three failure modes were chosen for each type of the automation component. For example, a valve actuator can be set to "failed open", "failed closed" or "no electric supply" failure modes which will respectively result in opening, closing or stopping to control the valve. In the context of this case study, a component failure mode pair (e.g., "Valve IDvalve" "failed open") defines a fault, which should be identified during the fault isolation process. Out of 348 potential faults (116 by 3), only 92 faults have actually changed the steady-state operation mode of the model. Therefore, these faults could be detected by datadriven fault isolation approaches.

During the data collection the model was executed for eleven power production levels in the range 90% to 100% in order to acquire a larger data set for faulty operation conditions. The 92 faults, which are detectable in the steadystate of the model, were simulated for each power level. Thus, the total number of simulations is 1012 (92 faults by 11 levels). Training and testing data sets used for the performance evaluation of different approaches were created from the results of simulations. The simulations were split so that data for six power production levels (54.5% of the dataset) was used for training while the data for other five power production levels (45.5% of the dataset) was used for testing. The separation was kept fixed for all experiments reported below. Each simulation includes signals traces logs for 180 seconds since the moment when a fault was introduced in the model. The model simulates monitoring signals for 37 sensors deployed all over the power plant. Because all methods considered in this article require a fixed set of features as an input, three statistical features (mean, max, and min) were extracted from each sensory signal leading to 111 features per simulation (37 signals by 3 features). Thus, the data set contains 1012 entries. Each entry consists of 111 features (extracted from 37 recorded signals). All entries are labeled with the corresponding fault (componentfailure mode pair).

All features in the data set were normalized because not all methods can handle non-normalized data equally well. The normalization range [-1,1] was used. Additionally, as HoloGN requires a discrete range of data, all features were quantized in even intervals (only for HoloGN) using a single hyperparameter - the number of quantization levels (M_i) . Given M_i , the quantization of a feature i normalized in the range [-1,1] is done as follows: $i_q =$ $\lfloor iM_i/2 \rfloor$, where i_a denotes the quantized feature i and $\lfloor * \rfloor$ denotes the nearest integer function. The optimal number of quantization levels was estimated using the training data. No intelligent analytics (e.g., fuzzy) were involved in the quantization process. Note that due to restrictions imposed by Facility owner, the data set is not available for an unrestricted access, but interested readers can contact authors in order to access it.

D. REFERENCE METHODS AND TOOLS FOR BENCHMARKING

The performance of the centralized and distributed HoloGN approaches was compared to the performance of several conventional classification methods: a multilayer perceptron ANN, a decision tree, a random forest, a Sequential minimal optimization (SMO), which is a fast algorithm for training Support Vector Machines, and a k-nearest-neighbors (kNN). The ANN and SMO were chosen as well-known machine learning algorithms, the decision tree as an algorithm which produces human-readable models and has shown in previous research to be better in fault isolation on the case study of this article [34]. The Random forest is an ensemble method using decision trees. The kNN was chosen because it is also a metric based method as the proposed approach. The performance of methods was assessed using MATLAB¹ and WEKA² tools. During the assessment ANN, decision tree, random forest, and SMO implemented in WEKA were used, while kNN and the proposed HoloGN methods are available from MATLAB. MATLAB implementation of the centralized HoloGN is available online (see [3]). For a given input, a single prediction is produced by a decision tree whilst ranked lists of possible alternatives are issues by the HoloGN and kNN. Faults in the ranked list are sorted in the ascending order according to their distances resembling the similarity to the provided input features. ANN and SMO could also provide a ranked list, but their implementations in WEKA are only able to issue the prediction with the highest score. Hence, the performance for ANN and SMO is presented only for the best prediction.

The power plant scenario was used to evaluate the performance of the methods in the fault isolation task. The performance was measured in terms of accuracy on the testing data set. The accuracy in turn was calculated as the proportion of the correctly isolated faults.

E. CALIBRATING THE COMPARED METHODS FOR THE FAIR BENCHMARKING

In order to get the optimized performance for each method, the corresponding hyperparameters were set with the help of 6-fold cross-validation on the training data. The number of fold for cross-validation matches the nature of the training data as it included simulations for six power production levels.

The applied ANN was feed-forward with three layers: input, output and a hidden layer in between. During the training the back propagation algorithm was used. The decision tree was trained using the J48 algorithm. The SMO method implements John Platt's Sequential minimal optimization algorithm for training a support vector classifier. The parameter k of the kNN providing the best performance during the cross-validation on the training data equals to 7. Because HoloGN requires finite alphabet of feature values,

¹https://mathworks.com.

²https://www.cs.waikato.ac.nz/ml/weka/.

the original values of the data set were quantized into finite number of levels that are uniformly distributed within the range of feature values. It was experimentally found that the approach demonstrated its best performance when each feature was quantized with $M_i = 60$. During the evaluation the proposed approach used HD vectors with 10, 000 elements. An initialization HD vector for each feature was generated at random.

F. AN EXAMPLE OF FAULT ISOLATION BY HOLOGN

Consider the situation when one of the valves entered the "failed closed" fault, i.e., the valve is closed at a time when it should be opened. The fault affects signals measured by 37 sensors. After 180 seconds since the fault start, the system extracts 3 features from each signal (111 in total). Table 1 illustrates normalized mean values for three different sensors.

TABLE 1. Normalized mean values for three different sensors when one of valves is in "failed closed" condition.

Sensor	Value
Pressurizer pressure	-0.058
Primary hot leg (right) temperature	0.860
Steam blow-out of steam generators liq mass flow	0.531

All available features are quantized into 60 levels. HoloGN uses all 111 quantized features to represent the system's state as 10, 000 dimensional HD vector. Next, HoloGN measures normalized Hamming distances between the formed HD vector and all available fault situations in the knowledge-base. The similarity score for each of 92 faults is the mean normalized Hamming distance for all entries of the fault in the knowledge-base. Finally, HoloGN issues list of faults sorted in the ascending order according to their similarity scores to the current state of the system. In the considered example the fault was isolated correctly. It is worth mentioning, that the second and the third closest faults also belong to the type "failed closed", but point to other valves.

The centralized mode is used to train the conventional machine learning methods. It also assumes that the structure of the evaluation data set assessing the performance of the fault isolation should have the same structure, i.e., the testing is also centralized. Because it is assumed that the interdependencies between different components of the case study model are not known, all methods were trained with the values of features extracted from all 37 model's components (sensors). From the implementation point of view, this corresponds to the case when all components communicate their values of features to a central processing node, which performs fault isolation.

G. PERFORMANCE ASSESSMENT: THE CENTRALIZED CASE

Fig. 4 presents the comparison between the benchmarked methods. The comparison shows that the centralized HoloGN performs well against the other methods even when considering only the best prediction (marked as 1st result





in Fig. 4). In particular, the accuracy of fault isolation achieved by SMO is approximately 0.6; ANN with one hidden layer achieved 0.54 while ANN without hidden layers (not depicted) achieved 0.60. The decision tree has demonstrated the performance close to 0.75 while random forest achieved only 0.62. kNN and the centralized HoloGN have shown results somewhat lower than the decision tree: 0.66 and 0.71 respectively. However, if top three predictions are considered then the accuracy of both the kNN and the centralized HoloGN are higher than 0.9.

Alternatively, the change in performance when considering several predictions can be seen using the Precision. In the case of top three predictions, if the correct fault type is among the predictions then the isolation is correct. For kNN when going from the best prediction to top three prediction the Precision has increased from 0.654 to 0.940 while for HoloGN the change was very similar 0.684 to 0.966. Thus, the results of comparison show that the proposed centralized approach demonstrates the performance of a par with other machine learning methods. Next, the centralized approach and its distributed version are compared.

H. PERFORMANCE ASSESSMENT: THE DISTRIBUTED CASE

In order to demonstrate the performance of the distributed HoloGN its accuracy is compared against the centralized version. The distributed version was considered to be deployed on each component of the studied model, thus, forming 37 predictors in total. Each predictor issues top three faults. Fig. 5 and 6 depict the accuracy for each result rank and their sum (denoted as 1+2+3). The majority voting was applied to predicted faults in order to get the final prediction. In order to justify the performance of the proposed approach, 50 independent simulations with different randomization seeds were performed. Each simulation randomly selected the neighboring control nodes for each component as well as randomly generated initialization HD vectors. The results presented in Fig. 5 and 6 depict the mean and standard deviation values. Fig. 5 demonstrates the results obtained by calculating the



FIGURE 5. Accuracy of the distributed HoloGN. The case of fault isolation using all features to create the knowledge-base.

accuracy of the fault isolation when in the operational phase each predictor is getting data from K components (i.e., K by 3 features), but the knowledge-base is created for all features. The list of components connected to each predictor was generated at random. The second assessment shown in Fig. 6 demonstrates the accuracy observed in the case when each predictor had its own knowledge-based created from features corresponding to the connected components.

Both bar graphs show the estimated accuracy and the corresponding standard deviations for different number of connected sensors included in each predictor as well as the performance for the centralized case (Fig. 4). For both scenarios the accuracy of the distributed HoloGN degrades compared to the centralized version with the number of connected components, but it is approaching 0.9 even with three sensors. We attribute the performance degradation to the fact that the distributed HoloGN has to operate in the absence of the full available information, i.e., in comparison to the centralized version, fault isolation is initially made by individual components using incomplete patterns. Secondly, the accuracy to a certain extend depends on the number of components connected to each predictor. For example, with nine connected sensors the total accuracy is 4.4% (Fig. 5) and 3.4% (Fig. 6) lower than for the centralized approach while with three connected sensors the difference is 8.7% (Fig. 5) and 7.5% (Fig. 6). However, beyond a certain point, the accuracy improves marginally with the increase in the number of connections. The study of the optimal number of connections is a subject for further investigation.

I. PERFORMANCE ASSESSMENT: COMPUTING TIME

While the main performance indicator in the scope of this article is the classification accuracy it is worth discussing the computational complexity of the proposed method. First, it should be noted that the fair comparison of the exact computing time would require implementations of all classifiers on the same platform using the same programming environment. Unfortunately, such implementations are



FIGURE 6. Accuracy of the distributed HoloGN. The case of fault isolation using individual knowledge-base for each predictor.

currently not available. However, qualitatively the complexity of the method in the operational phase linearly depends on the number of features, the dimensionality of HD vectors, and the size of the knowledge-base. Thus, it is comparable to the kNN except for the additional cost of mapping features to a high-dimensional space. On the other hand, once mapped operations are binary. Also, note that the dimensionality used in the experiments is 10, 000. Therefore, given the efficient implementation, one HD vector would require only 1250 bytes. For the kNN 111 real-valued features would require 888 bytes (assuming 8 bytes per number). Thus, the difference is not significant. An additional advantage of HD vectors is that the required operations (cyclic shift, Hamming distance, and majority sum) can be efficiently executed in one clock cycle (see [35]).

While hardware integration aspects are outside the main scope for this article, the computational energy-efficiency and computational complexity of hyperdimensional computing will be performed taking into account specific methods of hardware realization. The fundamental difference of hyperdimensional computing from the traditional computer architectures is the tight integration of logic with the memory during the processing. Therefore all its operations (binding, bundling, and permutation see Section V for details) can be implemented by manipulations and comparison of large patterns within the memory. The method for hardware integration of HD operations described in [36] and [37] is based on in-memory processing in 3D Resistive RAM, avoiding shuttling data back and forth between the memory and the processor as in conventional architectures. The proposed hardware implementation exhibits substantial energy savings due to accessing global interconnects at a relatively low frequency.

VII. CONCLUSION

This article presented the application of hyperdimensional computing for the problem of data-driven distributed

VOLUME 6, 2018

fault isolation. The application was demonstrated on the generic model of a nuclear power plant using the sensory measurements for training and assessing the method. The results of the use-case study show the accuracy of fault isolation comparable to the conventional machine learning methods applied in the same context.

It is worth mentioning two aspects, which could affect the results reported in this study. First, this study used the fixed setup for extracting the features. The search of optimal features for the considered problem was not the main focus. However, the exploration of effects of additional features and facilitating techniques such as dimensionality reduction and feature selection is a promising direction for the future work. Second, during the benchmarking several conventional methods were used but there are many other machine learning methods which were not assessed during the benchmarking. In particular, as the data in the studied application are timeseries a promising direction for the future work is to study the performance of methods for processing temporal data (e.g., Recurrent Neural Networks).

The advantages of the hyperdimensional computing-based solution are in its distributed operation and its potential for on-line learning. On-line learning would allow constantly keeping the updated model of a system. On the other hand, the distribution allows the fault isolation subsystem to be an integral part of the distributed automation aspect of the system. Moreover, it potentially enables the deployment of such subsystem in standard programmable logic controllers with predictable dependability characteristics which can be checked and proven using a variety of verification tools, including closed-loop simulation [38] and model-checking [39]. Alternatively, the subsystem can be implemented as a multiagent system.

From the practical point, the proposed approach can be used as an additional mechanism on top of the existing fault management system improving the robustness and the decision making for the whole system.

REFERENCES

- P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cogn. Comput.*, vol. 1, no. 2, pp. 139–159, Oct. 2009.
- [2] S. I. Gallant and T. W. Okaywe, "Representing objects, relations, and sequences," *Neural Comput.*, vol. 25, no. 8, pp. 2038–2078, 2013.
- [3] D. Kleyko, E. Osipov, A. Senior, A. I. Khan, and Y. A. Sekercioglu, "Holographic graph neuron: A bioinspired architecture for pattern processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 6, pp. 1250–1262, Jun. 2017.
- [4] D. Kleyko, A. Rahimi, D. Rachkovskij, E. Osipov, and J. Rabaey, "Classification and recall with binary hyperdimensional computing: Tradeoffs in choice of density and mapping characteristics," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.
- [5] International Electrotechnical Standard iec61499-1, Function Blocks— Part 1 Architecture, document 2005-01, Geneva, Switzerland, I. E. Commission, 2005.
- [6] D. Kleyko, E. Osipov, N. Papakonstantinou, V. Vyatkin, and A. Mousavi, "Fault detection in the hyperspace: Towards intelligent automation systems," in *Proc. IEEE Int. Conf. Ind. Informat. (INDIN)*, Jul. 2015, pp. 1219–1224.
- [7] F. J. Alexander, "Machine learning [guest editor's introduction]," Comput. Sci. Eng., vol. 15, no. 5, pp. 9–11, Sep./Oct. 2013.

- [8] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part I: Quantitative model-based methods," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 293–311, 2003.
- [9] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part II: Qualitative models and search strategies," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 313–326, 2003.
- [10] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis: Part III: Process history based methods," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 327–346, 2003.
- [11] R. Isermann, "Model-based fault-detection and diagnosis-status and applications," Annu. Rev. Control, vol. 29, no. 1, pp. 71–85, 2005.
- [12] V. Uraikul, C. W. Chan, and P. Tontiwachwuthikul, "Artificial intelligence for monitoring and supervisory control of process systems," *Eng. Appl. Artif. Intell.*, vol. 20, no. 2, pp. 115–131, 2007.
- [13] Z. Kurd, T. Kelly, and J. Austin, "Developing artificial neural networks for safety critical systems," *Neural Comput. Appl.*, vol. 16, pp. 11–19, Jan. 2007.
- [14] S. J. Lee and P. H. Seong, "A dynamic neural network based accident diagnosis advisory system for nuclear power plants," *Prog. Nucl. Energy*, vol. 46, nos. 3–4, pp. 268–281, 2005.
- [15] M. J. Embrechts and S. Benedek, "Hybrid identification of nuclear power plant transients with artificial neural networks," *IEEE Trans. Ind. Electron.*, vol. 51, no. 3, pp. 686–693, Jun. 2004.
- [16] T. Amraee and S. Ranjbar, "Transient instability prediction using decision tree technique," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3028–3037, Aug. 2013.
- [17] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2226–2238, Nov. 2013.
- [18] Y. Zhang, H. Zhou, S. J. Qin, and T. Chai, "Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares," *IEEE Trans. Ind. Informat.*, vol. 6, no. 1, pp. 3–10, Feb. 2010.
- [19] J. Neuzil, O. Kreibich, and R. Smid, "A distributed fault detection system based on IWSN for machine condition monitoring," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1118–1123, May 2014.
- [20] R. Gayler, "Vector symbolic architectures answer Jackendoff's challenges for cognitive neuroscience," in *Proc. Joint Int. Conf. Cogn. Sci. ICCS/ASCS*, 2003, pp. 133–138.
- [21] D. Rasmussen and E. Eliasmith, "A neural model of rule generation in inductive reasoning," *Topics Cogn. Sci.*, vol. 3, no. 1, pp. 140–153, 2011.
- [22] B. Emruli, R. W. Gayler, and F. Sandin, "Analogical mapping and inference with binary spatter codes and sparse distributed memory," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, 2013, pp. 1–8.
- [23] O. Rasanen and S. Kakouros, "Modeling dependencies in multiple parallel data streams with hyperdimensional computing," *IEEE Signal Process. Lett.*, vol. 21, no. 7, pp. 899–903, Jul. 2014.
- [24] O. J. Räsänen and J. P. Saarinen, "Sequence prediction with sparse distributed hyperdimensional coding applied to the analysis of mobile phone use patterns," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 9, pp. 1878–1889, Sep. 2016.
- [25] G. Recchia, M. Sahlgren, P. Kanerva, and M. Jones, "Encoding sequential information in semantic space models: Comparing holographic reduced representation and random permutation," *Comput. Intell. Neurosci.*, vol. 2015, 2015, Art. no. 58.
- [26] A. Rahimi, S. Benatti, P. Kanerva, L. Benini, and J. M. Rabaey, "Hyperdimensional biosignal processing: A case study for EMG-based hand gesture recognition," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, Oct. 2016, pp. 1–8.
- [27] A. Rahimi, A. Tchouprina, P. Kanerva, J. Millán, and J. Rabaey, "Hyperdimensional computing for blind and one-shot classification of eeg error-related potentials," *Mobile Netw. Appl.*, pp. 1–12, Oct. 2017. [Online]. Available: https://link.springer.com/article/10.1007/s11036-017-0942-6#citeas
- [28] D. Kleyko, S. Khan, E. Osipov, and S. P. Yong, "Modality classification of medical images with distributed representations based on cellular automata reservoir computing," in *Proc. IEEE Int. Symp. Biomed. Imag.*, Apr. 2017, pp. 1053–1056.
- [29] O. Rasanen, "Generating hyperdimensional distributed representations from continuous valued multivariate sensory input," in *Proc. 37th Annu. Meeting Cogn. Sci. Soc.*, 2015, pp. 1943–1948.

- [30] W. Bialek, A. Cavagna, I. Giardina, T. Mora, E. Silvestri, M. Viale, and A. M. Walczak, "Statistical mechanics for natural flocks of birds," *Proc. Nat. Acad. Sci.*, vol. 109, no. 13, pp. 4786–4791, 2012.
- [31] R. Oliva, "Model structure analysis through graph theory: Partition heuristics and feedback structure decomposition," *Syst. Dyn. Rev.*, vol. 20, no. 4, pp. 313–336, 2004.
- [32] D. Kleyko and E. Osipov, "On bidirectional transitions between localist and distributed representations: The case of common substrings search using vector symbolic architecture," *Procedia Comput. Sci.*, vol. 41, pp. 104–113, Dec. 2014.
- [33] E. P. Frady, D. Kleyko, and F. T. Sommer, "A theory of sequence indexing and working memory in recurrent neural networks," *Neural Comput.*, vol. 30, no. 6, pp. 1449–1513, Jun. 2018.
- [34] N. Papakonstantinou, S. Proper, B. O'Halloran, and I. Y. Tumer, "Simulation based machine learning for fault detection in complex systems using the functional failure identification and propagation framework," in *Proc. ASME CIE*, Buffalo NY, USA, 2014, pp. 1–10.
- [35] I. S. Haque, V. S. Pande, and W. P. Walters, "Anatomy of high-performance 2D similarity calculations," *J. Chem. Inf. Model.*, vol. 51, no. 9, pp. 2345–2351, 2011.
- [36] H. Li *et al.*, "Hyperdimensional computing with 3D VRRAM in-memory kernels: Device-architecture co-design for energy-efficient, error-resilient language recognition," in *IEDM Tech. Dig.*, Dec. 2016, pp. 1–4.
- [37] A. Rahimi et al., "High-dimensional computing as a nanoscalable paradigm," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 64, no. 9, pp. 2508–2521, Sep. 2017.
- [38] C.-H. Yang, V. Vyatkin, and C. Pang, "Model-driven development of Control Software for Distributed Automation: A Survey and an Approach," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 3, pp. 292–305, Mar. 2014.
- [39] C. Suender, V. Vyatkin, and A. Zoitl, "Formal verification of downtimeless system evolution in embedded automation controllers," ACM Trans. Embedded Comput. Syst., vol. 12, no. 1, 2013, Art. no. 17.



DENIS KLEYKO received the bachelor's degree (Hons.) in telecommunication systems and the master's degree (Hons.) in information systems from the Siberian State University of Telecommunications and Information Sciences, Novosibirsk, Russian, in 2011 and 2013, respectively. He is currently pursuing the Ph.D. degree with the Dependable Communication and Computation Systems Group, Department of Computer Science, Electrical and Space Engineering, Luleå University of

Technology, Luleå, Sweden. His current research interests include vector symbolic architectures, hyperdimensional computing, bio-inspired cognitive architectures, and machine learning.



EVGENY OSIPOV received the M.Sc. degree from EPFL, Swiss Federal Institute of Technology, in 1999, the Licentiate of Engineering degree from KTH Royal Institute of Technology, Sweden, in 2003, and the Ph.D. degree in computer science from the University of Basel, Switzerland, in 2005. He is currently a Full Professor in dependable communication and computation system with the Department of Computer Science and Electrical Engineering, Luleå University of Technology. His

major research interests are in application of cognitive computing and communication architectures to low-power embedded systems in the context of future cyber-physical systems, Internet-of-Things, and intelligent industries.



NIKOLAOS PAPAKONSTANTINOU received the Diploma degree in electrical & computer engineering from the University of Patras, Greece, and the Ph.D. degree in information technology in automation from Aalto University, Finland. He is currently a Senior Scientist with the VTT Technical Research Centre of Finland in the area of system modeling and simulations. He leads the Model-Driven Engineering and Industrial Information substance node which includes about

10 researchers. He leads the DECHEMA/OPC Foundation joint working group for the DEXPI OPC UA companion specification. He focuses on simulation and model driven approaches to system design, operation, and safety. His current interests also include and security of complex systems and industrial applications of data mining and machine learning.



VALERIY VYATKIN (M'03–SM'04) received the Ph.D. degree from the State University of Radio Engineering, Taganrog, Russia, in 1992. He was a Visiting Scholar with Cambridge University, Cambridge, U.K., and had permanent academic appointments with the University of Auckland, Auckland, New Zealand, the Martin Luther University of Halle-Wittenberg, Halle, Germany, and in Japan and Russia. He is on joint appointment as a Chaired Professor of dependable computa-

tion and communication systems with the Luleå University of Technology, Luleå, Sweden, and a Professor of information and computer engineering in automation with Aalto University, Helsinki, Finland. His current research interests include dependable distributed automation and industrial informatics, software engineering for industrial automation systems, and distributed architectures and multiagent systems applied in various industry sectors, including smart grid, material handling, building management systems, and reconfigurable manufacturing.

Dr. Vyatkin was a recipient of the Andrew P. Sage Award for the best IEEE Transactions paper in 2012.