Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2022) 000–000

www.elsevier.com/locate/procedia

16th CIRP Conference on Intelligent Computation in Manufacturing Engineering, CIRP ICME '22, Italy

A Concept for Dynamic and Robust Machine Learning with Contex Modeling for Heterogeneous Manufacturing Data

Simon Kamm[a]*, Nada Sahlab[a], Nasser Jazdi[a], Michael Weyrich[a]

[a]*Institute of Industrial Automation and Software Engineering, University of Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart, Germany*

* Corresponding author. Tel.: +49 711 685 67293; *E-mail address:* simon.kamm@ias.uni-stuttgart.de

Abstract

With the increasing amount of available and connected data sources, industrial automation applications such as condition monitoring of a production machine can be improved by considering various data. To gain insights from this data and make it useable, heterogeneous data has to be analyzed intensively. Limited machine learning approaches exist in industrial automation and manufacturing for analyzing data acquired from multiple sources. In this paper, first, a suitable concept for handling heterogeneous data from integration to analysis is presented as well as a multi-layer architecture for the concept's realization. The architecture encapsulates functionalities into the different layers and allows easy extendability and modifiability. Afterwards, a context modeling approach for managing heterogeneous data and existing approaches and algorithms for analyzing this data robustly and dynamically analyzing it are presented.

Peer-review under responsibility of the scientific committee of the 16th CIRP Conference on Intelligent Computation in Manufacturing Engineering.

Keywords: Machine Learning, Dynamic and Robust Machine Learning, Heterogeneous Data, Context Modeling

1. Introduction

With the potential of improving control and management of a system by acquiring more data reflecting its state as well as its environment and therefore generating a more holistic and virtual representation of it, adding more data sources e.g., sensors is necessary. In industrial automation and manufacturing, this can improve and enhance approaches, such as condition monitoring, anomaly detection, or failure analysis.

Data-driven approaches, especially machine learning (ML) algorithms gained attention within the last years for analyzing data and gaining knowledge about a system's state (e.g. predictive maintenance or condition monitoring) [1]. However, when it comes to heterogeneous data from multiple sources and of different types, data-driven approaches are often limited, since they typically focus on a specific type of data with a high volume (e.g. images (computer vision) [2, 3] or time-series data [4, 5]). Therefore, these approaches do not take advantage of the multimodality provided by heterogeneous data which enables different views of a system, since they are not designed to intelligently combine heterogeneous data sources [6].

To analyze heterogeneous data, it first has to be integrated, preprocessed, modeled, and managed suitably, allowing central data access with clearly defined interfaces. Different approaches exist, which aim at modeling and handling the complex relationships coming with the increased amount of data sources. One research direction is context modeling, where a semantic metamodel integrates a system's model with its operational environmental representation. Context is hereby defined as aggregated and time-continuous data outside the system's boundary, using which, a better understanding of the system's operation is possible [7]. To enable this understanding, a system's representation in terms of models and data is necessary, which is referred to as the internal context, which is then mapped to the external environmental context. Some of the characteristics of context modeling are the heterogeneity of context, as it is multimodal and dynamic, using a knowledge-based approach; a context model aggregates and unifies heterogeneous data to enable inference of complex

2212-8271 © 2022 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 16th CIRP Conference on Intelligent Computation in Manufacturing Engineering.

context, i.e. influencing contextual factors during the system's operation. Having a context model in combination with data-driven learning approaches enables efficient learning and an improved understanding of the learning results.

Data-driven learning approaches have challenges and limitations. When having, e.g., multiple views (sensors) on a system, the sensor configuration of an industrial automation system will likely vary during its lifetime. New sensors are installed or existing sensors are removed, modified, or fail [8]. This challenges a pre-trained data-driven model since they are usually fixed after training. To tackle this, the ML model needs to be adapted and made *robust* and *dynamic*. These terms will be introduced and defined in the later sections in detail.

The *objective of the paper* is to show a concept for analyzing heterogeneous data robustly and dynamically. Context modeling is used for integrating and handling the data first. A holistic architecture will be introduced, which handles the heterogeneous data from their sources to the final analysis and interface to the user.

The remaining paper is structured as follows: First, the basics and related work for the integration, modeling, and analysis of heterogeneous data in industrial automation and manufacturing are introduced. Based on that, the open challenges and requirements are discussed. Thereupon, an architecture is proposed which enables dynamic and robust machine learning for heterogeneous data. Subsequently, concrete approaches for realizing the introduced components of the architecture are presented and discussed. Finally, a conclusion and outlook on further work are given.

2. Basics and Related Work

2.1. Integrating and modeling heterogeneous data

To analyze data, first the integration, modeling, and thus provisioning of the heterogeneous data needs to be realized. Following [10], data processing for heterogeneous data can be divided into three steps: Data cleaning, data integration, and dimension reduction (including data normalization).

Data cleaning aims at improving the data quality by identifying incomplete, inaccurate, or unreasonable data and modifying or deleting such data [11]. Bad data quality can directly lead to inaccurate data analytics. Some of the most common and important techniques are outlier detection, data transformation, error repair (including the imputation of missing values), and data deduplication [12].

Clean data should then be integrated into a common dataset, stored in one (or multiple) database(s), and accessible for further processing and analytics. Multiple industrial data integration tools exist. One part of them is the so-called ETL (Extract, Transform, Load) tools. As the name states, these tools extract data from a source, transform it to the desired format, and load it to the defined target. In academia, ontology-based data integration (OBDI) gains increased interest, since it is theoretically able to handle semantic heterogeneity by resolving this heterogeneity into a uniform, formal, and semantic data description and managing the heterogeneous data while preserving the existing relations [6]. In [13], ontologies were developed for the manufacturing process of semiconductors to use the generated data for decision making. Ford investigated the potential of federated, distributed ontologies for the integration of big data from production [14].

Dimensionality Reduction and data normalization are crucial parts for heterogeneous data analysis. On the one side, dimensionality reduction is relevant, when multiple data sources are available and high dimensionality of the data is given. Uniform normalization is necessary to ensure common treatment of the data sources and not artificially highlight a data source due to its bigger data range.

Context-aware systems are systems that consider context, either in the form of a model or with context-aware computing methods to increase the system's sensitivity and adaptability to contextual changes. Depending on the domain, context has different definitions and realizations. [15] defines context as "any information that can be used to characterize the situation of an entity" during its interaction with a software application. For cyber-physical automation systems, context is defined as dynamic data representing flexibly connected entities around the system, sharing either a goal or an environment and contributing to an understanding of its operation [16]. Having a relational property as [17] describes, focusing on relations between different context information and how they relate to the system's operation is essential and can be represented within a context model.

Modeling context is predominantly realized semantically via ontologies or graphically via networks. An important aspect the context model should address is the dynamic evolvement of the model at runtime to reflect the dynamicity of the system and its environment as well as enable the addition of new data sources to model further context information. A further aspect to consider is the unified modeling of heterogeneous data within the context model, which necessitates a pre-processing and aggregation of acquired data and mapping approaches to the pre-defined schema or ontology. In [16], a structured data and metamodeling approach was used to acquire data and map it to a context graph. Another important aspect is the update of the context model, which based on the system can be time-based on event-based. To achieve usability of the context model and compensate for the invested modeling effort, an interface allowing for scalable and multiple access to the context model via applications and services is necessary.

2.2. Machine Learning for heterogeneous data

In the last years, ML and especially Deep Learning (DL) was used increasingly in manufacturing and industrial automation. Due to its capability of learning complex correlations between input and outputs from recorded, historical data, these algorithms are more and more used for specific tasks, such as object detection or failure analysis. Traditionally, these algorithms consume one kind of data, e.g. images or time series. In [18], time-series data is analyzed by a convolutional neural network to determine the failure class of an electronic device. [19] applied LSTM networks in discrete manufacturing for anomaly detection in a metal forming process. Since recent deep learning approaches are typically first implemented and evaluated on image data, multiple applications of deep learning for computer vision tasks are realized in manufacturing and industrial automation. [20], e.g., introduced a deep learning-based classification pipeline using a convolutional neural network for defect detection of solar cells based on electroluminescence images.

However, in nowadays systems, more data sources exist which acquire different views of the monitored system or component. These different views (multi-view) often complement each other and thus give additional information. The concept of multi-view data is defined as data collected in parallel using different methods or techniques [21]. The multiple views should be analyzed in a combined fashion to gain the best result. Current research in this direction is linked with the keywords sensor/data fusion and multi-view/multi-modal neural networks. These approaches are currently mainly the focus of the automotive domain, where different information, coming from diverse sensors (Lidar, Radar, Camera, Ego-data, etc.) should be fused to enable autonomous driving. In [22], lidar depth information and 2D image data are combined into an RGBD image, which is then processed by a convolutional network for image segmentation and object detection. Another approach [23] provided a fusion method for vehicle detection, which combines laser information, visual data, and GPS data.

In the manufacturing and industrial automation domain, just a limited amount of work was found, that analyzes heterogeneous data. [24] combined motor vibration and stator signals with current signals in a multi-view network architecture to fuse this information for motor fault diagnosis. In [25], an unsupervised machine learning approach is presented for the active perception in autonomous robots based on multimodal fusion. This approach was validated for a multimodal human activity sensor dataset. A multi-head CNN-RNN architecture was proposed by [8] for time series anomaly detection. First, the multi-head CNN is used to extract features of each sensor data independently to deal with heterogeneous data. Afterward, the time series are windowed and analyzed jointly by an RNN to detect an anomaly.

These approaches show that sensor/data fusion and multi-modal/multi-view neural networks are of interest for the manufacturing and industrial automation domain to handle the increasing variety and thus heterogeneity of the occurring data. However, limited research was so far conducted regarding a holistic approach for integrating, handling, and analyzing heterogeneous data in this domain.

3. Concept for handling and analyzing heterogeneous data

Different challenges arise in manufacturing and industrial automation when heterogeneous data is handled and analyzed.

Since data arise from different data sources, flexible data interfaces for data integration are required. Once heterogeneous data is integrated, the (often complex) relations within the data need to be considered and modeled, whereas classical and often used relational database systems are limited [26]. Data coming from different sources can have different meanings and interpretations of the same values and labels, which require a uniform formal and semantic data description to resolve this heterogeneity [27]. This allows the further usage of the data via a uniform interface. This gives the first challenge

C1: *Big semantic heterogeneity at the integration, description, and storage of the heterogeneous data*.

Once data is integrated, described, and modeled, it shall be analyzed. Nowadays, ML methods and algorithms are used for this step. Classical ML algorithms do not use the advantage of heterogeneous data and typically do not combine data coming from different data sources and are of different data formats [28]. Models specifically designed for heterogeneous data are lacking the ability to dynamically and robustly adapt to changes in the input, e.g. when data sources are missing or failing (*robust*) or when new data sources are added (*dynamic*). A brute-force approach requires re-training of the whole model, which requires a lot of computational overhead and capacity. In summary, the second challenge is

C2: *Complex data analytics for heterogeneous data*.

Based on these two challenges, concrete requirements (R) for a robust and dynamic ML approach for heterogeneous data in manufacturing and industrial automation are derived, which will be introduced in the following.

R1: *Heterogeneous data with different semantics and syntax shall be integrated*.

Interfaces for the data integration shall be flexible and extendable for new data sources.

R2: *A central data access in a uniform, machine-readable format shall be given under the preservation of available information*.

Known relations within the data shall be modeled and the data shall be described in a unique semantic.

R3*: The existing data variety shall be used for data analysis*.

I.e. heterogeneous data shall be analyzed and used and it shall intelligently be combined.

R4: *The built model/method shall be robust and dynamic to changes*.

It shall work in case of missing or failed (*robust*) and be extendible for new (*dynamic*) data sources.

In the following, an architecture is introduced, which is able to realize the concept fulfilling the four requirements. The architecture shall finally resolve the introduced and discussed challenges.

4. Architecture for Dynamic and Robust Machine Learning for heterogeneous Data

To address the derived requirements and realize the concept, a generic multi-layer architecture is first proposed. In the different layers, the required functionalities can be realized to fulfill the requirements. The architecture should, among others, enable reusability for different applications and domains and ensure functional suitability and extendibility/changeability. Therefore, software architecture aspects such as modularization, orchestration, and usage of service for encapsulation of functionality should be incorporated. The multi-layer architecture is shown in Figure 1. The different layers will be introduced in the following. Outside of the system architecture, the different heterogeneous data sources are given. Between the layers and even within the layer, the communication shall take place via services. With the help of orchestration, a higher level service can be offered to a user via

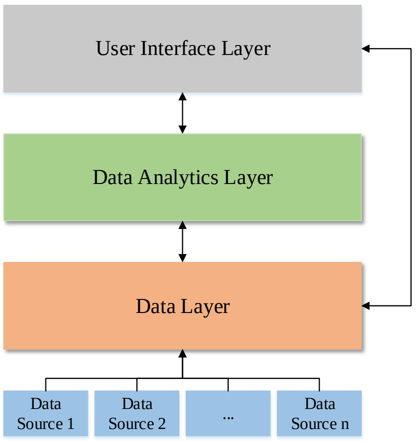

Figure 1: Generic multi-layer architecture for heterogeneous data analytics

the user interface layer. Due to the encapsulated modules and functionalities, new modules can easily be added. Authorization aspects can be added if some data or services shall be available solely to a specific user group.

4.1. Data Layer

The data layer shall integrate heterogeneous data with different semantics and syntax (R1) and allow central data access in a uniform and machine-readable format (R2). Thus, this layer needs to perform several tasks, e.g. data integration, data cleaning, and data storage. As a solution, a context model with pre-processing steps is proposed for the data layer.

The context model represents heterogeneous data uniformly within the data layer based on acquired data and enables knowledge-based learning through its access by the ML model. Heterogeneous data sources differ in type, sampling rate as well as the communication interface. Therefore, communication interfaces to different data sources are applied. These interfaces can for example be part of an integration platform. Configuring data sources as well as data sampling frequency then takes place. Data acquisition can be either event-based or time-based according to the system and its requirements. Each acquired data is represented by a JSON object according to a pre-defined schema defining different categories for the context model and having further metadata such as timestamps. Different data sources can represent a specific category, e.g. the category environment can be represented by multiple data such as temperature, humidity, and light intensity. Next, data is accumulated and integrated within a database, before further pre-processing takes place. The pre-processing checks the validity of data and aggregates it in 3 steps as shown in Fig.2. *First*, the validity and plausibility of the data is checked via Kalman filters (Data Validation). *Second*, outliers or values out of range are detected and filtered out (Outlier Detection) before (*third*) aggregating data and reducing their dimensionality (Dimensionality Reduction). The latter addresses the aspect of heterogeneous sampling rates, for which continuous and voluminous data series are aggregated and mapped to data with lower sampling frequency. Having this step is necessary for a coherent representation by the context model. After pre-processing, semantic annotation takes place based on the pre-defined categories and the aggregated data is represented as a CSV file. The context model has interfaces to the data acquisition and pre-processing modules and acquires CSV files for modeling an instance of the context model. A pre-defined but extendible metamodel describes the relationship between each category, therefore relating the system to its external environment and users. Based on the metamodel, a context model instance is created, denoting specific values for modeled entities i.e. categories. Each model instance is stored and related to the metamodel, so that an efficient query of either a specific entity or an instance of the complete context model is possible. Fig. 3 depicts the communication between the context model and the machine learning model.

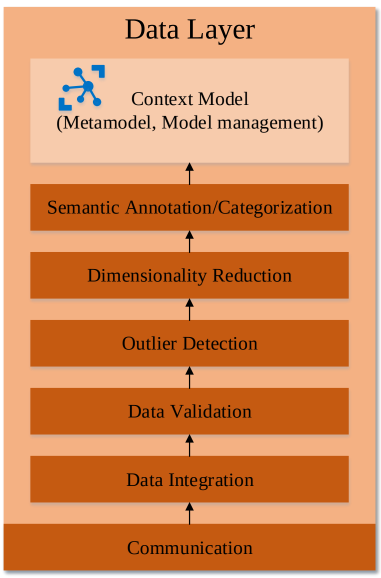

Figure 2: Data Layer approach to fulfill R1 and R2

4.2. Data Analytics Layer

Within the data analytics layer, the requirements R3 and R4 shall be addressed. First, the data variety shall be used for data analysis in an intelligent way (R3). Further, based on R3, the data analytics model shall be capable of dealing with missing/failing data sources (*robust*) and newly added or changed data sources (*dynamic*) (R4).

For the data analysis, the data modeled in the data layer is used and provided to the data analytics layer. The data is filtered, preprocessed and modeled in the context model at the data layer and then provided to the data analytics layer. In addition, the context model can provide information about sensor status, e.g. if a sensor is defective, missing or a new sensor is added. The interface between the data layer and the data analytics layer is shown in Figure 3.

To analyze the heterogeneous data, multi-view neural networks are proposed, which are capable of analyzing data coming from different sources in different formats, since e.g. image and time-series data will be in different formats, even after preprocessing and structuring within the data layer. Three possible realization structures for multi-view neural networks are shown in Figure 4, where (neural network) models are used in different ways (early, late, and joint fusion) to analyze data

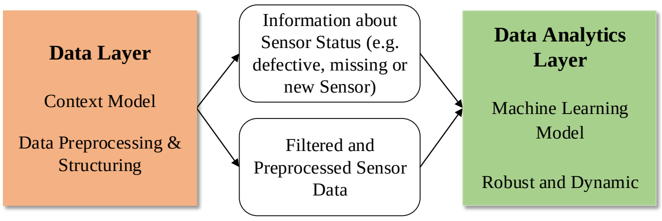

Figure 3: Interface between context model and robust and dynamic machine learning model

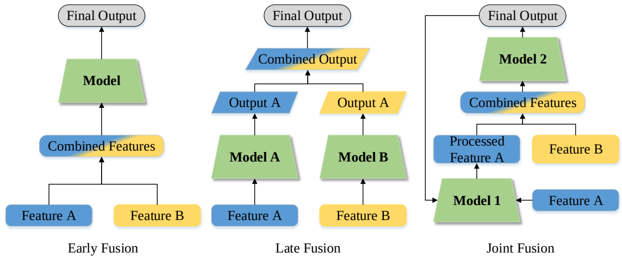

Figure 4: Possible structures for multi-view neural networks

coming from two sources (feature A and feature B). Early fusion first combines the features A and B and analyzes them jointly in a model within the data analytics layer (e.g. Machine Learning model) to obtain the final output. Late fusion first analyzes the features A and B separately and combines the outputs of the models for the final output. Joint fusion processes feature A in an early stage in model 1 and fuses the processed feature A with feature B for analysis in model 2 for the final output. The information from the output can be used as additional information in model 1 for processing feature A.

However, multi-view neural networks are not *robust* and *dynamic* by default. Further concepts are necessary to enhance the existing networks, which were reviewed and developed. The prototypical implementations are currently ongoing, which will be presented in the future. Nevertheless, first obtained outcomes of the analysis will be discussed. Table 1 lists the approaches and gives a preliminary rating of them for the *robustness* and *dynamics* of a neural network.

The first approach is input layer dropout (ILD), where dropout is used at the input layer. Typically, dropout is used within the hidden layers of a neural network to reduce the overfitting of the network. With ILD, single inputs are set to "0" randomly with a certain probability. This enhances the robustness of the model to missing inputs (inputs with "0"). This approach does not tackle the dynamic of the network.

A denoising autoencoder (DAE) learns to reconstruct the original input from a corrupted input (single inputs are set to "0" during training of the DAE). The DAE shall thus be capable to reconstruct an "original" input from a corrupted one, where single inputs are e.g. missing or failed. The DAE is used as a separate preprocessing step for the neural network. The DAE can reconstruct an original input, however, it artificially generates information, which can also be misleading for the following neural network. It does not cover the dynamic aspect.

Adversarial learning (AL) tries to find adversarial samples for the neural network during training to improve the robustness of the model. Typically, little noise is added to an input, which changes the output of the neural network, leading to a wrong prediction. In this case, the adversarial loss term is adjusted, so that single inputs are corrupted to "0", mimicking a missing or failed sensor. This incorporates the adversarial samples into the training process of the neural network. However, this approach has the problem of a vanishing gradient, since the adversarial loss is calculated based on the network's output then backpropagated through the network. The adjusted adversarial loss forces outputs to "0", leading to the vanishing gradient. This problem is under further research.

Transfer Learning (TL) aims at transferring existing knowledge learned in a source task to a new target task. Following [29], TL can be distinguished in different categories, where for the similarity of the input feature space, two categories are commonly distinguished:

- Homogeneous transfer learning: a setting in which the source and target input feature space are identical.
- Heterogeneous transfer learning: a setting in which the source and target input feature space differ.

Heterogeneous transfer learning directly tackles the previously introduced problem, where sensors are missing or failing (*robust*) or new sensors are added (*dynamic*). The input feature space differs, while the tasks remain the same.

Following the first investigations, input layer dropout and transfer learning seem to be the most promising approaches, where transfer learning seems suitable for both, the robustness and dynamics of the model, while input layer dropout seems to be suitable for increased robustness.

4.3. User Interface Layer

The user interface layer serves as the link between the developed concept and models and the user. The user can, e.g., be a domain expert, a data scientist, or a worker. Depending on the role, different views shall be given to enable the desired functionalities effetivly. A domain expert, e.g., wants detailed sensor values or analysis results for certain situations of the system, while the worker may just need high-level feedback

Table 1: Identified Approaches for robust and dynamic neural networks

Approach	**Description**	**Robustness**	**Dynamic**
Input Layer Dropout (ILD)	Dropout is used at the Input Layer of the neural network.	●	○
Denoising Autoencoder (DAE)	Inputs are corrupted. DAE learns to reconstruct original Input from corrupted input.	◑	○
Adversarial Leraning (AL)	Corruption of inputs through finding adversarial samples. Adversarial loss is adjusted, so that single inputs are "missing"	◑	○
Transfer Learning (TL)	Model is trained on original input. Can then be transferred to a new domain (either less-dimensional input or higher dimensional input)	●	●

about the system state, such as “OK” or “Not OK”. The data scientist may need the opportunity to train the neural network model and therefore define the training data and set relevant parameters for the training procedure. Therefore, an interface to the data analytics and the data layer is manadatory from the user interface layer. This will be developed upon the previously described layers and is part of future work since this is highly dependent on the functionalities of the lower layers.

5. Conclusion & Outlook

In this work, two challenges for handling heterogeneous data within the industrial automation and manufacturing domain were introduced based on previous work. Four requirements are then derived from literature to resolve the aforementioned challenges. To fulfill the requirements, a generic multi-layer architecture for heterogeneous data analysis was proposed, consisting of a data layer, a data analytics layer and a user interface layer. The architecture is modular and extendible and easily transferable to different applications. Concrete approaches are introduced and discussed for the realization of the multi-layer architecture to handle and analyze heterogeneous for industrial automation systems and manufacturing engineering applications:

- A unifying context model with defined pre-processing steps is introduced as a realization for the data layer, highlighting on relations between categorized data entities.
- Multi-view neural networks are then proposed for the analysis of heterogeneous data.
- To ensure robustness and dynamics of the neural network, different approaches are discussed, identifying input layer dropout and transfer learning as the most promising ones.

In future work, this defined architecture will be realized for an industrial automation system, to detailly prove the concept. Detailed research on the robustness and dynamics of neural networks is currently ongoing and will further be investigated.

References

[1] B. Maschler, S. Kamm, and M. Weyrich, “Deep industrial transfer learning at runtime for image recognition,” *at-Automatisierungstechnik*, vol. 69, no. 3, pp. 211–220, 2021.

[2] J. Villalba-Diez *et al.*, “Deep learning for industrial computer vision quality control in the printing industry 4.0,” *Sensors*, vol. 19, no. 18, p. 3987, 2019.

[3] H. Vietz *et al.*, “A Methodology to Identify Cognition Gaps in Visual Recognition Applications Based on Convolutional Neural Networks,” in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 2045–2050.

[4] B. Lindemann, N. Jazdi, and M. Weyrich, “Anomaly detection and prediction in discrete manufacturing based on cooperative LSTM networks,” in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 2020, pp. 1003–1010.

[5] S. Kamm *et al.*, “Hybrid Modelling for the Failure Analysis of SiC Power Transistors on Time-Domain Reflectometry Data,” in *2021 IEEE International Symposium on the Physical and Failure Analysis of Integrated Circuits (IPFA)*, pp. 1–6.

[6] S. Kamm, N. Jazdi, and M. Weyrich, “Knowledge Discovery in Heterogeneous and Unstructured Data of Industry 4.0 Systems: Challenges and Approaches,” *Procedia CIRP*, vol. 104, pp. 975–980, 2021.

[7] N. Sahlab, N. Jazdi, and M. Weyrich, “Dynamic Context Modeling for Cyber-Physical Systems Applied to a Pill Dispenser,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020, pp. 1435–1438.

[8] M. Canizo, I. Triguero, A. Conde, and E. Onieva, “Multi-head CNN - RNN for multi-time series anomaly detection: An industrial case study, ” *Neurocomputing*, vol. 363, pp. 246–260, 2019.

[9] V. Jirkovský and M. Obitko, “Semantic Heterogeneity Reduction for Big Data in Industrial Automation,” *ITAT*, vol. 1214, 2014.

[10] L. Wang, “Heterogeneous data and big data analytics,” *Automatic Control and Information Sciences*, vol. 3, no. 1, pp. 8–15, 2017.

[11] M. Chen, S. Mao, and Y. Liu, “Big data: A survey,” *Mobile networks and applications*, vol. 19, no. 2, pp. 171–209, 2014.

[12] I. F. Ilyas and X. Chu, *Data cleaning*: Morgan & Claypool, 2019.

[13] S. Schabus and J. Scholz, “Semantically Annotated Manufacturing Data to support Decision Making in Industry 4.0: A Use-Case Driven Approach,” in *Data Science - Analytics and Applications*: Springer, 2017, pp. 97–102.

[14] D. Ostrowski, N. Rychtyckyj, P. MacNeille, and M. Kim, “Integration of big data using semantic web technologies,” in *2016 IEEE Tenth International Conference on Semantic Computing (ICSC)*, 2016, pp. 382–385.

[15] A. K. Dey, “Understanding context,” *Personal and Ubiquitous Computing*, vol. 5, no. 1, pp. 1617–4917, 2001.

[16] N. Sahlab, N. Jazdi, and M. Weyrich, “An Approach for Context-Aware Cyber-Physical Automation Systems,” *IFAC-PapersOnLine*, vol. 54, no. 4, pp. 171–176, 2021.

[17] S. van Engelenburg, M. Janssen, and B. Klievink, “Designing context-aware systems: a method for understanding and analysing context in practice,” *Journal of logical and algebraic methods in programming*, vol. 103, pp. 79–104, 2019.

[18] S. Kamm, K. Sharma, N. Jazdi, and M. Weyrich, “A Hybrid Modelling Approach for Parameter Estimation of Analytical Reflection Models in the Failure Analysis Process of Semiconductors,” in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, 2021, pp. 417–422.

[19] B. Lindemann, F. Fesenmayr, N. Jazdi, and M. Weyrich, “Anomaly detection in discrete manufacturing using self-learning approaches,” *Procedia CIRP*, vol. 79, pp. 313–318, 2019.

[20] A. Bartler *et al.*, “Automated detection of solar cell defects with deep learning,” in *2018 26th European signal processing conference (EUSIPCO)*, 2018, pp. 2035–2039.

[21] J. Zhao, X. Xie, X. Xu, and S. Sun, “Multi-view learning overview: Recent progress and new challenges,” *Information Fusion*, vol. 38, pp. 43–54, 2017.

[22] B. Shahian Jahromi, T. Tulabandhula, and S. Cetin, “Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles,” *Sensors*, vol. 19, no. 20, p. 4357, 2019.

[23] F. Garcia, D. Martin, A. de La Escalera, and J. M. Armingol, “Sensor fusion methodology for vehicle detection,” *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 123–133, 2017.

[24] J. Wang *et al.*, “Multilevel information fusion for induction motor fault diagnosis,” *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2139–2150, 2019.

[25] M. Jayaratne, D. de Silva, and D. Alahakoon, “Unsupervised machine learning based scalable fusion for active perception,” *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1653–1663, 2019.

[26] R. Henkel, O. Wolkenhauer, and D. Waltemath, “Combining computational models, semantic annotations and simulation experiments in a graph database,” *Database*, vol. 2015, 2015.

[27] A. L’heureux, K. Grolinger, H. F. Elyamany, and M. am Capretz, “Machine learning with big data: Challenges and approaches,” *Ieee Access*, vol. 5, pp. 7776–7797, 2017.

[28] X. Wilcke, P. Bloem, and V. de Boer, “The knowledge graph as the default data model for learning on heterogeneous knowledge,” *Data Science*, vol. 1, 1-2, pp. 39–57, 2017.

[29] B. Maschler and M. Weyrich, “Deep transfer learning for industrial automation: a review and discussion of new techniques for data-driven machine learning,” *IEEE Industrial Electronics Magazine*, vol. 15, no. 2, pp. 65–75, 2021.