



Artificial-intelligence-enabled dynamic demand response system for maximizing the use of renewable electricity in production processes

Hendro Wicaksono¹ · Martin Trat² · Atit Bashyal¹ · Tina Boroukhanian¹ · Mine Felder² · Mischa Ahrens² · Janek Bender² · Sebastian Groß² · Daniel Steiner² · Christoph July³ · Christoph Dorus⁴ · Thorsten Zoerner⁴

Received: 13 February 2024 / Accepted: 14 February 2024

© The Author(s) 2024

Abstract

The transition towards renewable electricity provides opportunities for manufacturing companies to save electricity costs through participating in demand response programs. End-to-end implementation of demand response systems focusing on manufacturing power consumers is still challenging due to multiple stakeholders and subsystems that generate a heterogeneous and large amount of data. This work develops an approach utilizing artificial intelligence for a demand response system that optimizes industrial consumers' and prosumers' production-related electricity costs according to time-variable electricity tariffs. It also proposes a semantic middleware architecture that utilizes an ontology as the semantic integration model for handling heterogeneous data models between the system's modules. This paper reports on developing and evaluating multiple machine learning models for power generation forecasting and load prediction, and also mixed-integer linear programming as well as reinforcement learning for production optimization considering dynamic electricity pricing represented as Green Electricity Index (GEI). The experiments show that the hybrid auto-regressive long-short-term-memory model performs best for solar and convolutional neural networks for wind power generation forecasting. Random forest, k -nearest neighbors, ridge, and gradient-boosting regression models perform best in load prediction in the considered use cases. Furthermore, this research found that the reinforcement-learning-based approach can provide generic and scalable solutions for complex and dynamic production environments. Additionally, this paper presents the validation of the developed system in the German industrial environment, involving a utility company and two small to medium-sized manufacturing companies. It shows that the developed system benefits the manufacturing company that implements fine-grained process scheduling most due to its flexible rescheduling capacities.

Keywords Demand response system · Reinforcement learning · Artificial neural network · Energy-flexible production optimization · Semantic middleware · Ontologies · Green Electricity Index

Highlights

- Semantic middleware for semantic data integration in manufacturing demand response systems.
- Ontology-based integration model.
- Multiple machine learning models for renewable power generation forecasts and load prediction.
- Green Electricity Index indicating renewable electricity proportion and price.
- Real-time data acquisition system using the open-source solution OpenEMS.
- Reinforcement learning for production optimization under dynamic electricity pricing.

✉ Hendro Wicaksono
hwicaksono@constructor.university

✉ Martin Trat
trat@fzi.de

Extended author information available on the last page of the article

1 Introduction

Energy flexibility, particularly local flexibility, plays a decisive role in Europe's transition to renewable energy. According to the European Union's Energy Union Framework, consumers should engage in the energy transition by actively

participating in the energy market [1]. They can benefit from technological progress in energy cost reductions, and it can be realized, for instance, by participating in demand response (DR) programs. DR programs focus on changing energy consumption patterns on the end consumer's side by varying the electricity price over time or offering incentive payments [2]. Due to the increasing demand for personalized products and volatile market demand that lead to higher product complexity, industrial consumers must have more process flexibility [3]. It includes flexibility in achieving production objectives such as lower energy costs, higher quality, and higher profit.

In 2021, Germany's industrial sector had a share of roughly 44% of the country's total electricity consumption. Among sectors, this represents the largest share, followed by those of the commerce, household, and transportation sectors [4]. Additionally, industrial consumers faced steep rises in electricity prices [5]. Regarding gross electricity consumption across all sectors, the share of renewable energy was estimated to be around 41%. German legislation seeks to increase this share to around 80% by 2030 [4]. This process complicates electricity distribution and requires many stakeholders to embrace a paradigm shift: Contrary to traditional patterns, electricity consumption has to be dynamically balanced with highly volatile renewable generation. For instance, energy providers and distribution system operators need tools to predict generation and motivate flexible consumption behavior accurately. Apart from that, industrial processes cause large quantities of emissions. On a global scale, they generated around 2.54 Gt carbon dioxide equivalent of energy-related greenhouse gases in 2021, the highest level since 2000 [6].

Several studies have demonstrated the applicability and benefits of demand response for manufacturing consumers regarding growing energy-related needs, costs, and emission problems [7]. However, their participation in demand response programs has been much lower than that of the household and commercial sectors so far [8]. The implementation of demand response systems for manufacturing consumers is a challenging task due to the heterogeneous and large amount of data required by the systems, for example, for power generation forecasting [9], load prediction [10], and manufacturing process optimization [11].

This paper considers these aspects for designing and implementing a demand response system for the manufacturing industry. The proposed implementation optimizes industrial consumers' and prosumers' production-related electricity costs through electricity tariffs, which dynamically depend on the proportion of renewable energy in the procured mix. It also features a semantic middleware architecture that utilizes an ontology as the semantic integration model for heterogeneous data and exchanges the data for various components conducting power generation forecasting, load prediction, and production optimization. For these

components, a set of different machine learning models is developed. Accurate models are crucial as the electricity load can vary substantially depending on process characteristics and parametrizations [12].

This paper is organized as follows. Section 2 presents a literature review of the research related to the proposed solution's main parts. Then, the overall concept of the developed solution is described in Section 3. Section 4 focuses on semantic middleware as the data integration component of the solution. The system and processes that collect power consumption data based on the energy management solution OpenEMS are introduced in Section 5. The machine learning models for power generation and load prediction are elaborated in Section 6. The Green Electricity Index (GEI), representing the proportion of renewable electricity that is calculated using power generation and consumption forecast data, is explained in Section 7. Section 8 discusses mixed-integer linear programming and reinforcement learning approaches for production optimization under dynamic electricity pricing. Finally, Section 9 shows the implementation and validation of the developed solution, and Section 10 presents the conclusions and outlook.

2 Literature review

The literature review aims to analyze related research works and identify research gaps in semantic information integration, artificial intelligence in DR, and energy-flexible production. These areas are selected to represent the essential functions of using artificial intelligence to enable demand response systems in industrial settings. The review analyzed peer-reviewed journal and conference papers in the last 15 years. Scopus, IEEE Xplore, and Google Scholar databases were used to find the papers. The papers were grouped into the areas shown in the subsections below. They were analyzed in more detail to identify the research gaps in each of these. The research gaps and the contributions of this paper are summarized in Section 2.4.

2.1 Energy 4.0

Energy 4.0 refers to digital transformation in the energy and utilities industry by utilizing technologies such as the Internet of Things [13]. An example of the implementation of Energy 4.0 is the PEACEFULNESS software platform aiming at maximizing the use of renewable energy sources through demand-side management. PEACEFULNESS allows the interaction of multi-energy grids and simulations of operations and supervisions of hundreds of thousands of agents such as energy providers, distribution system operators, aggregators, and prosumers [14]. Furthermore, Mourtzis et al. [15] developed a business model that facilitates col-

laborations between energy providers and manufacturing power consumers. They focused on a production scheduling approach adaptive to dynamic energy prices calculated based on forecasts.

The implementation of Energy 4.0 has created information islands that can contribute to developing smart grids to go beyond the industrial sector by providing additional data [16]. As a consequence of Energy 4.0 implementation, a considerable amount of data is generated and must be managed in data centers. Energy consumption of data centers is increasing due to more energy being required to process and store large amounts of data combined with environmental factors. Yang et al. [17] developed an artificial-intelligence-based approach to intelligently schedule the control and refrigerating engine to reduce energy consumption. Similarly, researchers also engaged themselves in green artificial intelligence, which focuses on efficient methods to train and evaluate artificial-intelligence-based models [18].

2.2 Semantic information integration and interoperability

Semantic integration aims at efficiently integrating heterogeneous data from various sources using semantic technologies. Such technologies often resolve the problems of integration and interoperability within heterogeneous interconnected objects and systems by using a middleware component [19–21], which utilizes linked data to interconnect data sources. Information interoperability among different systems is done using ontologies [22, 23], which resemble the structure of information sources on the semantic level [24, 25].

Adamczyk et al. [26] present a knowledge-based expert system to support semantic interoperability in smart manufacturing based on an ontological approach. Similarly, semantic technologies are used in the RESPOND project to propose interoperable energy automation and deal with data heterogeneity as well as its consequent integration [27].

In his study, Mourtzis [23] explores the current industrial landscape to identify opportunities, challenges, and gaps in efficiently utilizing large amounts of data for process optimization. He specifically focuses on integrating semantics into this framework as part of his efforts.

There are various frameworks for ontology-based data exchange. Regarding energy and flexibility value chain data, several standards exist [28, 29], as well as a multi-agent-system approach building upon such to enable market-based flexibility [30]. Furthermore, semantic technologies are also applied to address building energy management [31, 32], water urban management [33], manufacturing energy management [34], and engineering processes [35]. Li et al. [36] also propose a semantic-based approach for a real-time demand response energy management system for a district heating network with monitoring and handling of hetero-

geneous data sources. It is employed to integrate a thermal network with other networks.

2.3 Artificial-intelligence-aided demand response

Machine learning (ML) methods can provide tools that function as vital entities in developing automated demand-side management resources. The tools are applied across various application areas such as forecasting (load, price, and renewable energy), clustering/identifying consumer groups, scheduling and load control, and pricing/incentive scheme design for DR programs. This section provides a brief overview of machine learning methods used as supporting tools in various application areas of DR. The authors do not discuss the relevance of each individual application area, but interested readers can find detailed discussions in [37].

2.3.1 Supervised machine learning

Primarily supervised ML techniques are applied across all aforementioned forecasting tasks [38]. These include kernel-based methods (e.g., support vector machines) [39–42], tree-based methods (e.g., random forests) [43–47], artificial neural networks (e.g., single-neuron architectures and deep learning) [48–54], and statistical time series models (e.g., variants of the auto-regressive integrated moving average (ARIMA) method) [55–59]. Artificial neural networks and deep learning methods are used more frequently in short-term forecasting tasks. If such are employed with recurrent network architectures, sequence memory and attention mechanisms can be realized.

Especially for the task of forecasting electricity loads, supervised ML-based approaches are popularly employed. In the past, neural networks [60, 61], support vector regression [62], and Gaussian processes for regression [63] were shown to achieve good load estimation performance for cutting, rotating, and drilling machinery. A more recent work of Ellerich [64] extends previous approaches, using a neural network and discretizing machine operations into temporal blocks. The approach of Mühlbauer et al. [65] also focuses on individual operations such as, for instance, spindle acceleration and deceleration of cutting machines. For such machines, they define numerical time and energy models, allowing for load prediction at the granularity of operations.

In contrast, calculating electricity load based on physically founded analytical approaches lacks accuracy. This can, in great measure, be attributed to the stochastic nature of machine processes [66, 67].

Other research works use supervised learning models for tasks other than those addressed above. Such models are employed, e.g., for learning comfort and behavior patterns of DR participants [68, 69].

2.3.2 Unsupervised machine learning

In DR, unsupervised learning algorithms have been predominantly used for clustering DR participants based on their consumption and appliances as well as on general load profiles [70–74]. Participant clusters can further be used to identify their flexibility potentials or resources and even to compensate consumers for participating in DR programs (assuming the cluster members share a similar baseline consumption).

2.3.3 Reinforcement learning

Due to varying complexity levels of scheduling and the often-delayed feedback of associated descriptive metrics, approaches based on reinforcement learning (RL) are particularly interesting for scheduling applications [75]. For instance, Lu et al. [7] suggest a multi-agent RL-based approach. Their goal is to minimize production processes' energy and material costs within flow-shop environments, which can be characterized by processes with constant step sequences. In contrast, Naimi et al. [76] focus on finding an optimal solution within the trade-off between minimizing energy consumption and makespan. They apply a single-agent RL approach. Apart from these, a large set of other works also pursue the goals of minimizing electricity costs or consumption via RL [77–87].

2.3.4 Other methods

Other approaches for optimizing energy-related metrics within the production context can be based on heuristics and mathematical methods. Heuristic approaches employ, for instance, simulated annealing [88], particle swarm optimization [89], ant colony optimization [90], backtracking search [91], and genetic optimization [92]. In general, such approaches are computationally efficient but cannot guarantee globally optimal solutions. Mathematical approaches, however, target globally optimal solutions specifically using exact calculation procedures. Therefore, such are often characterized by high time costs when compared to heuristics. Exemplary works apply linear programming [93] and constraint programming [94].

Other research works focus on optimizing peak load or peak to average ratio [79, 83, 84, 95], life cycle costs of energy storage systems [96], or environmental pollution using, e.g., carbon footprint measures [97]. Along with scheduling/controlling household appliances, some research works focus on the self-consumption of renewable energy [87, 98, 99]. This consumption strategy is an aspect of DR and is a catalyst that can increase profitability in grid-connected systems [98].

2.4 Summary of the research gaps and contributions of this research

This work develops a semantic middleware that utilizes ontology as the integrative information model of data from different domains required for the suggested demand response system implementation. The ontology is constructed by reusing and interconnecting well-established and recommended ontologies. None of the works mentioned in Section 2.2 focuses on semantic integration or ontology development for demand response systems except [27]. However, it does not consider reusing existing ontologies.

The research presented in this paper aligns with other works with respect to applying supervised learning for power generation forecasting and load prediction as described in Section 2.3. Nevertheless, none of those integrate power generation and load prediction in a demand response setting facilitated by a semantic middleware.

Several of the works on RL outlined in Section 2.3.3 aim to optimize manufacturing process scheduling. Each of these works focuses on a specific environment. The RL-based approach presented in this paper, however, is validated in two different shop floor environment types. The key contribution of the RL approach to the existing literature is that two new reward functions were developed regarding two machine states (running and idle) and dynamic electricity prices. This work also conducts experiments to qualitatively compare the performance of the developed RL approach to a mixed-integer linear programming model.

Furthermore, simulations have been used to support decision-making in manufacturing, considering multiple objectives, including energy efficiency. Digitalization implies more data being generated across different manufacturing processes, allowing simulation approaches to employ more complex models that integrate digital data across the manufacturing life cycle [100]. However, research on data-driven simulation in demand response contexts is still limited. This work develops a data integration approach across multiple organizations, such as energy providers and manufacturing power consumers. It also integrates data-driven simulations implemented using semantic information integration, ML, and RL.

3 Overall concept

Considering the research gaps in the field of DR and related concepts and solutions as summarized in Section 2.4, this research proposes an overall system architecture of the DR system depicted in Fig. 1. The architecture comprises components that represent the advancements beyond the works in the fields of Energy 4.0, semantic integration, and machine learning mentioned in Sects. 2.1, 2.2, and 2.3.

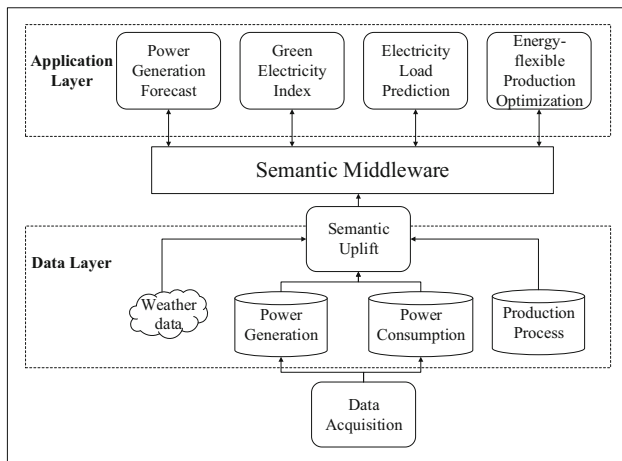


Fig. 1 System architecture of the developed demand response system

The architecture consists of a data layer, an application layer, and the semantic middleware as the connector between both layers. The data layer is responsible for storing and providing the required data sources, such as power generation and consumption data, production process data, and supporting data, such as weather data. The weather data is used as input for the power generation forecast and acquired from an open weather API [101]. The data acquisition system collects the actual generation and consumption data (cf. Section 5). Production process data are obtained using data acquisition systems in the electricity-consuming manufacturing companies’ IT landscape that log the activities of electricity-consuming machines. The data types involved in the data layer are described in Table 1.

The application layer contains four core components: power generation forecast (cf. Section 6.1), electricity load prediction (cf. Section 6.2), Green Electricity Index (GEI) (cf. Section 7), and energy-flexible production optimization (cf. Section 8). These four main components retrieve the required data from the semantic middleware through a RESTful API. Table 2 shows the details of each component, including its inputs and outputs.

Initially, the semantics of the collected data are extracted through a semantic uplift process and then integrated into the semantic middleware (cf. Section 4). To simplify user interaction, the semantic middleware provides a natural-

language query interface allowing data retrieval directly using queries formulated in English. It also allows knowledge graph queries using Neo4j’s Cypher query language syntax. The output data generated by components in the application layer, such as forecast data, GEI, and energy-optimized production schedules, are made available through the semantic middleware. It is represented by a set of bidirectional arrows in Fig. 1. As a result, data-consuming applications or users can retrieve data through interfaces provided by the semantic middleware.

Data flows associated with the abovementioned core components of the system are shown in Fig. 2. These components exchange data in a standardized fashion via the semantic middleware. On the one hand, the system requires consumers to provide primarily time and quantity information on their manufacturing processes and associated process parameters. The latter is also used as input for predicting individual load profiles for manufacturing each part or set of simultaneously manufactured parts. This enables the energy-flexible production optimization component to shift these manufacturing processes temporally based on their respective load profiles to optimize electricity costs. On the other hand, consumers are required to input user preferences that represent their scheduling needs. Based on this, the trade-off between makespan and electricity cost minimization can be parametrized for production optimization.

Apart from aforementioned inputs, the production optimization component requires electricity price profiles that are communicated by energy providers. In this context, a dynamic index variable, introduced in Section 7, represents energy pricing information and information on the share of energy from renewable sources among all sources within the purchased mix. Among other information, its computation is based on the predicted amount of renewable energy generated for the electricity grid environment the consumer belongs to. The production optimization component then determines and outputs schedules that are optimized with respect to the provided user preference, as described above, regarding electricity costs and makespan characteristics. Schedules are reported back to the consumers and, in an aggregated measure, supply energy providers with information on their customers’ electricity demand, which in turn supports supply planning.

Table 1 Description of data types on the data layer

Data type	Data source	Data communication interface
Weather	Open weather API [101]	REST API
Power generation	Energy management system of the utility company	REST API, CSV data import
Power consumption	Data acquisition system (cf. Section 5) at industrial consumer’s site	REST API
Production process	Computer-aided manufacturing programs	REST API, CSV data import

Table 2 Description of components in the application layer

Component	Description	Inputs	Outputs
Power generation forecast	A set of machine learning models to forecast electrical power generated from renewable sources, such as wind and solar, for a given time horizon	Weather data, past electrical power generation data, power plant types	Hourly power generation forecast with a given time horizon
Load prediction	A set of machine learning models to predict electrical power consumed by manufacturing machines for given process parameters	Process parameters and machine information	Predicted load of each machine based on its process parameters
Green electricity index	An indicator system that measures the amount of renewable energy in the electricity grid in comparison to the demand for electricity in Germany	Power generation forecast data, predicted load profiles	Hourly GEI
Energy-flexible production optimization	Production scheduling optimization system considering dynamic electricity pricing	Process parameters, predicted load profiles, GEI, and user preferences	Energy-flexible production schedules

This work evaluates the proposed system within the productive context of two manufacturing companies. Alongside other aspects, these companies can be characterized by different degrees of automation as well as production process schemes. More details of these companies' profiles as well as an assessment of the system's practical potential can be found in Section 9.2.

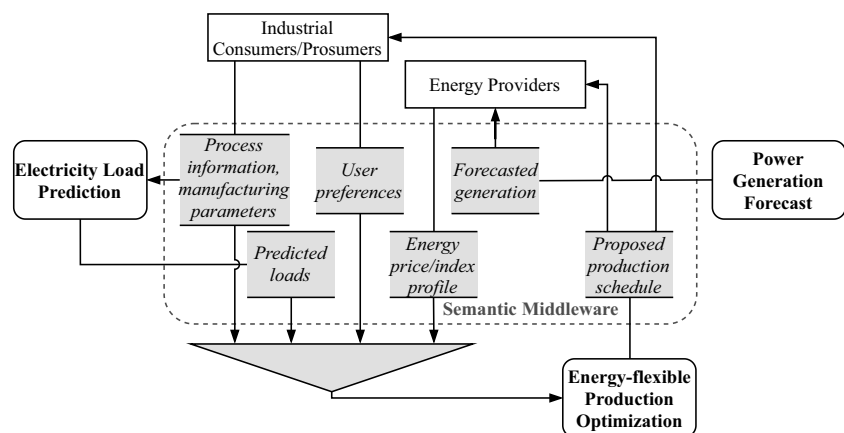
4 Semantic middleware

This section presents the semantic middleware that is developed to interface applications with multiple data sources in the proposed DR system. Generally, semantic middleware facilitates data integration across domains, including smart grids, renewable-energies-related fields, physical energy characteristics, sensors, and manufacturing. It is a software module using ontologies as the information inte-

gration model [102]. This work proposes an ontology to provide interoperability and facilitate data integration across various domains, ranging from power generation to the energy-efficient and intelligent use of production resources, by utilizing three different approaches, which are described as follows:

1. Ontology mapping and alignment: This approach interlinks related ontologies to facilitate continuous model integration across domains, avoiding data duplication. Initially, classes that are correlated across ontologies need to be analyzed regarding potential links. Then, specifically developed methods for ontology alignment can be applied for interlinking ontologies.
2. Developing a new ontology: Two methods can be employed to create a desired ontology. The first involves the reuse of existing ontologies, which can be extended by defining new concepts, properties, or relations. The

Fig. 2 Interaction of service components (represented by rounded rectangles with bold text) of the proposed system. The system's semantic middleware routes data flows (represented by partial rectangles with italicized text)



second method is to construct an entirely new ontology for a particular task while incorporating the data requirements of the defined use cases [103].

3. Mapping data to ontologies: An ontology utilizes formal semantics to explicitly define data, which provides a common understanding between stakeholders with different background knowledge [34]. It is achieved by mapping database schemes to semantically equivalent ontologies, which involves gathering data from multiple sources and linking it to the appropriate concepts defined in the ontology in a rule-based manner.

Overall, these three approaches can be used together or separately to develop an ontology that meets the data requirements of a particular task or domain. The ontology developed for the proposed DR system can help stakeholders to better understand data and manage complex data relationships across multiple domains.

The DR system is initially deployed in a cloud-edge architecture that employs semantic uplift processes to extract the semantics of data sources and connect them to the semantic middleware. With this, new knowledge can be derived from the semantically described data retrieved from the semantic middleware. The semantically described data can be retrieved using Neo4j’s Cypher query language, a REST-API, and a specifically developed natural-language interface. Cypher is a graph query language that allows retrieving data from graph-based structures and is generally applied as an expressive and efficient data querying instrument in property graphs [104]. Figure 3 illustrates the semantic middleware architecture.

The required data are primarily retrieved through REST API interfaces. Applications from different domains such as energy management, production planning, and control generate these data. They must be linked to relevant ontologies via data-to-ontology mappings. For this reason, after defining the data requirements, applicable ontology domains are spec-

Table 3 Considered ontologies and their corresponding domain scopes for different data

Data	Ontology	Domain	Reuse
Wind power	OntoWind	Renewable energy	✓
Solar power	Semanco	Energy	✓
Power consumption	Saref4ener	Energy	✓
GEI	GEI	Renewable energy	×

ified based on the data requirements [105]. Then, pre-existing ontologies corresponding to each relevant domain are identified and reused to fulfill the necessary data requirements. The domain scope of ontologies selected based on different data requirements is presented in Table 3. Once mapped, relations between the instances of ontology classes are established. A relation is a triple (subject, property, object) where property is a property class. The elements subject and object are sub-classed instances of the domain and range defined for the property class [106].

The results are imported into a Neo4j graph database, which is serialized using the Resource Description Framework (RDF). The representation of the mapping results for wind power data and the OntoWind ontology [107] in Neo4j, as listed in Table 3, can be seen in Fig. 4. A comprehensive search and review of various wind power-related ontologies was conducted to identify ontologies that meet the data requirements for the renewable energy domain. Based on the identified data requirements, the OntoWind ontology is found to be the best match and employed to map wind power data. The mapping resulting from the previous steps is imported and stored in the Neo4j graph database. The nodes in Fig. 4 represent the classes of the OntoWind ontology, while the edges represent the relations between instances/individuals of these classes. Nodes corresponding to the ns0_WindPower class are highlighted in red, and WindPower values are populated as instances/individuals within these classes. In the node

Fig. 3 The architecture of the proposed semantic middleware component

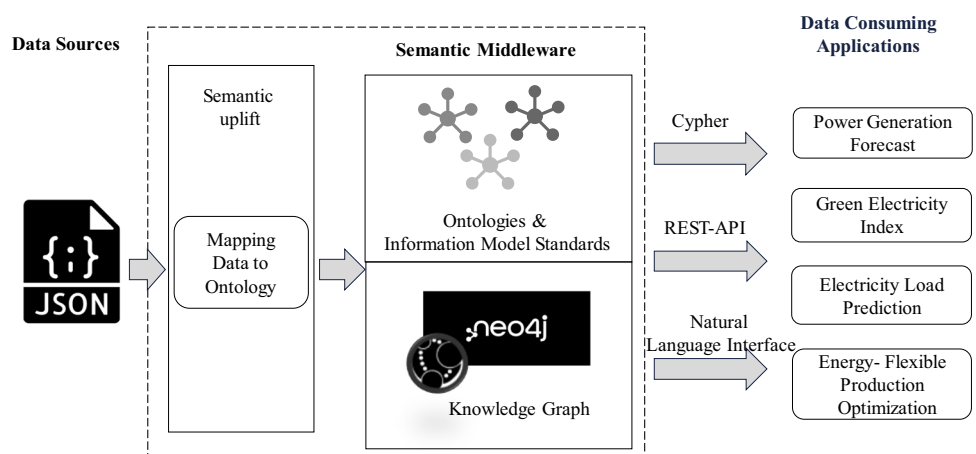
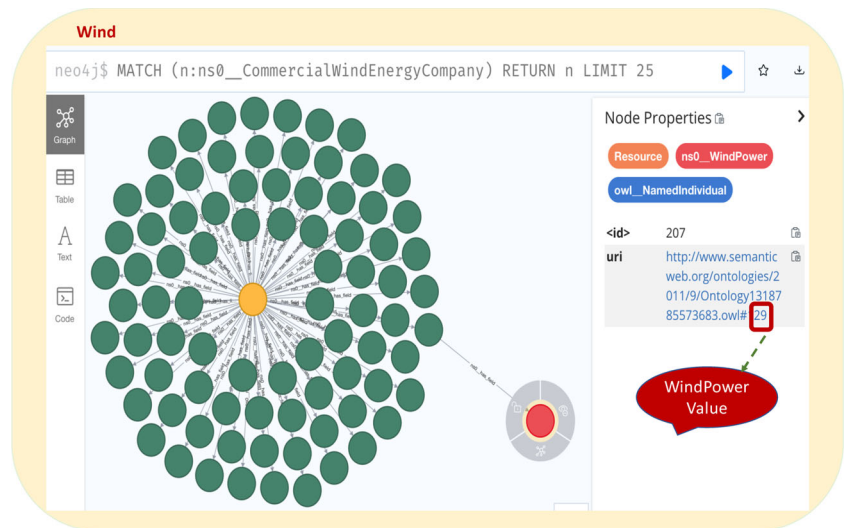


Fig. 4 Neo4j user interface for displaying an imported ontology



properties, the instances/individuals are displayed in the “uri” field. For instance, the value of the wind-power-related entry shown here is 29.

Once mapped, these ontologies are linked in the middle-ware using an ontology alignment method, which enables a common understanding of data from multiple domains and allows for fully integrated retrieval of that data. Since manually matching ontologies is time-consuming, developing automatic or semiautomatic techniques is an important research area in related fields. Bulygin et al. [108], for instance, propose an approach for ontology matching that employs ML methods by combining different similarity measures, including string-based, language-based, and structure-based ontology elements.

Ontologies are applied to support interoperability and inferencing. Among other activities, querying and reasoning are needed for deriving new knowledge, and inference allows inferring new knowledge from the existing. Using Cypher query, the data was exposed as RDF [109]. In this

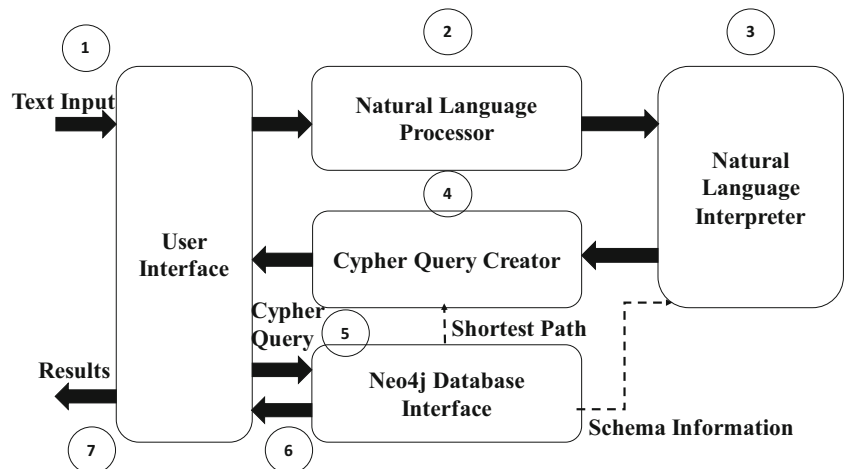
regard, a natural-language interface has been developed to translate natural-language queries into Cypher queries, making it possible to retrieve previously inaccessible information. The process for creating this interface is outlined in Fig. 5.

In the first step, the user enters the text query into the interface. The second step involves text preprocessing in a natural-language processor, which includes the following:

1. Splitting the string into tokens with natural-language tags (nouns, proper nouns, etc.)
2. Improving human readability by removing, e.g., camel case, hyphens, or punctuation
3. Creating sentence objects from the processed string for interpretation

In the third step, the authors use a natural-language interpreter and perform the following actions: (1) Comparing ontology components and (2) recursively splitting spans if no

Fig. 5 Architecture of the proposed system’s natural-language interface



match is found. In the fourth step, the Cypher query creator is used to rank span matches by confidence (or manual adjustment), create constraints from the most confident match of each span, identify the target constraint [110], and output a formatted Cypher query in the user interface in the fifth step. Finally, the Cypher query is executed via Neo4j, and the results are sent to the interface.

Figure 6 depicts the natural-language interface. Users can input a text query and receive the corresponding Cypher query upon confirmation. The application provides an option for users to select the most suitable query from the options displayed in the second field. Finally, the query results can be viewed in the Cypher result section.

5 Data acquisition

The deployed infrastructure for energy data acquisition relies on established and tested concepts. It enables easy control and acquisition of on-premise devices and meters, providing an established framework via the open-source energy management solution OpenEMS.¹ Figure 7 depicts the deployment architecture of OpenEMS. Data are retrieved from local systems, such as production machines and different metering systems, including energy meters. The data are then stored in a measurement database located in the back-end. Backend services such as predictive algorithms and visualization modules access the database through HTTP-REST or JSON-RPC interfaces.

In the case of the employed Janitza UMG 103 and UMG 604 energy meters, communication happens via Modbus. Modbus is a commonly and widely used communication protocol in industrial automation and building automation systems to monitor and control devices remotely. Therefore, it is easily integrated with the existing infrastructure without high opportunity costs. Using a Modbus energy meter also allows for seamless data exchange of the meter with a similar model from a different vendor. The OpenEMS edge instance deployed on a local Internet of Things device cyclically queries all attached devices via the respective communication protocol and transfers the data into a harmonized data model. The software is easily extendable by OSGi (Open Services Gateway initiative) modules. OSGi is a Java-based framework for modular software development that allows for the creation of highly dynamic and configurable applications. The main advantage of using OSGi in software development is that it allows creating modular, reusable, and loosely coupled components. Additionally, OSGi supports the creation of small, highly modular, self-contained modules (called “bundles”) that can be easily added, removed, or updated without affecting the rest of the application. It

facilitates developing, testing, and deploying new devices, features, and bug fixes.

Different configurations of the described acquisition setup are rolled out in the environment of the industrial companies used to validate the proposed system. Figure 8 shows a typical example: Two types of electrical meters are used. The UMG 604 is network capable and acts as a Modbus gateway that provides a transparent connection for UMG 103, which are integrated as Modbus slaves. Per device, power measurement data is locally stored in a time series database and simultaneously communicated to the back-end database in almost real-time via a mobile or wire-bound connection.

6 Power generation forecast and load prediction

This section discusses the authors’ approaches to forecasting power generation and predicting electricity loads associated with individual manufacturing processes.

6.1 Power generation forecast

The flexibility of grid operation and dynamic electricity pricing schemes can be improved by utilizing short-term and long-term power generation forecasts, as highlighted in the literature [111]. In this work, a dynamic index variable indicating the amount of renewable energy in the electricity grid (cf. Section 7) is developed utilizing short-term (24 h) forecasting models for both solar and wind power generation. The models rely on numerical weather data as well as historical wind and solar power generation data collected for the wider area around Trier, Germany, between January 2020 and October 2022.

Prior to forecast model training, the original power generation time series data undergoes simple preprocessing. Abnormal or anomalous values are identified and removed, while missing values are imputed through linear interpolation. This data is then decomposed into its trend, seasonality, and residual components. The resulting trend and residual components, along with numerical weather data, are again recombined to form multivariate time series data at an hourly resolution. Next, a windowing function is applied to create consecutive pairs. Each pair comprises a window of 48 historical data points as a forecasting basis and a subsequent window of 24 data points as forecasting target values. Finally, this set of data window pairs is split into 80% training and 20% test subsets.

In the following, the prediction capability of various neural network architectures, including dense neural network (DNN), long-short-term memory network (LSTM), convolutional neural network (CNN), and hybrid auto-regressive LSTM (AR-LSTM) models, is evaluated. Two metrics,

¹ <https://openems.io/>

Fig. 6 Natural-language interface for querying ontologies

Text Query	<p>What is the average of wind power uri from 20/09/2022 22:00:00 to 20/09/2022 22:30:00?</p> <p>Enter</p>						
	<table border="1"> <tr> <td>Target</td> <td>Query</td> </tr> <tr> <td> <input type="radio"/> is the <input type="radio"/> average <input checked="" type="radio"/> wind power uri <input type="radio"/> from 20/09/2022 22:00:00 to 20/09/2022 22:30:00 </td> <td> Select... Match(onto_match='Command(cypher=AVG(toInteger(split((name),'/')[-1],#)[-1]))', confidence=1.0) Match(onto_match='n_b:ns5_WindPower', confidence=5.0) Match(onto_match='n_ll:ns5_Timestamp') WITH *, split(split(n_ll.uri, '/')[-1],#)[-1] as n_ll_timestamp WITH *, datetime(</td> </tr> <tr> <td colspan="2" style="text-align: right;">Select Query</td> </tr> </table>	Target	Query	<input type="radio"/> is the <input type="radio"/> average <input checked="" type="radio"/> wind power uri <input type="radio"/> from 20/09/2022 22:00:00 to 20/09/2022 22:30:00	Select... Match(onto_match='Command(cypher=AVG(toInteger(split((name),'/')[-1],#)[-1]))', confidence=1.0) Match(onto_match='n_b:ns5_WindPower', confidence=5.0) Match(onto_match='n_ll:ns5_Timestamp') WITH *, split(split(n_ll.uri, '/')[-1],#)[-1] as n_ll_timestamp WITH *, datetime(Select Query	
Target	Query						
<input type="radio"/> is the <input type="radio"/> average <input checked="" type="radio"/> wind power uri <input type="radio"/> from 20/09/2022 22:00:00 to 20/09/2022 22:30:00	Select... Match(onto_match='Command(cypher=AVG(toInteger(split((name),'/')[-1],#)[-1]))', confidence=1.0) Match(onto_match='n_b:ns5_WindPower', confidence=5.0) Match(onto_match='n_ll:ns5_Timestamp') WITH *, split(split(n_ll.uri, '/')[-1],#)[-1] as n_ll_timestamp WITH *, datetime(
Select Query							
Cypher Query	<pre>MATCH (n_b:ns5_WindPower)-[:ns5_has_field]-(n_ll:ns5_Timestamp) WITH *, split(split(n_ll.uri, '/')[-1],#)[-1] as n_ll_timestamp WITH *, datetime({epochmillis: toInteger(n_ll_timestamp)}) as n_ll_datetime WHERE datetime('2022-09-20T22:00:00Z') <= n_ll_datetime <= datetime('2022-09-20T22:30:00Z') RETURN AVG(toInteger(split(split(n_b.uri, '/')[-1],#)[-1])) AS WindPower</pre>						
Cypher Result	<pre>[{ "WindPower": 76.0 }]</pre>						

Fig. 7 OpenEMS deployed on industrial IoT gateways connecting to a spectrum of local hardware such as meters, production machines, and energy meters relaying the gathered information to a central back-end database

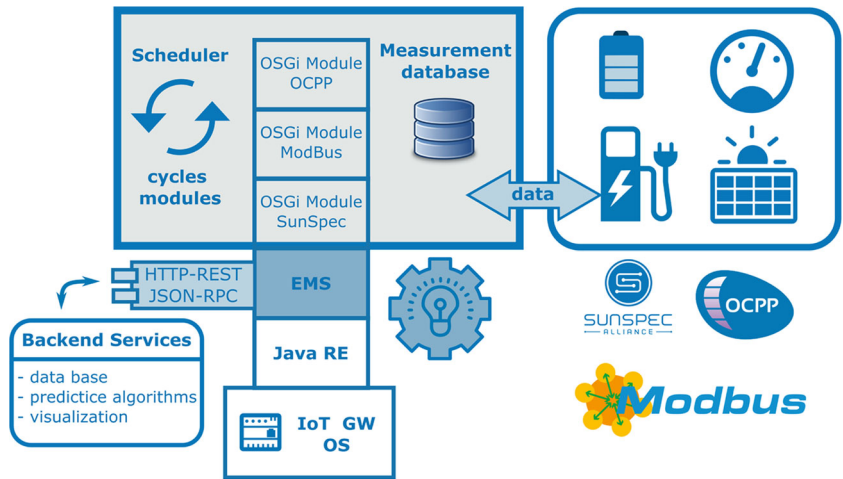
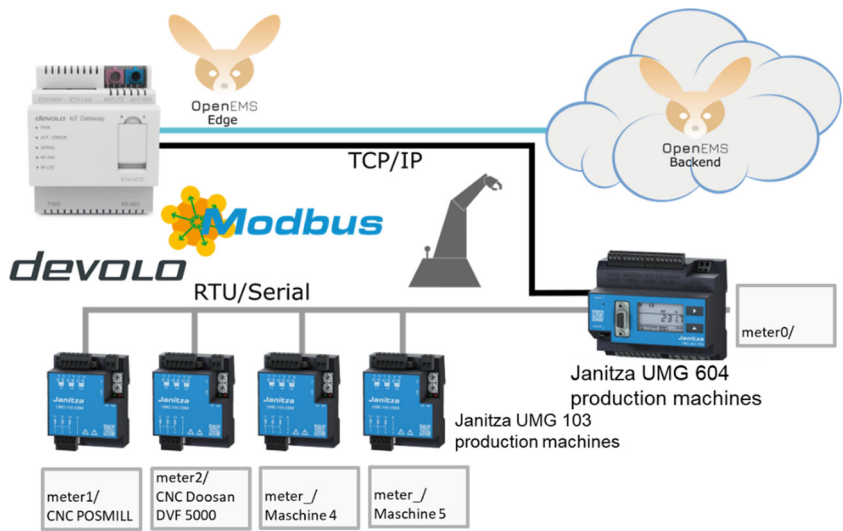


Fig. 8 Typical setup for data acquisition in the environment of the industrial companies used for validation of the proposed demand response system



namely the mean squared error (MSE) and the mean absolute percentage error (MAPE), are used to evaluate these models. Their associated formulae are shown in Eqs. 1 and 2, respectively.

$$MSE = \frac{1}{N} \sum_{n=1}^N (Actual - Predicted)^2 \tag{1}$$

$$MAPE = \frac{1}{N} \sum_{n=1}^N \frac{(Actual - Predicted)}{Actual} * 100 \tag{2}$$

Despite both MSE and MAPE being used to evaluate the prediction capability, it is important to mention that MSE values are used to quantify loss within the context of model weight optimization. Since over-predicted forecasts significantly impact MAPE values, optimizing model parameters based on MAPE would result in models that consistently tend to under-predict. Details on the considered model architectures and their prediction capability are presented in Tables 4 and 5, respectively.

The results in Table 5 indicate that the hybrid AR-LSTM model outperforms all other models and significantly improves prediction regarding both MSE and MAPE for solar power generation. Theoretically, this model can decompose the multi-step prediction into individual time steps. Each time step’s output is fed back into the model, and predictions are conditioned on the previous time step. The AR-LSTM model’s ability to predict solar energy is visualized in Fig. 9, which displays two example predictions in graphical form. The graphs indicate that the model can accurately capture the overall trend of solar power generation but struggles to capture the smaller fluctuations caused by intermittent weather conditions. Interestingly, the CNN model generalizes better than the AR-LSTM in predicting wind power generation. One possible explanation for this is that CNN layers are particularly effective at learning spatial correlations between input features, which may be further enhanced by the inclusion of weather variables. Figure 10 provides two examples of the CNN model’s predictions for wind energy. It is also evident that errors in predicting wind power generation are generally higher than that of its solar-related counterparts. This obser-

Table 4 Overview of deep learning architectures evaluated for short-term power generation forecasting

Model	Input	Hidden	Output
DNN	48	dense-512,264	dense-24
LSTM	48	lstm-32,dense-264	dense-24
CNN-1D	48	conv1D-256, dense-264	dense-24
AR-LSTM	48	lstm_rnn - 32	dense-24

Table 5 Model evaluations for forecasting experiments

	MSE		MAPE	
	Wind	Solar	Wind	Solar
DNN	0.294	0.177	37.6%	26.0%
LSTM	0.293	0.063	39.2%	16.1%
CNN	0.245	0.074	34.3%	17.6%
AR-LSTM	0.286	0.053	38.8%	14.3%

The best results for short-term forecasting are highlighted in boldface

vation could be attributed to the higher intermittent nature of wind. Figure 10 also highlights the impact of wind’s intermittent behavior, as the time series displays frequent fluctuations in contrast to the solar power generation seen in Fig. 9.

Although the above-outlined forecasting works focus primarily on evaluating various deep learning architectures, the results are promising. It is important for readers to understand that this evaluation is not intended to conclusively identify the optimal architecture. As more data is utilized in the future, the findings may change and help enhance the accuracy of the forecasts.

6.2 Load prediction

To enable an energy-flexible production optimization (cf. Section 8) and schedule manufacturing processes of individual or simultaneously manufactured parts based on their respective electricity load profiles, appropriate load prediction models are required. The authors develop load prediction models for four different industrial use cases. While the first two use cases require trained regression models due to the associated high variation in active power and processing time of the manufactured parts, the third and fourth use cases only require one representative load profile each to model their load characteristics. The training data for the regression models have been continuously collected on-site at the manufacturing companies used for validation (cf. Section 9.2), which is still ongoing at the time of writing this article. Consequently, one should see the following results as preliminary, with a chance of further improvements with more training data. The following descriptions are partly based on a work [112] also conducted within the context of this research. The results for the load prediction models developed for the first and second use cases are summarized in Table 6.

The first use case concerns a 5-axis machining center on which different parts are manufactured in small-batch or single-part production. This use case is relevant for company A, which is described in detail in Section 9.2. The goal is to predict the average active power and processing time required to manufacture individual parts. Consequently, separate models are trained for predicting average active power and processing time, which in combination constitute the

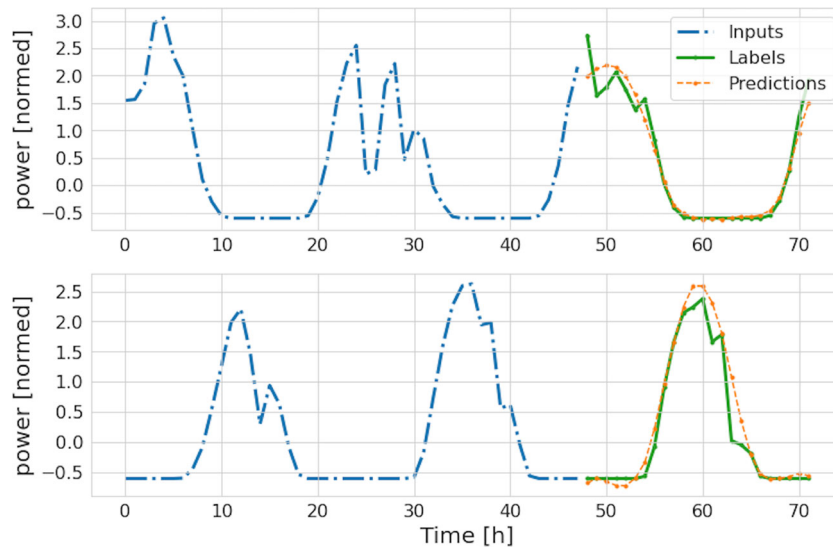


Fig. 9 Samples of one step of solar power forecasting using hybrid auto-regressive long-short-term memory neural networks

load prediction model. A schematic depiction of the load prediction for this use case is given in Fig. 11.

The prediction is always based on the computer-aided manufacturing (CAM) program used to manufacture a part. Aggregated data for the average active power and processing time, as well as twelve features extracted from each of the available 33 different CAM programs, are used for the training processes of the prediction models. Approximately 80% of these data points are utilized for the model training, while the other 20% serve as test data for the performance evaluation. The mean absolute error (MAE) and the mean absolute percentage error (MAPE) between the test data (actual data) and corresponding output data (predicted data) of the models are determined to evaluate the models. While the MAPE

is calculated according to Eq. 2, which is introduced in Section 6.1, the MAE is calculated according to Eq. 3.

$$MAE = \frac{1}{N} \sum_{n=1}^N |Actual - Predicted| \quad (3)$$

Different models are tested for the prediction of the average active power and the processing time. The random forest regression model performs best in predicting the average active power among the tested models. This model uses four of the twelve extracted features and achieves a MAPE of 19.68% and an MAE of 434.97 W on the test data. These results are compared to a basic model that always outputs

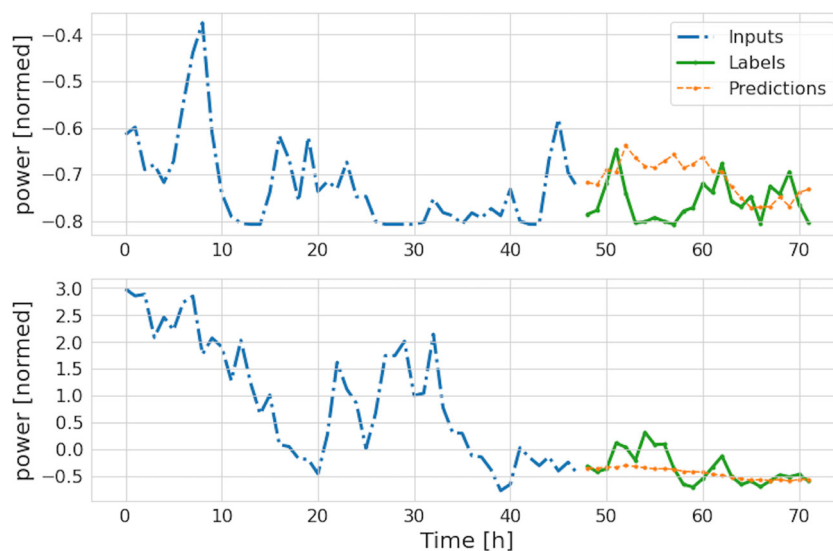


Fig. 10 Samples of one step of wind power forecasting using convolution neural networks

Table 6 Load prediction model evaluation for a 5-axis machining center and a rubber injection molding machine

	MAE		MAPE	
	Best model	Basic model	Best model	Basic model
5-AMC P	434.97 W	627.02 W	19.68%	28.12%
5-AMC t	250.33 s	735.61 s	44.37%	223.62%
RIMM P	471.83 W	884.00 W	4.56%	8.36%
RIMM t	37.64 s	50.93 s	16.73%	23.33%

(Abbreviations: 5 – *AMC P*, 5-axis machining center average active power; 5 – *AMC t*, 5-axis machining center processing time; *RIMM P*, rubber injection molding machine average active power; *RIMM t*, rubber injection molding machine cycle time)

the average active power for all training data, regardless of the inputs. The basic model achieves a MAPE of 28.12% and an MAE of 627.02 W on the test data and is thus significantly outperformed by the trained regression model. A *k*-nearest neighbors regression model performs best in predicting the processing time. This model also uses four of the 12 extracted features and achieves a MAPE of 44.37% and an MAE of 250.33 s on the test data. Another basic model, which always outputs the average processing time of the training data, leads to a MAPE of 223.62% and an MAE of 735.61 s and is substantially outperformed by the *k*-nearest neighbors model.

The authors expect the prediction accuracy to improve with more training data and plan to repeat the training processes once more data is available. The additional use of information on the material and the volume of the finished part is expected to improve the prediction accuracy as well. One way to improve the prediction accuracy for the processing time is to calculate a preliminary processing time based on the location and speed information contained in the CAM programs and use this as an additional feature. Furthermore, the authors determined a subsecond recording of the energy measurement data as a prerequisite to training separate regression models for each type of processing. The aggregation of the predictions of these models to a resulting electricity load profile could further improve the prediction accuracy and granularity. However, the utilized energy data

is gathered with a temporal resolution of one second, which the authors determined as insufficient for this method.

The second use case concerns a rubber injection molding machine on which different rubber molded parts are produced in series. This use case is relevant for company B, which is described in detail in Section 9.2. The objective of this use case is to predict the average active power and cycle times required for individual manufacturing cycles. During one manufacturing cycle, one or multiple parts are manufactured simultaneously using the same negative permanent mold. Similar to the first use case, separate models are trained to predict average active power and cycle time. A schematic depiction of the load prediction for this use case is shown in Fig. 12.

The prediction is based on aggregated process parameters, which are assumed to correspond to the programming parameters of the machine. Aggregated data for the average active power and cycle time, as well as 19 process parameters from each of the available 15 different negative permanent molds, are used to train the prediction models. As done for the model training for the 5-axis machining center, the available data points are split into 80% training data and 20% test data.

The ridge regression model produces the best results in predicting the average active power among the tested models. This model uses all 19 aggregated process parameters and

Use case 1: Load prediction for a 5-axis machining center

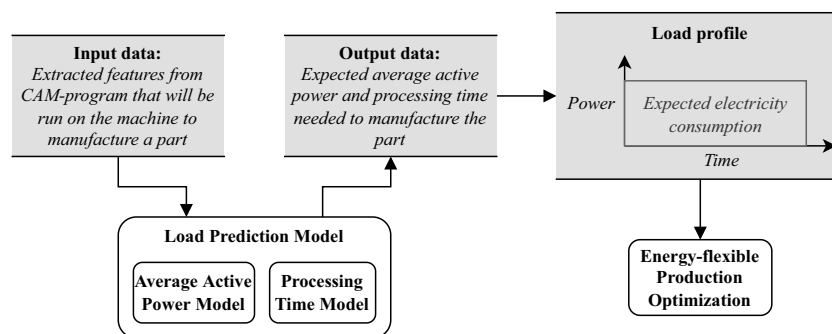


Fig. 11 Schematic depiction of the process for predicting the electricity load resulting from the use of a 5-axis machining center (use case 1, CAM = computer-aided manufacturing)

Use case 2: Load prediction for a rubber injection molding machine

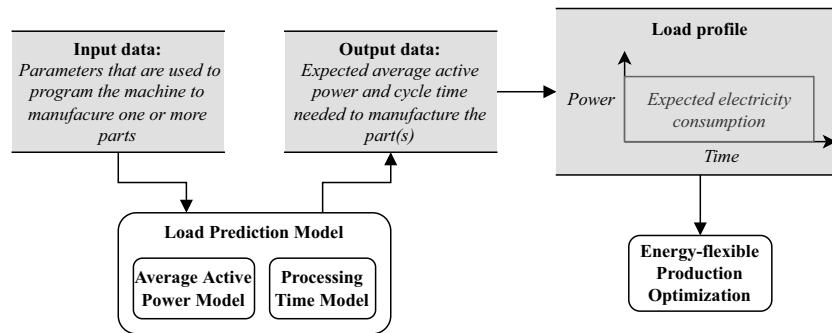


Fig. 12 Schematic depiction of the process for predicting the electricity load resulting from the use of a rubber injection molding machine (use case 2)

achieves a MAPE of 4.56% and an MAE of 471.83 W on the test data. A basic model, which functions analogously to the one for the 5-axis machining center, achieves a substantially higher MAPE of 8.36% and an MAE of 884.00 W on the same test data. The gradient-boosting regression model provided the best results for the cycle time prediction. This model uses two of the 19 aggregated process parameters and achieves a MAPE of 16.73% with an MAE of 37.64 s, which, again, represents a considerable improvement over the basic model, which achieves a MAPE of 23.33% and an MAE of 50.93 s.

Analogously to the first use case, the authors expect additional training data to improve the prediction accuracy further. Consequently, the training will be repeated with a more considerable amount of data once it is available.

The third and fourth use cases concern another rubber injection molding machine and an annealing furnace. These use cases are also relevant for company B (cf. Section 9.2). They do not require regression models since the considered machines are almost exclusively operated with the same parameterization. Consequently, one representative load profile can be determined for each machine using a

clustering-based approach and reused for all manufactured parts. A schematic depiction of the load prediction for use cases 3 and 4 is given in Fig. 13.

7 Green Electricity Index (GEI)

The Green Electricity Index (GEI) is a measure indicating the amount of renewable energy in the electricity grid in comparison to the demand for electricity in Germany. The value of the GEI indicates the percentage of German electricity that comes from renewable energy sources (solar, wind, water, and biomass) at a given time. It is calculated and updated daily based on data provided by transmission network operators. Additionally, short-term forecasts of renewable energy sources, as discussed in Section 6.1, are utilized to compute GEI values for the next hours up until a day. Values for the upcoming 72 h are determined using only weather data, while such lying more than 72 h in the future are increasingly based on climate data. For GEI values more than ten days in the

Use cases 3 and 4: Load prediction for a rubber injection molding machine or an annealing furnace

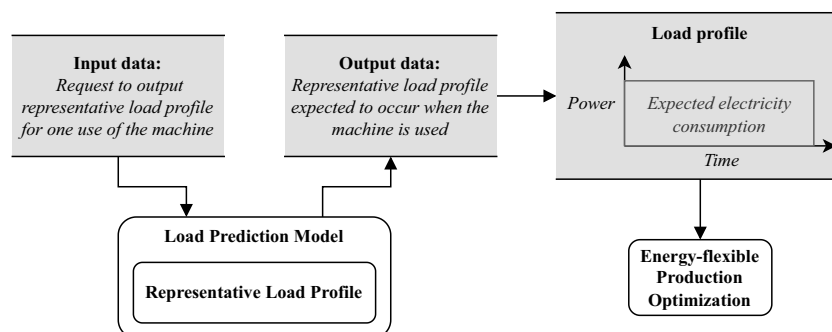


Fig. 13 Schematic depiction of the process for predicting the electricity load resulting from the use of a rubber injection molding machine or an annealing furnace (use cases 3 and 4)

future only climate models are used. Furthermore, the index calculation involves a measure of future uncertainty, which is learned automatically by including forecasting errors in the immediate past.

The GEI is calculated based on a complex graph with billions of nodes. Each node can represent, for example, a weather forecast for a specific location or the switching state of the power grid. As changes to one node impact every other node in the graph, the state of each node is crucial for determining the GEI and its impact on the power supply at a location. An important basis for the value of the GEI is the structure of the power grid and its nodes. Using pass-through elements as placeholders reduces the complexity of the power grid and is referred to as the Map-Reduce algorithm [113]. The structural data of the power grid is used as a database for the GEI. The GEI is issued as a forecast for the coming days and is based on a timetable of the power grid that is constantly updated by grid operators. Future switching states of the power grid can be learned from the past, and under specific conditions, certain switching states are more likely than others. The historical data the GEI is based on is crucial, and this data can come from various sources, such as weather forecasts, power consumption, and power generation. To improve the GEI, new data sources are continuously sought and evaluated. Equation 4 defines the computation of a GEI value.

$$GEI_{i,t} = 100 \cdot \left[\frac{R_{i,t}}{D_{i,t}} \cdot X_{i,t} + 1 - X_{i,t} \right] \quad (4)$$

with $\forall i, t \frac{R_{i,t}}{D_{i,t}} \geq 1 - X_{i,t}, X_{i,t} \in \{0, 1\}$

If the ratio of generated renewable energy R and electricity consumption D is 100% or higher, the GEI equates to 100, indicating that the electricity supply is entirely sourced from renewable sources. In this case, X is set to 0. Otherwise, X is set to 1, and GEI values indicate the percentage of electricity sourced from renewable sources, computed as the ratio of renewable energy R generation and electricity consumption D . Indices i and t represent graph node number and time, respectively.

GEI values are used to support short-term production planning decisions and identify potential areas for improvement. For example, suppose the GEI indicates much renewable energy available between 8 AM and 5 PM, with a peak at 1 PM. In that case, the production can be adjusted so that the most energy-consuming production runs during the peak at 1 PM and other less consuming productions run before and after. It would result in both increasing the use of renewable energy and lowering the cost of production. Thus, the GEI can also be considered a proxy dynamic pricing signal to motivate load reduction and shifting. In a broader context, the GEI is also an essential tool for monitoring the progress

of the energy transition in Germany. As noted at the beginning of this work, the German government has set a target to increase the share of renewable energy in the electricity supply to 80% by 2030. It would be supported by indicators like the GEI, which allows tracking each renewable energy source's contribution to electricity production and monitoring its development over time.

Although the algorithm of the GEI has been highly improved over time, it is essential to note that the GEI is based on publicly available data such as weather and climate data. Therefore, it should be used as a reference, not as a definitive measure of the percentage of renewable energy in the grid.

8 Production optimization under dynamic electricity pricing

The following section describes and compares the approaches for energy-flexible production optimization. It should be noted that the usable flexibility results exclusively from the possibility of shifting the starting times of individual manufacturing processes. A simplified schematic depiction of optimizing the production plan of three processes handled by a single machine is shown in Fig. 14. One can see the desired result: Processes are preferentially carried out at times when electricity prices are low, and the processes with the highest electricity load are shifted to times when electricity prices are lowest. The considered processes do not provide other types of flexibility, such as different operation modes, which would lead to different load profiles. The first developed approach uses mixed-integer linear programming (MILP) [114], while the second approach is based on reinforcement learning (RL) [115].

8.1 Mixed-integer linear programming approach

In this approach, the objective is to minimize the electricity costs for the considered planning horizon by temporally shifting flexible manufacturing processes to periods with lower electricity prices in the presence of dynamic electricity prices. The input data for the optimization are the electricity price profile, the predicted electricity load profiles for all manufacturing processes to be carried out, the planning horizon, and the machine type to be used for each process. Optionally, forecasted power generation for local generation capacities, such as photovoltaic systems, can also be considered. The decision variables are the starting times of the manufacturing processes and the respective machines to be used. The starting times can optionally be fixed to a specific value for each process individually to enable user interaction. Defined constraints ensure that individual manufacturing

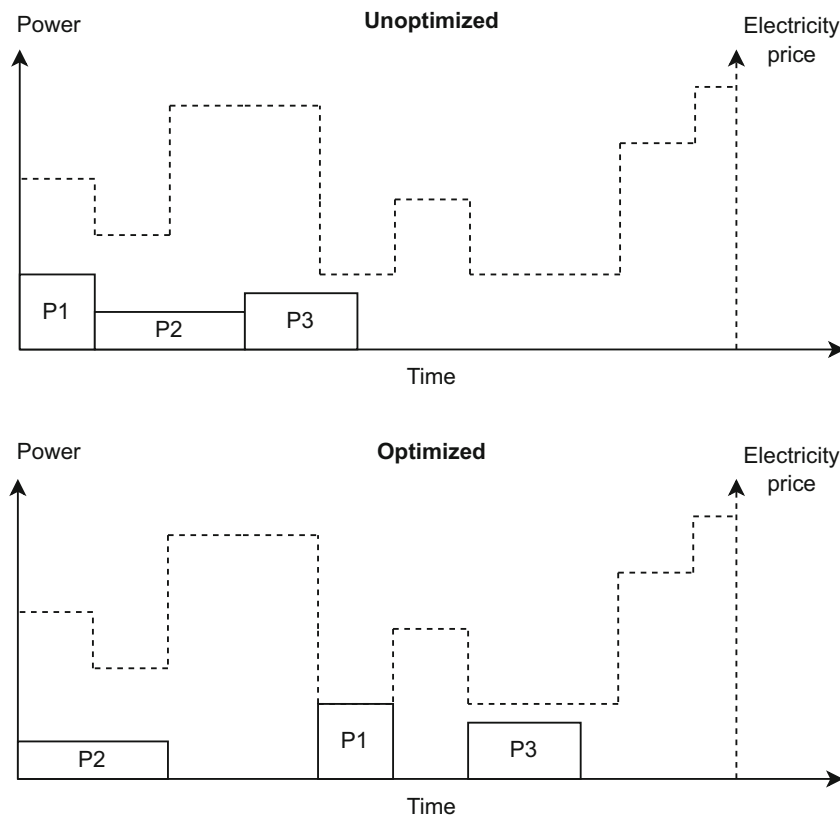


Fig. 14 Simplified schematic depiction of the production optimization under dynamic electricity pricing for a single machine and three processes P1, P2, and P3. The top and bottom sub-figures show the production plan before and after optimization, respectively

processes are not split up and that multiple processes are not simultaneously executed on the same machine.

8.2 Reinforcement learning approach

The second method for production optimization is based on RL. According to Mitchell [115], in RL, an agent learns to optimize its actions to reach objectives by maximizing the cumulated reward obtained from its environment. RL is applied to a multi-objective production scheduling problem to minimize electricity costs and makespan. The methodology is based on the work by Tassel et al. [116], who focus on minimizing the makespan. This research enhances their approach with DR and flexibility aspects. A previous work [117], also conducted within the context of this research, provides more details on the method and implementation than the following paragraphs. The selected RL algorithm is the proximal policy optimization (PPO), which is a deep-reinforcement-learning algorithm based on the actor-critic method, enabling a reliable learning performance for the agent [118].

The following paragraphs explain the three main components of the designed RL environments: action space, observation space, and reward functions. The action space

includes the set of jobs that can be assigned to the machines and the option to keep machines idle. The observation space includes time-related attributes, a Boolean attribute for the eligibility of an action employed by Tassel et al. [116], and electricity-related attributes, which are integrated into the environment. Time-related attributes are the percentage of finished operations and jobs, remaining time for operations and jobs, required time until a machine is free for the next job's operation to be scheduled, idle time since the last performed operation, and total idle time for a job. The Boolean attribute enables the elimination of ineligible actions, such as selecting already assigned jobs or occupied machines. The electricity-related attributes are the average electricity load and dynamic electricity prices. The electricity load data of each operation are generated uniformly over the interval between 1 and 20 kW, and German day-ahead electricity prices are obtained from [119] for the days between 11.10.2022 and 13.10.2022.

Two reward functions help to choose between operating and keeping a machine idle based on dynamic electricity prices. Equations 5 and 6 present the reward functions for selecting a job and keeping a machine idle, respectively. Both functions include objectives of minimizing electricity costs and makespan, which are weighted using a parameter α rang-

ing between 0 and 1. This parameter α enables customization in production planning regarding the manufacturer’s preferences towards electricity costs and makespan minimization. Within this context, the electricity cost for operation i of a job j is calculated by multiplying the operation’s electricity consumption, i.e., the product of the average electricity load and the processing time of operations, by the electricity prices that apply during its processing time. Since the processing time can coincide with multiple pricing periods, for each pricing period t , the electricity load is multiplied by the share of the processing time, which falls into the period t . Then, all the associated costs for the operation in these periods are summed up. Equation 5 associates electricity cost with a negative reward, indicating more penalty for selecting a job with high electricity consumption. At the same time, progressing the production schedule is rewarded by adding the processing time positively to the equation and punished by subtracting the resulting idle time for machine m . Equation 6 gives electricity prices as a positive reward to the agent for keeping the machine idle in periods with high electricity prices. The reward functions are defined as

$$Reward_1 = \alpha(1 - \sum_{t \in t_p} (e_{ij} \pi^t p_{ij}^t)) + (1 - \alpha)(p_{ij} - idle_m) \quad (5)$$

$$Reward_2 = -\alpha(1 - \sum_{t \in t_{idle}} \pi^t) + (1 - \alpha)(-idle_m) \quad (6)$$

where e_{ij} represents the average electricity load, π^t the electricity price, p_{ij} the processing time, p_{ij}^t the part of p_{ij} falling into the pricing period t , $idle_m$ is the idle time for machine m , t_p are the pricing periods during p_{ij}^t , and t_{idle} is the pricing periods during $idle_m$. The RL agent is trained by employing the same training time (10 min) and hyperparameters as Tassel et al. [116].

Figure 15 demonstrates the achieved results during the training with data of a scheduling case from Taillard [120]. During the beginning of the training, electricity costs are low since the agent still has tasks to complete. After their completion, the agent begins to reduce costs. When the optimal

results are achieved, the results remain stable until the end of the training. The trade-off situation between the objectives is noticeable concerning different weights. As designed, lower α values lead to higher electricity costs and shorter makespan values.

8.3 Approach comparison

While the MILP-based approach is implementable with relatively low effort and can potentially find exactly optimal production schedules, its scalability is limited. This approach could be a reasonable solution for smaller production environments and single optimization objectives. However, for larger factories with high numbers of machines and parts to manufacture, long planning horizons, and multiple optimization objectives, the exact solving of the optimization problem may not be efficient or feasible at all [121]. Furthermore, the optimization problem, parameters, decision variables, and constraints may vary for different manufacturers’ production environments and optimization objectives. Consequently, an individual program for each production environment may have to be implemented, which relativizes the ease of implementation argument compared to different, one-size-fits-all approaches.

In contrast, the RL method provides generic and scalable solutions for complex and dynamic production environments. The agent adapts the solution step-by-step to the environment by learning from the interaction with the surroundings, so the model does not have to be changed as the environment changes [8]. However, the initial modeling of the RL environment and training requires significant time and effort. Due to the outlined advantages, the RL approach is currently being applied as the energy-flexible production optimization component of the proposed system onto actual use case data, as further explained in Section 9.

Table 7 presents an additional overview summarizing the qualitative differences between the MILP and RL approaches regarding their features presented in this section. Time complexity refers to variations in the required time for finding a

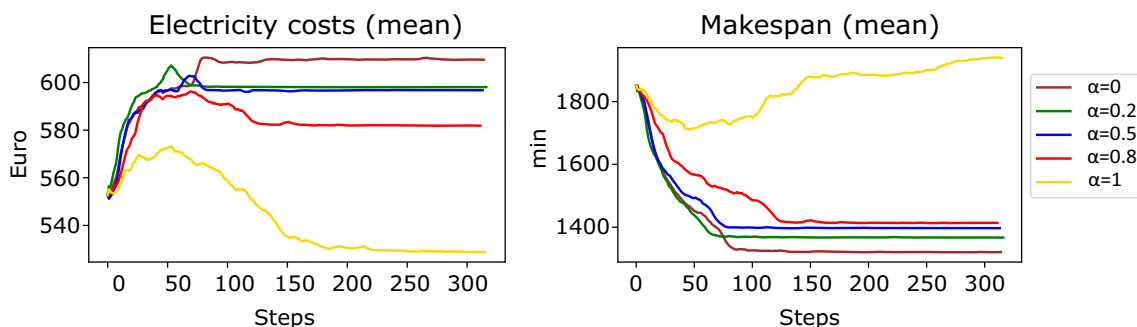


Fig. 15 Electricity costs and makespan for different weighting values during the training of the RL agent

Table 7 Comparison of two optimization approaches based on their qualitative features

Features	Mixed-integer linear programming (MILP)	Reinforcement learning (RL)
Optimization objective(s)	Electricity cost minimization	Electricity cost and makespan minimization
Problem environment	Simple and fixed production environment	Complex and variable production environment
Time complexity	Increases sharply with production environment complexity	Increases moderately with production environment complexity
Modeling effort	Low	High
Scalability	Limited	High
Optimal solution accuracy	Exact optimal solution	Depends on learning performance of agent(s)

solution in relation to the production environment complexity.

9 Implementation and validation

This section shows implementation-related information of various studies presented in earlier parts of this work. Additionally, results and observations of the holistic validation of all different aspects of the proposed DR system within the context of two manufacturing companies are provided.

9.1 Implementation

The neural network models described in Section 6.1 are developed and trained using TensorFlow, an open-source library focusing on training and inference of deep learning models [122]. Furthermore, the decomposition of time series data into its additive components of trend, seasonality, and residuals is obtained using the seasonal decompose method of the Python library statsmodels [123]. All other data preprocessing described in Section 6.1 is done using the Pandas library [124].

For the implementation of the load prediction models described in Section 6.2, the authors use the open-source Python library scikit-learn [125]. This library provides an extensive collection of various machine learning models and data processing tools.

The MILP-based production optimization, outlined in Section 8.1, is also Python-based and uses the open-source Pyomo software package [126]. It provides a high-level optimization modeling language and a wide range of tools to solve different types of optimization problems.

The RL approach described in Section 8.2 is implemented using Python, with the libraries Numpy [127] and Pandas [124] for structuring the data, OpenAI Gym [128] for creating the RL environment, and RLLib [129] for implementing PPO with Tensorflow [122] as the learning platform. The training is conducted on a computer with two Nvidia Titan RTX GPUs and an AMD Ryzen 9 3950X CPU.

9.2 Scenario description

The proposed system is validated within the production environments of two manufacturing companies, anonymously referred to as A and B in this work. Company A primarily produces metal components through forging, cutting, and lathing processes, characterized by a high degree of manual work. Orders traverse their computerized-numerical-control machine fleet in a mostly job-shop manner. The company operates in a single-shift mode. The load prediction for a 5-axis machining center, which represents the first use case described in Section 6.2, is applied for company A. Com-

pany B produces rubber parts using injection molding and annealing machinery. Generally, their orders traverse those machines in a flow-shop and partially automated manner. The company operates in three shifts. The load prediction models, presented in Section 6.2 for the second, third, and fourth use cases, are relevant for company B. These are applied for two rubber injection molding machines and one annealing furnace. Due to the high electricity consumption of some of their processes, both companies seek to economize energy-related costs and increase the use of renewable energy. Furthermore, they identified their need for intuitively comprehensible visual indicators to monitor the information mentioned above.

The GEI is proposed as a comprehensible indicator for these companies to monitor electricity prices and renewable energy share. As mentioned in Section 7, computing the index values requires short-term renewable energy forecasts, weather data, and climate data. Once the index values are computed, communication of the GEI values to the companies is facilitated by the semantic middleware described in Section 4. The semantic middleware also facilitates the acquisition of weather and historically generated data required for developing and training the forecasting models as described in Section 6.1.

Currently, the proposed system is being applied in a prototypic stage in company A's production environment. At this point, the system's benefit for company A is expected to be higher than for B. It is based on the fact that company A conducts fine-grained process scheduling daily and has inherently high planning flexibility due to its great product diversity and mostly project-based order planning and processing approach. For this reason, the proposed system is manually populated with data collected as described in Section 3. It suggests scheduling solutions that are optimized concerning provided user preferences towards the trade-off between makespan and electricity cost minimization. This way, it supports planning personnel in daily production scheduling activities. Planning personnel can deviate from proposed schedules using an interactive system interface and is informed on the impact of deviations on energy and emission-related measures.

In large part, company B lacks the flexibility A has at its disposal. One might attribute this to the fact that B plans its processes significantly further in advance and that rescheduling activities introduce considerable effort. However, it is assumed that company B will benefit significantly within the context of annealing resources, which can be scheduled more flexibly than their remaining machines. These machines exhibit a distinctively high load during the heat-up phase and a relatively low load in later phases of constant heat treatment of parts. This can be exploited by temporally shifting the initial heat-up-related loads into preferred periods. Therefore, the proposed system will be applied for company B analogously.

gously as it is for A while likely being limited to the context of annealing resources.

Regarding both companies' machinery and processes, it is assumed that higher digitization degrees, realized by standardized control and data interfaces and a manufacturing execution system, greatly benefit the system's application. Primarily, this would simplify the automatic collection of data required by the system. Secondly, this would support the autonomous introduction of system-proposed schedules in automated manufacturing environments. Thinking ahead, one can only begin to anticipate the impact of further system-external circumstances to be successively discovered during post-prototypical system application. For instance, occupational safety regulations might require personnel to start machines, which must operate under constant supervision manually. Consequently, this and potentially other factors might limit the extent of autonomous system application strategies.

10 Conclusions and outlook

This paper presents a novel implementation of a demand response system for manufacturing electricity consumers involving heterogeneous data sources from multiple domains and applications. It proposes a semantic middleware that enables seamless data exchange and facilitates a shared understanding of the data among the applications. The shared understanding of the semantic middleware works by reusing and linking ontologies from multiple established domains. The resulting ontology linking is used as the semantic description of data from multiple sources across domains through the semantic uplift process. This paper also describes the data flows through the semantic middleware that illustrates the data exchange among four main components, i.e., power generation forecast, load prediction, Green Electricity Index (GEI), and energy-flexible production optimization. To collect power generation and consumption data for the training of forecasting and predictive models, the semantic middleware is supported by a real-time data acquisition system developed by the authors based on the open-source solution OpenEMS.

Overall, the authors have developed multiple neural network models with different architectures for power generation forecasts, as well as multiple machine learning models for load prediction. The evaluations of these models have shown that convolutional neural networks are best suited for wind power generation forecasting, and hybrid autoregressive long short-term memory networks are optimal for solar power generation forecasting. Additionally, the evaluations of the load prediction models indicate that different machine learning models perform best for different machines. Looking ahead, the authors see great potential for

further improving these models by incorporating additional data sources and refining the model architectures to better capture the complexities of power generation and consumption.

The production optimization uses the GEI as input alongside predicted loads, user preferences, and production process parameters. The GEI represents the proportion of renewable electricity in the mix procured for production processes and indicates the electricity price level at the given time. It is calculated using forecasted power generation data. Furthermore, this research develops two models for production optimization based on mixed-integer linear programming and reinforcement learning. After conducting experiments to compare both alternatives, the authors decided to integrate the reinforcement learning-based approach into the system due to its capability to provide generic and scalable solutions for complex and dynamic production environments. This paper elaborates on implementing and validating the proposed system involving a utility company and two small to medium-sized manufacturing enterprises.

To summarize, the main outcomes of this work are the following:

- A concept and implementation of a system based on a semantic middleware and containing components dedicated to the prediction and optimization workloads to support production planning processes.
- An index measure making the share of renewable energy in a procured mix and the associated cost processable for humans and optimization processes.
- Insights into the practical applicability of the proposed system in an exemplary high-planning-flexibility environment.

The system can be improved by integrating more renewable energy sources, such as biomass and hydropower, into the power forecast module. Moreover, on the consumption side, industrial adoption can be increased by expanding the validation scenarios into more industrial types such as mass customization and process industry.

Author contribution The contribution of every author is specified within the following CRediT authorship contribution statement.

Hendro Wicaksono: conceptualization, methodology, writing—original draft, supervision, funding acquisition. Martin Trat: conceptualization, methodology, software, writing—original draft, supervision. Atit Bashyal: methodology, software, formal analysis, writing—original draft. Tina Boroukhian: methodology, software, formal analysis, writing—original draft. Mine Felder: methodology, software, formal analysis, writing—original draft. Mischa Ahrens: methodology, software, formal analysis, writing—original draft. Janek Bender: methodology, formal analysis. Sebastian Groß: methodology, software, formal analysis. Daniel Steiner: software, formal analysis. Christoph July: methodology, software, formal analysis, writing—original draft. Christoph Dorus: methodology, software, formal analysis, writing—original draft. Thorsten Zoerner: methodology, formal analysis.

Funding Open Access funding enabled and organized by Projekt DEAL. This work has been partially funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) through the research project “Delfine - Dynamic demand response system for sustainable production through information technology networking for efficient energy use, marketing and generation” that addresses the public call for “Digitization of the energy transition” formulated within the 7th energy research program (grant number 03EI4018C).

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Package EU (2015) A framework strategy for a resilient energy union with a forward-looking climate change policy (document 1). av
2. Albadi MH, El-Saadany EF (2007) Demand response in electricity markets: an overview, in: 2007 IEEE power engineering society general meeting. IEEE, p 1–5
3. Mourtzis D (2022) Chapter 4 - the mass personalization of global networks. In: Mourtzis D (ed) Design and operation of production networks for mass personalization in the era of cloud technology. Elsevier, p 79–116. <https://doi.org/10.1016/B978-0-12-823657-4.00006-3>. <https://www.sciencedirect.com/science/article/pii/B9780128236574000063>
4. Umweltbundesamt, Stromverbrauch (2023). <https://www.umweltbundesamt.de/daten/energie/energieverbrauch-nach-energetraegern-sektoren#entwicklung-des-endenergieverbrauchs-nach-sektoren-und-energetragern>
5. Statista (2023) Anteil am stromverbrauch nach sektoren in deutschland 2021. <https://de.statista.com/statistik/daten/studie/236757/umfrage/stromverbrauch-nach-sektoren-in-deutschland/>
6. IEA (2022) International Energy Agency, Global energy review: CO2 emissions in 2021. <https://www.iea.org/reports/global-energy-review-co2-emissions-in-2021-2>
7. Lu R, Li Y-C, Li Y, Jiang J, Ding Y (2020) Multi-agent deep reinforcement learning based demand response for discrete manufacturing systems energy management. *Appl Energy* 276:115473. <https://doi.org/10.1016/j.apenergy.2020.115473>
8. Huang X, Hong SH, Yu M, Ding Y, Jiang J (2019) Demand response management for industrial facilities: a deep reinforcement learning approach. *IEEE Access* 7:82194–82205. <https://doi.org/10.1109/ACCESS.2019.2924030>
9. Kuzlu M, Cali U, Sharma V, Güler Ö (2020) Gaining insight into solar photovoltaic power generation forecasting utilizing explainable artificial intelligence tools. *IEEE Access* 8:187814–187823
10. Javed F, Arshad N, Wallin F, Vassileva I, Dahlquist E (2012) Forecasting for demand response in smart grids: an analysis on use of anthropologic and structural data and short term multiple loads forecasting. *Appl Energy* 96:150–160
11. Ma S, Zhang Y, Liu Y, Yang H, Lv J, Ren S (2020) Data-driven sustainable intelligent manufacturing based on demand response for energy-intensive industries. *J Clean Prod* 274:123155
12. May G, Stahl B, Taisch M (2016) Energy management in manufacturing: toward eco-factories of the future - a focus group study. *Appl Energy* 164:628–638. <https://doi.org/10.1016/j.apenergy.2015.11.044>
13. Satuyeva B, Sauranbayev C, Ukaegbu IA, Nunna HK (2019) Energy 4.0: towards IoT applications in Kazakhstan. *Procedia Comput Sci* 151:909–915
14. Gronier T, Franquet E, Gibout S (2022) Platform for transverse evaluation of control strategies for multi-energy smart grids. *Smart Energy* 7:100079
15. Mourtzis D, Boli N, Xanthakis E, Alexopoulos K (2021) Energy trade market effect on production scheduling: an industrial product-service system (IPSS) approach. *Int J Comput Integr Manuf* 34(1):76–94
16. Mourtzis D, Angelopoulos J, Panopoulos N (2022) Smart grids as product-service systems in the framework of Energy 5.0-a state-of-the-art review. *Green Manuf Open* 1(1):5
17. Yang J, Xiao W, Jiang C, Hossain MS, Muhammad G, Amin SU (2018) AI-powered green cloud and data center. *IEEE Access* 7:4195–4203
18. Schwartz R, Dodge J, Smith NA, Etzioni O (2020) Green AI. *Commun ACM* 63(12):54–63
19. Katasonov A, Kaykova O, Khriyenko O, Nikitin S, Terziyan V (2008) Smart semantic middleware for the Internet of Things. In: International conference on informatics in control, automation and robotics, vol 2. ScitePress, pp 169–178
20. Terziyan V, Kaykova O, Zhovtobryukh D (2010) UbiRoad: semantic middleware for context-aware smart road environments. In: (2010) Fifth international conference on internet and web applications and services. IEEE p 295–302
21. Cruz IF, Xiao H et al (2005) The role of ontologies in data integration. *Eng Intell Syst Electric Eng Commun* 13(4):245
22. Ferchichi A, Bigand M, Lefebvre H (2008) An ontology for quality standards integration in software collaborative projects. In: First international workshop on model driven interoperability for sustainable information systems. Montpellier, France, sn, pp 17–30
23. Mourtzis D (2021) Towards the 5th industrial revolution: a literature review and a framework for process optimization based on big data analytics and semantics. *J Mach Eng* 21(3)
24. Borsato M (2017) An energy efficiency focused semantic information model for manufactured assemblies. *J Clean Prod* 140:1626–1643
25. Modoni GE, Doukas M, Terkaj W, Sacco M, Mourtzis D (2017) Enhancing factory data integration through the development of an ontology: from the reference models reuse to the semantic conversion of the legacy models. *Int J Comput Integr Manuf* 30(10):1043–1059
26. Adamczyk BS, Szejka AL, Júnior OC (2020) Knowledge-based expert system to support the semantic interoperability in smart manufacturing. *Comput Ind* 115:103161
27. Esnaola-Gonzalez I, Díez FJ, Berbakov L, Tomasevic N, Štorek P, Cruz M, Kirketerp P (2018) Semantic interoperability for demand-response programs: respond project’s use case. In: 2018 Global internet of things summit (GIoTS). IEEE p 1–6

28. Energy S (2019) The universal smart energy framework. Tech, Rep
29. Collective SE (2013) An introduction to the universal smart energy framework. Smart Energy Collective. Available online: https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group3_summary.pdf. Accessed 21 Sept 2018
30. Hippolyte J-L, Howell S, Yuce B, Mourshed M, Sleiman HA, Vinyals M, Vanhée L, Ontology-based demand-side flexibility management in smart grids using a multi-agent system. In: (2016) IEEE international smart cities conference (ISC2). IEEE, p 1–7. <https://doi.org/10.1109/ISC2.2016.7580828>
31. Howell SK, Wicaksono H, Yuce B, McGlinn K, Rezgui Y (2018) User centered neuro-fuzzy energy management through semantic-based optimization. IEEE Trans Cybern 49(9):3278–3292
32. Wicaksono H, Dobрева P, Häfner P, Rogalski S (2015) Methodology to develop ontological building information model for energy management system in building operational phase. In: International joint conference on knowledge discovery, knowledge engineering, and knowledge management. Springer, p 168–181
33. Howell S, Rezgui Y, Beach T (2017) Integrating building and urban semantics to empower smart water solutions. Autom Constr 81:434–448
34. Wicaksono H, Jost F, Rogalski S, Ovtcharova J (2014) Energy efficiency evaluation in manufacturing through an ontology-represented knowledge base. Intell Syst Account Financ Manage 21(1):59–69
35. Wicaksono H, Schubert V, Rogalski S, Laydi YA, Ovtcharova J (2012) Ontology-driven requirements elicitation in product configuration systems. In: Enabling manufacturing competitiveness and economic sustainability: proceedings of the 4th international conference on changeable, agile, reconfigurable and virtual production (CARV2011), Montreal, Canada, 2–5 October 2011. Springer, pp 63–67
36. Li Y, Rezgui Y, Kubicki S (2020) An intelligent semantic system for real-time demand response management of a thermal grid. Sustain Cities Soc 52:101857
37. Antonopoulos I, Robu V, Couraud B, Kirli D, Norbu S, Kiprakis A, Flynn D, Elizondo-Gonzalez S, Wattam S (2020) Artificial intelligence and machine learning approaches to energy demand-side response: a systematic review. Renew Sustain Energy Rev 130:109899. <https://doi.org/10.1016/j.rser.2020.109899>. <https://www.sciencedirect.com/science/article/pii/S136403212030191X>
38. Hong T, Pinson P, Wang Y, Weron R, Yang D, Zareipour H (2020) Energy forecasting: a review and outlook. IEEE Open Access J Power Energy 7:376–388. <https://doi.org/10.1109/OAJPE.2020.3029979>
39. Hong W-C (2009) Electric load forecasting by support vector model. Appl Math Modell 33(5):2444–2454. <https://doi.org/10.1016/j.apm.2008.07.010>. <https://www.sciencedirect.com/science/article/pii/S0307904X08001844>
40. Li J, Lei Y, Yang S (2022) Mid-long term load forecasting model based on support vector machine optimized by improved sparrow search algorithm. Energy Rep 8 491–497. iCPE 2021 - The 2nd international conference on power engineering. <https://doi.org/10.1016/j.egy.2022.02.188>. <https://www.sciencedirect.com/science/article/pii/S2352484722004358>
41. Fentis A, Bahatti L, Mestari M, Chouri B (2017) Short-term solar power forecasting using support vector regression and feed-forward nn. In: 2017 15th IEEE international new circuits and systems conference (NEWCAS), p 405–408. <https://doi.org/10.1109/NEWCAS.2017.8010191>
42. Zendejboudi A, Baseer M, Saidur R (2018) Application of support vector machine models for forecasting solar and wind energy resources: a review. J Clean Prod 199:272–285. <https://doi.org/10.1016/j.jclepro.2018.07.164>. <https://www.sciencedirect.com/science/article/pii/S095965261832153X>
43. Dudek G (2022) A comprehensive study of random forest for short-term load forecasting. Energies 15(20). <https://doi.org/10.3390/en15207547>. <https://www.mdpi.com/1996-1073/15/20/7547>
44. Lahouar A, Hadj Slama JB (2015) Random forests model for one day ahead load forecasting. <https://doi.org/10.1109/IREC.2015.7110975>
45. Raj MG, Pradip C, Saju N, Sangeetha SVT (2021) Random forest-based method for micro grid system in energy consumption prediction. J Phys Conf Ser 1964(5):052002. <https://doi.org/10.1088/1742-6596/1964/5/052002>
46. Khalyasmaa A, Eroshenko SA, Chakravarthy TP, Gasi VG, Bollu SKY, Caire R, Atluri SKR, Karrolla S (2019) Prediction of solar power generation based on random forest regressor model. In: 2019 International multi-conference on engineering, computer and information sciences (SIBIRCON), pp 0780–0785. <https://doi.org/10.1109/SIBIRCON48586.2019.8958063>
47. Lahouar A, Hadj Slama JB (2017) Hour-ahead wind power forecast based on random forests. Renew Energy 109(2017):529–541. <https://doi.org/10.1016/j.renene.2017.03.064>. <https://www.sciencedirect.com/science/article/pii/S0960148117302550>
48. Di Piazza A, Di Piazza M, La Tona G, Luna M (2021) An artificial neural network-based forecasting model of energy-related time series for electrical grid management. Math Comput Simul 184(2021):294–305. eLECTRIMACS 2019 Engineering - Modelling and computational simulation for analysis and optimisation in electrical power engineering. <https://doi.org/10.1016/j.matcom.2020.05.010>. <https://www.sciencedirect.com/science/article/pii/S0378475420301695>
49. Lee D (2021) Low-cost and simple short-term load forecasting for energy management systems in small and middle-sized office buildings. Energy Explor Exploit 39(2):637–656
50. Kalogirou SA (2001) Artificial neural networks in renewable energy systems applications: a review. Renew Sustain Energy Rev 5(4):373–401. [https://doi.org/10.1016/S1364-0321\(01\)00006-5](https://doi.org/10.1016/S1364-0321(01)00006-5). <https://www.sciencedirect.com/science/article/pii/S1364032101000065>
51. Runge J, Zmeureanu R (2019) Forecasting energy use in buildings using artificial neural networks: a review. Energies 12(17). <https://doi.org/10.3390/en12173254>. <https://www.mdpi.com/1996-1073/12/17/3254>
52. Jiao R, Huang X, Ma X, Han L, Tian W (2018) A model combining stacked auto encoder and back propagation algorithm for short-term wind power forecasting. IEEE Access 6:17851–17858
53. Niu Z, Yu Z, Tang W, Wu Q, Reformat M (2020) Wind power forecasting using attention-based gated recurrent unit network. Energy 196:117081
54. Wang H, Lei Z, Zhang X, Zhou B, Peng J (2019) A review of deep learning for renewable energy forecasting. Energy Convers Manag 198:111799
55. Chodakowska E, Nazarko J, Nazarko L (2021) Arima models in electrical load forecasting and their robustness to noise. Energies 14(23). <https://doi.org/10.3390/en14237952>. <https://www.mdpi.com/1996-1073/14/23/7952>
56. Yukseltan E, Yucekaya A, Bilge AH (2020) Hourly electricity demand forecasting using Fourier analysis with feedback. Energy Strat Rev 31:100524. <https://doi.org/10.1016/j.esr.2020.100524>. <https://www.sciencedirect.com/science/article/pii/S2211467X20300778>
57. Czapaj R, Kamiński J, Sołtysik M (2022) A review of autoregressive methods applications to short-term demand forecasting in power systems. Energies 15(18). <https://doi.org/10.3390/en15186729>. <https://www.mdpi.com/1996-1073/15/18/6729>

58. Abdelaziz A, Rahman M, El-Khayat M, Hakim M (2012) Short term wind power forecasting using autoregressive integrated moving average modeling
59. Pasari S, Shah A (2020) Time series auto-regressive integrated moving average model for renewable energy forecasting. In: Sangwan KS, Herrmann C (eds) *Enhancing Future Skills and Entrepreneurship*. Springer International Publishing, Cham, pp 71–77
60. Kant G, Sangwan KS (2015) Predictive modelling for energy consumption in machining using artificial neural network. *Proc CIRP* 37:205–210. <https://doi.org/10.1016/j.procir.2015.08.081>
61. Sangwan KS, Kant G (2017) Optimization of machining parameters for improving energy efficiency using integrated response surface methodology and genetic algorithm approach. *Proc CIRP* 61:517–522. <https://doi.org/10.1016/j.procir.2016.11.162>
62. Kant G, Sangwan KS (2014) Prediction and optimization of machining parameters for minimizing power consumption and surface roughness in machining. *J Clean Prod* 83:151–164. <https://doi.org/10.1016/j.jclepro.2014.07.073>
63. Park J, Law KH, Bhinge R, Biswas N, Srinivasan A, Dornfeld DA, Helu M, Rachuri S (2015) A generalized data-driven energy prediction model with uncertainty for a milling machine tool using gaussian process. In: *Proceedings of the ASME 10th international manufacturing science and engineering conference 2015*. The American Society of Mechanical Engineers, New York. <https://doi.org/10.1115/MSEC2015-9354>
64. Ellerich M (2021) *Potentiale von machine learning modellen zur prognose von lastgängen bei fertigungsprozessen*, Dissertation, Rheinisch-Westfälische Technische Hochschule Aachen
65. Mühlbauer M, Würschinger H, Polzer D, Hanenkamp N (2021) Energy profile prediction of milling processes using machine learning techniques. In: Beyerer J, Maier A, Niggemann O (eds) *Mach Learn Cyber Phys Syst*. Springer, Berlin Heidelberg, Berlin, Heidelberg, pp 1–11
66. Bhinge R, Park J, Law KH, Dornfeld DA, Helu M, Rachuri S (2017) Towards a generalized energy prediction model for machine tools. *J Manuf Sci Eng* 139(4). <https://doi.org/10.1115/1.4034933>
67. Duerden C, Shark L-K, Hall G, Howe J (2016) Prediction of granular time-series energy consumption for manufacturing jobs from analysis and learning of historical data. In: *2016 annual conference on information science and systems (CISS)*. IEEE, Piscataway, NJ, pp 625–630. <https://doi.org/10.1109/CISS.2016.7460575>
68. Goubko MV, Kuznetsov SO, Neznanov AA, Ignatov DI (2016) Bayesian learning of consumer preferences for residential demand response. *IFAC-PapersOnLine* 49(32):24–29
69. Shoji T, Hirohashi W, Fujimoto Y, Hayashi Y (2014) Home energy management based on Bayesian network considering resident convenience. In: *2014 International conference on probabilistic methods applied to power systems (PMAPS)*, IEEE, pp 1–6
70. Barreto R, Gonçalves C, Gomes L, Faria P, Vale Z (2022) Evaluation metrics to assess the most suitable energy community end-users to participate in demand response. *Energies* 15(7). <https://doi.org/10.3390/en15072380>. <https://www.mdpi.com/1996-1073/15/7/2380>
71. Lin S, Li F, Tian E, Fu Y, Li D (2019) Clustering load profiles for demand response applications. *IEEE Trans Smart Grid* 10(2):1599–1607. <https://doi.org/10.1109/TSG.2017.2773573>
72. Valdes J, Masip Macia Y, Dorner W, Ramirez Camargo L (2021) Unsupervised grouping of industrial electricity demand profiles: synthetic profiles for demand-side management applications. *Energy* 215: 118962. <https://doi.org/10.1016/j.energy.2020.118962>. <https://www.sciencedirect.com/science/article/pii/S0360544220320697>
73. Asghari P, Zakariazadeh A, Siano P (2022) Selecting and prioritizing the electricity customers for participating in demand response programs. *IET Gener Transm Distrib* 16(10):2086–2096. <https://doi.org/10.1049/gtd2.12417>. <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/gtd2.12417>
74. Silva C, Faria P, Vale Z (2018) Discussing different clustering methods for the aggregation of demand response and distributed generation, in: *IEEE Symp Ser Comput Intell (SSCI) 2018*:1645–1650. <https://doi.org/10.1109/SSCI.2018.8628781>
75. Vázquez-Canteli JR, Nagy Z (2019) Reinforcement learning for demand response: a review of algorithms and modeling techniques. *Appl Energy* 235:1072–1089. <https://doi.org/10.1016/j.apenergy.2018.11.002>. <https://www.sciencedirect.com/science/article/pii/S0306261918317082>
76. Naimi R, Nouri M, Cardin O (2021) A Q-learning rescheduling approach to the flexible job shop problem combining energy and productivity objectives. *Sustainability* 13(23):13016. <https://doi.org/10.3390/su132313016>
77. Huang X, Hong SH, Yu M, Ding Y, Jiang J (2019) Demand response management for industrial facilities: a deep reinforcement learning approach. *IEEE Access* 7:82194–82205. <https://doi.org/10.1109/ACCESS.2019.2924030>
78. Pallonetto F, Oxizidis S, Milano F, Finn D (2016) The effect of time-of-use tariffs on the demand response flexibility of an all-electric smart-grid-ready dwelling. *Energy Build* 128:56–67. <https://doi.org/10.1016/j.enbuild.2016.06.041>. <https://www.sciencedirect.com/science/article/pii/S0378778816305308>
79. Yoon JH, Baldick R, Novoselac A (2014) Dynamic demand response controller based on real-time retail price for residential buildings. *IEEE Trans Smart Grid* 5(1):121–129. <https://doi.org/10.1109/TSG.2013.2264970>
80. Chen Z, Wu L, Fu Y (2012) Real-time price-based demand response management for residential appliances via stochastic optimization and robust optimization. *IEEE Trans Smart Grid* 3(4):1822–1831. <https://doi.org/10.1109/TSG.2012.2212729>
81. Marzband M, Alavi H, Ghazimirsaeid SS, Uppal H, Fernando T (2017) Optimal energy management system based on stochastic approach for a home microgrid with integrated responsive load demand and energy storage. *Sustain Cities Soc* 28:256–264. <https://doi.org/10.1016/j.scs.2016.09.017>. <https://www.sciencedirect.com/science/article/pii/S2210670716304449>
82. Patyn C, Ruelens F, Deconinck G (2018) Comparing neural architectures for demand response through model-free reinforcement learning for heat pump control. *2018 IEEE international energy conference (ENERGYCON) 1–6*. <https://doi.org/10.1109/ENERGYCON.2018.8398836>
83. Khan IU, Ma X, Taylor CJ, Javaid N, Gamage KAA (2018) Heuristic algorithm based dynamic scheduling model of home appliances in smart grid. *2018 24th International conference on automation and computing (ICAC) 1–6*. <https://doi.org/10.1109/CISIS.2016.125>
84. Menos-Aikateriniadis C, Lamprinos I, Georgilakis PS (2022) Particle swarm optimization in residential demand-side management: a review on scheduling and control algorithms for demand response provision. *Energies* 15(6). <https://doi.org/10.3390/en15062211>. <https://www.mdpi.com/1996-1073/15/6/2211>
85. Bintoudi A, Bezas N, Zyglakis L, Isaoglou G, Timplalexis C, Gkaidatzis P, Tryferidis A, Ioannidis D, Tzovaras D (2021) Incentive-based demand response framework for residential applications: design and real-life demonstration. *Energies* 14:4315. <https://doi.org/10.3390/en14144315>
86. Ahmed M, Homod R, Ali J (2016). Artificial neural network based controller for home energy management considering demand response events. <https://doi.org/10.1109/ICAEEES.2016.7888097>
87. Croce V, Raveduto G, Verber M, Ziu D (2020) Combining machine learning analysis and incentive-based genetic algo-

- rithms to optimise energy district renewable self-consumption in demand-response programs. *Electronics* 9(6). <https://www.mdpi.com/2079-9292/9/6/945>
88. Keller F, Schultz C, Braunreuther S, Reinhart G (2016) Enabling energy-flexibility of manufacturing systems through new approaches within production planning and control. *Proc CIRP* 57:752–757. <https://doi.org/10.1016/j.procir.2016.11.130>
 89. Dababneh F, Li L, Shah R, Haefke C (2018) Demand response-driven production and maintenance decision-making for cost-effective manufacturing. *J Manuf Sci Eng* 140(6). <https://doi.org/10.1115/1.4039197>
 90. Jia Z-H, Wang Y, Wu C, Yang Y, Zhang X-Y, Chen H-P (2019) Multi-objective energy-aware batch scheduling using ant colony optimization algorithm. *Comput Ind Eng* 131:41–56. <https://doi.org/10.1016/j.cie.2019.03.033>
 91. Caldeira RH, Gnanavelbabu A, Vaidyanathan T (2020) An effective backtracking search algorithm for multi-objective flexible job shop scheduling considering new job arrivals and energy consumption. *Comput Ind Eng* 149:106863. <https://doi.org/10.1016/j.cie.2020.106863>
 92. Wei H, Li S, Quan H, Liu D, Rao S, Li C, Hu J (2021) Unified multi-objective genetic algorithm for energy efficient job shop scheduling. *IEEE Access* 9:54542–54557. <https://doi.org/10.1109/ACCESS.2021.3070981>
 93. Fang K, Uhan N, Zhao F, Sutherland JW (2011) A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. *J Manuf Syst* 30(4):234–240. <https://doi.org/10.1016/j.jmsy.2011.08.004>
 94. Raileanu S, Anton F, Iatan A, Borangiu T, Anton S, Morariu O (2017) Resource scheduling based on energy consumption for sustainable manufacturing. *J Intell Manuf* 28(7):1519–1530. <https://doi.org/10.1007/s10845-015-1142-5>
 95. Mohsenian-Rad A-H, Leon-Garcia A (2010) Optimal residential load control with price prediction in real-time electricity pricing environments. *IEEE Trans Smart Grid* 1(2):120–133. <https://doi.org/10.1109/TSG.2010.2055903>
 96. Alimohammadisagvand B, Jokisalo J, Kilpeläinen S, Ali M, Sirén K (2016) Cost-optimal thermal energy storage system for a residential building with heat pump heating and demand response control. *Appl Energy* 174:275–287. <https://doi.org/10.1016/j.apenergy.2016.04.013>. <https://www.sciencedirect.com/science/article/pii/S0306261916304640>
 97. da Silva IRS, Rabêlo RAL, Rodrigues JJ, Solić P, Carvalho A (2020) A preference-based demand response mechanism for energy management in a microgrid. *J Clean Prod* 255:120034. <https://doi.org/10.1016/j.jclepro.2020.120034>
 98. Luthander R, Widén J, Munkhammar J, Lingfors D (2016) Self-consumption enhancement and peak shaving of residential photovoltaics using storage and curtailment. *Energy* 112:221–231. <https://doi.org/10.1016/j.energy.2016.06.039>. <https://www.sciencedirect.com/science/article/pii/S0360544216308131>
 99. Lu Q, Yu H, Zhao K, Leng Y, Hou J, Xie P (2019) Residential demand response considering distributed PV consumption: a model based on China's PV policy. *Energy* 172:443–456. <https://doi.org/10.1016/j.energy.2019.01.097>. <https://www.sciencedirect.com/science/article/pii/S0360544219301173>
 100. Mourtzis D (2020) Simulation in the design and operation of manufacturing systems: state of the art and new trends. *Int J Prod Res* 58(7):1927–1949
 101. Weather data and weather API | visual crossing. <https://www.visualcrossing.com/>
 102. Wicaksono H, Boroukhian T, Bashyal A (2021) A demand-response system for sustainable manufacturing using linked data and machine learning. In: *Dynamics in logistics*. Springer, Cham, pp 155–181
 103. Noy NF, McGuinness DL et al (2001) *Ontology development 101: a guide to creating your first ontology*
 104. Francis N, Green A, Guagliardo P, Libkin L, Lindaaker T, Marsault V, Plantikow S, Rydberg M, Selmer P, Taylor A (2018) Cypher: an evolving query language for property graphs. In: *Proceedings of the 2018 international conference on management of data*, pp 1433–1445
 105. McGlenn K, Wicaksono H, Lawton W, Weise M, Kaklanis N, Petri I, Tzovaras D (2016) Identifying use cases and data requirements for BIM based energy management processes. In: *CIBSE technical symposium*
 106. Lamy J-B (2019) Owlready2 installation. <https://owlready2.readthedocs.io/en/latest/install.html#owlready2>
 107. Küçük D, Küçük D (2018) Ontowind: an improved and extended wind energy ontology. [arXiv:1803.02808](https://arxiv.org/abs/1803.02808)
 108. Bulygin L, Stupnikov SA (2019) Applying of machine learning techniques to combine string-based, language-based and structure-based similarity measures for ontology matching. In: *DAMDID/RCDL*, pp 129–147
 109. Barrasa J (2020) *Ontologies in neo4j: Semantics and knowledge graphs*. <https://neo4j.com/blog/ontologies-in-neo4j-semantics-and-knowledge-graphs/>
 110. irec conf.org (2020) Fantastiska resultat och fördelar med kamagra jelly: var kan man köpa läkemedlet i sverige? <https://www.irec-conf.org/>
 111. Dutta G, Mitra K (2017) A literature review on dynamic pricing of electricity. *J Oper Res Soc* 68(10):1131–1145
 112. Groß S (2022) *Lastprognose für industrielle fertigungsprozesse mittels maschineller lernverfahren*. Master's thesis, Karlsruhe University of Applied Science
 113. Dean J, Ghemawat S (2004) MapReduce: simplified data processing on large clusters. In: *OSDI'04: sixth symposium on operating system design and implementation*, San Francisco, CA, pp 137–150
 114. Eiselt HA, Sandblom C-L (2022) *Integer linear programming*. In: *Operations research: a model-based approach*. Springer International Publishing, Cham, pp 161–213. https://doi.org/10.1007/978-3-030-97162-5_5
 115. Mitchell TM (1997) *Mach learn*. McGraw-Hill series in computer science, McGraw-Hill, New York
 116. Tassel P, Gebser M, Schekotihin K (2021) A reinforcement learning environment for job-shop scheduling. <http://arxiv.org/pdf/2104.03760v1>
 117. Felder M, Steiner D, Busch P, Trat M, Sun C, Bender J, Ovtcharova J (2023) Energy-flexible job-shop scheduling using deep reinforcement learning. 4th Conference on production systems and logistics CPSL (in press)
 118. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. <http://arxiv.org/pdf/1707.06347v2>
 119. ENTSO-E European network of transmission system operators for electricity. Entso-e transparency platform: transmission: day-ahead prices. <https://transparency.entsoe.eu/transmission-domain/r2/dayAheadPrices/show>
 120. Taillard E (1993) Benchmarks for basic scheduling problems. *Eur J Oper Res* 64(2):278–285. [https://doi.org/10.1016/0377-2217\(93\)90182-M](https://doi.org/10.1016/0377-2217(93)90182-M)
 121. Eiselt HA, Sandblom C-L (2022) *Machine scheduling*. In: *Operations research: a model-based approach*. Springer International Publishing, Cham, pp 333–352. https://doi.org/10.1007/978-3-030-97162-5_9
 122. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M, Ghemawat S, Goodfellow I, Harp A, Irving G, Isard M, Jia Y, Jozefowicz R, Kaiser L, Kudlur M, Levenberg J, Man' e D, Monga R, Moore S, Murray D, Olah C, Schuster M, Shlens J, Steiner B, Sutskever I, Tal-

- war K, Tucker P, Vanhoucke V, Vasudevan V, Vi'egas F, Vinyals O, Warden P, Wattenberg M, Wicke M, Yu Y, Zheng X (2015) TensorFlow: large-scale machine learning on heterogeneous distributed systems. <https://www.tensorflow.org>
123. Seabold S, Perktold J (2010) Statsmodels: econometric and statistical modeling with Python. In: 9th Python in science conference
124. McKinney W (2010) Data structures for statistical computing in Python. In: Proceedings of the 9th Python in science conference, proceedings of the Python in science conference, SciPy, pp 56–61. <https://doi.org/10.25080/Majora-92bf1922-00a>
125. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay É (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12(85): 2825–2830. <http://jmlr.org/papers/v12/pedregosa11a.html>
126. Hart WE, Watson J-P, Woodruff DL (2011) Pyomo: modeling and solving mathematical programs in python. *Math Program Comput* 3(3):219–260. <https://doi.org/10.1007/s12532-011-0026-8>
127. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, Wieser E, Taylor J, Berg S, Smith NJ, Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, Del Río JF, Wiebe M, Peterson P, Gérard-Marchant P, Sheppard K, Reddy T, Weckesser W, Abbasi H, Gohlke C, Oliphant TE (2020) Array programming with numpy. *Nature* 585(7825):357–362. <https://doi.org/10.1038/s41586-020-2649-2>
128. Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W (2016) OpenAI Gym. <http://arxiv.org/pdf/1606.01540v1>
129. Liang E, Liaw R, Moritz P, Nishihara R, Fox R, Goldberg K, Gonzalez JE, Jordan MI, Stoica I (2017) RLlib: abstractions for distributed reinforcement learning. <http://arxiv.org/pdf/1712.09381v4>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Hendro Wicaksono¹ · Martin Trat²  · Atit Bashyal¹ · Tina Boroukhan¹ · Mine Felder² · Mischa Ahrens² · Janek Bender² · Sebastian Groß² · Daniel Steiner² · Christoph July³ · Christoph Dorus⁴ · Thorsten Zoerner⁴

¹ School of Business, Social and Decision Sciences, Constructor University, Campus Ring 1, 28759 Bremen, Germany

² Intelligent Systems and Production Engineering, FZI Research Center for Information Technology,

Haid-und-Neu-Str. 10-14, 76131 Karlsruhe, Baden-Württemberg, Germany

³ devolo AG, Charlottenburger Allee 67, 52068 Aachen, North Rhine-Westphalia, Germany

⁴ StromDAO GmbH, Gerhard Weiser Ring 29, 69256 Mauer, Baden-Württemberg, Germany