IOP Conference Series:
Materials Science and
Engineering

Identifying Energy Bottlenecks in Manufacturing Systems through an Integrated Dashboard

To cite this article: Sunith Bandaru *et al* 2026 *IOP Conf. Ser.: Mater. Sci. Eng.* **1342** 012060

IOP Conf. Series: Materials Science and Engineering **1342** (2026) 012060 doi:10.1088/1757-899X/1342/1/012060

Identifying Energy Bottlenecks in Manufacturing Systems through an Integrated Dashboard

Sunith Bandaru[1], **Carlos Alberto Barrera Diaz**[2], **Amos H.C. Ng**[1,3], **and Lars Hanson**[1,4]

[1]School of Engineering Science, University of Skövde, Sweden
[2]Volvo Group Trucks Operations, Skövde, Sweden
[3]Department of Civil and Industrial Engineering, Uppsala University, Sweden
[4]Volvo Construction Equipment, Arvika, Sweden

E-mail: sunith.bandaru@his.se

Abstract. Manufacturing companies are gradually moving from Industry 4.0's technology focus to Industry 5.0's sustainability focus, and identifying and addressing energy bottlenecks is a part of this transition. In practice, this is challenging due to limited availability of energy data and its poor integration with systems like MES and SCADA. Energy dashboards are capable of consolidating energy data, visualizing consumption patterns, and tracking related KPIs for sustainability. However, most existing implementations are limited to facility-level overviews or machine-specific views without consideration of operational details. To identify energy bottlenecks, the dashboards must also analyze machine states, batch sizes, product mixes, and cycle times. Therefore, this paper presents a Python-based web application built with the Dash framework and open-source packages. The application integrates data from EMS, MES, and SCADA systems. It is capable of performing statistical time-series analysis, joint energy-stop analysis, state-based mapping of energy use, and visualizing various Key Performance Indicators. The proposed integrated dashboard targets discrete manufacturing and is demonstrated on a gear machining line at Volvo Group Trucks Operations. The dashboard currently operates offline with data from enterprise systems, but aims for real-time API integration as a digital twin in the future. This could support simulation for detecting inefficiencies, predicting energy bottlenecks, and optimizing energy consumption.

1 Introduction

Monitoring, analyzing, and reducing energy consumption in manufacturing operations is gaining importance as businesses shift from the technology-centric view of Industry 4.0 to the sustainability-centric view of Industry 5.0. A central challenge in this transition is identifying and mitigating so-called *energy bottlenecks* to maximize the energy efficiency of manufacturing systems. However, achieving this in practice is often difficult due to constraints such as the limited scope and availability of energy consumption data and the lack of seamless integration of this data with common enterprise systems like MES and SCADA. Even when such integration exists, discovering energy inefficiencies requires specialized contextual analysis that considers not only machine operational states but also variations in batch sizes, product mix, setup times, and, consequently, cycle times.

Energy dashboards have emerged as a practical solution to this challenge by aggregating energy data, visualizing consumption patterns, and enabling decision-makers to link energy use with production contexts. Existing approaches in the literature demonstrate how dashboards can improve energy transparency, support benchmarking, and track KPIs to achieve sustainability goals. However, most implementations remain limited either to aggregated facility-level monitoring or to machine-level tracking without integration of detailed operational data. To carry out targeted improvements under budget constraints, there is a need for dashboards that combine machine states, process variability, and production parameters to provide actionable insights for reducing inefficiencies and identifying energy bottlenecks in real manufacturing environments.

At the same time, existing energy dashboards are often proprietary licensed products. Customizing them to include specialized analyses or functionalities typically requires additional development time and costs, as such modifications must be made by the service provider. This dependency poses significant challenges for internal pilot projects and hinders rapid iteration and adaptation to specific organizational needs, thereby delaying the implementation of energy optimization initiatives.

In this paper, we present a Python-based web application, developed using the Dash framework and built entirely on openly available Python packages, that enables: (i) integration of relevant data from multiple sources, including EMS (Energy Management System), MES (Manufacturing Execution System), and SCADA (Supervisory Control and Data Acquisition); (ii) statistical time-series analysis of energy consumption data; (iii) joint analysis of energy consumption and machine stop data; (iv) synchronization of energy data with machine operational states; and (v) calculation and visualization of various KPIs and their trends.

While the dashboard is developed with typical discrete manufacturing systems in mind and is therefore company-independent, we demonstrate its application through a real industrial use-case of a gear machining line at Volvo Group Trucks Operations (Volvo GTO). Currently, the dashboard operates offline using data exported from the aforementioned enterprise systems. However, the ultimate goal is to integrate it directly with these systems via APIs or standard industrial messaging protocols to enable real-time monitoring and analysis. Once integrated, the dashboard has the potential to serve as a digital twin of the production line, providing a synchronized, real-time representation of machine states, energy consumption, and other operational parameters. This capability, when combined with discrete-event simulation, can support operators and engineers in identifying inefficiencies, predicting both energy and productivity bottlenecks, and testing potential 'what-if' scenarios and optimization strategies.

The paper is structured as follows. Section 2 reviews related work on energy dashboards and energy bottlenecks in manufacturing. Section 3 describes the design and implementation of the proposed integrated dashboard, detailing its various modules. Section 4 presents the use case of a gear machining line at a major Swedish vehicle manufacturer. Section 5 demonstrates the dashboard's functionality on this use case and Section 6 discusses the results from applying the dashboard to the use case data, highlighting key insights on energy consumption and operational efficiency. Finally, Section 7 concludes the paper and outlines future work directions.

2 Related Work

This section reviews recent literature on energy dashboards and the identification of energy bottlenecks in manufacturing systems. It covers key advancements in dashboard technologies, from IoT-based monitoring to integrated decision-support systems, and discusses various methodologies for defining and detecting energy bottlenecks.

2.1 Energy Dashboards and Analytical Tools

In recent years, digital tools for monitoring and analyzing energy consumption in manufacturing have moved from simple monitoring to integrated analytical platforms. Some studies indicate the use of IoT for energy data acquisition and processing. For example, Uhlmann et al. [1] describe an IoT-based system that links various energy indicators to digital twins, where machine learning methods are used to detect patterns of energy use. Similarly, Mirani et al. [2] present an industrial IoT system that uses edge processing for real-time energy monitoring of both fixed and mobile assets. Their system also provides a master user interface for comparative analysis and a dedicated interface for in-depth analysis. Webb et al. [3] propose a specific architecture for energy monitoring and visualization in smart factories. Their solution provides analytics and insights about energy consumption at multiple user levels and can be used within SMEs. They demonstrate its application in a robotic material handling case study.

The concept of the digital twin also appears often in the literature due to its ability to provide additional insights through predictive and prescriptive analytics. Fan et al. [4] propose a hybrid digital twin that combines real-time physical data with simulations of factory operations, including soft systems like human decision-making, and uses genetic algorithms to optimize for green manufacturing metrics. This allows for not only monitoring but also simulating 'what-if' scenarios to test energy-saving strategies without disrupting production. In the same vein, Khodadadi et al. [5] present a methodology for data-driven extraction of simulation models for energy-oriented digital twins and apply it to assembly process of a quadcopter drone part. Process mining, akin to digital twins, is being increasingly used for analyzing event logs to discover actual workflows. As reviewed by Kurniasih et al. [6], integrating process mining with discrete-event simulation helps address issues like resource bottlenecks and energy efficiency.

The most advanced tools have evolved into integrated decision-support systems (DSS) that embed energy considerations into broader operational planning. Some dashboards are designed as comprehensive DSS for specific industries, such as the one developed by Hernández et al. [7] for the forest products sector, which uses performance indicators and statistical methods to aid decision-making. Other examples, though from different industries, exemplify the core components of a manufacturing DSS: integration of field data, simulation tools, and interactive dashboards to forecast challenges and evaluate system modifications [8]. In manufacturing, this translates to frameworks that balance multiple objectives. Bocewicz et al. [9] developed a declarative framework for production line balancing that includes energy consumption as a key criterion alongside cost and schedule continuity, allowing for interactive trade-off analysis. Guendouli et al. [10] propose an integrated production-maintenance strategy that also considers energy consumption. In addition to these, AI-driven systems have also been used to optimize scheduling and resource deployment for energy efficiency [11], including bi-objective optimization of cost and energy emissions in human-robot collaborative assembly lines [12].

In summary, the literature shows a clear trend toward integrated and intelligent energy analysis tools. These systems are evolving from passive displays into active decision-support platforms that leverage IoT, digital twins, and AI to provide comprehensive insights for optimizing energy use in manufacturing.

2.2 Energy Bottlenecks

The idea of an energy bottleneck in manufacturing extends the traditional bottleneck concept, which identifies the process or machine that constrains overall system throughput [13]. As industries place greater emphasis on energy efficiency and sustainability, this definition has evolved to include components that not only limit production flow but also cause excessive energy use. Ghatorha et al. [14] describe an energy bottleneck as a resource whose inefficiency leads to both reduced productivity and increased energy consumption. Similarly, Wang et al. [15] highlight that such bottlenecks often appear in processes with poor energy-to-output ratios, particularly in energy-intensive sectors like steelmaking.

Methods to identify energy bottlenecks generally combine production and energy performance metrics. The Energy Focused Bottleneck Analysis (EFBA) framework, for example, integrates classic throughput analysis with energy and carbon data to locate the stages that consume the most energy relative to their output [14]. Data-driven studies, such as those by Lai et al. [16], show that sensor data and process mining can reveal shifting or "dynamic" bottlenecks as production conditions change. More recent work links these approaches with Industry 4.0 technologies. Keramati Feyz Abadi et al. [17] and Ragazzini et al. [18] demonstrate how IoT systems and digital twins enable real-time monitoring and prediction of energy bottlenecks across complex lines.

Optimization models also contribute to this field. Asghar et al. [19] use mixed-integer linear programming to identify bottlenecks affecting both production scheduling and energy consumption, incorporating stochastic failure and repair rates across working, idle, and repair states. Building on this, Cagno et al. [20] emphasize that IoT technologies provide the real-time data infrastructure required for such stochastic models. This digital integration enhances visualization and connectivity, enabling a shift from reactive maintenance to predictive, energy-aware production scheduling.

While these studies show clear progress towards energy monitoring, analysis, and reduction, there is still no industry-standard definition for energy bottlenecks, nor a widely accepted methodology for integrating data needed to calculate KPIs relevant to detecting such bottlenecks. This gap highlights the need for practical tools that can combine energy consumption data with operational data to identify and address energy bottlenecks effectively in real-world manufacturing settings.

3 Design and Implementation of the Proposed Integrated Dashboard

The development of an integrated dashboard is motivated by the practical challenges of data integration in discrete manufacturing, where energy data, machine stop data, and cycle time data exhibit fundamentally different temporal structures, as illustrated in Figure 1. Energy consumption is typically acquired continuously through sensors and aggregated at regular intervals by the EMS. This reflects the analog nature of power usage. In contrast, both machine stop data and cycle time data are event-driven and recorded by MES and SCADA as discrete occurrences representing various machine states such as producing, waiting, fault, or setup. These states are triggered by operational changes rather than fixed time intervals. To facilitate joint analysis of these three data sources, a carefully designed data integration approach is needed.

The dashboard developed in this work uses a systematic data processing pipeline that aggregates both stop data and cycle time data from event-based production logs, and synchronizes them with continuous energy measurements. As the dashboard currently works offline, the analysis begins with CSV files exported from EMS, MES, and SCADA systems. These files are uploaded and parsed to extract relevant fields such as timestamps, resource identifiers, and operational states. The following sections describe the main modules of the dashboard, which provide comprehensive tools for energy data analysis, machine stop data analysis, cycle time analysis, and Key Performance Indicator (KPI) calculation.

3.1 Energy Data Analysis Module

The energy data analysis module provides a range of tools for processing and analyzing aggregated energy consumption data, similar to those represented by the blue bars in Figure 1. After

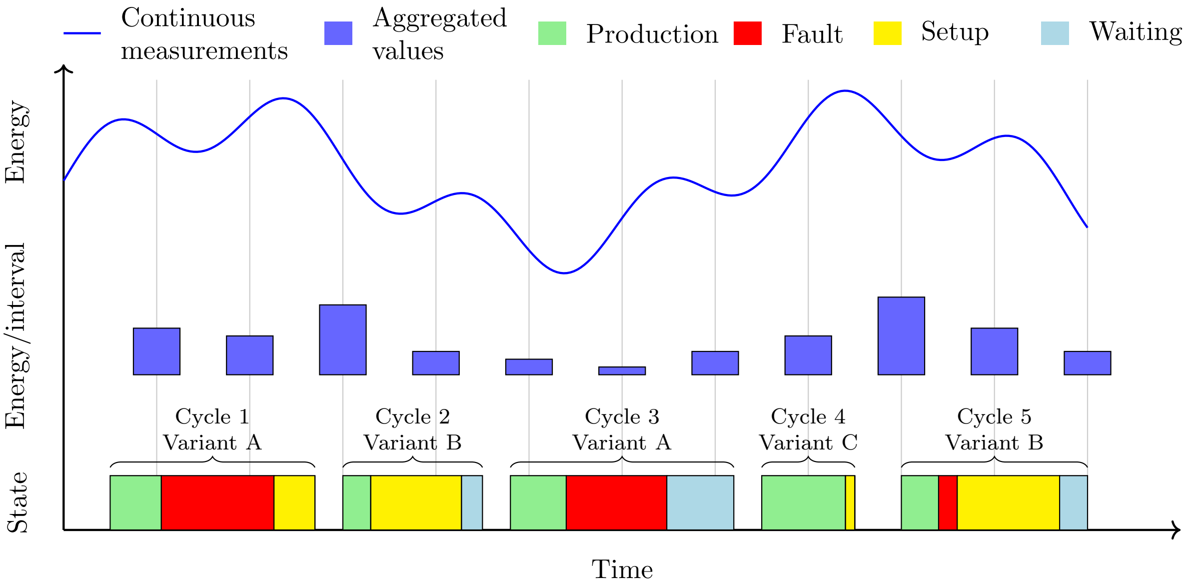


Figure 1: Illustration of the challenges in integrating data with different temporal structures in energy dashboards. Energy consumption data is measured continuously (blue curve) and aggregated at regular intervals (blue bars), while operational state data is stored as events based on machine states: producing (green), fault (red), setup (yellow), and waiting (light blue).

the user uploads the data, the dashboard first generates summary statistics and basic visualizations for an initial overview and verification by the user. The module's more advanced features are organized into two main sub-tabs: individual series analysis and multivariate analysis. For analyzing individual time series, the dashboard uses Python's ‘statsmodels‘ library to perform several key functions. It can apply seasonal decomposition (STL) to separate an energy signal into its trend, seasonal, and residual components, which helps in identifying regular cycles in energy use, such as those occurring daily or weekly. The module also generates autocorrelation (ACF) and partial autocorrelation (PACF) plots, which visualize how current energy consumption values of a machine relate to its past values, offering insight into the temporal dependencies within the signal.

For multivariate analysis, the module uses correlation heatmaps and scatter plots to visualize relationships between energy consumption values of different machines. It also implements Principal Component Analysis (PCA) using ‘scikit-learn‘ to reduce the dimensionality of the data, typically to two principal directions, which can help in visually identifying clusters of time intervals when the machines have similar energy profiles. The resulting PCA biplot is interactive, i.e., it allows the user to select a cluster (or a group of time periods) to understand how it differs from other times periods in terms of energy consumption of the machines. To do this, the module uses scikit-learn's decision tree classifier to generate human-readable rules that describe the conditions associated with specific energy consumption levels. For instance, a rule might state that “during the selected intervals, Machine A consumes less than X kWh while Machine B consumes more than Y kWh”.

Discovering patterns in energy consumption is an important step for identifying inefficiencies. However, to link these patterns to operational states, the dashboard provides two additional modules that analyze machine stop data and cycle time data, respectively, and synchronize them with the energy data for joint analysis.

3.2 Machine Stop Data Analysis Module

The machine stop data analysis module processes and analyzes production interruptions captured by the SCADA system to provide insights into downtime patterns. After the user uploads stop data containing stop-start and stop-end timestamps for all resources, the module calculates a range of summary statistics for stop durations. These basic results are presented as bar charts and sortable tables, which allows users to quickly identify machines with the most downtime.

For a more in-depth analysis, the module provides two key visualization sub-tabs. The first visualization splits, if necessary, and maps stop events to corresponding time intervals in the energy data, highlighting them on time-series plots of energy consumption for various machines. This allows users to correlate stop events with energy consumption patterns, revealing any anomalous periods of excessive consumption. The second visualization uses grouped bar charts to show how the energy consumption differs during time intervals corresponding to stops versus during normal operation for various machines, enabling users to quantify the energy impact of downtime and identify opportunities for energy savings.

3.3 Cycle Time Analysis Module

The cycle time analysis module processes machine state data to analyze manufacturing performance and its relationship with energy consumption. The module starts with event-based production logs captured by the MES, which contain details like start and end times, resource identifiers, and operational states for each cycle. Similar to the previous module, the cycles are split, if necessary, and mapped to corresponding time intervals in the energy data, and the durations of various states like producing, waiting, setup, or inactive are calculated within each interval. This critical function of aligning the event-based cycle data with the interval-based energy measurements is referred to as 'Cycle Time Aggregation' in the dashboard, and it is essential for later KPI calculations.

For visualization, the module provides interactive plots that allow users to compare different operational states across various machines. This includes overlaid time-series plots showing how the durations of chosen states vary over time for different machines, as well as comparative boxplots that summarize the distribution of these durations, enabling quick identification of manufacturing performance variability across machines.

3.4 Key Performance Indicator (KPI) Module

The KPI module serves as the analytical hub of the dashboard, bringing together data from the energy, stop, and cycle time analyses to calculate a comprehensive set of performance indicators. The data integration step combines the aggregated energy data from the first module and the aggregated cycle time data from the third module, resulting in synchronized energy consumption and operational state information across all time intervals. The module provides integrated visualization feature to plot one or more synchronized parameters for various machines.

With the integrated data in place, the KPI calculation engine computes 15 distinct performance indicators, which are centrally managed in the 'KPI_DEFINITIONS' dictionary within the dashboard's Python code. As shown in Table 1, these KPIs are grouped into three categories to provide a holistic view of system performance. The first category, **Energy KPIs**, focuses on consumption patterns, with metrics like the Peak Energy Factor and Base Load Factor that help identify spikes and idle energy use. The second, **Operational KPIs**, measures production efficiency through standard metrics like Availability Rate and Overall Utilization. The third, **Consolidated KPIs**, merges energy and production data to create hybrid indicators such as Producing Energy Rate and Idle Energy Rate, which directly link energy use to specific operational contexts.

Table 1: Key Performance Indicators implemented in the KPI module.

KPI Name	**Formula and Description**
Energy KPIs (Calculated by averaging quantities computed individually in each interval)	
Mean Energy Consumption	$mean(E_{total})$ Description: Typical energy consumption in any interval (kWh)
Peak Energy Factor	$mean(E_{max}/E_{total})$ Description: Indicator for energy consumption spikes
Base Load Factor	$mean(E_{min}/E_{total})$ Description: Indicator for idle energy consumption
Energy Stability Factor	$mean(E_{min}/E_{max})$ Description: Indicator for energy consumption fluctuations
Energy Variability Range	$mean(E_{max} - E_{min})$ Description: Typical energy variation in any interval (kWh)
Operational KPIs (Calculated after adding machine state times over all intervals)	
Availability Rate	$(T_{total} - \Sigma\, T_{nonactive})/T_{total} \times 100$ Description: Component of Overall Equipment Effectiveness
Overall Utilization	$\Sigma\, T_{producing}/T_{tracking} \times 100$ Description: Indicator for overall productive time
Performance Rate	$\Sigma\, T_{producing}/\Sigma\, T_{active} \times 100$ Description: Indicator for overall efficiency when active
Non-Productive Rate	$\Sigma\, T_{nonproductive}/T_{tracking} \times 100$ Description: Indicator for overall non-productive time
Average Downtime (min/hr)	$\Sigma\, T_{nonactive}/T_{tracking} \times 60$ Description: Indicator for overall wasted and maintenance time
Consolidated KPIs (Estimated by minimizing $\Sigma\, [E_{total} - (E_{producing} + E_{partchange} + E_{waiting} + E_{nonactive})]^2$)	
Producing Energy Rate	$P_{producing} = \Sigma\, E_{producing}/(\Sigma\, T_{producing}/3600)$ Description: Energy consumption rate when producing (kWh/hr)
Setup Energy Rate	$P_{setup} = \Sigma\, E_{partchange}/(\Sigma\, T_{partchange}/3600)$ Description: Energy consumption rate when setting-up (kWh/hr)
Idle Energy Rate	$P_{idle} = \Sigma\, (E_{waiting} + E_{nonactive})/(\Sigma\, (T_{waiting} + T_{nonactive})/3600)$ Description: Energy consumption rate when waiting or inactive (kWh/hr)
Wasted Energy Rate	$P_{wasted} = \Sigma\, E_{nonactive}/(\Sigma\, T_{nonactive}/3600)$ Description: Energy consumption rate when inactive (kWh/hr)
Non-Productive Energy Rate	$P_{nonproductive} = \Sigma\, (E_{partchange}+E_{waiting}+E_{nonactive})/(\Sigma\, (T_{nonproductive})/3600)$ Description: Energy consumption when not producing (kWh/hr)

E_{total}, E_{min}, E_{max} = total, minimum, and maximum energy consumption per interval (kWh);
$E_{producing}$, $E_{partchange}$, $E_{waiting}$, $E_{nonactive}$ = estimated energy consumption per interval in different machine states (kWh);
$T_{tracking}$ = total tracking time over all intervals (seconds);
$T_{total} = T_{producing} + T_{partchange} + T_{waiting} + T_{nonactive}$;
$T_{active} = T_{producing} + T_{partchange} + T_{waiting}$;
$T_{nonproductive} = T_{partchange} + T_{waiting} + T_{nonactive}$

Before calculating these KPIs, the user is required to select relevant columns from the uploaded datasets. This includes specifying the columns corresponding to energy consumption values (total, minimum, and maximum) and selecting/grouping columns corresponding to each of the operational states (producing, setup, inactive, and waiting). The module then adds grouped columns, if any, thus creating a dataset suitable for KPI calculations.

The calculated KPIs are presented to the user through interactive charts and sortable data tables, allowing for easy comparison across different machines or time periods. The module also includes a trend visualization feature that enables users to understand how these indicators change over different time scales (e.g., hourly, daily, weekly). This functionality is useful for tracking performance over time and assessing the impact of any process improvements or operational changes made on the factory floor. By providing multiple ways to view and analyze the KPIs, the module supports a more data-driven approach to identifying energy bottlenecks and opportunities for improvement.

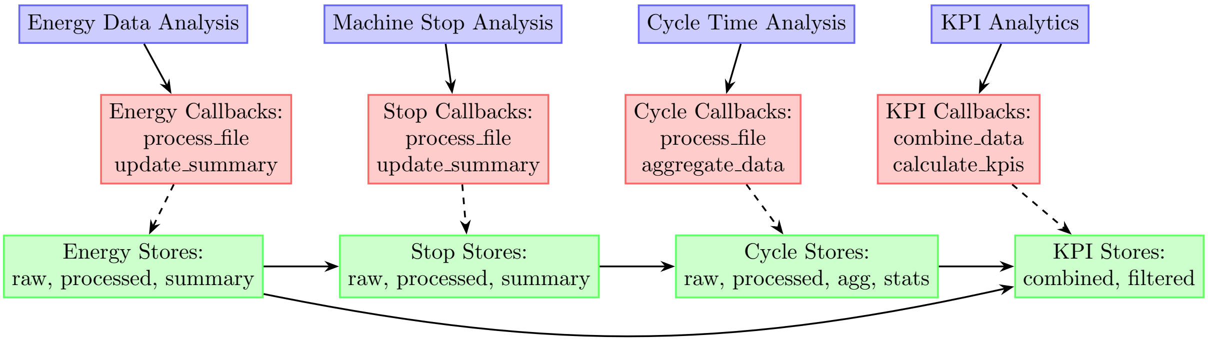


Figure 2: High-level architecture of the integrated dashboard application, showing the main analytical modules (blue), their dedicated data stores (green), and the flow of data through key callbacks (red).

3.5 Implementation

The dashboard is a Python-based web application that uses the Dash framework by Plotly. The architecture is modular and scalable. It organizes around the main analytical functions. The application's structure, as depicted in Figure 2, consists of the four primary modules described above: Energy Data Analysis, Machine Stop Analysis, Cycle Time Analysis, and KPIs.

Dash's client-side 'dcc.Store' components handle data management. Each module uses a set of dedicated stores to maintain the state of the application. These stores hold raw uploaded data, processed dataframes, and summary statistics. For instance, the Energy Data Analysis module uses 'energy-raw-store' for the initial data and 'energy-processed-store' for the cleaned, analyzed data. This separation ensures that data processing pipelines are logical and traceable.

A series of callbacks enables the application's interactivity. These callbacks trigger functions based on user input. They process data from the stores, perform calculations, and update the UI components, such as graphs and tables. For example, when a user uploads a file, a callback parses the data and saves it to the corresponding raw data store. Subsequent callbacks process this data and generate visualizations. The final KPI module integrates data from the energy and cycle time stores to compute and display the consolidated performance indicators. This reactive, callback-driven model provides an interactive and responsive user experience.

4 Use Case Description

The use case selected for this study focuses on a machining line dedicated to the production of a timing gear used in a heavy-duty truck internal combustion engine (ICE). The timing gear is a critical component responsible for synchronizing the rotation of the crankshaft and camshaft, ensuring precise valve timing and optimal engine performance. Due to its functional importance, the gear requires high dimensional accuracy, surface integrity, and consistency, which makes the line a representative and technically relevant case for studying energy consumption and process efficiency in precision machining.

The machining line consists of a sequence of automated and semi-automated operations, each performed by a designated machine or cell, as shown in the layout below. The production starts with raw gear blanks that are loaded into the Robotcell Bin-picking, where a robotic system identifies and picks parts for feeding into subsequent operations. The EMAG Lathe (EMAG Svarv) performs the turning operations, shaping the gear blank to its initial geometry. This is followed by the Liebherr Gear Hobbing Machine (Kuggfräsmaskin), which generates the gear

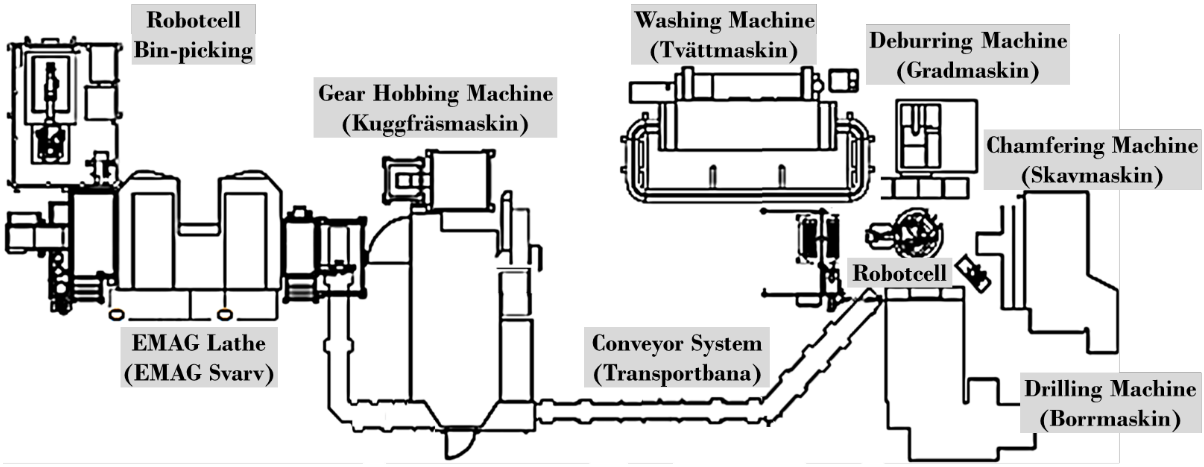


Figure 3: Layout of the machining line for timing gear production at Volvo GTO.

teeth through high-precision hobbing. The machined components are then transferred via a conveyor system (Transportbana) to the next station. Subsequent processes refine the geometry and surface quality. The Drilling Machine (Borrmaskin) performs necessary bore and hole operations, while the Deburring Machine (Gradmaskin) and Chamfering Machine (Skavmaskin) remove sharp edges and residual burrs to ensure correct meshing performance and assembly safety. A dedicated Robotcell handles intermediate part transfer and orientation adjustments between stations. Finally, the components are cleaned in the Washing Machine including pallet conveyor (Tvättmaskin), ensuring removal of metal chips, oil, and other residues before inspection and assembly.

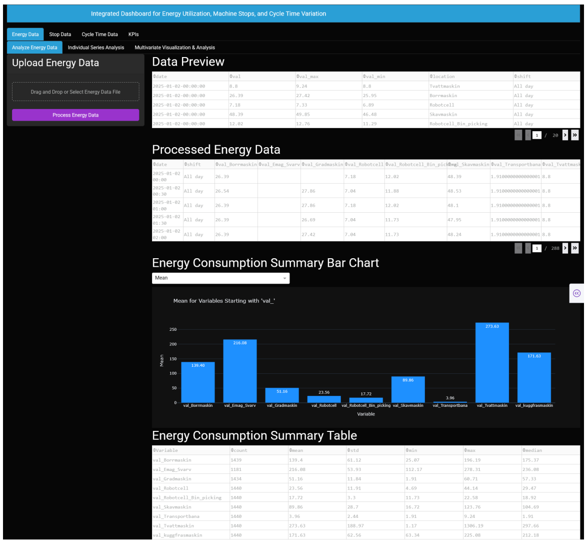
This machining line was selected as a use case for the study of energy consumption in discrete manufacturing systems, as it combines multiple machining, handling, and auxiliary operations representative of industrial production environments. The diversity of machine types and automation levels provides a suitable context for analyzing energy profiles across different process categories; cutting, handling, and auxiliary activities; allowing for detailed evaluation of a variety of energy Key Performance Indicators.

The data for the use case was collected over a period of one month (January 2025) during regular production operations. Energy consumption data was obtained from the facility's Energy Management System (EMS), which records power usage at 30-minute intervals for each machine. Machine stop data, including timestamps and reasons for stoppages, was extracted from the SCADA system, while cycle time data detailing machine states and durations was obtained from the MES. The datasets were provided as CSV files by the company, and used directly in integrated dashboard to generate the figures and analyses presented in the following sections.

4.1 Demonstration of the Integrated Dashboard

The dashboard's energy data analysis module begins with Figure 4, which displays the upload interface, data preview, and summary statistics for user verification. For deeper analysis, Figure 5 illustrates the multivariate analysis tab with interactive scatter plots. Users can select clusters of data points to highlight corresponding periods on time-series plots of energy consumption for selected machines. The PCA biplot offers similar selection capabilities, allowing users to derive decision tree rules that describe energy consumption patterns.

In Figure 6, the machine stop analysis module shows how stop events are mapped onto time-



Data Preview

date	val	val_max	val_min	location	shift
2025-01-02-00:00:00	8.8	9.24	8.8	Tvattmaskin	All day
2025-01-02-00:00:00	26.39	27.42	25.95	Borrmaskin	All day
2025-01-02-00:00:00	7.18	7.33	6.89	Robotcell	All day
2025-01-02-00:00:00	48.39	49.85	46.48	Skavmaskin	All day
2025-01-02-00:00:00	12.02	12.76	11.29	Robotcell_Bin_picking	All day

Processed Energy Data

date	shift	val_Borrmaskin	val_Emag_Svarv	val_Gradmaskin	val_Robotcell	val_Robotcell_Bin_pick	val_Skavmaskin	val_Transportbana	val_Tvattmask
2025-01-02 00:00	All day	26.39			7.18	12.02	48.39	1.9100000000000001	8.8
2025-01-02 00:30	All day	26.54		27.86	7.04	11.88	48.53	1.9100000000000001	8.8
2025-01-02 01:00	All day	26.39		27.86	7.18	12.02	48.1	1.9100000000000001	8.8
2025-01-02 01:30	All day	26.39		26.69	7.04	11.73	47.95	1.9100000000000001	8.8
2025-01-02 02:00	All day	26.39		27.42	7.04	11.73	48.24	1.9100000000000001	8.8

Energy Consumption Summary Bar Chart

Energy Consumption Summary Table

Variable	count	mean	std	min	max	median
val_Borrmaskin	1439	139.4	61.12	25.07	196.19	175.37
val_Emag_Svarv	1181	216.08	53.93	112.17	278.31	236.08
val_Gradmaskin	1434	51.16	11.84	1.91	60.71	57.33
val_Robotcell	1440	23.56	11.91	4.69	44.14	29.47
val_Robotcell_Bin_picking	1440	17.72	3.3	11.73	22.58	18.92
val_Skavmaskin	1440	89.86	28.7	16.72	123.76	104.69
val_Transportbana	1440	3.96	2.44	1.91	9.24	1.91
val_Tvattmaskin	1440	273.63	188.97	1.17	1306.19	297.66
val_kuggfrasmaskin	1440	171.63	62.56	63.34	225.08	212.18

Figure 4: Screenshot of the data upload and summary statistics tab in the Energy Data Analysis module.

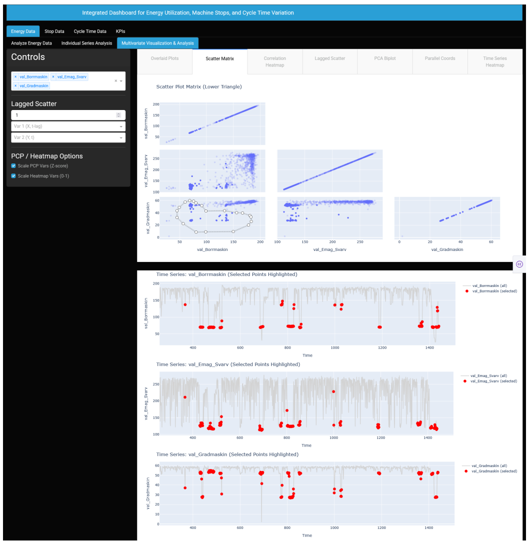


Figure 5: Screenshot of the multivariate analysis tab in the Energy Data Analysis module.

series plots of energy use for different machines. This helps users identify connections between interruptions and energy patterns. It highlights any unusual spikes in consumption. On the right, Figure 7 shows how aggregated cycle time data can be visualized. It uses overlaid time-series plots of chosen operational states across various machines. The box plots allow visual comparison of performance variations for the chosen machines.

The KPI module is shown in Figure 8, highlighting the data integration step that merges processed energy data from the first module with aggregated cycle time data from the third module to synchronize energy consumption and operational states across intervals. On the right, Figure 9 shows the user interface for defining columns related to energy metrics (such as total, min, and max consumption) and operational states (like producing, setup, inactive, and waiting), which facilitates the computation of the 15 KPIs outlined in Table 1.

5 Results and Discussion

Table 2 presents the calculated Key Performance Indicators (KPIs) for each resource in the machining line over the one-month tracking period. The results highlight significant variations in energy consumption patterns and operational efficiency across different machines, providing valuable insights into potential energy bottlenecks and areas for optimization.

Looking at the energy consumption metrics, we see that the lathe (Svarv) uses the most energy on average at 216.33 kWh. Right behind it are the Gear Hobbing Machine (Kuggfräs) at 193.22 kWh and the Drilling Machine (Borrmaskin) at 162.12 kWh. This is reasonable because these are the heavy-duty machining operations that involve cutting and removing material. On the other hand, auxiliary machines like the Robotcell Bin-picking (Robotcell_BP) and the dedicated Robotcell use much less energy, 18.63 kWh and 26.88 kWh respectively, since they are mainly for handling and transferring parts. The KPI trend for Mean Energy Consumption is shown in Figure 10 for all machines over 24 hour periods during the entire tracking period. The Energy Variability Range highlights these differences even more, with Svarv showing the biggest variation at 252.13 kWh, which could point to changing energy needs from different production

Figure 6: Screenshot of the combined visualization tab in the Machine Stop Data Analysis module.

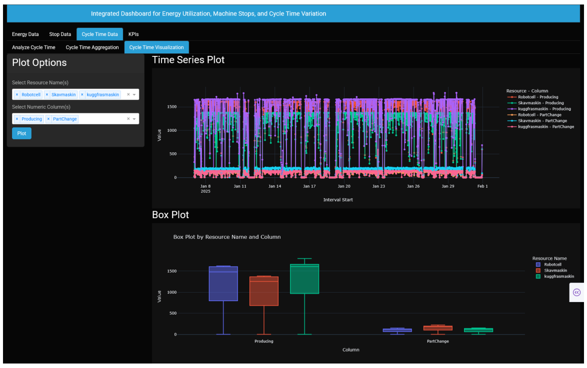


Figure 7: Screenshot of the cycle time visualization tab in the Cycle Time Analysis module.

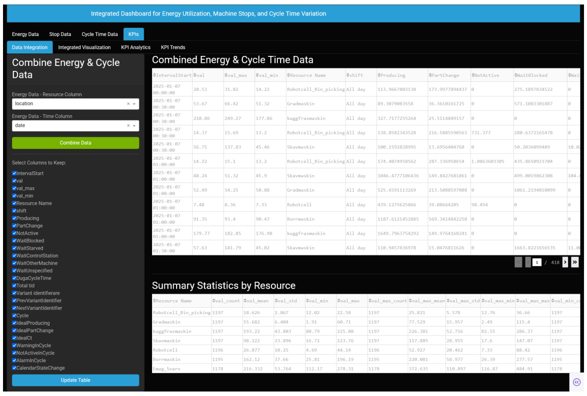


Figure 8: Screenshot of the data integration tab in the KPI module.

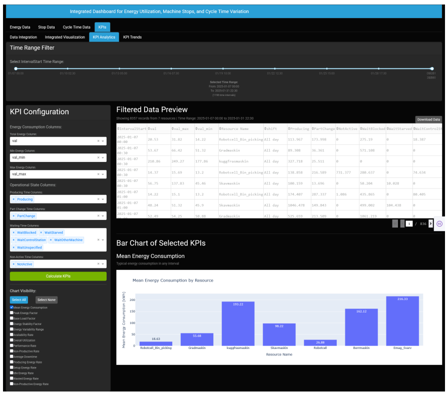


Figure 9: Screenshot of the KPI visualization tab in the KPI module.

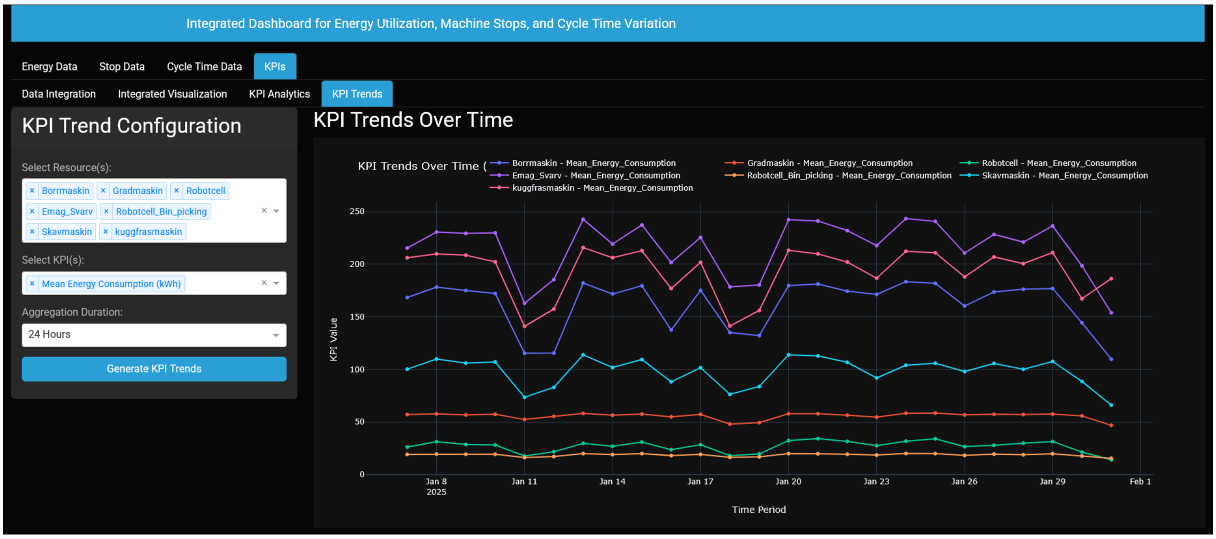
loads or inefficiencies. The Base Load Factor, which mostly relates to idle energy use, is low for the Robotcell at 0.54, showing good power management when not active, while Svarv's 0.6 suggests higher baseline consumption.

Operational KPIs reveal disparities in machine availability and utilization. The Chamfering Machine (Skav) achieves the highest availability rate at 99.39%, indicating minimal downtime, while Svarv lags significantly at 68.41%, pointing to frequent interruptions that could stem from maintenance issues or operational faults. Overall Utilization is highest for Svarv (80.27%) and the Robotcell (63.42%), but this must be interpreted alongside availability. Svarv's high utilization occurs despite low availability, suggesting that when operational, it is heavily used. The trend for this KPI is shown in Figure 11, again over 24 hour periods during the entire month. The Non-Productive Rate for Svarv exceeds 100% because there are actually two lathes in the machining line (see Figure 3) connected to the same energy meter, causing their state times to be added within each interval. This is also why the Average Downtime for Svarv is significantly higher at 18.95 min/hr compared to other machines.

Consolidated KPIs, which integrate energy and operational data, provide critical insights into energy bottlenecks by linking consumption to specific machine states. The Producing Energy Rate, which measures energy use during active production, varies significantly across machining stations: the Gear Hobbing Machine (Kuggfräs) exhibits the highest rate at 42.77 kWh/hr,

Table 2: Key Performance Indicators by Resource (names have been shortened for brevity).

KPI	**Robotcell_BP**	**Grad**	**Kuggfräs**	**Skav**	**Robotcell**	**Borr**	**Svarv**
Mean Energy Consumption (kWh)	18.63	55.68	193.22	98.22	26.88	162.12	216.33
Peak Energy Factor (ratio)	1.38	1.38	1.17	1.2	1.96	1.34	1.71
Base Load Factor (ratio)	0.81	0.89	0.85	0.76	0.54	0.78	0.6
Energy Stability Factor (ratio)	0.61	0.66	0.74	0.64	0.33	0.6	0.39
Energy Variability Range (kWh)	10.95	27.91	63.55	44.8	39.92	96.86	252.13
Availability Rate (%)	95.85	97.91	91.73	99.39	88.77	89.93	68.41
Overall Utilization (%)	8.05	27.66	67.52	54.11	63.42	64.34	80.27
Performance Rate (%)	9.16	30.34	79.37	58.6	81.68	77.49	51.77
Non-Productive Rate (%)	83.94	65.6	25.82	38.85	25.45	28.76	106.37
Average Downtime (min/hr)	2.49	1.25	4.96	0.37	6.74	6.04	18.95
Producing Energy Rate (kWh/hr)	6.01	12.14	42.77	21.94	5.15	35.52	33.60
Setup Energy Rate (kWh/hr)	4.89	11.99	35.52	30.85	19.39	33.11	0.00
Idle Energy Rate (kWh/hr)	3.38	10.85	29.89	14.89	4.28	21.34	19.15
Wasted Energy Rate (kWh/hr)	3.12	8.09	20.29	6.53	2.78	21.34	16.29
Non-Productive Energy Rate (kWh/hr)	3.63	11.04	31.05	18.14	7.32	28.99	13.74



Select Resource(s): Borrmaskin, Gradmaskin, Robotcell, Emag_Svarv, Robotcell_Bin_picking, Skavmaskin, kuggfrasmaskin

Select KPI(s): Mean Energy Consumption (kWh)

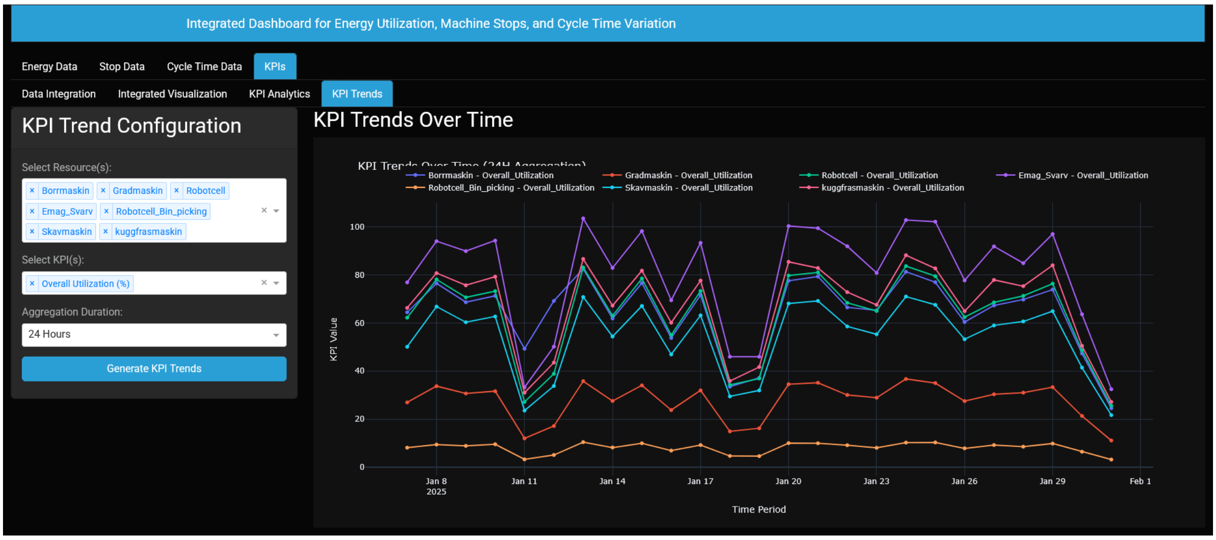
Aggregation Duration: 24 Hours

Figure 10: KPI trend for Mean Energy Consumption for all machines except the conveyor system and the washing machine.

followed by the Drilling Machine (Borrmaskin) at 35.52 kWh/hr, the Lathe (Svarv) at 33.60 kWh/hr, the Chamfering Machine (Skav) at 21.94 kWh/hr, and the Deburring Machine (Grad) at 12.14 kWh/hr. In contrast, the Wasted Energy Rate, which measures energy consumed during inactive periods, is notably higher for Borrmaskin at 21.34 kWh/hr, Kuggfräs at 20.29 kWh/hr, and Svarv at 16.29 kWh/hr. This highlights significant potential for energy savings during waiting and inactive states. The Setup Energy Rate is unusually high for Kuggfräs, Skav, and Borrmaskin. It even exceeds Producing Energy Rate for some machines, indicating potential for flow optimization. The Non-productive Energy Rate is a good overall indicator of energy inefficiencies in the system, and it highlights Kruggfräs and Borrmaskin as the main energy bottlenecks that are constraining system efficiency and sustainability.

6 Conclusions and Future Work

The integrated dashboard presented in this paper provides a comprehensive solution for energy analysis in discrete manufacturing environments. Its core features include modular components



Integrated Dashboard for Energy Utilization, Machine Stops, and Cycle Time Variation

KPI Trend Configuration

Select Resource(s): Borrmaskin, Gradmaskin, Robotcell, Emag_Svarv, Robotcell_Bin_picking, Skavmaskin, kuggfrasmaskin

Select KPI(s): Overall Utilization (%)

Aggregation Duration: 24 Hours

KPI Trends Over Time

Figure 11: KPI trend for Overall Utilization for all machines except the conveyor system and the washing machine.

for energy data analysis, machine stop data analysis, cycle time analysis, and KPI analytics, enabling users to perform statistical time-series analysis, joint energy-stop analysis, state-based mapping of energy use, and interactive KPI visualizations. By integrating data from EMS, MES, and SCADA systems, the dashboard synchronizes continuous energy measurements with event-based operational data, thus providing a holistic view of system performance. Implemented using the Dash framework and open-source Python packages, it ensures accessibility, customizability, and scalability for industrial pilot projects.

The dashboard was demonstrated on a real-world use case involving the timing gear machining line at Volvo GTO, where it processed one month of data to calculate 15 KPIs across multiple machines. Results show significant variations in energy consumption patterns and operational efficiency, with machines like the Gear Hobbing Machine (Kuggfräs) and Drilling Machine (Borrmaskin) identified as the main energy bottlenecks due to high non-productive energy rates. Overall, the results demonstrate the practical value of the integrated dashboard.

Future work will involve improving the dashboard's capabilities. Once the company is satisfied with its features, real-time data integration will be implemented to enable continuous monitoring and provide direct feedback to energy managers. This will involve using existing or developing new APIs to connect with EMS, MES, and SCADA systems for live data streaming. More advanced features such as time-series forecasting and machine learning-based anomaly detection will need to be incorporated to proactively identify potential issues before they lead to unplanned downtime or excessive energy consumption. Finally, the dashboard will be tested and validated in additional industrial settings, including at other companies, to assess its generalizability and effectiveness in diverse operational contexts.

Acknowledgments

The authors acknowledge the financial support received from KK-stiftelsen (The Knowledge Foundation, Stockholm, Sweden) for the Synergy research project *LITMUS: Leveraging Industry 4.0 Technologies for Human-Centric Sustainable Production* (grant no. 2024-0013).

References

[1] E. Uhlmann, J. Polte, and C. Geisert, “IoT-based energy monitoring in production,” *Measurement: Sensors*, vol. 38, p. 101591, 2025.

[2] A. Mirani, A. Awasthi, N. O’Mahony, and J. Walsh, “Industrial IoT-based energy monitoring system: Using data processing at edge,” *Internet of Things*, vol. 5, no. 4, pp. 608–633, 2024.

[3] L. Webb, M. Tokhi, and B. Alkan, “A solution architecture for energy monitoring and visualisation in smart factories with robotic automation,” in *Lecture Notes in Networks and Systems*, vol. 811, pp. 383–394, 2024.

[4] S. Fan, H. Tong, and S. Wang, “A system dynamics-based hybrid digital twin model for driving green manufacturing,” *Systems*, vol. 13, no. 8, p. 651, 2025.

[5] A. Khodadadi and S. Lazarova-Molnar, “Data-driven extraction of simulation models for energy-oriented digital twins of manufacturing systems: an illustrative case study,” in *Proceedings - Winter Simulation Conference*, pp. 1669–1680, 2024.

[6] J. Kurniasih, Z. Abas, S. Asmai, and A. Wibowo, “Process mining and simulation modeling in production systems: a review,” *Logforum*, vol. 20, no. 4, pp. 429–453, 2024.

[7] T. Hernández, N. Lehoux, and L. Gosselin, “Development of an energy monitoring and control interface for the forest products industry,” pp. 719–730, 2021.

[8] A. Loomba, G. Aaboen, P. Soni, and M. Stanko, “Transforming produced water management with a decision support system,” in *Proceedings of the Annual Offshore Technology Conference*, 2025.

[9] G. Bocewicz, G. Radzki, M. Piechowski, M. Jasiulewicz–Kaczmarek, and Z. Banaszak, “A declarative framework for production line balancing with disruption-resilient and sustainability-focused improvements,” *Sustainability*, vol. 17, no. 15, p. 6747, 2025.

[10] E. Guendouli, L. Mifdal, S. Dellagi, E. Kibbou, and A. Moufki, “Integrated production-maintenance strategy considering energy consumption and recycling constraints in dry machining,” *Journal of Cleaner Production*, vol. 469, p. 142493, 2024.

[11] Z. Ge, “Applications and challenges of artificial intelligence-driven intelligent manufacturing systems in production line automation,” in *Frontiers in Artificial Intelligence and Applications*, vol. 405, pp. 881–887, 2025.

[12] O. Yilmaz, N. Aydin, and I. Kucukkoc, “Bi-objective optimization of human-robot collaborative mixed-model multiple assembly lines considering model assignment and energy consumption,” *Journal of Computational and Applied Mathematics*, vol. 473, p. 116876, 2026.

[13] C.-T. Kuo, J.-T. Lim, and S. Meerkov, “Bottlenecks in serial production lines: A system-theoretic approach,” *Mathematical Problems in Engineering*, 1996.

[14] K. Ghatorha, P. Sharma, and A. Kang, “Optimizing carbon footprint and operational productivity of horizontal boring machines: An EFBA-based approach,” in *IOP Conference Series: Earth and Environmental Science*, 2024.

[15] G. Wang, H. Bai, D.-Q. Cang, and Y.-B. Zong, "Mathematical model for diagnosis of energy consumption bottlenecks in steel plants and its application," *Journal of University of Science and Technology Beijing*, 2009.

[16] X. Lai, H. Shui, D. Ding, and J. Ni, "Data-driven dynamic bottleneck detection in complex manufacturing systems," *Journal of Manufacturing Systems*, 2021.

[17] M. Keramati Feyz Abadi, C. Liu, M. Zhang, and Y. Xu, "Leveraging ai for energy-efficient manufacturing systems: Review and future prospectives," *Journal of Manufacturing Systems*, 2025.

[18] L. Ragazzini, E. Negri, L. Fumagalli, and M. Macchi, "Digital twin-based bottleneck prediction for improved production control," *Computers and Industrial Engineering*, 2024.

[19] I. Asghar, B. Sarkar, and S.-J. Kim, "Economic analysis of an integrated production–inventory system under stochastic production capacity and energy consumption," *Energies*, 2019.

[20] E. Cagno, D. Accordini, P. Thollander, and A. Trianni, "Energy management and industry 4.0: Analysis of the enabling effects of digitalization on the implementation of energy management practices," *Applied Energy*, 2025.