

Article

Resource-Efficient Telemetry-Based Condition Monitoring with Digitally Configurable DC/DC Converters and Embedded AI

Andreas Federl ^{1,2,*} , Markus Böhmisch ^{1,2}, Valentin Sagstetter ¹, Gerhard Fischerauer ³  and Robert Bösnecker ¹

¹ Department of Electrical Engineering and Media Technology, Deggendorf Institute of Technology, Dieter-Görlitz-Platz 1, 94469 Deggendorf, Germany; valentin.sagstetter@th-deg.de (V.S.); robert.boesnecker@th-deg.de (R.B.)

² Elec-Con Technology GmbH, Alte Straße 68, 94034 Passau, Germany

³ Chair of Measurement and Control Systems and Center of Energy Technology (ZET), Universität Bayreuth, Universitätsstr. 30, 95447 Bayreuth, Germany

* Correspondence: andreas.federl@th-deg.de

Abstract

Digitally configurable DC/DC converters provide built-in telemetry signals that offer new opportunities for operational data-driven monitoring in embedded energy systems. However, exploiting these signals for intelligent condition monitoring remains challenging due to limited computational resources and the need to preserve the safety and determinism of power supply control. This work investigates the combination of digitally configurable DC/DC converters and embedded artificial intelligence for resource-efficient load and condition monitoring based exclusively on converter-side power telemetry. A lightweight, feature-based current analysis pipeline is proposed, incorporating domain-informed temporal and electric features. Three representative machine learning model classes, Random Forest, Support Vector Machine, and a Neural Network, are evaluated. The approach is implemented on an ESP32-class microcontroller operating as a dedicated monitoring unit, fully separated from the safety-critical power supply control. Experimental validation on a laboratory demonstrator shows that classification accuracies of up to 99% can be achieved for four system states using only five features at a 100 Hz telemetry sampling rate, while remaining within typical embedded memory constraints. The results demonstrate that converter-internal telemetry enables effective and scalable condition monitoring without additional sensors, supporting the combination of embedded intelligence and digitally configurable power supplies for industrial applications.

Keywords: digitally configurable power supply; condition monitoring; load monitoring; operational data analysis; machine learning; embedded artificial intelligence



Academic Editor: Francisco Javier Ruiz-Rodríguez

Received: 2 February 2026

Revised: 12 February 2026

Accepted: 13 February 2026

Published: 18 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Digitalization is increasingly shaping modern energy and industrial systems, with a growing emphasis on data-driven monitoring, adaptability, and efficiency. While system-level concepts such as demand-side management or energy-aware control are often discussed at the grid or infrastructure level, their practical realization is facilitated by reliable operational data at the device level [1]. In this context, the International Energy Agency (IEA) highlights the potential of digital technologies to improve efficiency and operational flexibility while emphasizing reliability and cybersecurity requirements in data-driven architectures [1]. These observations call for a closer examination of digitalization at the level of power electronics systems. DC/DC converters play a central role in supplying,

conditioning, and distributing energy in modern electronic systems and increasingly expose operational data via digital configuration and telemetry interfaces, enabling data-driven analysis directly at the source [2].

Power electronics is increasingly evolving beyond the optimization of efficiency and power density toward improved configurability, monitoring capabilities, and tighter integration into system-level power management architectures. This development is driven by ongoing digitalization trends in energy and industrial systems [1] and is reflected in recent review studies that emphasize the growing role of digitally assisted converter systems with enhanced observability and software-based parameterization [3]. Accordingly, DC/DC converters are increasingly implemented using digitally controlled or mixed-signal architectures, enabling converter parameters to be modified, monitored, and coordinated in software rather than fixed hardware components [2]. This programmable configurability forms the basis for adaptive operating modes, telemetry-based supervision, and data-driven diagnostic functions across a wide range of applications [4,5]. In addition, prior research has demonstrated that condition monitoring and diagnostic tasks can be realized using intrinsic electrical signals already available within configurable power electronic systems. By exploiting current- and voltage-based features derived from existing measurements, operational deviations and degradation effects can be detected without introducing additional external sensors or modifying the physical system architecture [6]. These developments motivate the investigation of telemetry-based monitoring concepts that exploit converter-internal measurement data as a readily available information source for embedded condition monitoring and diagnostics.

A central aspect of this development is the use of real-time operational data for condition-based monitoring and early anomaly detection in power-electronic systems. Continuously acquired operational data can be leveraged to build data-driven representations of converter behavior and to detect deviations caused by component aging or parameter drift [5]. Data-driven condition monitoring and prognostics of power-electronic systems have increasingly employed deep learning techniques, particularly convolutional and recurrent Neural Network architectures. In selected studies, including CNN–LSTM-based approaches, high predictive accuracy has been demonstrated for remaining useful life estimation and fault-related pattern recognition under controlled experimental conditions [7,8]. However, several review studies highlight that end-to-end deep learning models generally demand substantial computational resources and large amounts of labeled data, while also exhibiting non-deterministic execution behavior. These factors considerably restrict their practical deployment in resource-constrained embedded systems [9–11]. As a result, lightweight model structures and feature-based representations remain a key design principle for embedded condition monitoring, particularly in safety-critical and real-time-constrained environments.

In parallel, the data-driven evaluation of operational and telemetry signals has been shown to improve fault detection and diagnostic capabilities in power-electronic systems. Machine learning approaches have been applied to identify abnormal operating patterns and fault-related transients in converters and connected components under controlled experimental conditions [12,13]. By exploiting characteristic changes in measured electrical signals, such methods enable early detection of deviations from nominal behavior and can support diagnostic functions at the system level. These developments motivate the use of converter-internal telemetry as an additional information source for condition monitoring and anomaly detection [14]. It is important to distinguish the proposed telemetry-based monitoring approach from related concepts such as non-intrusive load monitoring (NILM) and power side-channel analysis. While NILM aims to identify multiple unknown loads from aggregated power measurements at higher system levels [15], and side-channel

analysis (SCA) typically exploits high-frequency leakage phenomena for security-related inference [16], the present work focuses on device-level monitoring using converter-internal telemetry signals. These signals are digitally available within the power supply unit and represent well-defined electrical quantities such as current and voltage and also temperature, sampled at comparatively low rates. As a result, the proposed approach does not rely on blind load disaggregation or high-frequency signal acquisition but instead leverages structured, converter-provided operational data for condition monitoring of a known connected system. Beyond classical SCA, power supply monitoring has also been investigated in the context of security-related applications, where characteristic variations in supply current are analyzed to enhance device-level protection mechanisms [15]. Beyond condition and fault monitoring, power supply telemetry has also been investigated as a potential information source for security-related monitoring and anomaly detection in embedded systems [17].

The growing availability of communication and configuration interfaces has expanded the functional role of modern DC/DC converters beyond fixed-point power conversion. Digital communication enables parameterization, monitoring, and coordination within higher-level power management architectures. At the device level, this development is reflected in a shift from rigid analog control structures toward mixed-signal and software-configurable architectures. Standardized interfaces such as PMBus provide a widely adopted mechanism to access operational parameters and telemetry data and thus form a practical foundation for adaptive control strategies and data-driven diagnostic functions in digitally configurable power supplies [18]. While earlier generations of DC/DC converters predominantly relied on fixed analog control structures, microcontrollers and mixed-signal controllers have become increasingly common. Advances in semiconductor integration and decreasing costs enable sophisticated digital and hybrid control functions, including adaptive efficiency optimization, online parameter adjustment, and automated commissioning procedures [2,19]. In parallel, wide-bandgap (WBG) semiconductors such as gallium nitride (GaN) further accelerate digitally assisted converter architectures. Higher switching speeds and improved loss characteristics support increased power density and dynamic performance while imposing stricter requirements on control robustness, timing determinism, and system integration [19,20].

The ongoing industrial transformation within the frameworks of digital transformation increasingly requires intelligent data processing close to the point of data generation to enable timely and autonomous decision-making [9,21]. While cloud-based architectures offer scalable computing resources, their applicability in industrial monitoring is often limited by communication latency, bandwidth requirements, availability constraints, and security requirements. [10,11,22]. Consequently, edge-based processing has gained significant attention for time-critical and resource-efficient applications. A key enabler is Tiny Machine Learning (TinyML), which allows trained models to be executed on resource-constrained microcontrollers [9,10]. Such devices typically provide limited memory and computing resources and thus necessitate lightweight models and efficient feature representations [9,11]. By performing inference locally, latency can be reduced while improving robustness and data privacy [10,11]. Recent studies further emphasize the growing relevance of TinyML for industrial monitoring applications, highlighting trends toward increasingly autonomous and distributed intelligence at the device level [23]. In the context of power electronics, these developments motivate the use of converter-internal telemetry as a readily available data source for data-driven condition monitoring and anomaly detection [8]. To avoid interference with safety-critical control and protection and to reduce validation effort, telemetry processing and inference can be executed on a dedicated monitoring unit that is architecturally separated from the power supply control logic [24,25]. Building upon these

concepts, this work demonstrates telemetry-based, resource-efficient condition monitoring using digitally configurable DC/DC converters and embedded intelligence.

Despite the increasing availability of converter-internal telemetry (e.g., via PMBus), its systematic use for embedded condition monitoring has so far been insufficiently addressed. Existing approaches often assume additional sensing or host-level compute resources or do not explicitly consider the separation between safety-critical power control and data-driven inference. As a result, it remains largely unclear how telemetry sampling rate, feature design, and model class jointly influence classification performance and deployability on microcontroller-class devices. This work addresses this gap by proposing a telemetry-only, feature-based monitoring pipeline designed for deterministic execution and predictable memory use, and by quantifying the accuracy–memory trade-off across representative model classes under controlled operating conditions.

The main contributions of this work are as follows:

1. We present a telemetry-based analysis pipeline primarily relying on converter-side power telemetry signals for operational data-driven condition monitoring of a connected load, designed for execution on microcontroller-class hardware.
2. We evaluate three representative model classes (Random Forest, Support Vector Machine, and a Neural Network) and analyze the trade-off between classification accuracy and embedded memory footprint.
3. We demonstrate an architectural separation between power supply control and data-driven monitoring, supporting safe deployment and independent model updates in embedded systems [24,25].

In contrast to prior telemetry-based monitoring studies that primarily target host-side analysis or assume additional sensing and computing resources, this work introduces a telemetry-only pipeline explicitly optimized for microcontroller deployment. The novelty lies in systematically quantifying the trade-off among sampling rate, feature count, and model class under fixed embedded constraints and in embedding the approach in a safety-oriented architecture that isolates inference from power supply control, enabling safe deployment and independent model updates.

2. Materials

This section describes the hardware platforms, power converter architectures, and monitoring infrastructure used in this work. The focus is placed on the concrete realization of digitally configurable DC/DC converters and the associated embedded monitoring hardware, providing the basis for reproducibility of the experimental results presented in the subsequent sections.

2.1. Digitally Configurable DC/DC Converter Platforms

The hardware basis of this study consists of digitally configurable DC/DC converters from the digital intelligent Power Supply Unit (diPSU) product family manufactured by Elec-Con Technology GmbH, Passau, Bavaria, Germany [26]. These converters combine analog power control with digital configuration, monitoring, and communication interfaces. Within this platform, two hybrid converter architectures are available: Digitally Enhanced Power Analog (DEPA) [27,28] and Core Independent Peripherals (CIP) [29,30]. Both platforms provide access to internal operational parameters, including input and output voltage, input and output current, and characteristic temperature values, via a digital communication interface. These telemetry signals are continuously available at runtime and can be externally accessed for monitoring and analysis purposes.

2.2. Digitally Enhanced Power Analog (DEPA) Converter Implementation

The DEPA-based DC/DC converter employs a hybrid control architecture in which the primary voltage and current control is implemented in the analog domain, while a digital supervisory unit provides parameter configuration, monitoring, and communication capabilities [27,31]. The analog control loop operates independently of the digital subsystem, which is responsible for supervisory and interface-related functions.

The digital subsystem provides runtime access to configurable parameters such as output voltage setpoints, current limits, soft-start behavior, and protection thresholds [28]. Operational telemetry data are acquired internally and made available via the digital communication interface for external access. A representative DEPA-based buck converter from the diPSU family is shown in Figure 1, while the corresponding block diagram illustrating the separation of analog control and digital supervision is depicted in Figure 2.



Figure 1. diPSU buck converter based on the DEPA concept.

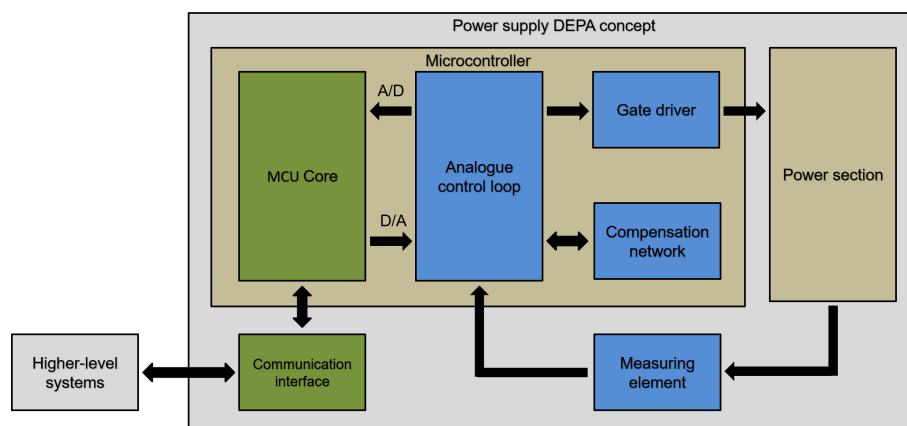


Figure 2. Block diagram of the diPSU buck converter based on the DEPA concept.

Due to the supervisory role of the digital subsystem, computational and memory resources are primarily allocated to configuration, monitoring, and communication tasks. As a result, the available digital resources are primarily dimensioned for configuration, monitoring, and communication tasks [26]. The DEPA architecture is included as a representative reference concept for digitally configurable power supplies and to contrast alternative hybrid control approaches with the CIP-based implementation used in the experimental evaluation.

2.3. Core Independent Peripheral (CIP) Converter Implementation

The CIP-based DC/DC converter is based on a hybrid architecture in which real-time control and protection functions are implemented using autonomous hardware peripherals within a microcontroller [29,30]. These peripherals operate independently of the CPU core

and can be interconnected through internal hardware signal paths. In the CIP-based implementation, functions such as pulse-width modulation, timing, protection logic, and event handling are implemented using dedicated hardware peripherals. The CPU is primarily responsible for initialization, communication, diagnostics, and higher-level system coordination [25]. This architecture is characterized by deterministic timing behavior and reduced interrupt latency compared to CPU-driven control approaches [25].

A buck-boost DC/DC converter from the diPSU product family based on the CIP concept is shown in Figure 3, and its corresponding block diagram is illustrated in Figure 4. Compared to the DEPA-based platform, the CIP implementation provides increased on-chip memory resources and supports higher telemetry update rates [26,32]. The experimental evaluation presented in this work is conducted using a CIP-based DC/DC converter.

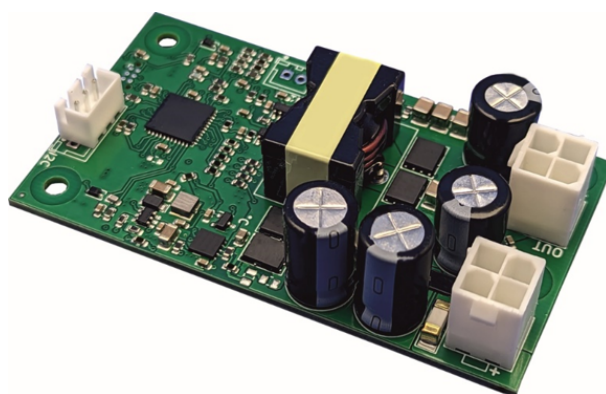


Figure 3. diPSU buck-boost converter based on the CIP concept.

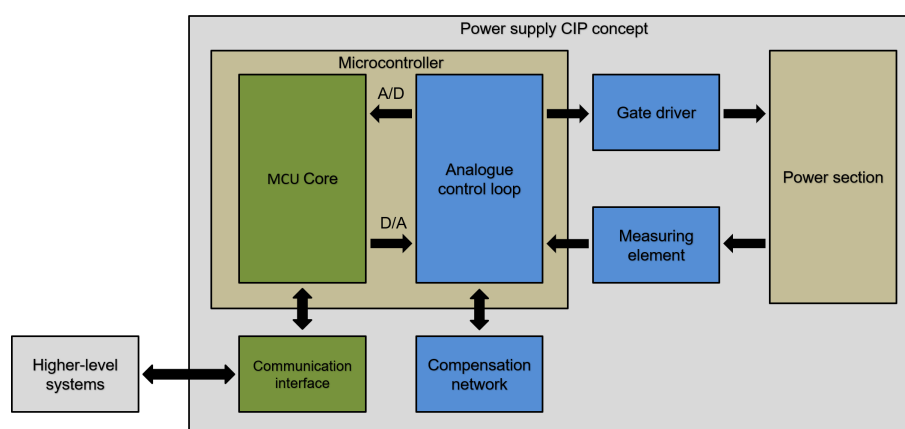


Figure 4. Block diagram of the diPSU buck-boost converter based on the CIP concept.

2.4. Comparison of DEPA and CIP Platforms

Both converter platforms support digitally configurable operation and telemetry access but differ in their underlying control architectures. The DEPA platform combines continuous-time analog control with digital supervisory functions, whereas the CIP platform is based on hardware-peripheral-driven control with limited CPU involvement [26]. In the implemented diPSU systems, DEPA-based converters provide telemetry update rates of approximately 400 Hz, whereas CIP-based converters support update rates of up to 1000 Hz depending on configuration [26,32]. In addition, CIP-based controllers offer larger available RAM and flash memory resources. A structured comparison of both architectures is summarized in Table 1. The comparison reflects architectural characteristics and implementation-related observations reported in the referenced literature and development documentation; it is not intended as a quantitative performance benchmark.

Table 1. Comparison of DEPA and CIP architectures.

Comparison Criterion	DEPA Architecture (e.g., MCP19119)	CIP Architecture (e.g., PIC16F1779)
System Architecture	Analog control loop with a digital interface for configuration and monitoring [33,34].	Autonomous hardware modules (CIPs) interconnected to form a closed-loop control system [35].
Control Domain	Continuous analog domain for error amplification and PWM modulation [34].	Continuous analog domain through the interconnection of peripherals [35].
CPU Load and Autonomy	CPU is required for monitoring, communication, and active parameter adjustments [36].	CPU is primarily used for initialization, the control loop runs core-independently in hardware [35].
Topology Flexibility	Largely optimized for fixed converter topologies [36].	Very high flexibility, enables the implementation of complex topologies like four-switch buck-boost (FSBB) on a standard platform [35].
Parameterization	Digital adjustment of predefined registers for gain, zeros, and compensation [36].	Digital configuration of electrical parameters and protection features (internal DAC registers via the I ² C interface) [35].
Scalability	Limited by the hardware-fixed peripherals of the specific controller family [36].	High; additional functions (monitoring, protection) can be added using available computing/communication resources and modules [35].
Design Complexity	Moderate; focus lies on the stability analysis of the analog PI controller [36].	High; requires detailed knowledge of Configurable Logic Cells (CLC) and hardware event sequencing [35].

2.5. Embedded Monitoring Hardware

For embedded condition monitoring of the realized prototype, an ESP32-class microcontroller from Espressif Systems (Shanghai) Co., Ltd., Shanghai, China, is used as an external monitoring unit. [37]. The monitoring unit operates independently of the power supply control hardware and provides a programmable platform for external data processing. The digitally configurable DC/DC converter supplies power to the monitored system and provides telemetry data to the monitoring unit via a digital communication interface. Power supply control and monitoring functions are implemented on separate hardware units.

2.6. Demonstrator System

The Fischertechnik Model Factory for Quality Assurance with AI serves as electrical load of the representative demonstrator system and is manufactured by fischertechnik GmbH, Waldachtal, Bavaria, Germany [38]. The model factory integrates mechanical transport, sensing, actuation, and AI-based image classification, resulting in dynamic and repeatable electrical load behavior; it is shown in Figure 5. AI-based visual quality inspection has become an important application domain in modern manufacturing systems, enabling automated classification and sorting of workpieces based on visual features [39,40]. The Fischertechnik model factory represents a scaled laboratory demonstrator of such inspection pipelines.

In the model factory, workpieces are assigned to appropriate storage locations using image recognition. Three “OK” states are distinguished by color, and there is also a state for “not OK.” An overview of the considered system states is provided in Table 2.

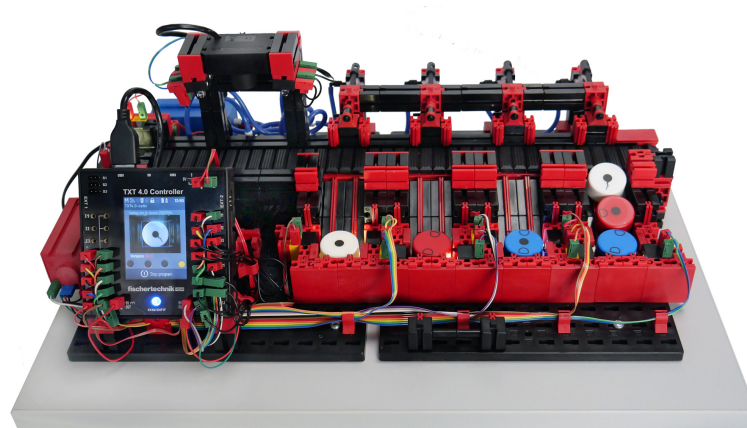


Figure 5. Fischertechnik Model Factory for Quality Assurance with AI.

Table 2. Definition of system states.

Class	System State	Description
wio	Storage position white okay	Workpiece transported and stored at position one after classification
rio	Storage position red okay	Workpiece transported and stored at position two after classification
bio	Storage position blue okay	Workpiece transported and stored at position three after classification
nio	Storage position not okay	Workpiece transported and stored at position four after classification

The demonstrator system is supplied with a CIP-based buck-boost DC/DC converter from the diPSU family. The converter provides telemetry signals, including input voltage, output voltage, input current, output current, and module temperature, which are accessible via its digital interface. Since telemetry is acquired from a converter platform used in industrial contexts, the measurements inherently include realistic converter-side noise and communication effects, although they were not isolated and quantified separately. Figure 6 illustrates the overall system architecture and the interaction between the power converter, monitoring hardware, and demonstrator system.

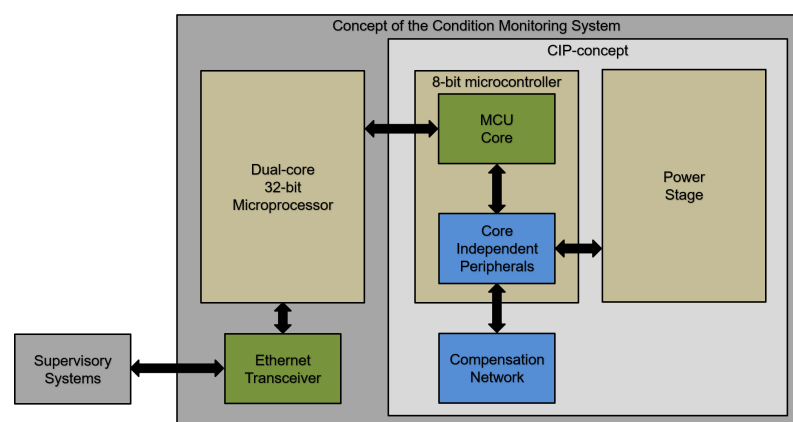


Figure 6. Overall system concept of the realized demonstrator.

The experimental system consists of a digitally configurable DC/DC converter, an external monitoring and logging unit, and the Fischertechnik model factory acting as a dynamic load. Telemetry signals are transmitted from the converter to the monitoring unit

via a digital communication interface, while power conversion and data processing remain architecturally separated to ensure safe operation.

2.7. Electrical Operating Conditions

All experiments were conducted under fixed electrical operating conditions to ensure reproducibility and comparability across all measurement runs. The digitally configurable DC/DC converter was operated with identical firmware and parameterization throughout the complete experimental campaign. The converter was supplied with a constant input voltage of 12 V and configured to an output voltage setpoint of 9 V. These operating conditions were maintained for all recorded datasets and were not modified between individual measurement runs. Digitally configurable power supplies enable precise and repeatable electrical parameterization through software-defined setpoints. This capability represents a key advantage for experimental evaluation, as identical operating points can be restored reliably across repeated measurements and extended data acquisition sessions. By maintaining constant electrical boundary conditions, the influence of external variability is minimized, allowing the investigation to focus exclusively on the effects of telemetry sampling rate, feature configuration, and model selection.

3. Methods

Based on the described hardware configuration, telemetry-based condition monitoring is evaluated using a feature-based embedded machine learning approach. This section describes the methodological framework used to analyze operational data from digitally configurable DC/DC converters and to perform embedded condition monitoring. The focus is placed on signal acquisition, preprocessing, feature extraction, and the application of machine learning models under embedded system constraints. The objective of the applied methodology is to acquire reproducible operational telemetry data from a digitally configurable DC/DC converter supplying a dynamic industrial-like load under controlled laboratory conditions. The resulting datasets are used to evaluate the influence of the telemetry sampling rate and feature selection on the classification performance and memory requirements of the models.

3.1. Data Acquisition

In the data acquisition process of this work, a clear distinction must be made between dataset generation and the inference phase used during operation. During data collection for the creation of the training dataset, external start and stop signals from the Fischertechnik model factory are employed solely to simplify the acquisition process and to ensure a precise and reproducible data basis, particularly with regard to reference labeling. These signals are not required during inference and are not used as input features for the machine learning models. In an operational monitoring scenario, cycle segmentation is performed exclusively based on converter-side current telemetry by detecting transitions between idle and active load states. This approach enables fully telemetry-based monitoring in systems where no explicit process-level synchronization signals are available. The operational data are acquired directly via the telemetry interface of the digitally configurable DC/DC converters described in Section 2. The primary signal used for analysis is the output current together with its timestamp, as it reflects the dynamic interaction between the power supply and the connected load. The digitally configurable DC/DC converter supplies power to the monitored system while continuously transmitting telemetry data via a digital communication interface. Within each segmented cycle, internal operating phases are identified directly from the current signal and are not obtained from the demonstrator's control system. Raw telemetry data were acquired at a sampling rate of 800 Hz,

and lower sampling rates of 400 Hz, 200 Hz, and 100 Hz were generated synthetically by uniform downsampling of the original dataset to ensure identical operating conditions across all configurations.

3.2. Dataset Generation Protocol

For dataset generation, a structured and repeatable measurement protocol was applied to ensure consistent operating conditions across all recorded samples. Prior to data acquisition, the demonstrator system was operated for approximately five minutes to reach thermal and operational steady-state conditions. Following the warm-up phase, consecutive process cycles of the Fischertechnik model factory were recorded under identical electrical and mechanical configurations. Each complete process cycle corresponds to one experimental run and produces a characteristic current consumption pattern at the output of the supplying DC/DC converter. Telemetry data were continuously acquired during normal system operation until the predefined number of cycles per class was reached. The classification outcome of the demonstrator determines the resulting storage position and thereby defines the associated system state label. No manual intervention or modification of system parameters was performed during the acquisition process. This protocol ensures that all datasets were generated under controlled and comparable conditions, while preserving the natural variability of the dynamic load behavior inherent to the demonstrator system.

3.3. Signal Preprocessing

The recorded electrical signals contain high-frequency noise components. To suppress these components while preserving the load-dependent dynamics, a second-order Butterworth low-pass filter with a cutoff frequency of 10 Hz is applied to the raw signal. This filtering step improves the robustness of subsequent feature extraction against measurement noise in digitally acquired telemetry signals. The analysis intentionally focuses on event- or timing-related characteristics that remain observable under bandwidth-limited telemetry. Although the applied low-pass filter limits the spectral content of the current signal to low-frequency components, the telemetry sampling rate remains a relevant design parameter for the proposed feature-based analysis. The extracted features are predominantly time-domain and event-based, including runtime estimation, peak timing, and transition localization, rather than frequency-domain characteristics. A higher sampling rate improves the temporal discretization of the filtered signal and enables more precise localization of transient events and phase boundaries in the discrete-time representation. Consequently, the benefit of increased telemetry sampling rate in this study arises primarily from improved temporal resolution rather than from the availability of higher-frequency signal components. This distinction is particularly relevant for embedded systems, where timing-related features can provide significant discriminative information under constrained signal bandwidth.

3.4. Segmentation of Process Cycles

The Fischertechnik model factory operates in discrete process cycles whose duration depends on mechanical movements, sensor feedback, and an internal AI-based image classification. Consequently, the recorded current signals do not represent stationary time series but consist of segments with variable length, each corresponding to a single process execution. For dataset generation, each process cycle is segmented using the factory's start and stop signals to ensure reproducible temporal boundaries. These signals are provided by the demonstrator's control system and are used exclusively for offline labeling and evaluation. The resulting time-series segments form the basis for feature extraction and subsequent classification. While external start and stop signals are applied

to delimit complete process cycles, the identification of internal process phases, such as the classification interval, is performed solely based on characteristic patterns in the current signal. Although external synchronization signals are used in this study to guarantee reproducible segmentation during dataset generation, equivalent cycle boundaries can also be derived directly from the current signal by detecting transitions between idle and active operating states. This enables fully telemetry-based operation in scenarios where explicit process-level synchronization is not available.

3.5. Feature-Based Time-Series Representation

To enable efficient embedded inference, a feature-based analysis approach is adopted instead of directly processing raw time-series data. Feature extraction reduces data dimensionality, improves robustness, and significantly lowers computational and memory requirements compared to end-to-end waveform-based models. General-purpose automated feature extraction frameworks for time-series analysis, such as tsfresh, have been widely applied in data-driven monitoring tasks [41]. However, their computational complexity and large feature sets typically exceed the constraints of microcontroller-class systems, motivating the use of domain-informed and lightweight feature designs in this work. The following features are used in this study:

- Runtime represents the temporal duration of a complete processing cycle. It is calculated as the product of the total number of acquired data samples and the sampling interval.
- Total Electric Charge refers to the cumulative electrical charge transferred during a single operation cycle. It is obtained by numerical integration of the measured output current over time.
- Maximum Peak Time indicates the exact temporal position within the recorded time series at which the current signal reaches its global maximum value.
- Peak Count represents the total number of current peaks observed within the signal and reflects the frequency of characteristic load transitions during the operational process.
- Classification Time describes the temporal duration of the internal image-based classification process of the Fischertechnik model factory; this is determined from the characteristic current curve.

A detailed statistical summary of the observed characteristics is presented in Table 3. It should be noted that the Classification Time feature represents a system-specific characteristic of the demonstrator setup. While this feature is derived exclusively from converter-side current telemetry, its interpretation relies on prior knowledge of the operational sequence of the Fischertechnik model factory. The inclusion of this feature does not require access to external process signals or controller-level information, but it is not intended as a universally transferable indicator. Rather, it serves to evaluate the achievable classification performance when domain-specific knowledge is available. More generic features, such as runtime, electric charge, and peak-related metrics, remain applicable across different load scenarios.

Table 3. Statistical summary of extracted features.

Feature	Min	Max	Mean $\pm$ Std
Runtime [s]	5.49	9.82	7.29 $\pm$ 0.98
Total Electric Charge [C]	1.68	3.24	2.24 $\pm$ 0.30
Maximum Peak Time [s]	2.80	9.54	6.99 $\pm$ 0.96
Peak Count [-]	146	303	221 $\pm$ 33
Classification Time [s]	2.04	6.41	4.84 $\pm$ 0.23

3.6. Feature Scaling, Normalization and Feature Relevance Analysis

For machine learning models that are sensitive to feature scaling, namely SVMs and Neural Networks, Min–Max normalization is applied to map each feature to the interval $[0, 1]$. Tree-based models such as Random Forests are invariant to monotonic feature scaling and are therefore trained using unnormalized feature values [42]. To assess the contribution of individual features to classification performance, feature relevance is evaluated using the Gini impurity criterion of a Random Forest classifier, with the results reported in Table 4 [42]. This analysis provides insight into which features contribute most strongly to class separability and supports informed feature selection for embedded deployment.

Table 4. Normalized mean decrease in Gini impurity 100 Hz.

Feature	Normalized Gini Decrease
Maximum Peak Time	0.36
Runtime	0.31
Total Electric Charge	0.23
Peak Count	0.07
Classification Time	0.03

Feature relevance analysis is shown exemplarily for 100 Hz, similar trends were observed at higher sampling rates.

3.7. Machine Learning Models

Three machine learning model classes are evaluated in this work: a Random Forest, a SVM, and a fully connected Neural Network. All models are trained offline on a host system and subsequently deployed using platform-specific representations. The Random Forest and SVM models are trained using the scikit-learn machine learning framework (version 1.5.0), which provides standardized and well-established reference implementations of classical learning algorithms [43]. For embedded deployment, both models are converted into static C representations using microMLgen (version 1.1.28), a Python (version 3.12.2) library that creates finished C header files with a predict function for scikit-learn models. The Neural Network is trained using TensorFlow (version 2.17.0) and converted to the TensorFlow Lite model format (version tflite Micro 1.3.1). TensorFlow Lite is a smaller version of TensorFlow developed for deploying TensorFlow models to embedded platforms. An even more lightweight version is TensorFlow Lite Micro, which is a TensorFlow Lite implementation specifically for microcontroller architectures.

3.7.1. Random Forest Classifier

The Random Forest classifier consists of an ensemble of decision trees trained on randomly sampled subsets of the data and feature space [42]. The final prediction is obtained via majority voting across all trees. In this work, the model is constrained to 10 trees with a maximum depth of 10 and a maximum of 10 leaf nodes per tree to limit memory consumption and ensure feasibility on embedded hardware.

3.7.2. Support Vector Machine Classifier

A SVM with a Radial Basis Function (RBF) kernel is employed to model non-linear decision boundaries [44]. An SVM grid search was conducted separately for each telemetry sampling rate for the minimum and maximum feature-set sizes (two and five features) to identify the best-performing hyperparameters for each rate. The resulting optimal parameters are reported in Table 5.

For the comparison across sampling rates and feature-set sizes, a single fixed SVM configuration ($C = 1000$, $\gamma = 10$) was intentionally used instead of optimizing hyperparameters for each configuration individually. The value of C was selected based on the grid search results of the two-feature configurations, as this choice also keeps the memory requirements broadly comparable to those of the other two models. This approach enhances the comparability of the results and supports an application-oriented evaluation. By avoiding configuration-specific tuning, the analysis does not introduce an additional degree of freedom, enabling a clearer interpretation of the observed performance trends. For embedded deployment, the trained SVM model is converted into optimized C code using microMLgen, enabling efficient inference without dynamic memory allocation.

Table 5. Grid-search results: SVM hyperparameters for different sampling rates and feature counts.

Sampling Rate	Number of Features	C	γ	Kernel
100 Hz	2	1000	10	rbf
200 Hz	2	1000	10	rbf
400 Hz	2	1000	10	rbf
800 Hz	2	1000	10	poly
100 Hz	5	10	10	rbf
200 Hz	5	10	10	rbf
400 Hz	5	1	10	rbf
800 Hz	5	0.1	10	poly

3.7.3. Neural Network Classifier

The Neural Network is implemented as a fully connected multilayer perceptron using the TensorFlow framework [45]. The architecture comprises two hidden layers with 32 and 64 neurons, respectively, using ReLU activation functions, followed by a Softmax output layer. After training, the model is converted to TFLite Micro format and optimized for inference on microcontroller-class hardware following established TinyML deployment practices. The neural network architecture was chosen to satisfy the constraints of TinyML deployment and to ensure comparability with the other models in terms of memory footprint and inference cost, rather than performing an extensive hyperparameter search for each evaluated configuration.

3.8. Embedded Deployment Considerations

The input data consists of the converter's time domain output current signal, from which a deterministic feature extraction step is applied prior to classification. The machine learning models operate exclusively on the resulting feature vectors and do not process raw time-series data directly. To investigate the influence of input dimensionality on memory consumption and classification performance, four feature configurations are evaluated with $d \in \{2, 3, 4, 5\}$. The complete feature set comprises the Runtime, Total Electric Charge, Maximum Peak Time, Peak Count, and Classification Time of the factory. This order also corresponds to the structure of the feature set. All models perform a four-class classification task, corresponding to four distinct system states of the Fischertechnik model factory. Feature extraction is identical for all models to ensure a fair comparison. All preprocessing, feature extraction, and inference steps are designed to operate within the computational and memory constraints of an ESP32-class microcontroller. Model conversion and optimization are performed using TensorFlow Lite and microMLgen [37,46,47]. To ensure system safety and maintainability of the demonstrator, all machine learning inference is executed on a dedicated monitoring microcontroller, fully separated from the safety-critical power supply control logic [24,25].

3.9. Evaluation Protocol and Memory Analysis Methodology

Model evaluation is performed using five-fold cross-validation across all four classes. For each fold, the dataset is partitioned into training and validation subsets while preserving class balance. Model hyperparameters are selected based on validation performance within the cross-validation procedure. Reported classification results correspond to the mean and standard deviation of classification accuracy across all five folds. In addition to classification performance, confusion matrices are evaluated to analyze class-wise misclassification behavior.

This paper deliberately focuses on classification performance and static deployability constraints (flash footprint). Measuring feature-extraction time, inference latency, peak RAM/stack usage, and energy consumption on target hardware is essential for a full real-time assessment but requires dedicated embedded benchmarking and power-measurement instrumentation and is therefore treated as out of scope for this publication. These aspects will be addressed in future work focusing specifically on device timing and energy measurements.

The memory analysis focuses exclusively on the static storage requirements of model parameters required for inference. Memory usage related to training, runtime activations, intermediate buffers, stack usage, and framework overhead is explicitly excluded. The focus on static model parameters reflects a design-relevant metric for embedded deployment, as flash memory consumption directly limits model complexity on microcontroller-class hardware. Runtime memory requirements strongly depend on the selected inference framework, compiler configuration, and platform-specific buffering strategies and are therefore not considered comparable across systems. Two complementary memory perspectives are considered:

1. Host-based model storage refers to the persistence of trained models as serialized Python objects or parameter arrays on a development system, typically using pickle for evaluation [48]. The resulting file encapsulates model parameters, hyper-parameters and object-related metadata. However, this approach introduces significant storage overhead due to object structure and type information and requires the corresponding runtime environment and library versions for successful deserialization, thereby limiting portability and reproducibility.
2. Embedded model storage for deployment on microcontrollers involves converting trained models into a static, platform-adapted representation during an offline compilation step. Only the numerical model parameters, such as feature indices, decision thresholds, or weights, are retained, while all dynamic structures and runtime dependencies are removed. The resulting parameter arrays are stored compactly in non-volatile memory (flash) and accessed directly during inference, eliminating deserialization and runtime overhead and enabling deterministic execution under strict memory constraints.

4. Results

This section presents the experimental results obtained from telemetry-based condition monitoring of the Fischertechnik model factory. The objective is to quantify classification performance and model memory requirements as a function of telemetry sampling rate, feature set size, and machine learning model class under identical experimental conditions.

4.1. Evaluation Metrics and Reporting Convention

Classification performance is quantified using overall classification accuracy. All reported accuracy values correspond to the mean and standard deviation obtained from five-fold cross-validation. To ensure consistent comparison across experiments, all models

were evaluated using identical datasets, feature extraction procedures, and class definitions as described in Section 3. Process-phase signals provided by the fischertechnik model are used exclusively for dataset generation and are not required for power supply operation or telemetry-based condition monitoring.

In addition to overall classification accuracy, we report a confusion matrix for a representative configuration. For a fixed test-split fold (Random Forest, 400 Hz—5F), the confusion matrix in Figure 7 reports the sample counts for each combination of ground-truth and predicted class labels.

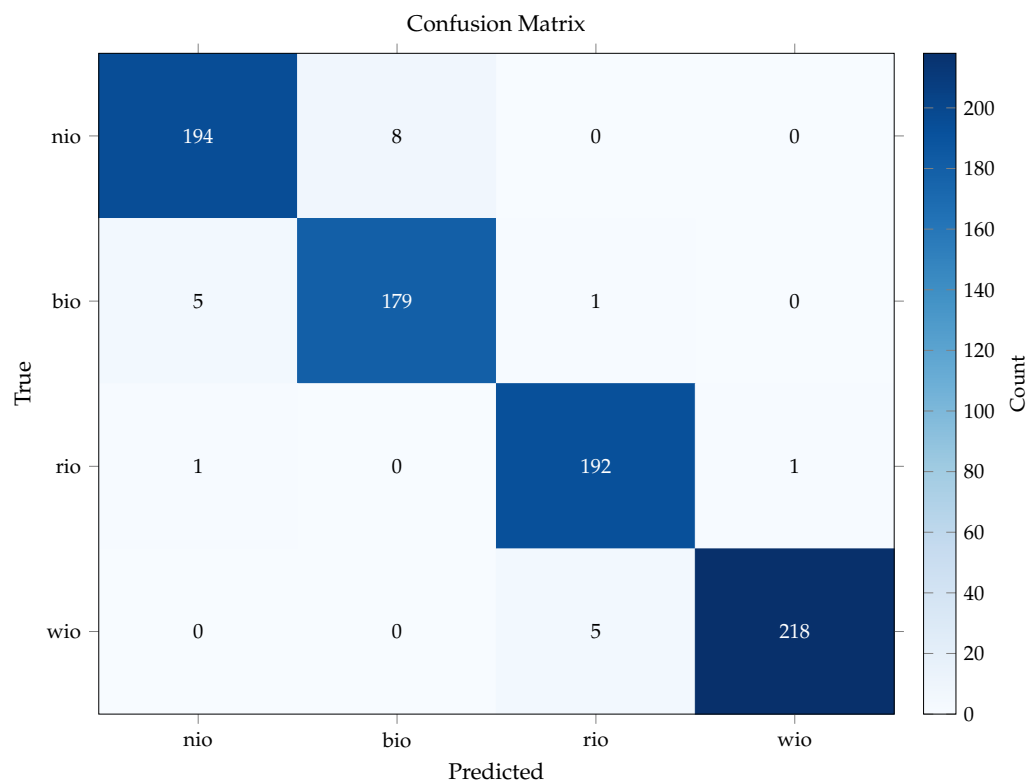


Figure 7. Confusion matrix for the Random Forest model (example fold, 400 Hz—5F).

The corresponding Table 6 provides class-wise precision, recall, and F1-score, as well as the macro-averaged and weighted-averaged F1-scores, all computed on the same fold. Random Forest 400 Hz—5F was selected for illustrative purposes because it exhibits a substantially lower accuracy compared to the other two models.

Table 6. Class-wise precision, recall, and F1-score for the Random Forest model (400 Hz—5F, 5-fold CV, fixed test split).

Class	Support	Precision	Recall	F1
nio	202	0.9700	0.9604	0.9652
bio	185	0.9572	0.9676	0.9624
rio	194	0.9697	0.9897	0.9796
wio	223	0.9954	0.9776	0.9864
Macro-F1				0.9734
Weighted-F1				0.9739

4.2. Dataset Overview

For each of the four storage positions of the model factory, 1005 process cycles were recorded, resulting in four balanced classes. All measurements were acquired at a telemetry sampling rate of 800 Hz. Lower sampling rates of 400 Hz, 200 Hz, and 100 Hz were

subsequently generated synthetically by downsampling the original 800 Hz dataset to ensure identical operating conditions across all configurations. All datasets were collected under identical laboratory conditions and processed using the feature extraction pipeline described in Section 3. An overview of the entire dataset is presented in Table 7. No data augmentation or class rebalancing techniques were applied. The four classes investigated in this study correspond to distinct system states of the Fischertechnik model factory, which arise from different storage positions selected after the internal image-based classification process. Each class therefore represents a unique sequence of mechanical movements and actuator activations, resulting in characteristic current consumption patterns at the power supply output.

Table 7. Tabular overview of the dataset.

Parameter	Value
Number of classes	4
Process cycles per class	1005
Total number of samples	4020 cycles
Sampling rate (raw)	800 Hz
Derived sampling rates	400 Hz, 200 Hz, 100 Hz
Average cycle duration	7.29 s
Warm-up duration	5 min
Class balance	fully balanced

All features were computed exclusively within individual process cycles. No temporal overlap or shared samples occur between training and validation folds, ensuring strict separation of data across cross-validation splits.

4.3. Classification Accuracy Results

Figure 8 presents the classification accuracy obtained for all evaluated models as a function of telemetry sampling rate and number of extracted features. Reported values correspond to the mean accuracy across five cross-validation folds, together with the associated standard deviation. The results allow an assessment of the individual and combined influence of sampling rate and feature configuration on classification performance.

Using the system-consistent class order (wio–rio–bio–nio), the confusion matrix (Figure 7) shows pronounced diagonal dominance (783/804 correct; accuracy 97.39%). Aggregate metrics confirm robust and balanced performance (Macro-F1 = 0.9734; Weighted-F1 = 0.9739). Class-wise results remain uniformly high, with the strongest performance for wio (F1 = 0.9864) and rio (F1 = 0.9796) and slightly lower yet still strong scores for bio (F1 = 0.9624) and nio (F1 = 0.9652), consistent with boundary-related confusions. Importantly, 20 of 21 errors (95.24%) occur between adjacent system states, suggesting that residual mistakes reflect local decision-boundary overlap rather than broad class confusion.

4.3.1. Results Random Forest

For the Random Forest classifier, classification accuracy generally increases with the number of features used. The influence of higher sampling rates on classification accuracy is rather low. The highest accuracies are observed at sampling rates of 400 Hz and 800 Hz with five extracted features, where performance differences between both rates remain small and largely within the range of the observed standard deviations. Regarding feature count, the inclusion of additional features generally improves classification accuracy. The inclusion of the fifth feature further increases the classification accuracy significantly.

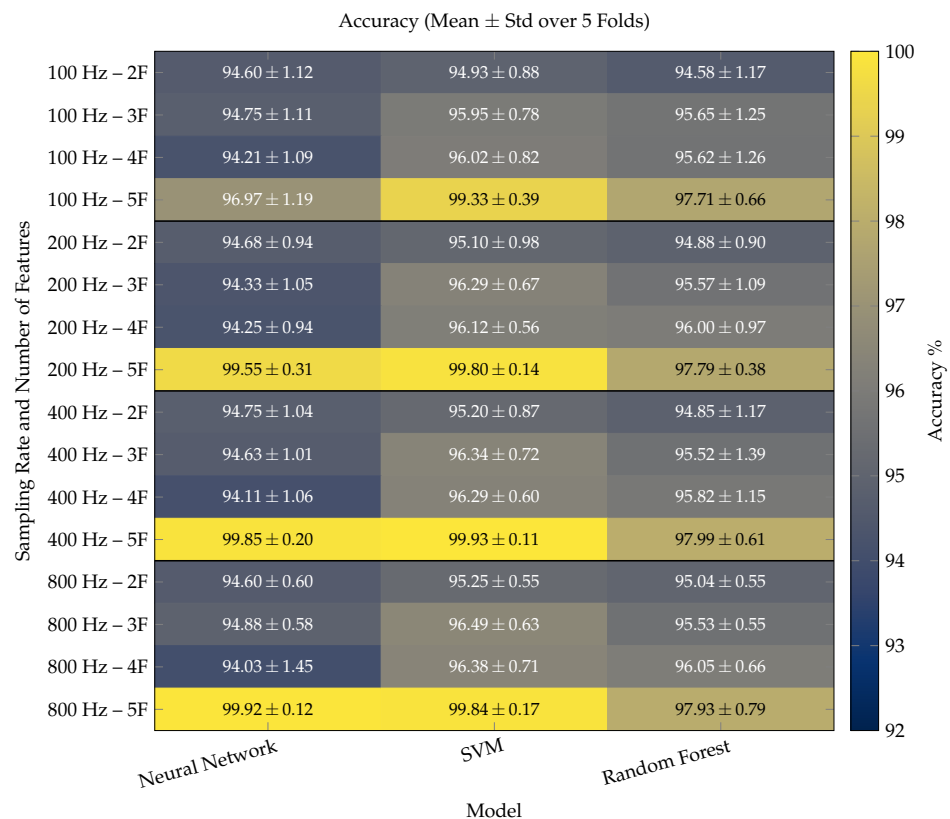


Figure 8. Classification accuracy as a function of telemetry sampling rate and number of extracted features for all evaluated model classes. Reported values represent the mean accuracy across five-fold cross-validation together with the corresponding standard deviation.

4.3.2. Results Support Vector Machine

In terms of classification accuracy, the SVM performs better than the Random Forest, which is particularly evident when all features are used, regardless of the data rate. The influence of the number of features is particularly evident in the transition from four to five features. It can also be observed here that the standard deviation is lowest with five features. At 100 Hz, the SVM outperforms the Random Forest and Neural Network across all evaluated feature configurations, demonstrating a greater robustness to reduced temporal resolution. With increasing sampling rate, classification accuracy further improves and approaches a plateau at 400 Hz and 800 Hz. Differences between these two sampling rates remain small, suggesting diminishing returns from further increases in temporal resolution.

4.3.3. Results Neural Network

Below the five feature configuration, the classifier based on a Neural Network lags behind the results of the other two models at all data rates, and there are no significant improvements due to higher data rates. Only when using five features and a 200 Hz data rate is a classification accuracy comparable to that of SVM achieved. The variability across cross-validation folds decreases notably for configurations achieving the highest accuracies, suggesting more stable model behavior under these conditions.

4.3.4. Influence of Sampling Rate and Feature Count

Across all models examined, classification accuracy varies with both the telemetry sampling rate and the number of features. Different combinations of sampling rate and feature configuration lead to different levels of accuracy depending on the model, with the respective effects strongly dependent on the model chosen. In general, increasing the

sampling rate often leads to only comparatively minor improvements, especially when increasing from 400 Hz to 800 Hz. Increasing from 100 Hz to 200 Hz, on the other hand, brings a clearly visible improvement. The number of features significantly influences the performance of the classification, especially the use of five features. Higher telemetry sampling rates are associated with increased classification accuracy across most of the configurations examined, even though identical low-pass filtering was applied to all signals. This behavior is consistent with the preprocessing strategy described in Section 3.3. Even under low-pass filtered conditions, higher sampling rates enable more precise localization of transient events and time-related features, which play a central role in the proposed feature-based classification approach.

4.4. Model Memory Footprint

To evaluate the suitability of the proposed models for embedded deployment, model memory requirements were analyzed from two complementary perspectives.

4.4.1. Host-Based Model Storage Footprint

The host-based model storage footprint represents the serialized size of the trained model representations obtained on the development system. The corresponding results are shown in Figure 9. For the Random Forest and Neural Network models, memory consumption remains nearly constant across different sampling rates, as model size is primarily determined by fixed structural configurations and hyperparameters. In contrast, the SVM exhibits notable variations in storage requirements, which can be attributed to differences in the number of support vectors retained after training. The values are highest for Random Forest, with SVM falling below the five feature configuration between Random Forest and Neural Network. From the five-feature configuration onward, the value is lowest for SVM.

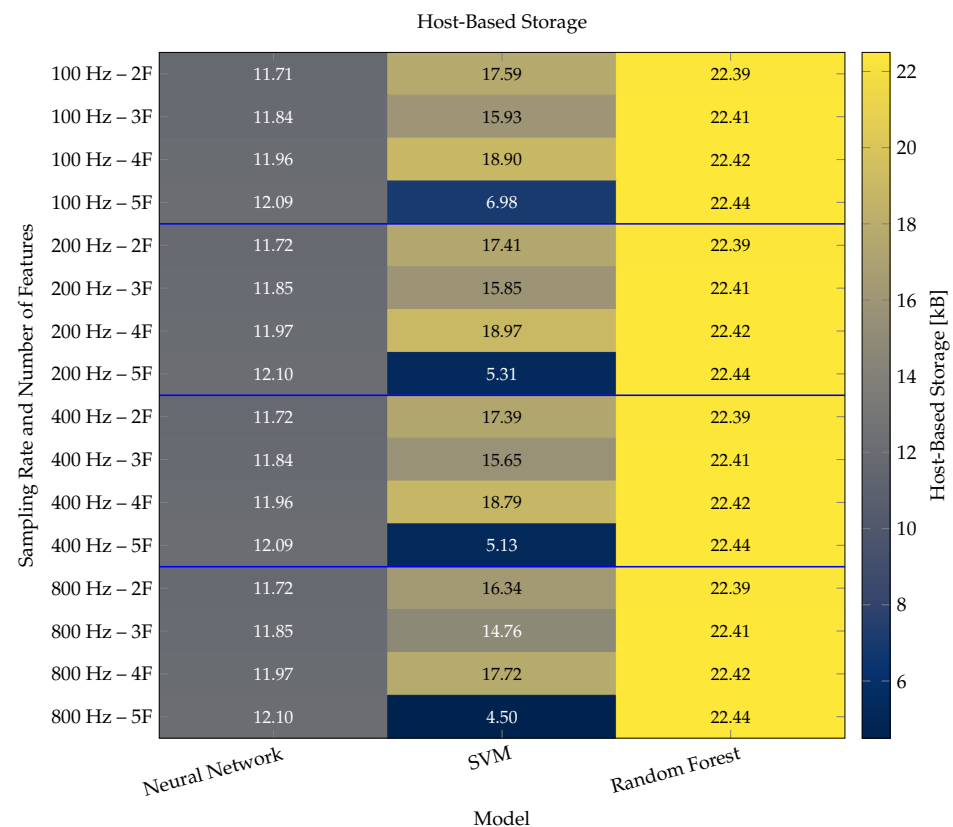


Figure 9. Host-based model storage footprint of the evaluated machine learning models for different feature configurations and telemetry sampling rates.

4.4.2. Embedded Flash Memory Footprint

The embedded flash memory footprint denotes the non-volatile memory required by platform-specific static model representations generated for microcontroller deployment. The corresponding results are shown in Figure 10. As expected, telemetry sampling rate does not influence embedded model size of Random Forest and Neural Network, since learned model parameters are independent of temporal data resolution. Observed differences between configurations arise primarily from model structure and feature dimensionality, which indirectly affect the number of stored model components, particularly in the case of the SVM. Below the five feature configuration, the size of the SVM is again greater than that of the Random Forest and Neural Network. When considering the five feature configuration, the value of the SVM is the lowest across all sampling rates considered.

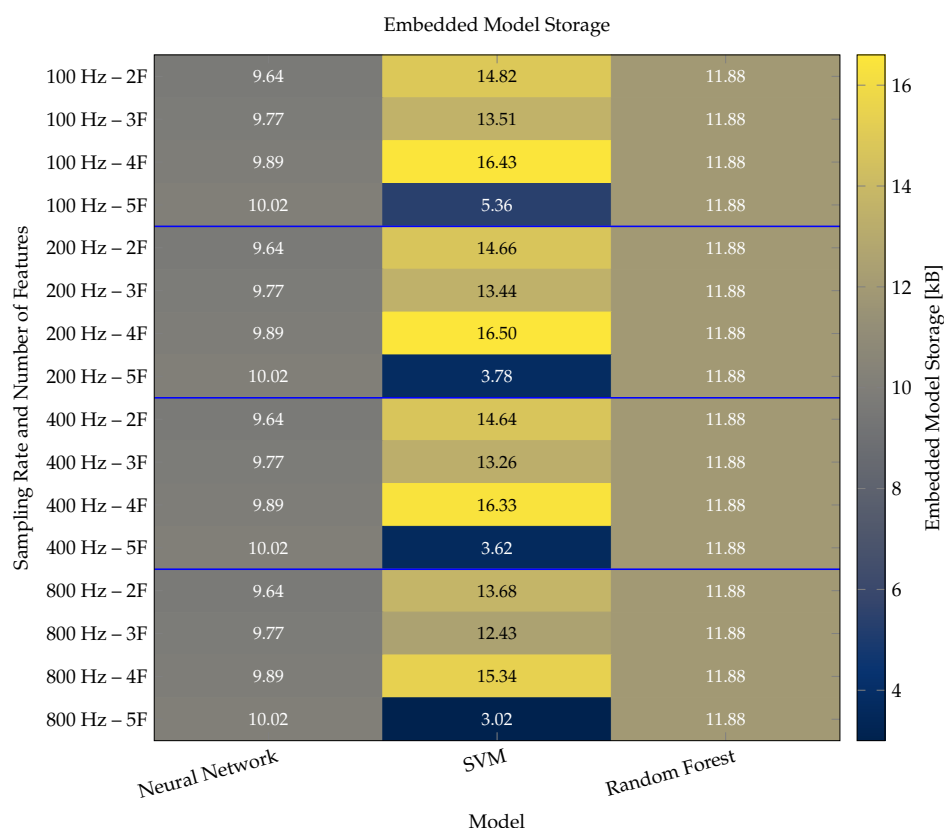


Figure 10. Embedded flash memory footprint of the deployed model representations on microcontroller-class hardware for different feature configurations and telemetry sampling rates.

5. Discussion and Conclusions

This section discusses the experimental results presented in Section 4 and places them within the context of telemetry-based condition monitoring using digitally configurable DC/DC converters. The discussion focuses on the influence of telemetry sampling rate, feature selection, model characteristics, and architectural design choices, followed by limitations and future research directions.

5.1. Interpretation of Classification Performance

The results demonstrate that operational telemetry acquired directly from digitally configurable DC/DC converters contains sufficient information to distinguish discrete system states of a connected load. Similar observations have been reported in prior work on telemetry-based monitoring and power-side signal analysis, where characteristic load-dependent patterns are reflected in electrical quantities [8,15,16]. The observed performance

confirms that characteristic timing information and transient load behavior are preserved in current measurements, even when no additional external sensors are employed.

Based on the exemplar confusion matrix, the observed predominance of confusions between neighboring states (e.g., wio to rio and bio to nio) is compatible with the physical proximity of these system regimes. This local error pattern suggests that class-conditional feature distributions overlap primarily near transition boundaries, indicating that residual errors are more likely attributable to thresholding and transition-region effects than to systematic confusion of non-adjacent states.

5.2. Impact of Telemetry Sampling Rate

The telemetry sampling rate affects both the classification accuracy and, in the case of the SVM, the model memory footprint. The smallest SVM storage size is observed at a sampling rate of 800 Hz when using five features. In particular, the Neural Network model exhibits a pronounced improvement in classification accuracy when increasing the sampling rate from 100 Hz to 200 Hz with five selected features. Similar trends are observed for the other two models, where the most significant performance gains occur within this sampling rate range. Further increases in sampling rate, especially from 400 Hz to 800 Hz, result in only minor additional improvements.

5.3. Role of Feature-Based Representation

The feature-based representation employed in this work enables efficient inference under embedded constraints while retaining discriminative information. Feature extraction has been widely recognized as a key design principle in TinyML systems, where computational and memory resources prohibit direct processing of high-dimensional raw signals [9–11]. The experimental results demonstrate that domain-informed temporal and electrical features provide sufficient discriminative information for reliable classification under the investigated conditions, which is consistent with established approaches in time-series feature engineering and embedded machine learning [23,41]. Although end-to-end deep learning approaches based on raw time-series data, such as CNN or CNN-LSTM architectures, have demonstrated strong performance in condition monitoring and prognostics, their direct application was not the primary objective of this work but remains an important direction for future work and will be evaluated under identical segmentation and cross-validation splits to enable a fair comparison.

Such models typically require significantly higher computational resources, memory capacity, and training data volume, which limits their applicability in resource-constrained microcontroller environments. The focus of this study lies on resource-efficient embedded inference using converter-side telemetry that is readily available in digitally configurable power supplies. Feature-based representations enable deterministic computational complexity, predictable memory requirements, and transparent interpretability, which are critical design aspects for safety-related and industrial embedded systems. Consequently, the proposed approach prioritizes lightweight feature extraction combined with classical machine learning models to evaluate the achievable trade-off between classification performance and embedded deployability. The investigation of end-to-end time-series learning approaches remains an important direction for future work, particularly for platforms with increased computational capabilities.

When looking at classification accuracy, it becomes apparent across all three models that increasing the number of features used has a significantly greater impact on accuracy than a higher sampling rate.

5.4. Comparison of Machine Learning Models

The observed differences between model classes reflect well-known characteristics of the respective algorithms. SVMs are known for strong generalization performance in low-dimensional feature spaces, particularly when decision boundaries are non-linear [44]. Neural Networks provide increased representational flexibility but typically require higher model complexity and memory resources to achieve comparable performance [9,11]. Under the applied structural constraints, the Random Forest classifier shows limited representational flexibility compared to the SVM and Neural Network models [42].

5.5. Architectural Implications for Embedded Monitoring

The architectural separation between power supply control and data-driven monitoring represents an important design principle for safety-critical embedded systems. Prior work has emphasized that isolating machine-learning inference from real-time control loops significantly reduces validation complexity and operational risk [24,25]. All features used in this work are computed exclusively from converter-side current telemetry. However, their definition and interpretation are informed by prior knowledge of the system's operational structure, which enables meaningful segmentation and feature extraction without requiring additional sensors. Edge-based analytics executed on dedicated monitoring hardware further align with established embedded intelligence and TinyML concepts, enabling low-latency inference while maintaining system robustness and maintainability [9,10,22].

5.6. Limitations and External Validity

All reported results were obtained on a single laboratory demonstrator (Fischertechnik model factory) supplied by a single converter type under fixed electrical parameterization. Consequently, the study does not yet demonstrate transfer to other load categories, other converter topologies, or industrial environmental conditions. This limits the degree to which absolute accuracy values can be generalized. The primary contribution of this work is the controlled analysis of sampling-rate, feature, and model-class trade-offs under identical conditions. Generalization across systems requires additional datasets and validation campaigns. All experiments were conducted under controlled laboratory conditions with identical converter firmware, electrical parameterization, and demonstrator configuration. While the employed system represents a scaled laboratory setup, it captures essential characteristics of industrial automation processes, including dynamic load behavior and discrete operational phases. The presented results are obtained under controlled laboratory conditions with a limited number of predefined system states. While this enables systematic investigation of telemetry sampling rate and feature selection, transfer to industrial environments may require retraining and validation under increased variability, including temperature drift, aging effects, and multi-load interactions. In addition, the behavior under communication-related non-idealities and under systematic sensor drift was not explicitly quantified in this study. Future work will therefore include dedicated robustness tests under controlled temperature variation, induced offset/gain drift, and emulated communication disturbances to assess their impact on feature stability and classification performance. While absolute classification accuracy values are therefore system-specific, the observed trends regarding telemetry sampling rate, feature dimensionality, and model selection are expected to generalize to other telemetry-based monitoring scenarios. While flash footprint is a necessary feasibility constraint for microcontroller deployment, a complete real-time qualification additionally requires measurements of feature extraction time, inference latency, peak RAM/stack usage, and energy per inference on the target hardware. Providing such benchmarks is beyond the scope of this manuscript and will be addressed in future work.

Furthermore, this study does not include an end-to-end deep learning baseline (e.g., a 1D-CNN or CNN-LSTM) trained directly on raw telemetry time series. End-to-end architectures are commonly evaluated in related condition-monitoring settings; however, this work focuses on feature-based representations to enable deployment on microcontroller-class hardware with predictable inference latency and constrained resources. As a result, the empirical comparison is limited to feature-based approaches. Future work will evaluate compact 1D-CNN and CNN-LSTM baselines under the same experimental protocol.

5.7. Conclusions and Outlook

Using only converter-side telemetry, classification accuracies of over 99% were achieved, particularly with SVM, which was already possible at a data rate of 100 Hz. In this case, SVM also requires significantly less memory than Random Forest and Neural Network. The investigations clearly show that process-specific features have a significantly greater influence on classification accuracy than higher data rates. Thus, by selecting suitable features, it is possible to implement very resource-efficient and telemetry-based monitoring of connected loads using digitally configurable power supplies. The findings support current trends toward distributed edge intelligence and TinyML-based monitoring architectures, which aim to exploit existing data sources close to the physical process while minimizing system overhead [9–11,21,22]. Further research will quantify feature-extraction runtime, inference latency, and energy consumption for both feature calculation and model inference across multiple microcontroller and system-on-a-chip platforms. The approach will be validated using additional loads and converter types under more diverse environmental conditions. In addition, a future comparison with neuromorphic hardware is conceivable to assess potential benefits in terms of latency and energy efficiency.

Author Contributions: Conceptualization, A.F., R.B. and G.F.; data curation, V.S. and A.F.; formal analysis, A.F. and R.B.; investigation, A.F., M.B. and V.S.; methodology, A.F. and M.B.; project administration, R.B. and A.F.; validation, A.F., V.S. and G.F.; visualization, M.B. and A.F.; writing—original draft, A.F., M.B. and V.S.; writing—review and editing, R.B. and G.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets generated and analyzed during the current study are publicly available in a data repository under <https://doi.org/10.57880/rdspace-ubt-50>.

Acknowledgments: The authors gratefully acknowledge the support of the Bavarian State Ministry of Education, Science and the Arts within the framework Graduiertenkolleg Energieautarke Gebäude of the TechnologieAllianz Oberfranken (TAO). The authors used AI-based language models (ChatGPT 5.2 and Gemini 3) solely for language editing and improvement of readability. All scientific content, interpretations, and conclusions are entirely the responsibility of the authors.

Conflicts of Interest: Andreas Federl and Markus Böhmisch report part-time employment with Elec-Con Technology GmbH. All other authors declare no commercial or financial relationships that could be perceived as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADC	Analog-to-Digital Converter
AI	Artificial Intelligence
CIP	Core Independent Peripherals
CLC	Configurable Logic Cell
CNN	Convolutional Neural Network

CPU	Central Processing Unit
DAC	Digital-to-Analog Converter
DC	Direct Current
DEPA	Digitally Enhanced Power Analog
diPSU	digital intelligent Power Supply Unit
DSP	Digital Signal Processor
ESP32	ESP32-class microcontroller (Espressif Systems)
FSBB	Four-Switch Buck-Boost
GaN	Gallium Nitride
GPIO	General-Purpose Input/Output
Hz	Hertz
I ² C	Inter-Integrated Circuit
ILM	Intrusive Load Monitoring
IoT	Internet of Things
kB	kilobyte
LSTM	Long Short-Term Memory
MCU	Microcontroller Unit
NILM	Non-Intrusive Load Monitoring
PI	Proportional–Integral (controller)
PMBus	Power Management Bus
PWM	Pulse-Width Modulation
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RUL	Remaining Useful Life
SCA	Side-Channel Analysis
Std	Standard deviation
TinyML	Tiny Machine Learning
WBG	Wide-Bandgap

References

1. International Energy Agency (IEA). *Digitalisation and Energy*; IEA: Paris, France, 2017. Available online: <https://www.iea.org/reports/digitalisation-and-energy> (accessed on 19 December 2025).
2. Liu, Y.; Meyer, E.; Liu, X. Recent Developments in Digital Control Strategies for DC/DC Switching Power Converters. *IEEE Trans. Power Electron.* **2009**, *24*, 2567–2577. [[CrossRef](#)]
3. Sarvi, M.; Zohdi, H.Z. A Comprehensive Overview of DC-DC Converters Control Methods and Topologies in DC Microgrids. *Energy Sci. Eng.* **2024**, *12*, 2017–2036. [[CrossRef](#)]
4. Kumaraguruparan, S.; Konguvel, E. Optimal Control Strategies for High-Efficiency Non-Isolated DC-DC Buck Converters in IoT Applications: A Comparative Study. *Heliyon* **2024**, *10*, e38119. [[CrossRef](#)] [[PubMed](#)]
5. Lei, Z.; Zhou, H.; Dai, X.; Hu, W.; Liu, S. Digital Twin Based Monitoring and Control for DC-DC Converters. *Nat. Commun.* **2023**, *14*, 5604. [[CrossRef](#)]
6. Yüce, F.; Hiller, M. Condition Monitoring of Power Electronic Systems Through Data Analysis of Measurement Signals and Control Output Variables. *IEEE J. Emerg. Select. Top. Power Electron.* **2022**, *10*, 5118–5131. [[CrossRef](#)]
7. Rojas-Dueñas, G.; Riba, J.; Moreno-Eguilaz, M. CNN-LSTM-Based Prognostics of Bidirectional Converters for Electric Vehicles' Machine. *Sensors* **2021**, *21*, 7079. [[CrossRef](#)]
8. Li, D.; Kakosimos, P.; Peretti, L. Machine-Learning-Based Condition Monitoring of Power Electronics Modules in Modern Electric Drives. *IEEE Power Electron. Mag.* **2023**, *10*, 58–66. [[CrossRef](#)]
9. Capogrosso, L.; Cunico, F.; Cheng, D.S.; Fummi, F.; Cristani, M. A Machine Learning-Oriented Survey on Tiny Machine Learning. *IEEE Access* **2024**, *12*, 23406–23426. [[CrossRef](#)]
10. Abadade, Y.; Temouden, A.; Bamoumen, H.; Benamar, N.; Chtouki, Y.; Hafid, A. A Comprehensive Survey on TinyML. *IEEE Access* **2023**, *11*, 96892–96922. [[CrossRef](#)]
11. Heydari, S.; Mahmoud, Q.H. Tiny Machine Learning and On-Device Inference: A Survey of Applications, Challenges, and Future Directions. *Sensors* **2025**, *25*, 3191. [[CrossRef](#)] [[PubMed](#)]
12. Kou, L.; Liu, C.; Cai, G.; Zhang, Z. Fault Diagnosis for Power Electronics Converters Based on Deep Feedforward Network and Wavelet Compression. *Electr. Power Syst. Res.* **2020**, *185*, 106370. [[CrossRef](#)]

13. Araújo, A.C.S.; Baccarini, L.M.R.; Filho, P.C.M.L.; Caminhas, W.M. A Digital Twin Approach to Smart Monitoring and Fault Diagnosis. *IEEE Access* **2025**, *13*, 148384–148395. [\[CrossRef\]](#)
14. Jain, P.; Poon, J.; Singh, J.P.; Spanos, C.J.; Sanders, S.R.; Panda, S.K. A Digital Twin Approach for Fault Diagnosis in Distributed Photovoltaic Systems. *IEEE Trans. Power Electron.* **2020**, *35*, 940–956. [\[CrossRef\]](#)
15. Zhuang, M.; Shahidehpour, M.; Li, Z. An Overview of Non-Intrusive Load Monitoring: Approaches, Business Applications, and Challenges. In Proceedings of the IEEE International Conference on Power System Technology, Guangzhou, China, 6–8 November 2018. [\[CrossRef\]](#)
16. Randolph, M.; Diehl, W. Power Side-Channel Attack Analysis: A Review of 20 Years of Study for the Layman. *Cryptography* **2020**, *4*, 15. [\[CrossRef\]](#)
17. Myridakis, D.; Spathoulas, G.; Kakarountas, A.; Schinianakis, D. Smart Devices Security Enhancement via Power Supply Monitoring. *Future Internet* **2020**, *12*, 48. [\[CrossRef\]](#)
18. System Management Interface Forum, Inc. PMBus™ Specification Part II: Command Language, Revision 1.3.1. Available online: <https://pmbus.org/wp-content/uploads/2022/01/PMBus-Specification-Rev-1-3-1-Part-II-20150313.pdf> (accessed on 12 January 2026).
19. Kroičs, K.; Gaspersons, K.; Elkhateb, A. Response Time Reduction of DC–DC Converter in Voltage Mode with Application of GaN Transistors and Digital Control. *Electronics* **2024**, *13*, 901. [\[CrossRef\]](#)
20. Button, R.M.; Kascak, P.; Lebron-Velilla, R. Digital Control Technologies for Modular DC-DC Converters. In Proceedings of the 2000 IEEE Aerospace Conference, Big Sky, MT, USA, 25 March 2000. [\[CrossRef\]](#)
21. Zhang, Z.; Li, J. A Review of Artificial Intelligence in Embedded Systems. *Micromachines* **2023**, *14*, 897. [\[CrossRef\]](#)
22. Ang, L.-M.; Seng, J.K.P. Embedded Intelligence: Platform Technologies, Device Analytics, and Smart City Applications. *IEEE Internet Things J.* **2021**, *8*, 13165–13182. [\[CrossRef\]](#)
23. Han, H.; Trimi, S.; Lee, S.M. Tiny Machine Learning (TinyML): Research Trends and Future Application Opportunities. *Array* **2026**, *29*, 100674. [\[CrossRef\]](#)
24. Paleyes, A.; Urma, R.-G.; Lawrence, N.D. Challenges in Deploying Machine Learning: A Survey of Case Studies. *ACM Comput. Surv.* **2023**, *55*, 1–29. [\[CrossRef\]](#)
25. Kopetz, H. *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 2nd ed.; Springer: New York, NY, USA, 2011.
26. Böhmisch, M.; Federl, A.; Sulzinger, M.; Bauerfeind, D. Stromversorgungen mit digital konfigurierbarer Regelung für Embedded-Systeme: Anwendungen/Grundlagen/Ausblick. In Proceedings of the 3rd Symposium Elektronik und Systemintegration (ESI 2022), Landshut, Germany, 6 April 2022; pp. 21–30.
27. Microchip Technology Inc. MCP19118/19: Digitally Enhanced Power Analog Controller with Integrated Synchronous Driver; Data Sheet DS20005350A; Microchip Technology Inc.: Chandler, AZ, USA, 2014. Available online: <https://ww1.microchip.com/downloads/aemDocuments/documents/OTH/ProductDocuments/DataSheets/20005350A.pdf> (accessed on 12 January 2026).
28. Microchip Technology Inc. Digitally Enhanced Power Analog (DEPA). Microchip Developer Help Documentation. Available online: <https://developerhelp.microchip.com/xwiki/bin/view/applications/power/depa/> (accessed on 12 January 2026).
29. Microchip Technology Inc. What Is a Core Independent Peripheral (CIP)? Microchip Developer Help Documentation. Available online: <https://developerhelp.microchip.com/xwiki/bin/view/products/mcu-mpu/8bit-pic/cip/> (accessed on 12 January 2026).
30. Microchip Technology Inc. Configurable Logic Cell (CLC). Microchip Developer Help Documentation. Available online: <https://developerhelp.microchip.com/xwiki/bin/view/products/mcu-mpu/8bit-pic/peripherals/clc/> (accessed on 12 January 2026).
31. Erickson, R.W.; Maksimović, D. *Fundamentals of Power Electronics*, 3rd ed.; Springer: Cham, Switzerland, 2020. [\[CrossRef\]](#)
32. Obermeier, S. Konzeption, Implementierung, Evaluierung und Optimierung eines Digital Konfigurierbaren Buck-Boost Wandlers auf Basis eines CIP-Mikrocontrollers. Bachelor’s Thesis, Technische Hochschule Deggendorf, Deggendorf, Germany, 2023. (In German)
33. Federl, A. Digital Konfigurierbares Akku-Ladesystem. Master’s Thesis, Technische Hochschule Deggendorf, Deggendorf, Germany, 2018. (In German)
34. Cleveland, T. Digitally Enhanced Analog Power Control. Presentation Slides; Microchip Technology Inc.: Chandler, AZ, USA, 2013. Available online: https://iee.li/pdf/viewgraphs/digitally_enhanced_analog_power_control.pdf (accessed on 12 January 2026).
35. Böhmisch, M. Entwurf, Simulation und Evaluation eines Digital Konfigurierbaren Four-Switch Buck-Boost Wandlers. Master’s Thesis, Technische Hochschule Deggendorf, Deggendorf, Germany, 2021. (In German)
36. Böhmisch, M. Analyse und Optimierung der Regelung eines Digital Konfigurierbaren Tiefsetzstellers. Bachelor’s Thesis, Technische Hochschule Deggendorf, Deggendorf, Germany, 2020. (In German)
37. Immonen, R.; Hämmäläinen, T. Tiny Machine Learning for Resource-Constrained Microcontrollers. *J. Sens.* **2022**, *2022*, 7437023. [\[CrossRef\]](#)
38. fischertechnik GmbH. Quality Inspection with AI, 9 V—Simulation Model (Art. No. 568416), Product Documentation; fischertechnik GmbH: Waldachtal, Germany, 2024. Available online: <https://www.fischertechnik.de/en/products/industry-and-universities/training-models/568416-quality-control-with-ai-9v> (accessed on 12 January 2026).

39. Sundaram, S.; Zeid, A. Artificial Intelligence-Based Smart Quality Inspection for Manufacturing. *Micromachines* **2023**, *14*, 570. [[CrossRef](#)] [[PubMed](#)]
40. Zhu, X.; Mårtensson, P.; Hanson, L.; Björkman, M.; Maki, A. Automated Assembly Quality Inspection by Deep Learning with 2D and 3D Synthetic CAD Data. *J. Intell. Manuf.* **2024**, *36*, 2567–2582. [[CrossRef](#)]
41. Christ, M.; Braun, N.; Neuffer, J.; Kempa-Liehr, A.W. Time Series Feature Extraction on Basis of Scalable Hypothesis Tests (Tsfresh—A Python Package). *Neurocomputing* **2018**, *307*, 72–77. [[CrossRef](#)]
42. Breiman, L. Random Forests. *Mach. Learn.* **2004**, *45*, 5–32. [[CrossRef](#)]
43. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Müller, A.; Nothman, J.; Louppe, G.; et al. Scikit-Learn: Machine Learning in Python. *arXiv* **2012**, arXiv:1201.0490.
44. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
45. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.B.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In Proceedings of the USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016.
46. Banbury, C.; Reddi, V.J.; Lam, M.W.Y.; Fu, W.; Fazel, A.; Holleman, J.; Huang, X.; Hurtado, R.; Kanter, D.; Lokhmotov, A.; et al. Benchmarking TinyML Systems: Challenges and Direction. *arXiv* **2020**, arXiv:2003.04821.
47. Tsoukas, V.; Gkogkidis, A.; Boumpa, E.; Kakarountas, A. A Review on the Emerging Technology of TinyML. *ACM Comput. Surv.* **2024**, *56*, 259. [[CrossRef](#)]
48. Kellas, A.D.; Christou, N.; Jiang, W.; Li, P.; Simon, L.; David, Y.; Kemerlis, V.P.; Davis, J.C.; Yang, J. PickleBall: Secure Deserialization of Pickle-Based Machine Learning Models. *arXiv* **2025**, arXiv:2508.15987. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.