Enhancing Energy Efficiency in AI: A Multi-Faceted Analysis across Time Series, Semantic AI and Deep Learning Domains

Lejla Begic Fazlic[1], Berkay Cetkin[1], Achim Guldner[1], Matthias Dziubany[2], Julian Heinen[2], Stefan Naumann[1], and Guido Dartmann[1]

[1] Institute for Software Systems (ISS), Trier University of Applied Sciences, Birkenfeld, Germany,
l.begic@umwelt-campus.de,
https://www.umwelt-campus.de/
[2] BITO CAMPUS GmbH, Meisenheim, Germany

Abstract. This research investigates strategies to enhance the energy efficiency of artificial intelligence (AI) algorithms, focusing on three pivotal domains: time series analysis, semantic AI, and deep learning (DL). Through a comprehensive examination of variables such as data size and the impact of hyper-parameter adjustments, the study aims to uncover nuanced insights into the relationship between algorithmic performance and energy consumption. By exploring the unique challenges and opportunities within each use case, this research provides valuable guidance for practitioners seeking to optimize energy efficiency in AI applications. The findings contribute to the ongoing discourse on sustainable AI development, offering practical overview to balance computational power with environmental considerations.

Keywords: artificial intelligence, energy efficiency, machine learning, sustainability

1 Introduction

The rapid advancement of artificial intelligence (AI) has brought changes across various sectors. However, the increasing computational demands of AI algorithms pose significant energy efficiency challenges. Addressing these challenges is crucial to ensure sustainable AI development. This research focuses on analysing the resource and energy efficiency of AI algorithms, a key aspect of sustainable AI. It targets three pivotal domains: time series analysis, semantic AI, and deep learning (DL). Time series analysis is crucial in fields like finance and weather forecasting, where AI can offer valuable insights. Each field offers distinct challenges and opportunities regarding energy efficiency. The goal of this research is to offer a detailed understanding of how different factors interact and affect the performance and energy efficiency of AI algorithms. We first conducted research on the energy aspects of natural language processing (NLP) in semantic AI.

Next, we examined energy consumption related to complex neural network operations in DL. Finally, we focused on the computational demands of time series analysis. A key part of the study is to explore how the size of datasets affects energy efficiency, given that larger datasets generally demand more computational resources. The research also examines how changes in hyper-parameters, like learning rate and batch size, can influence both algorithm performance and energy consumption, aiming to make AI models more energy-efficient. We conducted a study of the available literature to understand the relationship between energy use and algorithm efficiency. This research contributes to the discussion on sustainable AI by providing a practical view on balancing computing needs and environmental concerns. The structure of this paper is outlined as follows: Section 2 offers a review of relevant literature. The methodology and use case design is detailed in Section 3. Section 4 focuses on the evaluation and discussion, and the conclusions and the future work are provided in Section 5.

2 Related Work

In recent years, advancements in energy efficiency in the field of AI have focused on reducing the significant energy consumption of AI models. Advances in Green AI initiatives have been pivotal, emphasizing sustainable AI development by integrating energy-efficient practices in model training and deployment, thus balancing computational power with environmental considerations. Recent advancements not only focus on reducing the energy footprint of DL models [1,2,3,4,5,6,7] but also extend to optimizing semantic AI algorithms for better language understanding and enhancing time series models for more energy-efficient processing in different applications. In this study, we delve into the often-overlooked influence of software on the energy usage and overall environmental footprint of hardware systems. The rapidly growing field of AI, with a focus on machine learning (ML) and DL, has sparked a keen interest in evaluating their energy demands and ecological impacts during the training phase [8,9,5]. The assessment of the carbon footprint of AI [9], measurement of the energy requirements of AI systems [10], and evaluation of the efficiency of AI platforms such as PyTorch and TensorFlow [11] present one of the important research studies. Authors in [5] employ a life-cycle approach to estimate the carbon emissions generated from training NLP models. Recent advancements in Green AI emphasize the importance of energy-efficient ML where the contributions include the study on the impact of data preprocessing and feature selection [12], exploration of model optimization through weight pruning [13], and the discussion on Green AI's role in sustainable computing by Wang et al. [14]. In the field of energy- and resource-efficient software, various methodologies have been developed to assess the environmental sustainability of software products, as exemplified by the research of Naumann et al. [15] and Mancebo et al. [16]. Our analysis places special emphasis on energy consumption during the training and testing phases of various usage scenarios, comparing them using the energy and resource consumption metrics established by the authors in [17]. Recent advancements in

time series analysis, especially regarding energy consumption, are key for efficient resource management and environmental sustainability. Sentiment analysis is a ML technique that interprets and classifies emotions expressed in text data, often used to understand opinions in customer feedback, social media, and other written sources. Recent research in sentiment analysis has explored the capabilities of large language models (LLMs). Studies like Zhong et al. [18] compared the zero-shot performance of LLMs with fine-tuned BERT models, while researchers in [19] investigated ChatGPT's proficiency in handling various sentiment analysis tasks, including polarity shifts and sentiment inference. Deng et al. [20] delved into fine-tuning a smaller model using a LLM to generate weak labels, achieving performances comparable to supervised models. These studies indicate a growing interest in understanding LLMs' effectiveness in sentiment analysis, but they also highlight the need for more comprehensive evaluations across diverse tasks and datasets. Recent study [21] introduces a new way to classify emotions in text using spiking neural networks (SNNs) to enhance energy efficiency. To the best of our knowledge, the investigation of energy efficiency in sentiment analysis tasks still remains a relatively untapped area in the field. Recent progress in image classification, led by convolutional neural networks (CNNs) models like ResNet [22] and VGG-16 [23], has significantly improved accuracy in various domains.

In our study, we examined three different use-case scenarios, exploring into aspects such as the volume of data and the consequences of modifying hyperparameters. This approach enabled us to discover complex details about the connection between the performance of algorithms and their energy consumption. In our first sentiment analysis use case scenario, we specifically investigate the energy efficiency of BERT models, examining how different data volumes and batch sizes influence their energy consumption. This focused approach enables us to understand the nuances of energy usage in these NLP models, ensuring that they not only maintain high accuracy but also adhere to sustainable computational practices. As a second use case scenario in our study, we extended our analysis of resource and energy efficiency to encompass various epoch and batch sizes while working with neural network models like ResNet [22], DenseNet [24], MobileNet [25], Inception [26], VGG-16 [23] and VGG-19 [23] for image classification. This approach allowed us to comprehensively assess how different training configurations impact the energy and computational demands of these models. Our findings offer a detailed perspective on the balance between training efficiency, model accuracy, and energy use, providing important guidance for enhancing neural network training in scenarios where energy efficiency is crucial. Transitioning to our third scenario, we shift our focus to the extensive landscape of sensor data. This case study involved analyzing time series data using tools from sktime, a Python library for time series analysis [27], specifically the Random Interval Spectral Ensemble (RISE), the KNeighbors Time Series Classifier (kNNTime) and the Time Series Forest Classifier (TSFC). We aimed to evaluate the accuracy of detecting fill levels, while also incorporating energy assessments to determine the system's energy efficiency.

3 Methodology and Design

In this section, we outline the research questions we aim to address, describe the practical experiment conducted as part of this research, which includes a description of the case study's design, the experimental procedures employed, and the methods used to analyze the collected results. We assess algorithm efficiency by monitoring hardware usage and power consumption in the aforementioned scenarios. For reference, we have prepared a comprehensive replication package, which includes detailed system specifications, the scenarios explored, the data recorded, and the results of our analyses. All these materials are accessible in our Git repository https://gitlab.rlp.net/rgdsai/mfa.

3.1 Research Questions: AI Model Efficiency

In light of our research objectives, we structured our inquiry into the energy consumption and efficiency of AI models, particularly in NLP and DL, through the following research questions:

RQ1: How does batch size influence energy consumption and accuracy in NLP model training? This research question aims to understand the relationship between batch size during the fine-tuning phase of NLP models and its impact on energy efficiency and model accuracy. By answering this, we intend to identify optimal batch sizes that balance energy consumption with performance efficacy.

RQ2: What is the effect of training data size on the energy consumption and accuracy of NLP models? We investigated the energy consumption and accuracy of models by varying the proportion of the training data, exploring how these metrics change with increasing data volumes. This seeks to determine the optimal data size that ensures efficient energy use without compromising the model's accuracy.

RQ3: How do different DL architectures compare regarding energy consumption and model performance? Focusing on architectures like DenseNet, ResNet, VGG-16, VGG-19, and Inception, this question explores how the structural variances in these models affect their energy efficiency and overall performance. The goal is to provide a comparative analysis that guides the selection of the most energy-efficient model without sacrificing performance.

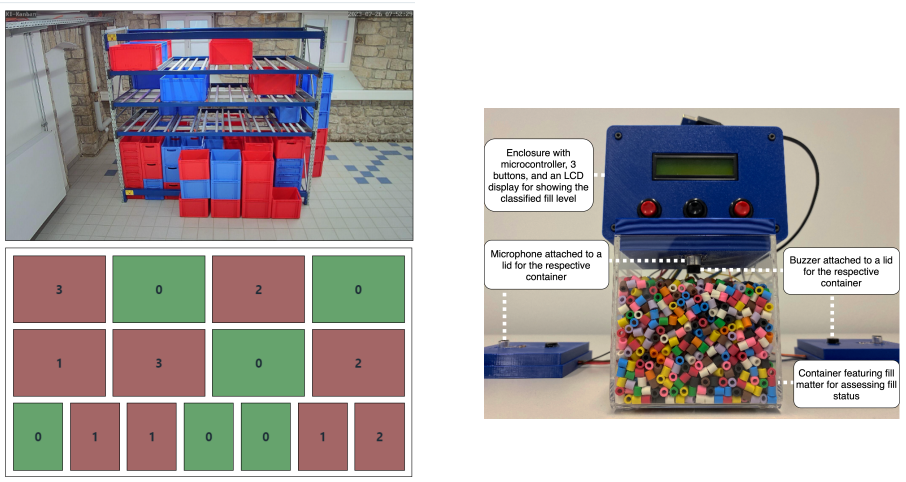
RQ4: In the context of time series approaches using sensor data, what is the best configuration that exhibit the most energy-efficient processing? This question extends the study to time series analysis, evaluating various AI model scenarios on their energy efficiency when processing sensor data. The objective is to identify models that offer an optimal balance between energy efficiency and effective time series analysis.

By exploring these research questions, our study aims to provide meaningful contributions into the energy-efficient implementation of AI systems, especially in a time where the environmental impact of technology is of paramount concern. The findings are expected to guide both practitioners and researchers in making informed decisions about AI model selection and optimization for sustainable usage.

3.2 Data Aquisition

Our research embarks on a multifaceted exploration, traversing three distinct scenarios. For our first use case scenario we used Stanford's Large Movie Review Dataset IMDB [28,29] that contains 50,000 movie reviews. Reviews are labeled as 1 or 0 corresponding to positive or negative sentiment, respectively. A minimal data preprocessing are done prior to tokenization as BERT was trained on complete sentences. To effectively utilize pre-trained BERT, we must utilize the library's tokenizer due to BERT's specific, fixed vocabulary and the tokenizer's unique handling of out-of-vocabulary words. Additionally, it's essential to add special tokens at the start and end of each sentence, standardize sentence length through padding or truncation, and explicitly identify padding tokens using the "attention mask". The developed code supports sentiment analysis on a variety of CSV datasets, making it adaptable to any text classification task, with the ability to analyze textual data and assign sentiment labels, regardless of the specific dataset's structure or content.

In our vision use-case a simple webcam took pictures of a flowrack with various number of bins in each lane. As the bins move forward after putting them into a lane, it is possible to determine the number of bins from the backside of the rack. The pictures of the whole rack were devided into smaller ones showing only one separated lane, which can store a maximal number of tree bins. In order to manage different lane sizes and camera angels the pictures of the lanes were mathematically transformed to equal size. Fig. 1 (a) shows a picture of the flowrack and the determined number of bins in each lane.



(a) (b)

Fig. 1. Demonstrators utilized in the experimental use case

In our third application scenario, we developed a demonstrator for generating time series data. It's designed to determine the fill level of a small container through acoustic sound waves, with the capability to differentiate between five distinct levels of fill: 0%, 25%, 50%, 75%, and 100%. The construction of this demonstrator involved the use of 3D printing to create a structure that houses a central unit. This central unit incorporates an ESP32 [30] microcontroller, equipped with three buttons and a display that shows the classified fill level. Attached to this central unit are three lids, each fitted with a buzzer to generate acoustic signals and a microphone for recording the corresponding time-series data. The operational principle involves placing a lid over a container filled with material. Upon pressing the corresponding button associated with that lid, a sinusoidal sweep is emitted as an acoustic signal. The recording of this signal commences before the emission of the sound to capture ambient noise for noise reduction purposes and continues after the signal has ended to include sound reflections. This process facilitates the approximate calculation of the room's impulse response, which is then transmitted to a Raspberry Pi via MQTT [31]. This Raspberry Pi is responsible for managing and storing the data. The impulse response of the room, captured in this scenario, serves as time-series data and forms the foundation for training the ML algorithms: RISE, kNNTime, and TSFC. Fig. 1(b) illustrates the setup of the demonstrator, providing a visual representation of its configuration and components.

3.3 Case Study Overview

In our first use case scenario, we conducted series of experiments specifically focusing on text classification using a BERT model. We iterate through a predetermined number of experiments, adjusting the dataset and feature set sizes based on specified percentages, to evaluate the impact on model performance. The process includes data preprocessing, model training, and validation stages, logging each step's start and end times for energy performance tracking and reproducibility. The different adjustments of the batch size (16 or 32) are used during training to maintain a balance between gradient noise and memory efficiency. The Adam optimizer with default hyperparameters was used in all scenarios. Additionally, we emphasize memory management through explicit garbage collection and system calls to clear RAM, ensuring efficient resource utilization during the experiments. Our methodological framework involves a detailed comparison of energy consumption metrics such as mean power (W) and energy usage (Wh), for the preprocessing and training phases, as well as for the GPU utilization. By incrementally increasing dataset sizes from 10% to 100% of the total volume, we could simulate different training intensities to observe their impacts on energy efficiency and model performance. This approach allowed us to capture a range of performance metrics, including processing times, CPU and GPU usage percentages, RAM and GRAM usage, and GPU temperature.

In our study's second use case scenario, we extended our analysis of resource and energy efficiency by examining various epoch and batch sizes when working

with neural network models like ResNet[23], DenseNet [25], MobileNet [26], Inception [27], VGG-16 [24], and VGG-19 [24] for image classification, enabling a comprehensive assessment of the impact of different training configurations on energy and computational demands.

In our third scenario, we explore the application of Edge AI to determine container fill levels using acoustic analysis. Our analysis spans across distinct settings, aiming to identify the most efficient combination for precise fill level classification while optimizing energy consumption. We utilized three ML algorithms: RISE, kNNTime, and TSFC. These were employed to adjust two key parameters: sample length (the length of the room impulse response considered by the models) and the number of estimators/neighbors. Analogous to the second use case, we measured energy consumption using metrics such as mean power, energy usage during training and testing phases, as well as training and testing duration, CPU usage, and accuracy using the F1 score metric.

3.4 Tailored Hardware for Different Use Cases and Measurement Methodology

For the semantic analysis and DL use-cases, we utilized a high-powered server configuration, powered by an Intel Xeon W-2295 processor with 18 cores running at 3.0 GHz, paired with 131.56 GB of DDR4-2933 RAM. It features an NVIDIA GeForce RTX 4090 24 GB GPU and is built on an ASUS WS C422 Pro/SE mainboard. This setup was specifically selected to optimize the processing power and memory requirements needed for these complex tasks, ensuring efficient and effective analysis. Conversely, for the time series analysis use-case, we choose a standard PC configuration. The experiments were carried out on a computer setup that included 4 GB of RAM, arranged in two modules of 2 GB each, and was driven by an Intel Core i5-650 processor. This system also boasted a dual-storage configuration, combining a 500 GB hard disk drive (HDD) for extensive storage capacity with a 250 GB solid-state drive (SSD) for rapid data retrieval and system responsiveness. This choice reflects the comparatively lower computational demands of time series analysis, which, while still requiring precision and accuracy, can be effectively conducted on less powerful hardware. This distinction in hardware choice between the use-cases allowed us to not only tailor our computational resources to the specific needs of each task but also to investigate the potential impacts of hardware capabilities on the efficiency and outcomes of different types of data analysis.

The measurement methodology for AI-based method was described in previous work [32] and in context of AI methods in [33]. The measurements are based on the methodology and guidelines outlined in the Green Software Measurement Model [34]. Data aggregation is automated, with users logging process start times, end times, and labels in a CSV file called the action log. A Standard Usage Scenario (SUS) outlines the basic workflow. Resource data, including CPU and RAM usage, is recorded using the Linux performance reporting tool collectl on a Linux Ubuntu system. Energy consumption data is obtained from a power meter, requiring synchronization with the executing computer's time.

The accurate calculation of energy consumption for the process requires recording the baseline consumption of the executing system through a corresponding measurement, with the measured value adjusted accordingly, ideally conducted for a duration comparable to that of the actual process.

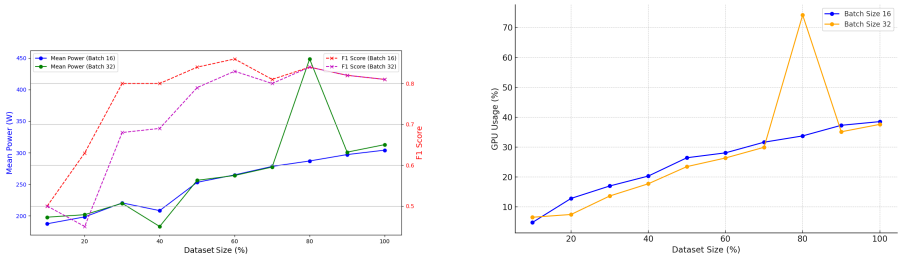
4 Results and Discussion

To address our first research question (**RQ1**), we aim to explore the impact of batch size on energy consumption and accuracy during the fine-tuning phase of NLP model training. Our goal is to try to identify the optimal batch sizes that balance energy efficiency with performance efficacy. We conducted a comprehensive analysis by systematically varying batch sizes and measuring their effects on energy consumption and accuracy for the BERT model across different dataset sizes. Detailed performance metrics, including energy efficiency (measured in Wh) and model accuracy (indicated by the F1 score) for two batch sizes (16 and 32), are presented in Table 1.

Table 1. Combined performance metrics for Bert model

Performance Metrics for Bert Model - Batch Size 16										
Metric	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Mean Power [W]	187.516	198.40	220.544	208.318	253.173	264.890	278.309	286.963	297.075	304.070
Time Preprocessing [s]	3.941	3.973	3.954	4.002	3.990	3.993	4.040	4.041	4.035	4.067
Time Train [s]	1.410	1.911	2.433	3.044	3.421	3.971	4.426	4.949	5.379	5.891
Time Test [s]	0.020	0.028	0.041	0.057	0.070	0.081	0.091	0.103	0.118	0.130
Energy Preprocessing [Wh]	0.080	0.075	0.073	0.103	0.076	0.074	0.073	0.074	0.074	0.079
Energy Train [Wh]	0.041	0.073	0.123	0.088	0.222	0.266	0.315	0.365	0.407	0.451
CPU Usage [%]	3.250	3.448	3.437	3.307	3.464	3.410	3.409	3.385	3.350	3.266
GPU Usage [%]	4.827	12.848	17.005	20.319	26.442	28.080	31.641	33.742	37.290	38.523
RAM Usage [%]	1.072	1.122	1.141	1.242	1.180	1.189	1.242	1.228	1.294	1.306
GRAM Usage [%]	12.338	12.329	12.630	12.399	12.490	12.488	12.492	12.548	12.595	12.582
GPU Temp [°C]	53.177	42.558	46.417	49.798	47.827	45.068	45.756	46.346	47.625	40.608
GPU Energy [Wh]	0.055	0.084	0.129	0.149	0.224	0.263	0.311	0.358	0.412	0.453
F1 Score	0.5	0.63	0.80	0.80	0.84	0.86	0.81	0.84	0.82	0.81
Performance Metrics for Bert Model - Batch Size 32										
Metric	**10%**	**20%**	**30%**	**40%**	**50%**	**60%**	**70%**	**80%**	**90%**	**100%**
Mean Power [W]	197.94	201.840	219.91	183.181	256.350	263.929	277.332	448.520	300.924	312.802
Time Preprocessing [s]	3.923	3.940	3.964	3.988	4.020	4.015	3.994	4.016	4.042	4.062
Time Train [s]	1.253	1.560	1.921	2.463	2.581	2.873	3.246	3.282	3.906	4.208
Time Test [s]	0.016	0.027	0.039	0.050	0.064	0.074	0.085	0.204	0.108	0.122
Energy Preprocessing [Wh]	0.076	0.071	0.078	0.117	0.077	0.076	0.074	0.083	0.078	0.077
Energy Train [Wh]	0.047	0.075	0.104	0.011	0.227	0.225	0.264	0.318	0.351	0.362
CPU Usage [%]	3.197	3.331	3.417	3.374	3.393	3.518	3.449	3.678	3.417	3.408
GPU Usage [%]	6.553	7.482	13.658	17.744	23.481	26.364	29.937	74.155	35.123	37.605
RAM Usage [%]	1.135	1.230	1.178	1.374	1.290	1.281	1.329	1.563	1.347	1.333
GRAM Usage [%]	17.160	17.404	17.315	17.198	17.442	17.499	17.442	18.708	17.630	17.641
GPU Temp [°C]	54.491	42.318	46.500	47.243	47.956	48.306	47.994	64.481	50.425	49.755
GPU Energy [Wh]	0.055	0.059	0.088	0.113	0.169	0.198	0.237	9.247	0.316	0.346
F1 Score	0.5	0.45	0.68	0.69	0.79	0.83	0.80	0.84	0.82	0.81

The relationships between batch size, mean power consumption and F1 score are presented in Fig. 2 (a). Our final insights from this examination of batch



(a) (b)

Fig. 2. Performance metrics and resource utilization for BERT models

size impacts on BERT model training underscore the intricate balance between computational resource utilization and model effectiveness. Our key findings are outlined below:

- We found a direct correlation between batch size and energy consumption. Larger batch sizes generally resulted in higher mean power consumption but also offered more efficient processing in terms of energy per data point processed, particularly with larger datasets. The analysis revealed that batch sizes of 32 often resulted in shorter training and preprocessing times compared to a batch size of 16, especially as the dataset size increased.
- Optimal model performance was observed at varying dataset percentages, highlighting the importance of selecting batch size based on the specific context of the model's application. The F1 score analysis indicated that while larger batch sizes can enhance training efficiency, they do not always correlate with improved model accuracy.
- Batch size 32 (orange line in Fig. 2b) typically exhibits lower or comparable GPU usage compared to batch size 16 for most dataset sizes, except for a significant spike at the 80% data size. This suggests that using a larger batch size can be more GPU efficient for certain dataset sizes but may encounter inefficiencies or bottlenecks at specific points (like 80%). We found that as the dataset size increases, so does the energy required for training, particularly evident in smaller batch sizes.
- Identifying the optimal batch size for NLP model training involves balancing energy efficiency, training speed, and model accuracy. Our findings suggest that while larger batch sizes may enhance computational efficiency, they require careful consideration of the trade-offs involved, particularly regarding model performance and hardware limitations.

Our investigation into the effects of batch size on the energy consumption and accuracy of NLP model training with a specific focus on the BERT model provides critical insights for optimizing training processes. These insights emphasize the necessity of a nuanced approach to batch size selection, tailored to the specific goals of energy efficiency, computational resource management, and model accuracy.

In addressing our second research question (**RQ2**), we investigate the effects of scaling training dataset size on the energy consumption and accuracy of a BERT model. This study aims to elucidate the dual impact of dataset size on model efficiency—quantified through energy requirements—and on model performance, as gauged by the F1 score. To systematically explore these relationships, we employ Pearson correlation analysis. This method allows us to quantitatively assess the linear correlation between training data size and two key outcomes: the energy consumed throughout the training phase and the accuracy of the model. Our analysis revealed significant relationships for a batch size of 16:

- We found a strong positive correlation of 0.98 between the training data size and the energy consumed during training, highlighting a substantial increase in energy requirements as dataset size expands.
- We found a moderate positive link with a correlation of about 0.70, between the size of the training data and the F1 score. This means that making the dataset larger tends to improve how accurate the model is, but not as much as it increases the energy needed for training.

From our analysis, we also found the following correlations for a batch size of 32:

- While a strong positive correlation of 0.93 was observed between the training data size and the energy consumed during training for batch size 32, this is slightly lower than the correlation noted for batch size 16, suggesting that while energy demands still increase with larger datasets, the rate of increase may be less steep for larger batch sizes.
- A strong positive correlation (0.85) between the training data size and the F1 score for batch size 32 was found, indicating a more distinct improvement in model accuracy with larger training datasets, more so than observed with batch size 16.

Our third research question (**RQ3**) focused on unraveling the intricate balance between energy consumption and model performance across prevalent DL architectures. This inquiry aimed to dissect how structural variations in models such as DenseNet, ResNet, VGG-16, VGG-19, and Inception influence their operational efficiency and effectiveness. We thoroughly compiled data reflecting each model's energy usage during training and testing phases, accompanied by their performance metrics, primarily measured through F1 scores. Additionally, we considered the computational time and resource utilization, to paint a comprehensive picture of each model's energy profile as it is shown in Table 2. The energy consumption across different models and the trade-off between global energy consumption and the best F1 score are depicted in Figures 3 and 4, respectively.

Through this comparative analysis, our goal was to offer practical guidance for choosing the most energy-efficient model without sacrificing performance. In our analysis of energy efficiency across various DL architectures, we discovered that each model exhibits its own optimal configuration that harmonizes energy

Table 2. Comparison of algorithm performance and energy consumption - scenario image classification

Model Name	Batch Size	Epoch Numb.	Mean Power[W]	Time[s] Train	Time[s] Test	Energy[Wh] Train	Energy[Wh] Test	CPU[%] Usage	GPU[%] Usage	RAM [%] Usage	GRAM[%] Usage	GPU[C°] Temp.	GPU[Wh] Energy	F1 Score(%)
DenseNet	32	9	271.06	104.7	74.35	6.39	1.74	11.85	40.26	1.40	21.17	44.52	5.92	99.65
	16	9	224.61	165.10	76.44	6.09	1.73	12.84	31.17	1.37	12.25	39.81	5.12	99.54
	32	5	245.50	53.66	73.80	3.20	1.67	9.76	33.71	1.40	21.24	40.90	3.30	99.34
	16	5	159.2	83.10	76.15	3.07	1.73	10.75	28.16	1.40	12.24	40.30	3.00	98.80
ResNet	32	9	282.32	46.65	14.76	2.58	0.40	26.77	34.64	0.72	6.69	44.86	1.77	99.09
	16	9	279.14	52.27	14.97	2.82	0.39	28.11	34.03	0.77	4.05	44.59	1.83	99.74
	32	5	265.14	24.70	15.04	1.33	0.40	22.77	31.55	0.80	6.64	49.00	1.02	99.80
	16	5	263.88	27.32	14.75	1.43	0.40	24.16	31.59	0.79	4.04	48.69	1.04	99.76
MobileNet	32	9	268.86	54.83	30.00	3.08	0.70	20.82	32.84	2.00	12.91	45.47	2.39	99.77
	16	9	232.04	72.46	30.65	2.85	0.71	22.25	26.82	2.06	7.74	44.61	2.11	99.56
	32	5	244.09	28.63	30.52	1.54	0.70	16.67	27.20	2.09	12.66	43.11	1.32	99.35
	16	5	226.24	36.00	29.65	1.44	0.70	18.43	24.30	2.11	7.74	47.77	1.24	99.82
Inception	32	9	286.27	138.85	53.84	7.72	1.82	19.64	43.25	0.85	18.81	47.20	6.95	99.82
	16	9	270.72	165.77	54.70	8.27	1.70	18.68	39.57	0.79	10.52	45.73	7.13	99.81
	32	5	253.88	71.30	54.16	3.25	1.83	16.48	37.67	0.87	18.79	44.77	3.72	99.31
	16	5	247.05	83.80	54.50	3.65	1.68	16.07	36.00	0.80	9.96	43.65	3.83	99.84
VGG-16	32	9	355.50	140.54	14.26	9.68	0.95	13.61	74.27	1.02	30.88	58.16	9.23	64.05
	16	9	346.86	159.69	14.26	10.62	0.91	16.07	77.12	1.00	19.25	57.29	10.11	35.16
	32	5	333.10	71.18	14.11	4.36	0.98	13.34	71.65	1.05	31.05	56.46	4.97	71.68
	16	5	327.79	81.37	14.22	4.93	0.91	15.37	73.69	1.03	19.27	55.59	5.40	39.19
VGG-16bn	32	9	372.18	159.50	16.36	11.66	1.24	12.51	75.92	1.10	33.90	61.07	12.03	99.40
	16	9	370.19	176.72	16.14	12.88	1.19	14.59	78.68	1.07	23.06	60.98	13.14	98.43
	32	5	338.18	81.96	16.27	5.03	1.25	12.24	72.32	1.14	33.88	59.13	6.29	99.56
	16	5	340.11	89.79	16.37	5.64	1.20	14.03	75.09	1.13	23.04	59.35	6.83	93.75
VGG-19	32	9	349.90	156.09	15.92	10.39	1.16	13.66	76.42	1.66	27.30	58.68	10.84	49.34
	16	9	271.38	44.26	15.88	2.01	0.72	12.67	71.91	1.22	19.57	53.16	3.29	33.15
	32	5	321.77	80.00	15.95	4.97	0.73	12.30	73.35	1.19	26.73	57.14	5.80	63.23
	16	5	316.62	90.95	15.90	5.80	0.43	14.00	76.47	1.16	19.54	56.08	6.20	28.71
VGG-19bn	32	9	376.54	178.10	17.99	13.28	1.35	11.64	77.52	1.16	35.35	61.00	13.58	92.14
	16	9	373.22	198.73	17.80	14.63	1.31	13.45	80.40	1.12	23.76	61.20	14.86	92.18
	32	5	348.06	90.72	18.00	5.89	1.36	11.26	74.39	1.11	35.33	60.15	7.18	98.00
	16	5	348.36	100.26	17.92	6.56	1.34	12.90	77.51	1.09	23.75	60.15	7.80	93.20

consumption with performance. For DenseNet and ResNet, the optimal configuration was determined to be a batch size of 32 with 5 training epochs, which significantly reduced energy usage during both the training and testing phases

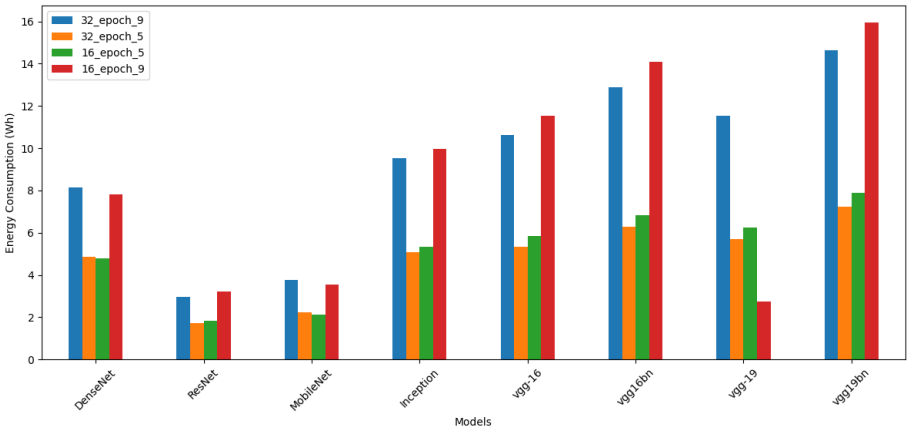


Fig. 3. Total energy consumption vs. models

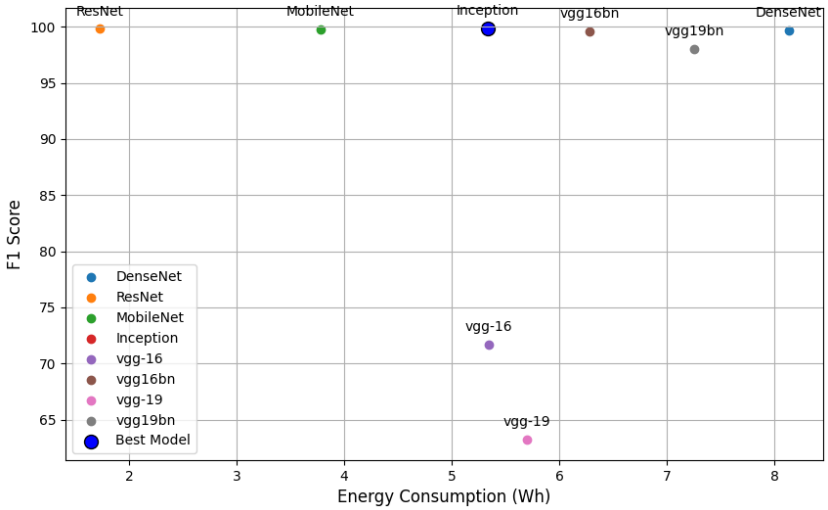


Fig. 4. Trade-off between global energy consumption and best F1 score

without compromising the models' accuracy, as evidenced by their F1 scores. Similarly, MobileNet demonstrated its energy efficiency under the same parameters, indicating a consistent pattern among these architectures for achieving operational efficiency. The Inception model, however, diverged slightly, finding its most energy-efficient configuration with a batch size of 16 and 5 epochs. This adjustment offered substantial energy savings across the board, while still securing a high F1 score, illustrating that a slight reduction in batch size could yield notable efficiency gains without sacrificing performance. VGG-16bn followed a similar pattern to Inception, opting for a batch size of 16 and 5 epochs, which was effective in reducing energy consumption while maintaining a commendable level of accuracy. Conversely, VGG-19bn aligned with DenseNet and ResNet, favoring a batch size of 32 and 5 epochs for its most energy-efficient performance. This setup allowed for minimized energy usage during operations while achieving a robust F1 score. This affirms that even among models with varying complexities, there is an opportunity to achieve equilibrium between energy efficiency and model accuracy. Our key insights are summarized as follows:

- The findings indicate a complex relationship between model depth and energy consumption, challenging the conventional belief that more complex models are always more energy-intensive.The investigation into F1 scores highlighted that architectural sophistication does not always translate to enhanced model accuracy. For instance, despite VGG-19's depth, it did not consistently outperform the less complex VGG-16 in terms of accuracy.
- Through the lens of our analysis, it became evident that there isn't a universal optimal batch size or epoch count that maximizes energy efficiency across all models. Instead, each architecture demands a tailored approach to find its equilibrium point that harmonizes energy consumption with model per-

formance. Across all models, a trend towards selecting a moderate batch size and a lower number of epochs (5) appears to be the sweet spot for optimizing energy efficiency without significantly impacting the model's performance. This analysis underscores the critical insight that, despite the diversity in architecture and design, DL models can achieve a delicate balance between energy efficiency and performance through strategic adjustments in batch size and epoch count.
– The research also highlighted a connection between the efficient use of hardware resources and energy consumption. Models that were able to make more effective use of hardware resources frequently demonstrated improved energy efficiency without necessarily sacrificing performance.

In addressing our last research question (**RF4**) regarding the energy efficiency of AI solution for processing time series sensor data, Table 3 outlines five distinct configurations of the kNNTime, TSFC and RISE algorithms. Each provides insights into how variations in sample lengths and the number of estimators/neighbors impact the algorithm's energy consumption and classification performance. Figure 5 illustrates the energy consumption and performance analysis of ML algorithms, using the first three entries for each algorithm from Table 3. The kNNTime algorithm demonstrates robust performance, particularly evident through its consistently high F1 scores, which peak at 0.92 for both 400 and 100 sample lengths when utilizing just one neighbor.

Table 3. Comparison of algorithm performance and energy consumption across different parameter configurations

Algorithm Name	Sample Length	N-Estim./ Neighbors	Mean Power [W]	Energy [Wh]		Time [s]		CPU Usage [%]	F1 Score
				Training	Testing	Training	Testing		
kNNTime	700	1	83.77	0.015	7518.32	0.00	174.56	24.67	0.87
	400	1	83.40	0.011	2466.45	0.00	57.02	24.67	0.92
	100	1	80.17	0.013	153.32	0.00	3.410	24.59	0.92
	700	3	83.56	0.017	7605.13	0.00	176.09	24.69	0.75
	400	3	83.23	0.012	2449.15	0.00	56.50	24.67	0.73
RISE	700	100	77.93	47.87	31.16	1.00	0.69	24.71	0.92
	400	100	76.89	42.97	28.06	0.89	0.62	24.52	0.92
	100	100	71.58	18.99	12.33	0.41	0.21	24.37	0.92
	700	50	73.75	22.92	14.83	0.45	0.32	24.43	0.92
	400	50	76.51	21.19	13.81	0.45	0.29	24.27	0.95
TSFC	700	100	57.32	1.90	0.89	0.03	0.00	20.61	0.98
	400	100	60.20	1.40	0.62	0.02	0.00	20.58	0.98
	100	100	62.40	0.77	0.28	0.00	0.00	15.93	0.95
	700	50	60.72	0.99	0.45	0.017	0.00	18.07	0.97
	400	50	56.57	0.74	0.31	0.00	0.00	15.38	0.97

When examining energy and time efficiency, we observe that energy consumption during testing presents considerable variation, escalating as sample length increases. The energy required for training remains consistently low across various configurations, emphasizing the algorithm's efficiency during the learning phase.

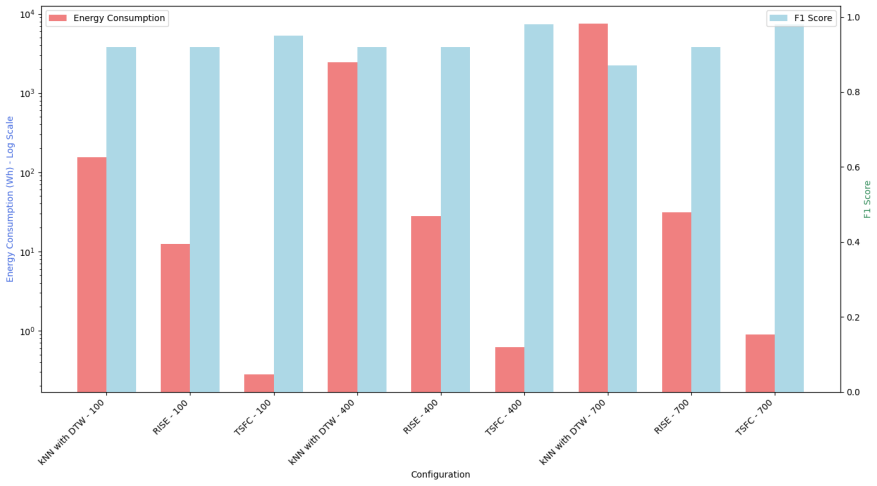


Fig. 5. Energy consumption and performance analysis of ML algorithms

This is primarily because kNN, unlike many other ML algorithms, does not require a separate training step to create a model; it simply stores the data and makes inferences directly from the entire dataset during the prediction phase. The most efficient configuration for kNNTime emerges with a sample length of 100 paired with a single neighbor. This setup also minimizes energy consumption and reduces testing time, representing an optimal balance for those seeking both precision and efficiency. For algorithm RISE, reducing the sample length from 700 to 100 leads to a decrease in energy consumption throughout the training and testing phase, aligning with expectations that less data requires less computational power. The F1 score remains consistently high across different sample lengths, suggesting that RISE effectively maintains predictive performance even with reduced data. Reducing the number of estimators from 100 to 50 decreases energy consumption in both training and testing phases without significantly compromising the F1 score. This indicates an efficient use of computational resources by RISE, as it maintains high accuracy with fewer estimators. RISE shows a consistent pattern of CPU usage across different configurations. A sample length of 400 with 50 estimators provides the best balance of high accuracy (F1 score of 0.95) with reduced energy and time consumption. TSFC shows remarkable energy efficiency across all sample lengths, with a significant decrease in energy consumption as sample length decreases. This suggests that TSFC is particularly suited for energy-efficient processing of time series data. The F1 score is very high across different configurations, indicating that TSFC does not compromise on predictive performance even when optimizing for energy efficiency. Similar to RISE, reducing the number of estimators for TSFC results in lower energy consumption without a notable drop in F1 score. This efficiency is especially remarkable, given the already low energy consumption of TSFC, underscoring its suitability for energy-constrained scenarios. TSFC presents an

optimal scenario where energy efficiency and high predictive performance coexist. It demonstrates that careful algorithm design and parameter tuning can achieve high accuracy in ML tasks without incurring high computational costs.

5 Conclusion and Future Work

Our findings revealed a nuanced relationship between batch size and energy efficiency, where larger batch sizes led to increased mean power consumption but also enhanced energy efficiency per data point, especially with larger datasets. The study highlighted the importance of context-specific batch size selection, as the optimal balance between energy efficiency and performance varies across different scenarios. Investigating the impact of training dataset size, we observed a direct correlation between increased dataset sizes and higher energy requirements, alongside an improvement in model accuracy. This underscores a crucial trade-off between energy consumption and model performance, indicating that optimizing training processes necessitates a careful consideration of dataset size. Our analysis across various DL architectures demonstrated that each model exhibits its own optimal configuration that harmonizes energy consumption with performance. This finding challenges the conventional wisdom that more complex models are inherently more energy-intensive, advocating for a tailored approach to training parameter selection. In examining AI solutions for time series sensor data, strategic adjustments in sample lengths and the number of estimators/neighbors showed significant impacts on energy efficiency and model accuracy. This suggests that for energy-sensitive applications, choosing the right parameters can lead to substantial energy savings without major accuracy sacrifices.The collective insights from our research emphasize the critical role of strategic parameter selection in achieving energy-efficient ML practices. In our future work, we will investigate the cause of the observed GPU usage peak at the 80% dataset size, examining factors such as hardware configuration and potential system bottlenecks. Additionally, since our current energy efficiency assessment is based on specific hardware configurations, we recognize the importance of broadening the scope to improve generalizability. To achieve this, we plan to expand the datasets and include additional models in our analysis.

Acknowledgements

This project was funded by German Federal Ministry for the Environment, Nature Conservation, Nuclear Safety, and Consumer Protection (BMUV) project “KIRA” under Grant 67KI32013 and Climate Action (BMWK) project “EASY” under Grant 01MD22002D. Parts of the text have been enhanced and linguistically revised using AI tools. We gratefully acknowledge students Marvin Schacht for providing necessary resources to accomplish this research work.

References

1. T. Yarally, L. Cruz, D. Feitosa, J. Sallou, and A. van Deursen, "Uncovering energy-efficient practices in deep learning training: Preliminary steps towards green ai," in *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*, pp. 25–36, IEEE Computer Society, may 2023.
2. T. Xu, "These simple changes can make ai research much more energy efficient." https://www.technologyreview.com/2022/07/06/1055458/ai-research-emissions-energy-efficient/. Accessed: 2023-12-01.
3. "Shrinking deep learning's carbon footprint." https://news.mit.edu/2020/shrinking-deep-learning-carbon-footprint-0807/. Accessed: 2023-11-01.
4. "Power-hungry ai: Researchers evaluate energy consumption across models." https://cse.engin.umich.edu/stories/power-hungry-ai-researchers-evaluate-energy-consumption-across-models. Accessed: 2023-12-11.
5. W. Shiqiang, "Efficient deep learning," in *Nature Computational Science*, vol. 1, pp. 181–182, 2021.
6. T. Luo, W.-F. Wong, R. S. M. Goh, A. T. Do, Z. Chen, H. Li, W. Jiang, and W. Yau, "Achieving green ai with energy-efficient deep learning using neuromorphic computing," *Commun. ACM*, vol. 66, p. 52–57, jun 2023.
7. X. Tu, A. Mallik, D. Chen, K. Han, O. Altintas, H. Wang, and J. Xie, "Unveiling energy efficiency in deep learning: Measurement, prediction, and scoring across edge devices," in *2023 IEEE Symposium on Edge Computing*, pp. 80–93, 2023.
8. E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 13693–13696, Apr. 2020.
9. D. Lazzaro, A. Cinà, M. Pintor, A. Demontis, B. Biggio, F. Roli, and M. Pelillo, *Minimizing Energy Consumption of Deep Learning Models by Energy-Aware Training*, pp. 515–526. arXiv e-prints, 09 2023.
10. E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, "Estimation of energy consumption in machine learning," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 75–88, 2019.
11. S. Georgiou, M. Kechagia, T. Sharma, F. Sarro, and Y. Zou, "Green ai: do deep learning frameworks have different costs?," in *Proceedings of the 44th International Conference on Software Engineering*, ICSE '22, (New York, NY, USA), p. 1082–1094, Association for Computing Machinery, 2022.
12. R. Verdecchia, L. Cruz, J. Sallou, M. Lin, J. Wickenden, and E. Hotellier, "Data-centric green ai an exploratory empirical study," in *2022 International Conference on ICT for Sustainability (ICT4S)*, (Los Alamitos, CA, USA), pp. 35–45, IEEE Computer Society, jun 2022.
13. S. Gholami and M. Omar, "Can pruning make large language models more efficient?," 2023.
14. X. Wang, H. Wang, B. Bhandari, and L. Cheng, "Ai-empowered methods for smart energy consumption: A review of load forecasting, anomaly detection and demand response," *International Journal of Precision Engineering and Manufacturing-Green Technology*, pp. 1–31, 09 2023.
15. S. Naumann, M. Dick, E. Kern, and T. Johann, "The greensoft model: A reference model for green and sustainable software and its engineering," *Sustainable Computing: Informatics and Systems*, vol. 1, no. 4, pp. 294–304, 2011.
16. J. Mancebo, C. Calero, and F. García, *GSMP: Green Software Measurement Process*, pp. 43–67. Cham: Springer International Publishing, 2021.

17. T. Johann, M. Dick, S. Naumann, and E. Kern, "How to measure energy-efficiency of software: Metrics and measurement results," *2012 1st International Workshop on Green and Sustainable Software, GREENS 2012 - Proceedings*, 06 2012.
18. Y. S. Zhong, T. Zhang, and G. Ronzoni, "Sentimental analysis of facebook reviews: Does hospitality matter in senior living?," *International Journal of Hospitality Management*, vol. 112, p. 103384, 2023. Hospitality in Healthcare.
19. Y. Ji, W. Wu, H. Zheng, Y. Hu, X. Chen, and L. He, "Is chatgpt a good personality recognizer? a preliminary study," 2023.
20. Y. Deng, W. Zhang, S. J. Pan, and L. Bing, "SOUL: Towards sentiment and opinion understanding of language," in *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.
21. J. Chen, X. Ye, J. Sun, and C. Li, "Towards energy-efficient sentiment classification with spiking neural networks," in *Artificial Neural Networks and Machine Learning – ICANN 2023* (L. Iliadis, A. Papaleonidas, P. Angelov, and C. Jayne, eds.), (Cham), pp. 518–529, Springer Nature Switzerland, 2023.
22. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
23. K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.
24. G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269, IEEE Computer Society, jul 2017.
25. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017.
26. J. Tang, *Intelligent Mobile Projects with TensorFlow*, ch. 2. Pact Publishing, 2018.
27. "sktime." https://www.sktime.net/en/stable/. Accessed: 2024-01-08.
28. "imdb-dataset-of-50k-movie-reviews." https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews. Accessed: 2023-11-01.
29. "Bert for sentiment-analysis." https://github.com/chriskhanhtran/bert-for-sentiment-analysis. Accessed: 2023-10-01.
30. "Esp32." https://www.espressif.com/en/products/socs/esp32.
31. "Mqtt." https://mqtt.org/. Accessed: 2024-01-08.
32. E. Kern, L. M. Hilty, A. Guldner, Y. V. Maksimov, A. Filler, J. Gröger, and S. Naumann, "Sustainable software products—towards assessment criteria for resource and energy efficiency," *Future Generation Computer Systems*, vol. 86, pp. 199–210, 2018.
33. A. Guldner and J. Murach, "Measuring and assessing the resource and energy efficiency of artificial intelligence of things devices and algorithms," in *Advances and New Trends in Environmental Informatics* (V. Wohlgemuth, S. Naumann, G. Behrens, H.-K. Arndt, and M. Höb, eds.), (Cham), pp. 185–199, Springer International Publishing, 2023.
34. A. Guldner, R. Bender, C. Calero, G. S. Fernando, M. Funke, J. Gröger, L. M. Hilty, J. Hörnschemeyer, G.-D. Hoffmann, D. Junger, T. Kennes, S. Kreten, P. Lago, F. Mai, I. Malavolta, J. Murach, K. Obergöker, B. Schmidt, A. Tarara, J. P. De Veaugh-Geiss, S. Weber, M. Westing, V. Wohlgemuth, and S. Naumann, "Development and evaluation of a reference measurement model for assessing the resource and energy efficiency of software products and components—green software measurement model (gsmm)," *Future Generation Computer Systems*, vol. 155, pp. 402–418, 2024.